

Ablaufplanung bei Reihenfertigung mit mehrfacher Zielsetzung auf der Basis von Ameisenalgorithmen

Zur Erlangung des akademischen Grades eines
Doktors der Wirtschaftswissenschaften

(Dr. rer. pol.)

der Wirtschaftswissenschaftlichen Fakultät
der Universität Passau
vorgelegte Dissertation

von

Dipl.-Kfm. Christian Petri

Gutachter: Prof. Dr. H. Ziegler
Prof. Dr. P. Kleinschmidt
Tag der letzten Fachprüfung: 14.12.2006

Danksagung

Die vorliegende Arbeit ist in der Zeit von Juli 2002 bis Dezember 2006 unter Betreuung von Prof. Dr. H. Ziegler am Lehrstuhl für Betriebswirtschaftslehre mit Schwerpunkt Produktion und Logistik der Universität Passau entstanden.

Zunächst möchte ich meinem Doktorvater Herrn Prof. Dr. H. Ziegler herzlichst für die freundliche Betreuung, die vielen anregenden Diskussionen und den großen Freiraum, den er mir eingeräumt hat, danken. Seine Unterstützung sowohl während meiner Zeit als wissenschaftlicher Mitarbeiter als auch als Stipendiat hat zum Gelingen dieser Arbeit maßgeblich beigetragen.

Ebenso gilt mein Dank Herrn Prof. Dr. P. Kleinschmidt, der die Arbeit als Zweitgutachter betreut und mich insbesondere bei der Beantragung der Stipendien unterstützt hat.

Ein Teil der Arbeit ist während eines 6-monatigen Forschungsaufenthaltes am Indian Institute of Technology in Chennai, Indien entstanden. Hier gilt mein Dank insbesondere Herrn Prof. C. Rajendran und seinen Mitarbeitern für die hervorragende fachliche Betreuung und hilfreichen Anregungen. Ihm, seiner Familie, seinen Mitarbeitern und Studenten danke ich für die intensive persönliche und medizinische Betreuung während dieser Zeit.

Dem DAAD danke ich für das Forschungsstipendium, welches mir die Forschung am IIT aber auch die schönen und unvergesslichen Eindrücke über dieses beeindruckende Land ermöglicht hat.

Daneben haben viele weitere Personen zum Gelingen dieser Arbeit beigetragen, denen allen zu Danken hier der Platz nicht ausreicht. Der letzte und größte Dank gilt meinen Eltern, die mich stets uneingeschränkt und mit viel Liebe unterstützt haben. Ihnen möchte ich diese Arbeit widmen.

Inhaltsverzeichnis

1	Einleitung	1
2	Charakterisierung des Permutation Flow Shop Problems	5
2.1	Grundlagen und Prämissen	5
2.1.1	Klassifikation von Reihenfolgeproblemen	5
2.1.2	Spezifikation des in der Arbeit betrachteten Permutation Flow Shop Problems	8
2.2	Das betrachtete Zielsystem	10
2.2.1	Ziele für das Permutation Flow Shop Problem	10
2.2.1.1	Die Gewinnmaximierung als Beurteilungsmaßstab	10
2.2.1.2	Lagerkosten des Fertigungsbestandes	11
2.2.1.3	Terminabweichungskosten	14
2.2.1.4	Leerzeitkosten und Kapazitätsauslastung	15
2.2.1.5	Zielsystem und Berechnung der Fertigstellungszeitpunkte der Aufträge	17
2.2.2	Analyse der Zielbeziehungen	18
3	Das multikriterielle PFSP als Vektorminimierungsproblem	22
3.1	Einführung	22
3.2	Methoden zur Ermittlung effizienter Lösungen	30
3.2.1	Definition Kompromißmodell	30
3.2.2	Diskussion verschiedener Kompromißmodelle	31
3.2.2.1	Lexikographische Optimierung	31
3.2.2.2	Konvexkombination	32
3.2.2.3	Epsilon-Constraint Ansatz	33
3.2.2.4	Kompromißmodell von Soland (1979)	34
3.2.2.5	Abstandsfunktionen	35
3.2.2.6	Goal-Programming	39

3.3	Lösungsansätze für das PFSP und Single Machine Problem bei mehrfacher Zielsetzung	40
3.3.1	Exakte und heuristische Verfahren	40
3.3.1.1	Exakte Verfahren	40
3.3.1.2	Heuristische Lösungsansätze	41
3.3.2	A-priori Ansätze	42
3.3.3	A-posteriori Ansätze	43
3.3.4	ACO Algorithmen für das multikriterielle PFSP und Single Machine Problem	44
4	Lösungsqualität heuristisch effizienter Mengen	46
4.1	Einführung	46
4.1.1	Grundsätzliche Überlegungen zur Lösungsqualität heuristisch effizienter Mengen	46
4.1.2	Klassifikation von und Anforderungen an Maße zur Qualitätsanalyse	50
4.1.2.1	Klassifikation von Qualitätsmaßen	50
4.1.2.2	Die TMPS als Referenzlösung	51
4.1.2.3	Anforderungen an geeignete Maße: Monoto- nie und Relativität	53
4.2	Beurteilung von Qualitätsmaßen	55
4.2.1	Distanz zur Menge aller effizienten Lösungen	55
4.2.1.1	Kompatibilität mit den Outperformance Re- lationen als zusätzliche Anforderung	55
4.2.1.2	Beurteilung von Maßen	59
4.2.2	Größe der aufgezeigten Variationsbreiten der Zielfunktionswerte	79
4.2.3	Verteilung der Zielfunktionsvektoren im Zielraum . . .	83
4.2.4	Optimierungseigenschaft bezüglich der einzelnen Ziele	89
4.3	Aufbau des verwendeten Beurteilungssystems	92
5	Zielraum- und Lösungsraumanalyse	93
5.1	Grundlegende Aspekte	93
5.1.1	Stand der Forschung	93
5.1.2	Weiterführende Experimente	95
5.2	Nachbarschaften im Lösungsraum	99
5.2.1	Nachbarschaftsoperatoren	99
5.2.2	Beziehungen zwischen Nachbarschaftsoperatoren . . .	101
5.3	Nachbarschaft im Zielraum	102
5.3.1	Nachbarschaftskonzepte	102

5.3.1.1	Allgemeine Überlegungen	102
5.3.1.2	Nachbarschaft im Zielraum: Konzept 1 (NSZR1)	103
5.3.1.3	Nachbarschaft im Zielraum: Konzept 2 (NSZR2)	105
5.3.1.4	Nachbarschaft im Zielraum: Konzept 3 (NSZR3)	106
5.3.1.5	Nachbarschaft im Zielraum: Konzept 4 (NSZR4)	107
5.3.2	Experimentelle Analyse	113
5.3.2.1	Experiment 1: Nachbarschaft im Lösungsraum gleich Nachbarschaft im Zielraum?	113
5.3.2.2	Experiment 2: Abweichung NSLR von NSZR4	117
5.3.2.3	Zusammenfassung	123
5.4	Analyse der Qualität der Distanzapproximationen	124
5.4.1	Zur Notwendigkeit der Approximation	124
5.4.2	Maße zur Distanzapproximation zwischen zwei Auftragsfolgen	125
5.4.3	Analyse der Approximationsqualität	127
5.4.3.1	Allgemeine Anforderungen an ein Maß	127
5.4.3.2	Differenzierungstiefe und Korrelation mit der echten Distanz	129
5.5	Analyse der Topographie im Lösungsraum	134
5.5.1	Die Korrelation von Distanz und Zielfunktionswert	134
5.5.2	Analyse der Distanz globaler Optima	139
5.5.2.1	Isolierte Analyse multipler globaler Optima der einzelnen Ziele	139
5.5.2.2	Integrierte Analyse aller globalen und der effizienten globalen Optima aller Ziele	142
5.5.3	Analyse der Distanz der Elemente der Menge aller effizienten Lösungen	144
5.6	Zusammenfassung	145
6	ACO Algorithmen für das multikriterielle PFSP	147
6.1	Grundlagen der ACO Metaheuristik	147
6.1.1	Die Natur als Vorbild	147
6.1.2	Definition und Eigenschaften der ACO-Metaheuristik	151
6.1.3	Ausgestaltungsmöglichkeiten für das PFSP	155
6.1.3.1	Die Pheromonmatrix T	155

6.1.3.2	Anzahl der Ameisen <i>Ants</i> in einer Kolonie . . .	156
6.1.3.3	Konstruktion einer Auftragsfolge	156
6.1.3.4	Modellierung des autokatalytischen Prozesses	161
6.1.4	Ausgestaltungsmöglichkeiten für a-posteriori Ansätze	163
6.1.4.1	Allgemeine Ausgestaltungsmöglichkeiten . . .	163
6.1.4.2	Multi Colony Ansätze und Kommunikation . .	166
6.1.4.3	Der Multi Colony Ansatz von Iredi u. a. (2001)	167
6.2	Lokale Suchstrategie in zwei Phasen	169
6.2.1	Motivation der zwei Phasen-Strategie	169
6.2.2	Lokale Suche in Phase 1: Job Swap Schema (JSS) und Job Insertion Schema (JIS)	171
6.2.3	Lokale Suche in Phase 2: Das Job Swap Schema mit rekursiver API Suche (JSS-RAPI)	173
6.3	Der SCMO-Ansatz	174
6.3.1	Beschreibung des Algorithmus	174
6.3.1.1	Allgemeine Eigenschaften	174
6.3.1.2	Initialisierungsphase und Ermittlung der en- dogenen kleinsten oberen Schranken	176
6.3.1.3	Suchschema im Zielraum	179
6.3.1.4	Die SCMO-SUCHE-REGION Prozedur	181
6.3.2	Experimentelle Analyse	184
6.3.2.1	Aufbau der Experimente und Bewertungsme- thoden	184
6.3.2.2	Ausgangsparametereinstellungen	186
6.3.2.3	Experiment 1: Systematische Suche im Ziel- raum und Verwendung der LNDS	189
6.3.2.4	Experiment 2: Einfluß der Neuinitialisierung der Pheromonmatrix	195
6.3.2.5	Experiment 3: Variation von B	198
6.3.2.6	Experiment 4: Variation der Parameter <i>Iter</i> und <i>Ants</i>	202
6.3.2.7	Experiment 5: Variation des Parameters ρ . .	206
6.4	Der MCMO-Ansatz	209
6.4.1	Beschreibung des Algorithmus - Variante 1 (MCMO1) .	209
6.4.1.1	Allgemeine Eigenschaften	209
6.4.1.2	MCMO-MASTER - Suche im Zielraum	210
6.4.1.3	MCMO-CONSTRUCT - Konstruktion von Auftragsfolgen	222
6.4.1.4	MCMO-SUB1, MCMO-SUB2 und MCMO-SUB3	224

6.4.2	Modifikation des MCMO-Ansatz	
	- Variante 2 (MCMO2)	229
6.4.3	Experimentelle Analyse	230
6.4.3.1	Experiment 1: Kommunikationsstrategien . .	230
6.4.3.2	Experiment 2: Variation von B	235
6.4.3.3	Experiment 3: Variation der Parameter <i>Iter</i> und <i>Ants</i>	243
6.4.3.4	Experiment 4: Variation des Parameters ρ . .	248
6.5	Vergleich mit einem Referenzverfahren	253
6.5.1	Grundsätzliche Überlegungen	253
6.5.2	Experimentelle Analyse	254
6.5.2.1	Vergleich der Lösungsqualität von SCMO, MCMO und MOGLS	254
6.5.2.2	Verbesserung der kleinsten oberen Schranken durch SCMO und MCMO	259
6.6	Zusammenfassung	266
7	Zusammenfassung und Ausblick	268
	Literaturverzeichnis	272
A	Generierung von Testproblemen	287
B	Generierung von Fertigstellungsterminen	288
C	Verwendete kleinste obere Schranken für C_{max}, F_{mid}, und T_{max}	290

Abkürzungsverzeichnis

<i>ACO</i>	Ant Colony Optimization
<i>ACO2</i>	ACO-Ansatz von Rajendran u. Ziegler (2002)
<i>ADJ1</i>	Unidirektional Adjacency Metric
<i>ADJ2</i>	Bidirectional Adjacency Metric
<i>API</i>	Adjacent Pairwise Interchange Nachbarschaft
<i>BSH</i>	Backward Shift Nachbarschaft
<i>DSH</i>	Double Shift Nachbarschaft
<i>ENGA</i>	Elitist Nondominated Sorting Genetic Algorithm
<i>EX</i>	Exchange Nachbarschaft
<i>FSH</i>	Forward Shift Nachbarschaft
<i>GPWGA</i>	Gradual-Priority Weighting Genetic Algorithm
<i>GNDF</i>	Globale heuristisch effiziente Front
<i>GNDS</i>	Globale heuristisch effiziente Menge
<i>INV</i>	Inversions Nachbarschaft
<i>JIS</i>	Job Insertion Schema
<i>JSP</i>	Job Shop Problem
<i>JSS</i>	Job Swap Schema
<i>JSS – RAPI</i>	Job Swap Schema mit rekursiver API-Suche
<i>LNDF</i>	Lokale heuristisch effiziente Front
<i>LNDS</i>	Lokale heuristisch effiziente Menge
<i>MCMO</i>	Multi-Colony-Multi-Objective
<i>MCMO1</i>	MCMO Variante 1
<i>MCMO2</i>	MCMO Variante 2
<i>MOGLS</i>	Multi Objective Genetic Local Search
<i>MOSA</i>	Multi Objective Simulated Annealing
<i>NDF</i>	Heuristisch effiziente Front (Non Dominated Front)
<i>NDS</i>	Heuristisch effiziente Menge (Non Dominated Set)
<i>NSGA</i>	Nondominated Sorting Genetic Algorithm
<i>NSLR</i>	Nachbarschaft im Lösungsraum
<i>NSLR'</i>	Reduzierte Nachbarschaft im Lösungsraum

<i>NSZR</i>	Nachbarschaft im Zielraum
<i>NSZR1</i>	Nachbarschaft im Zielraum Konzept 1
<i>NSZR2</i>	Nachbarschaft im Zielraum Konzept 2
<i>NSZR3</i>	Nachbarschaft im Zielraum Konzept 3
<i>NSZR4</i>	Nachbarschaft im Zielraum Konzept 4
<i>PFSP</i>	Permutation Flow Shop Problem
<i>POSN</i>	Position Based Metric
<i>PREC</i>	Precedence Metric
<i>RAPI</i>	Rekursive API Suche
<i>SACO</i>	ACO Ansatz von T'kindt u. a. (2002)
<i>SCMO</i>	Single-Colony-Multi-Objective
<i>SMTEMDP</i>	Single Machine Total Earliness with Multiple Due Dates Problem
<i>SMTTP</i>	Single Machine Total Tardiness Problem
<i>SMTWTP</i>	Single Machine Total Weighted Tardiness Problem
<i>SMTWDP</i>	Single Machine Total Weighted Deviation Problem
<i>SSDP</i>	Single Start mit dynamischem Parametersatz
<i>SSFP</i>	Single Start mit fixiertem Parametersatz
<i>TMPF</i>	Total Meta Pareto Front
<i>TMPS</i>	Total Meta Pareto Set
<i>TSP</i>	Traveling Salesman Problem
<i>UMSS</i>	Unverbundene Multi Start Strategie
<i>VMSS</i>	Verbundene Multi Start Strategie

Mathematische Symbole und Schreibweisen

$ADJ1(\pi, \pi')$	Unidirectional Adjacency Metric zwischen π und π'
$ADJ2(\pi, \pi')$	Bidirectional Adjacency Metric zwischen π und π'
$Ants$	Anzahl der Ameisen in einer Kolonie
ant	Ameisenindex, $ants = 1(1)Ants$
AEX_l^k	Anteil gefundener Extrempunkte des Ziels k durch Lösung l
$AEX_{v,run}^k$	AEX^k -Maß der Lösung aus Rechenlauf run von Lösungsverfahren v
$AEX_v^{\sigma k}$	Standardabweichung des AEX^k -Maßes von Lösungsverfahren v
$AEX_v^{\mu k}$	Durchschnittswert des AEX^k -Maßes von Lösungsverfahren v
$API(\pi)$	Adjacent Pairwise Interchange Nachbarschaft von π
AVG_k^1	Durchschnittliche Abweichung von $NSZR4$ bzgl. Ziel k bezogen auf die gesamte reduzierte Nachbarschaft im Lösungsraum
AVG_k^2	Durchschnittliche Abweichung von $NSZR4$ bzgl. Ziel k bezogen auf die Anzahl der Auftragsfolgen, die eine positive Abweichung aufweisen
α	Parameter des Ameisenalgorithmus
$BC(NDS_l, NDS_w)$	Binary Coverage Metric zwischen Lösung l und w $TMPS$ als Referenzlösung
$BSH(\pi)$	Backward Shift Nachbarschaft von π
B	Parameter im SCMO- und MCMO-Ansatz, auf dessen

	Grundlage die Anzahl der Suchbereiche im Zielraum (\hat{B}) bestimmt wird
\hat{B}	Anzahl Suchbereiche im Zielraum beim SCMO- und MCMO-Ansatz
b	Beschränkungsvektor im Kompromißmodell von Soland (1979)
β	Parameter des Ameisenalgorithmus
C_{max}	Zykluszeit
C_n	Fertigstellungszeitpunkt von Auftrag n
$C_{[\pi_h]}$	Fertigstellungszeitpunkt des Auftrags auf Position h in Auftragsfolge π
C_{nm}	Fertigstellungszeitpunkt von Auftrag n auf Maschine m
$C(NDS_l, NDS_w)$	Coverage-Maß (C-Maß) zwischen Lösung l und w
CL_l^μ	Cluster Maß der Lösung l
CLR_l^μ	Cluster Ratio der Lösung l
c	Vektor zur Identifikation einer Indifferenzregion bei der Berechnung von $NT_l^\mu(c)$
$c(x, y)$	Distanz im Zielraum zwischen je einem Zielfunktionsvektor zweier heuristisch effizienter Mengen. Verwendet bei der Berechnung von $Dist1_l$ und $Dist2_l$
$cand$	Größe von π^{cand}
$D1_l$	Modifikation von $Dist1_l$
$D2_l$	Modifikation von $Dist2_l$
$DEV_k(\pi', \pi^\circ)$	Abweichung eines Elements π' aus $NSLR^r(\pi^\circ)$ von der $NSZRA(z(\pi^\circ))$
$Dist1_l$	Durchschnittliche Distanz der Lösung l zur Referenzlösung Ref
$Dist2_l$	Größte Distanz eines Elements aus Ref zum jeweils nächsten Element aus NDS_l
$DSH(\pi)$	Double Shift Nachbarschaft von π
$DTMPF_l^k$	Relative Abweichung der Lösung l vom Minimum der $TMPF$ bzgl. Ziel k
DUB_l^k	Relative Abweichung der Lösung l von der besten bekannten kleinsten oberen Schranke bzgl. Ziel k
$DUB_{v,run}^k$	DUB^k -Maß der Lösung aus Rechenlauf

	<i>run</i> von Lösungsverfahren v
$DUB_v^{\sigma k}$	Standardabweichung des DUB^k -Maßes von Lösungsverfahren v
$DUB_v^{\mu k}$	Durchschnittswert des DUB^k -Maßes von Lösungsverfahren v
$d(x)$	Euklidische Distanz im Zielraum, von Element $x \in NDS_l$ zum nächstliegenden Element der $E(\Pi, z)$.
$d(z(x))$	Euklidische Distanz im Zielraum zwischen zwei direkt aufeinanderfolgenden Elementen einer NDF
\bar{d}	Durchschnittswert über alle $d(z(x))$ der Elemente x einer NDF
d_n	Liefertermin ; Spätestzulässiger Fertigstellungstermin von Auftrag n
Δ_k	Variationsbreite des Ziels k in der Referenzlösung <i>Ref</i>
$\Delta AEX_v^{\mu k}$	Verhältnis von AEX_v^{μ} zu einem Referenzwert
$\Delta AEX_v^{\sigma k}$	Verhältnis von AEX_v^{σ} zu einem Referenzwert
$\Delta DUB_v^{\mu k}$	Verhältnis von DUB_v^{μ} zu einem Referenzwert
$\Delta DUB_v^{\sigma k}$	Verhältnis von DUB_v^{σ} zu einem Referenzwert
ΔSQ_v^{μ}	Verhältnis von SQ_v^{μ} zu einem Referenzwert
ΔSQ_v^{σ}	Verhältnis von SQ_v^{σ} zu einem Referenzwert
$\Delta Ticks_v^{\mu}$	Verhältnis von $Ticks^{\mu}$ zu einem Referenzwert
$\Delta Ticks_v^{\sigma}$	Verhältnis von $Ticks^{\sigma}$ zu einem Referenzwert
$\Delta \tau_{nh}(t)$	Hinzuzufügende Pheromonmenge bei der Pheromonaktualisierung zum Zeitpunkt t
δ	Parameter in der erweiterten gewichteten Tchebycheff-Distanz
e	Rangindex bei der lex. Optimierung, $e = 1(1)K$
$[e]$	Index, des in Rang e eingeordneten Ziels bei der lexikographischen Optimierung, $e = 1(1)K$
$E(\Pi, z)$	Menge aller (streng) effizienten Lösungen
$E(z)$	Menge aller effizienten Zielfunktionsvektoren; Effiziente Front
$ENDF_l^k$	Menge der Zielfunktionsvektoren, die für Ziel k den kleinsten Wert innerhalb von NDF_l aufweisen
ER_l	Error Ratio der Lösung l
ER_l^{VEL}	Error Ratio der NDF_l , wie von

$ETMPF^k$	Van Veldhuizen (1999) vorgestellt Menge der Zielfunktionsvektoren, die für Ziel k den kleinsten Wert innerhalb der $TMPF$ aufweisen
$EX(\pi)$	Exchange Nachbarschaft von π
ϵ_i^k	Element des Vektors ϵ^k
ϵ^k	Beschränkungsvektor im Epsilon-Constraint Modell
η_{nh}	Lokale Komponente bei der Berechnung der Auswahlwahrscheinlichkeit p_{nh}
F_{mid}	Mittlere Durchlaufzeit
F_n	Durchlaufzeit von Auftrag n
F_{sum}	Gesamtdurchlaufzeit
$FSH(\pi)$	Forward Shift Nachbarschaft von π
g	Funktion der Menge G_z
G_z	Menge der streng monoton fallenden Funktionen
GD_l	Generational Distance von Lösung l
h	Positionsindex in Auftragsfolge π , $h = 1(1)N$
HD_l	Hyperareadifference der Lösung l
\bar{I}	Durchschnittliche Maschinenleerzeit
I_m	Leerzeit auf Maschine m
$InfI_l$	Inferiority Index der Lösung l
$InMax$	Innerhalb von $MaxDist$ erreichbare Zahl von Auftragsfolgen
$INV(\pi)$	Inversionsnachbarschaft von π
$Iter$	Anzahl Iterationen beim SCMO- und MCMO-Ansatz bei gleichem Gewichtungsvektor λ
$iter$	Iterationenindex, $iter = 1(1)Iter$
$[i]_\pi$	Position von Auftrag i in Auftragsfolge π
k	Zielindex, $k = 1(1)K$
K	Anzahl der Ziele im Zielsystem
l	Lösungsindex, $l = 1(1)L$
L	Anzahl der zur Ermittlung der $TMPS$ bzw. $TMPF$

	verwendeten Lösungen
$L(\pi^{ant})$	Allgemeine Notation für die Lösungsqualität der von der Ameise ant konstruierten Lösung π^{ant}
LB^{Cmax}	Untere Grenze der Zykluszeit
$LEVEL$	Aktuelle Tiefe der Rekursion bei $RAPI$
λ	Zielgewichtungsvektor in Kompromißmodellen, $\lambda \in \mathbb{R}^K$
λ_k	Element des Vektors λ , Gewicht von Ziel k
m	Maschinenindex, $m = 1(1)M$
M	Anzahl Maschinen
$Match$	Mächtigkeit der Schnittmenge von NSLR und NSZR $Match = NSLR \cap NSZR $
$MaxDist$	Maximale Distanz zwischen den Elementen einer Menge von Auftragsfolgen
MAX_LEVEL	Maximal zulässige Tiefe der Rekursion beim $RAPI$
MOS_l^k	Modified Kth Objective Pareto Spread der Lösung l
MOS_l	Modified Overall Pareto Spread der Lösung l
$MPFE_l$	Maximum Pareto Front Error der Lösung l
MPS_l	Meta Pareto Solution Set von NDS_l
n	Auftragsindex, $n = 1(1)N$
N	Anzahl Aufträge
N_T	Anzahl terminüberschreitender Aufträge
NDC_l^μ	Number of Distinct Choices der Lösung l
NDF_l	Heuristisch effiziente Front (Non Dominated Front) l
NDS_l	Heuristisch effiziente Menge (Non Dominated Solutions) l
$NNDS_l$	Number of Non Dominated Solutions der Lösung l
$NS_k^+(z(\pi))$	Menge aller Auftragsfolgen, deren Zielfunktionswerte die kleinste nichtnegative Abweichung vom Zielfunktionsvektor $z(\pi)$ der Auftragsfolge π bzgl. Ziel k aufweisen
$NS_k^-(z(\pi))$	Menge aller Auftragsfolgen, deren Zielfunktionswerte die kleinste nichtpositive Abweichung vom Zielfunktionsvektor $z(\pi)$ der Auftragsfolge π bzgl. Ziel k aufweisen
$NS_k(z(\pi))$	Nachbarschaft der Auftragsfolge π im Zielraum bzgl. Ziel k

$NS_k^{+E}(z(\pi))$	Menge aller bezogen auf den Abstandsvektor z^a effizienten Elemente aus $NS_k^+(z(\pi))$
$NS_k^{-E}(z(\pi))$	Menge aller bezogen auf den Abstandsvektor z^a effizienten Elemente aus $NS_k^-(z(\pi))$
$NS_k^E(z(\pi))$	Nachbarschaft der Auftragsfolge π im Zielraum (NSZR3) bzgl. Ziel k
$NS^r(z(\pi))$	Nachbarschaft von $z(\pi)$ in Region r bei NSZR4
$NSLR^r(\pi)$	Reduzierte Nachbarschaft im Lösungsraum von π
$NSLR(\pi)$	Nachbarschaft im Lösungsraum von π
$NSZR1(z(\pi))$	Nachbarschaft von π im Zielraum - Konzept 1
$NSZR2(z(\pi))$	Nachbarschaft von π im Zielraum - Konzept 2
$NSZR3(z(\pi))$	Nachbarschaft von π im Zielraum - Konzept 3
$NSZR4(z(\pi))$	Nachbarschaft von π im Zielraum - Konzept 4
$NT_l^\mu(c)$	Binärvariable, die anzeigt, ob $T_\mu(c)$ ein Element enthält oder nicht
OS_l^k	Kth Objective Pareto Spread der Lösung l
OS_l	Overall Pareto Spread der Lösung l
p	Parameter der l_p -Distanzen
p_{nm}	Bearbeitungszeit von Auftrag n auf Maschine m
P_n	Gesamtbearbeitungszeit von Auftrag n
Ψ_ϵ	Kompromißmodell Epsilon-Constraint-Ansatz
$\Psi_{(g,b)}$	Kompromißmodell von Soland (1979)
Ψ_λ	Kompromißmodell Gewichtete Tchebycheff-Distanz
Ψ_λ^g	Kompromißmodell Goal Programming mit l_p -Maß
Ψ_λ^{konvex}	Kompromißmodell Konvexkombination
Ψ_λ^{steuer}	Abwandlung von Steuer (1986) für $\Psi_{(\lambda,\delta)}$
Ψ_Θ	Kompromißmodell ungewichtete Tchebycheff Distanz
$\Psi_{(\lambda,\delta)}$	Kompromißmodell erweiterte gewichtete Tchebycheff-Distanz
PES_l	Proportion of Efficient Solutions der Lösung l
$POSN(\pi, \pi')$	Position Based Metric zwischen π und π'
$PREC(\pi, \pi')$	Precedence Metric zwischen π und π'
p_{nh}	Wahrscheinlichkeit für Auftrag n auf Position h plaziert zu werden
p_0	Parameter der Pseudo Random

	Proportionate Rule
p_1	Parameter der erweiterten Pseudo Random Proportionate Rule
p_2	Parameter der erweiterten Pseudo Random Proportionate Rule
$\psi(\pi)$	Kompromißzielfunktion (allgemeine Form)
π	Permutation von Aufträgen, Auftragsfolge, $\pi \in \Pi$
$\tilde{\pi}$	aktuelle Teillösung / aktuelle Teilauftragsfolge
$\pi_{[h]}$	Auftrag auf Position h in Auftragsfolge π
π_k^*	Auftragsfolge, die Zielgröße k minimiert
π^{ant}	Von Ameise <i>ant</i> konstruierte Auftragsfolge
π_k^{aid}	Auftragsfolge, die Zielfunktionswert k maximiert
π^{cand}	Kandidatenliste bei der Lösungskonstruktion durch eine Ameise
π^{glb}	Global beste bekannte Lösung einer Kolonie
π_k^{id}	Auftragsfolge, die Zielfunktionswert k minimiert
π^{init}	Initialisierungsauftragsfolge für die Pheromonmatrizen im SCMO- und MCMO-Ansatz
π_k^{nad}	Auftragsfolge aus $E(\Pi, z)$, die Zielgröße k maximiert
Π	Lösungsraum, Menge aller $N!$ möglichen Permutationen von N Aufträgen
Π_k^*	Menge individuell optimaler Lösungen bzgl. Ziel k
$\Pi_{[k]}$	Lösungsmenge in der lexikographischen Optimierung, nachdem nach Ziel $[k]$ optimiert wurde, mit $[k]$ als Ziel auf Rang k
$\Pi(\Psi_\lambda)$	Lösungsmenge des Kompromißmodells Ψ_λ
q	Hyperrechteck
Q	Parameter beim Pheromonupdate nach dem Ant Cycle Prinzip
r_n	Ankunftszeitpunkt von Auftrag n
r	Regionenindex, $r = 1(1)R$
r'	Regionenindex, $r' = 1(1)R$
R	Anzahl Regionen in <i>NSZRA</i>
R_C	Vollständige (complete) Outperformance Relation
R_O	Outperformance Relation
R_S	Starke (strong) Outperformance Relation

R_W	Schwache (weak) Outperformance Relation
Ref	Referenzlösung
$range$	Parameter der lokalen Suchverfahren. Anzahl Positionen vor und nach der aktueller Position des aktuellen Auftrags, die bei der Überprüfung auf einen vorteilhaften Zug berücksichtigt werden
$rank$	Anzahl der Eliteameisen im AS^{rank} Ansatz
$Runs$	Anzahl der Lösungen, die für eine Probleminstanz von einem Lösungsverfahren v bestimmt werden
$runs$	Index eines Rechenlaufs, $runs = 1(1)Runs$
ρ	Korrelationskoeffizient , Trail-Persistence-Rate
ρ_{korr}	Korrektur des Korrelationskoeffizienten von OLKIN U. PRATT (1958)
ρ^{real}	Korrelation zwischen Zielfunktionswerten und echter Distanz
ρ^{ADJ1}	Korrelation zwischen Zielfunktionswerten und ADJ1
ρ^{ADJ2}	Korrelation zwischen Zielfunktionswerten und ADJ2
ρ^{POSN}	Korrelation zwischen Zielfunktionswerten und POSN
ρ^{PREC}	Korrelation zwischen Zielfunktionswerten und PREC
$step$	Schrittweite, mit der die Elemente von λ im SCMO- und MCMO-Ansatz variiert werden
S_l	Size of Dominated Space (S-Maß) der Lösung l
\tilde{S}_l	Mit normierten Zielfunktionsvektoren ermitteltes S-Maß der Lösung l
$SCZY_l$	Distanzquotient der Lösung l von Czyzak u. Jaszkiwicz (1998)
SD_l	Spacing-Maß der Lösung l
$SolT_v$	Insgesamt von Lösungsverfahren v generierte Anzahl von Auftragsfolgen
$SolLS_v$	Von Lösungsverfahren v in der lokalen Suche generierte Anzahl von Auftragsfolgen
SQ_l	S-Maß Quotient der Lösung l
$SQ_{v,run}$	S-Maß Quotient der Lösung aus Rechenlauf run von Lösungsverfahren v
SQ_v^σ	Standardabweichung des S-Maß Quotienten des Lösungsverfahrens v
SQ_v^μ	Durchschnittlicher S-Maß Quotient des Lösungsverfahrens v

T	Pheromonmatrix (allgemeine Schreibweise)
T_{nh}	Summen-Pheromonwert
T_{max}	Maximale Terminüberschreitung
T_{mid}	Mittlere Terminüberschreitung
T_{sum}	Summe der Terminüberschreitungen
$T_{\mu}(c)$	Indifferenzregion von c
$TicksLS_v$	Von Lösungsverfahren v für die lokale Suche benötigte Rechenzeit
$TicksT_v$	Benötigte Gesamtrechenzeit eines Lösungsverfahrens v
τ_{ij}	Pheromonwert, der ausdrückt, wie vorteilhaft es ist, Auftrag j direkt nach Auftrag i zu fertigen
τ_{max}	Oberer Grenzwert für Pheromone beim MAX-MIN Ant System
τ_{min}	Unterer Grenzwert für Pheromone beim MAX-MIN Ant System
τ_{nh}	Pheromonwert, der ausdrückt, wie vorteilhaft es ist, Auftrag n auf Position h zu plazieren
Θ	Vektor zur Bestimmung des Tchebycheff-Punktes, $\Theta \in \mathbb{R}^K$
$\Theta_{[k]}$	Element von Θ
U	Menge nicht ausgewählter Elemente / Aufträgen
\bar{U}	Durchschnittliche Kapazitätsauslastung des Fertigungsbereichs
u	Gleichverteilte Zufallszahl aus dem Intervall $[0;1]$, $u \leftarrow U[0; 1]$
μ	Parameter von NDC_l^{μ}
v	Index des Lösungsverfahrens, $v = 1(1)V$
V	Anzahl der in den Vergleich aufgenommenen Lösungsverfahren
w_{nm}	Wartezeit von Auftrag n vor Maschine m
W_n	Wartezeit von Auftrag n
W_{sum}	Summe der Wartezeiten aller N Aufträge
WC_k	Größte Abweichung von $NSZR4$ bzgl. Ziel k in der reduzierten Nachbarschaft im Lösungsraum

x_i	(Zufalls-)Variable bei der Berechnung des Korrelationskoeffizienten ρ
x_{ij}	Binärvariable
y_i	(Zufalls-)Variable bei der Berechnung des Korrelationskoeffizienten ρ
y_{ij}	Binärvariable
y_k	Anzahl an Auftragsfolgen π' in $NSLR^r$ mit $z_k(\pi') < z_k^{grenze}$
$z^a(z(\pi), z(\pi'))$	Vektor der absoluten Abstände der Vektoren $z(\pi)$ und $z(\pi')$
z^{aid}	Antiidealzielpunkt
z^b	Badpoint
\tilde{z}^b	Normierter Badpoint
z^g	Goodpoint
\tilde{z}^g	Normierter Goodpoint
z_k^{grenze}	Kleinster in $NSZR4$ auftretender Wert für Ziel k
z^{id}	Idealzielpunkt
z^{lex}	Zielfunktionsvektor der Lösung einer lexikographischen Optimierung
z^{max}	Beschränkungsvektor beim S-Maß (S_l)
z^{nad}	Nadirpunkt
z^{ref}	Referenzpunkt
z^{ut}	Utopiepunkt
$z(\pi)$	Zielfunktionsvektor der Auftragsfolge π , $z(\pi) \in \mathbb{R}^K$
$z_k(\pi)$	Zielfunktionswert bzgl. Ziel k , von Auftragsfolge π
z_k^{EUB}	Endogene kleinste obere Schranken von Ziel k
z_k^{NORM}	Wert zur Normierung der Zielgröße k bei der Ermittlung von $\psi_\lambda(\pi)$ im SCMO- und MCMO-Ansatz
$\tilde{z}_k(\pi)$	Normierter Zielfunktionsvektor von π
z_k^*	Zielfunktionswert der individuell opt. Lösung bzgl. Ziel k
z_{UB}^k	Exogene kleinste obere Schranke für Ziel k
z_l^{k*}	Kleinster Zielfunktionswert bzgl. Ziel k in Lösung l
$Z(\Pi)$	Zielraum, Menge der Zielfunktionsvektoren $z(\pi)$ aller Elemente π aus Π
ζ^k	Wert bei der Berechnung von DUB_l^k
ζ_k^{NORM}	Wert bei der Berechnung von z_k^{NORM}

<i>%InMax</i>	Anteil von <i>InMax</i> am gesamten Lösungsraum
<i>%LR</i>	Anteil der Auftragsfolgen in NSLR, die auch in NSZR sind
<i>%Ticks_v</i>	Anteil der Rechenzeit der lokalen Suche an der Gesamtrechenzeit von Lösungsverfahren <i>v</i>
<i>%SolT_v</i>	Anteil der während der lokalen Suche generierten Auftragsfolgen an der insgesamt generierten Anzahl von Auftragsfolgen von Lösungsverfahren <i>v</i>
<i>%ZR</i>	Anteil der Auftragsfolgen in NSZR, die auch in NSLR sind
<i>#O</i>	Anzahl globaler Optima für ein Ziel in einer Probleminstanz
<i>#P</i>	Anzahl Probleminstanzen mit multiplen globalen Optima

Operatoren in Algorithmen

==	Test auf Gleichheit
=	Zuordnung

Abbildungsverzeichnis

2.1	Menge aller effizienten Lösungen	9
3.1	Arten effizienter Punkte	25
3.2	Besondere Punkte im Zielraum	28
4.1	Das Effizienzkriterium bei der Beurteilung der Lösungsqualität	47
4.2	Ermittlung der TMPS	53
4.3	Schwache Outperformance	56
4.4	Starke Outperformance	56
4.5	Vollständige Outperformance	57
4.6	Gegenseitige Dominanz	58
4.7	Number of non dominated Solutions, Fall: $L \geq 3$	60
4.8	Error Ratio, Fall: $L \geq 3$	63
4.9	Proportions of efficient Solutions, Fall: $L \geq 3$	64
4.10	$D1_l$ und $D2_l$, Fall: $L \geq 3$	67
4.11	Berechnung der Generational Distance GD_l	69
4.12	Inkompatibilität der Generational Distance mit R_O	71
4.13	Inkompatibilität des Maximum Pareto Front Errors mit R_O	72
4.14	Coverage of two sets	74
4.15	S-Maß	77
4.16	Kth-Objective Pareto Spread	81
4.17	Cluster-Maß	87
4.18	Cluster Ratio	88
5.1	Motivation der Zielraum- und Lösungsraumanalyse	97
5.2	Nachbarschaft im Zielraum - Konzept 1	104
5.3	Nachbarschaft im Zielraum - Konzept 2	105
5.4	Nachbarschaft im Zielraum - Konzept 3	108
5.5	Nachbarschaft im Zielraum - Konzept 4	109
5.6	Symmetrie von NSZR4	112
5.7	Abweichungsanalyse von NSZR4	119

6.1	Doppelbrückenexperiment von Goss u. a. (1989)	148
6.2	Indirekter Kommunikationsmechanismus	149
6.3	Kodierungsalternativen der Pheromonmatrix	155
6.4	Aktualisierungsstrategien nach Iredi u. a. (2001)	168
6.5	SCMO - Suchschema im Zielraum	180
6.6	SCMO - Exp. 3: Trade Off: ΔSQ_v^μ - $\Delta Ticks_v^\mu$	199
6.7	MCMO1 - Übersicht der Prozeduren	210
6.8	MCMO - Suchschema im Zielraum	211
6.9	MCMO: i=0; j=6	215
6.10	MCMO: i=0; j=6; s=5; Kolonie 1 & 2	216
6.11	MCMO: i=0; j=6; s=5; Kolonie 1 & 3	216
6.12	MCMO: i=0; j=6; s=5; Kolonie 2 & 3	217
6.13	MCMO: i=0; j=6; s=4; Kolonie 1 & 2	217
6.14	MCMO: i=0; j=6; s=4; Kolonie 1 & 3	218
6.15	MCMO: i=0; j=6; s=4; Kolonie 2 & 3	218
6.16	MCMO: i=0; j=6; s=3	219
6.17	MCMO: i=1; j=4	219
6.18	MCMO: i=1; j=4; s=3; Kolonie 1 & 2	220
6.19	MCMO: i=1; j=4; s=3; Kolonie 1 & 3	220
6.20	MCMO: i=1; j=4; s=3; Kolonie 2 & 3	221
6.21	MCMO: i=2; j=2	222
6.22	MCMO2 - Übersicht der Prozeduren	229
6.23	MCMO1 - Trade Off: ΔSQ^μ - $\Delta Ticks_v^\mu$	236
6.24	MCMO2 - Trade Off: ΔSQ^μ - $\Delta Ticks_v^\mu$:	238

Tabellenverzeichnis

2.1	Klassifikation von Reihenfolgeproblemen	6
2.2	Korrelation der Ziele im betrachteten Zielsystem	20
4.1	Eigenschaften der Qualitätsmaße zur Erfassung der Distanz zur Menge aller effizienten Lösungen	79
4.2	Eigenschaften der Qualitätsmaße zur Erfassung der Variati- onsbreite der Zielfunktionswerte	83
4.3	Eigenschaften der Qualitätsmaße zur Erfassung der Vertei- lung der Elemente der NDS im Zielraum	89
4.4	Multiple Optima der Zykluszeit in TAIL001	91
5.1	NSZR4: Definition der Regionen bei $K=2$	110
5.2	NSZR4: Definition der Regionen bei $K=3$	110
5.3	Definition Region r'	111
5.4	Gemeinsame Elemente in NSZR4 und API, EX und INV . . .	115
5.5	Gemeinsame Elemente NSZR4 und FSH, BSH und DSH . . .	116
5.6	Gemeinsame Elemente in der reduzierten NSLR und NSZR4.	121
5.7	Durchschnittliche Abweichungen AVG_k^1 u. AVG_k^2	122
5.8	Größte Abweichung WC_k	123
5.9	Anforderungen an ein Distanzmaß $N=2$	127
5.10	Anforderungen an ein Distanzmaß $N=3,4,5,6,7$	128
5.11	Differenzierungstiefe Ersatzmaße $N=4$	132
5.12	Differenzierungstiefe Ersatzmaße $N=6$	132
5.13	Differenzierungstiefe Ersatzmaße $N=8$	132
5.14	Korrelationen zwischen echter Distanz und den Ersatzma- ßen, $N=4$	133
5.15	Korrelationen zwischen echter Distanz und den Ersatzma- ßen, $N=6$	133
5.16	Korrelationen zwischen echter Distanz und den Ersatzma- ßen, $N=8$	133

5.17	Korrelation zwischen echter Distanz, Ersatzmaßen und Zykluszeit	136
5.18	Korrelation zwischen echter Distanz, Ersatzmaßen und mittlerer Durchlaufzeit	137
5.19	Korrelation zwischen echter Distanz, Ersatzmaßen und maximaler Terminüberschreitung	137
5.20	Häufigkeitsverteilung der Korrelationen	138
5.21	Isolierte Analyse globaler Optima - Zykluszeit	140
5.22	Isolierte Analyse globaler Optima - Mittlere Durchlaufzeit	140
5.23	Isolierte Analyse globaler Optima - Maximale Terminüberschreitung	141
5.24	Integrierte Analyse - Alle globalen Optima	142
5.25	Integrierte Analyse - Effiziente globale Optima	144
5.26	Distanzanalyse der effizienten Menge	145
6.1	ACO Algorithmen für Scheduling Probleme	154
6.2	SCMO - Alternative Suchrichtungen im Zielraum	189
6.3	SCMO - Exp. 1: SQ -Maß	190
6.4	SCMO - Exp. 1: Variante ohne LNDS: DUB^k -Maß	191
6.5	SCMO - Exp. 1: Variante mit LNDS: DUB^k -Maß	192
6.6	SCMO - Exp. 1: Variante ohne LNDS: AEX^k -Maß	193
6.7	SCMO - Exp. 1: Variante mit LNDS: AEX^k -Maß	194
6.8	SCMO - Exp. 2: SQ -Maß	195
6.9	SCMO - Exp. 2: DUB^k -Maß	196
6.10	SCMO - Exp. 2: AEX^k -Maß	196
6.11	SCMO - Exp. 3: ΔSQ_v und $\Delta Ticks_v$	199
6.12	SCMO - Exp. 3: ΔDUB_v^k	200
6.13	SCMO - Exp. 3: ΔAEX_v^k	201
6.14	SCMO - Konfigurationen Experiment 4	202
6.15	SCMO - Exp. 4: SQ -Maß	203
6.16	SCMO - Exp. 4: DUB^k -Maß	204
6.17	SCMO - Exp. 4: AEX^k -Maß	205
6.18	SCMO - Exp. 5: SQ -Maß	206
6.19	SCMO - Exp. 5: DUB^k -Maß	207
6.20	SCMO - Exp. 5: AEX^k -Maß	208
6.21	MCMO - Exp. 1: SQ -Maß	231
6.22	MCMO1 - Exp. 1: DUB^k -Maß	232
6.23	MCMO2 - Exp. 1: DUB^k -Maß	232
6.24	MCMO1 - Exp. 1: AEX^k -Maß	233
6.25	MCMO2 - Exp. 1: AEX^k -Maß	233

6.26 MCMO1 - Exp. 2: ΔSQ_v und $\Delta Ticks_v$	235
6.27 MCMO2 - Exp. 2: SQ -Maß	237
6.28 MCMO1 - Exp. 2: ΔDUB_v^k -	239
6.29 MCMO2 - Exp. 2: ΔDUB_v^k	240
6.30 MCMO1 - Exp. 2: ΔAEX_v^k	241
6.31 MCMO2 - Exp. 2: ΔAEX_v^k	242
6.32 MCMO - Exp. 3: SQ_v	243
6.33 MCMO1 - Exp. 3: DUB^k -Maß	244
6.34 MCMO2 - Exp. 3: DUB^k -Maß	245
6.35 MCMO1 - Exp. 3: AEX^k -Maß	246
6.36 MCMO2 - Exp. 3: AEX^k -Maß	247
6.37 MCMO - Exp. 4: SQ -Maß	248
6.38 MCMO1 - Exp. 4: DUB^k -Maß	249
6.39 MCMO2 - Exp. 4: DUB^k -Maß	249
6.40 Wahl von ρ in Abhängigkeit von der Problemgröße	250
6.41 MCMO1 - Exp. 4: AEX^k -Maß	251
6.42 MCMO2 - Exp. 4: AEX^k -Maß	251
6.43 Konfigurationen von SCMO, MCMO1 und MCMO2	253
6.44 Vergleich SCMO, MCMO und MOGLS: SQ -Maß	254
6.45 Vergleich SCMO, MCMO und MOGLS: DUB^k -Maß	255
6.46 Vergleich SCMO, MCMO und MOGLS: AEX^k -Maß	256
6.47 Vergleich SCMO, MCMO und MOGLS: Struktur der Rechen- zeiten	257
6.48 Vergleich SCMO, MCMO und MOGLS: Anzahl generierter Auftragsfolgen	258
6.49 Verbesserte kleinste obere Schranken - F_{mid} : TAIL001-TAIL030260	
6.50 Verbesserte kleinste obere Schranken - F_{mid} : TAIL031-TAIL060261	
6.51 Verbesserte kleinste obere Schranken - F_{mid} : TAIL061-TAIL090262	
6.52 Verbesserte kleinste obere Schranken - T_{max} : TAIL001- TAIL030	263
6.53 Verbesserte kleinste obere Schranken - T_{max} : TAIL031- TAIL060	264
6.54 Verbesserte kleinste obere Schranken - T_{max} : TAIL061- TAIL090	265
A.1 Kleine Testprobleme	287
C.1 Parametereinstellungen MOGLS und GPWGA	291
C.2 Kleinste obere Schranken: TAIL001-TAIL030	292
C.3 Kleinste obere Schranken: TAIL031-TAIL060	293

C.4 Kleinste obere Schranken: TAIL061-TAIL090 294

Kapitel 1

Einleitung

In den letzten Jahren hat sich die Wettbewerbssituation der Unternehmen aufgrund der Globalisierung stark verändert. Die Konkurrenz zwischen den Unternehmen nimmt zu. Gleichzeitig wachsen die Ansprüche der Kunden. Letzteres äußert sich z.B. im Wunsch nach individuell angepassten Produkten. Das Erfüllen der Kundenwünsche sowie die Fähigkeit schnell und termingerecht zu liefern, sind zu einem entscheidenden Wettbewerbsfaktor geworden. Dies hat auch Auswirkungen auf das Zielsystem der Produktion. Im Gegensatz zum früher vorherrschenden Ziel der Kapazitätsauslastung, nimmt die Bedeutung von zeitorientierten Zielsetzungen, wie z.B. Fertigung nahe am Liefertermin oder kurze Durchlaufzeiten zu. Das Erreichen dieser Ziele wird unter anderem auch von der Fertigungsreihenfolge der Aufträge durch das Produktionssystem beeinflusst.

Bei mehrstufigen Produktionssystemen lassen sich die Werkstattfertigung (Job Shop) und die Reihenfertigung (Flow Shop) als Organisationsprinzip unterscheiden. In dieser Arbeit wird das sogenannte Permutation Flow Shop Problem (PFSP) betrachtet, wobei die in der Literatur vorherrschenden Annahmen zugrunde gelegt werden.¹ Permutation Flow Shop bedeutet dabei, daß alle Aufträge die gleiche Maschinenfolge besitzen (Flow Shop) und daß auf allen Maschinen die gleiche Auftragsfolge gilt (Permutation Flow Shop). Aus betriebswirtschaftlicher Sicht hat sich die Lösung des Planungsproblems² an den entscheidungsrelevanten Kosten zu orientieren. Relevant sind dabei z.B. Lager-, Rüst-, Leer- sowie Anpassungskosten. Auch Konventionalstrafen für die Überschreitung eines Liefertermins gehören dazu. Die Komponenten der entscheidungsrelevanten Kosten lassen

¹Vgl. zu diesen Annahmen Abschnitt 2.1.2.

²Ziel der Planungsaufgabe ist es, bei gegebener Maschinenfolge, die Bearbeitungsreihenfolge der Aufträge festzulegen (vgl. Seelbach (1975, S.15), Zäpfel (1982, S.247f.)).

sich jedoch nur schwer erfassen. Daher werden üblicherweise Ersatzziele verwendet, die aus dem Ziel der Minimierung der entscheidungsrelevanten Kosten abgeleitet werden. Das in der Arbeit behandelte Problem hat eine multikriterielle Zielsetzung. Das Zielsystem besteht aus den drei Zielen der Minimierung der Zykluszeit, der Minimierung der mittleren Durchlaufzeit sowie der Minimierung der maximalen Terminüberschreitung. Ziel ist die Bestimmung der Menge aller effizienter Auftragsfolgen. Eine Klassifikation von Reihenfolgeproblemen und die Charakterisierung des hier betrachteten Permutation Flow Shop Problems erfolgt ebenso wie eine Diskussion des Zielsystems in Kapitel 2.

Seit der Veröffentlichung des Johnson Algorithmus (1954)³, hat das Problem der Reihenfolgeplanung in der wissenschaftlichen Arbeit erhebliche Aufmerksamkeit erfahren. Dabei lag das Hauptaugenmerk jedoch auf Lösungsansätzen, die nur ein Ziel, wie z.B. die Minimierung der Zykluszeit, verfolgen. Dies mag auch daran liegen, daß das Problem bereits bei einer Zielgröße in der Regel NP-schwer ist. In den 70er Jahren wurden erste sehr allgemeine und theoretische Ansätze für das Reihenfolgeproblem mit mehrfacher Zielsetzung veröffentlicht.⁴ Erst in den letzten Jahren jedoch wurde das Thema wieder international aufgegriffen. Insgesamt handelt es sich beim vorliegenden Problem um ein Vektorminimierungsproblem. Eine Diskussion von Lösungsmöglichkeiten mit Hilfe von Kompromißmodellen für diese Probleme erfolgt in Kapitel 3. Die vorhandenen Ansätze zur Problemlösung lassen sich in 2 Gruppen aufteilen. In die erste Gruppe fallen Ansätze, die zum Ziel haben, lediglich eine effiziente Auftragsfolge zu ermitteln. Diese Ansätze spielen insbesondere dann eine Rolle, wenn bereits vor der Problemlösung Informationen über die Präferenzstruktur des Entscheidungsträgers vorliegen und diese daher bei der Problemlösung berücksichtigt werden können. Derartige Ansätze werden daher als a-priori Ansätze bezeichnet. Die zweite Gruppe wird von Ansätzen gebildet, die als Ziel die Ermittlung der Menge aller effizienter Auftragsfolgen haben. Da bei diesen Ansätzen die Präferenzen erst nach der Lösungsermittlung eingehen, werden die Ansätze auch als a-posteriori Ansätze bezeichnet. Der in dieser Arbeit zu entwickelnde Algorithmus gehört der zweiten Gruppe an.

Für die Ermittlung der Lösung des Problems, stehen grundsätzlich zwei Arten von Lösungsmöglichkeiten zur Verfügung. Zum einen können exakte Verfahren wie z.B. Branch and Bound verwendet werden. Diese bestimmen die Menge aller effizienten Lösungen. Die exakten Verfahren sind jedoch mit einem extrem hohen Rechenaufwand verbunden und daher für pra-

³Siehe Johnson (1954, S.64).

⁴Siehe z.B. Dinkelbach (1977, 546-564) und Dinkelbach (1982).

xisrelevante Problemgrößen nicht anwendbar. Aus diesem Grund richtet sich das Hauptaugenmerk der Forschung auf die Entwicklung von heuristischen Verfahren. Diese finden nicht zwingend die exakte Lösung. Sie können jedoch vergleichsweise schnell gute Lösungen finden. Das bedeutet, anstatt der Menge echt effizienter Punkte ermitteln Heuristiken eine Menge heuristisch effizienter Punkte⁵. Zu den am häufigsten eingesetzten Heuristiken gehören Metaheuristiken, wie Tabu Search, Simulated Annealing und Genetische Algorithmen. Zur Lösung kombinatorischer Optimierungsprobleme bei einfacher Zielsetzung hat sich in letzter Zeit eine neue Klasse von Metaheuristiken, die Ant Colony Optimization (ACO), als effektiv und effizient erwiesen. Bislang gibt es jedoch nur sehr wenige ACO-Ansätze für Reihenfolgeprobleme bei mehrfacher Zielsetzung. Eine kurze Übersicht über diese Ansätze wird am Ende von Kapitel 3 gegeben.

In den Kapiteln 4 und 5 werden vorgelagerte Aspekte der Problemlösung mit Hilfe von Heuristiken betrachtet. Diese sind unabhängig von der verwendeten ACO Metaheuristik und ebenso für die Problemlösung mit Hilfe anderer Metaheuristiken, wie z.B. Simulated Annealing oder Genetischen Algorithmen relevant.

Kapitel 4 beschäftigt sich zunächst mit der Beurteilung der Lösungsqualität von heuristisch effizienten Mengen. Die Überlegungen sind unabhängig vom betrachteten Permutation Flow Shop Problem. Beim Vergleich der Lösungsgüte ist, im multikriteriellen Fall, in der Literatur noch kein einheitliches Vorgehen erkennbar. Neben dem visuellen Vergleich, der nur sehr begrenzt bei zwei oder maximal drei Zielen anwendbar ist, definiert in der Regel jeder Autor eigene Qualitätsmaße, um einen Vergleich durchzuführen. Der Vergleich stellt keine Schwierigkeit dar, solange eine Lösung eine andere Lösung vollständig dominiert. Ist dies jedoch nicht der Fall, ist ein Vergleich mittels des Effizienzkriteriums nicht mehr möglich. Um diese Situation aufzulösen, werden in der Literatur bisher im wesentlichen zwei Ansätze verfolgt. Der erste Ansatz erweitert die Annahmen über die Präferenzen des Entscheidungsträgers und entwickelt darauf aufbauend eine skalare Maßzahl. Die Arbeit folgt dem zweiten in der Literatur verfolgten Ansatz und berücksichtigt mehrere Qualitätsaspekte, welche jeweils mit eigenen Maßen erfasst werden. Neben dem Ziel, Lösungen zu vergleichen, ist die Beschäftigung mit Qualitätsmaßen insbesondere notwendig, um die Parameter der Heuristik gut einzustellen.

Kapitel 5 untersucht Grundlagen für die Verwendung von lokalen Suchverfahren. Ein gängiges Vorgehen bei der Problemlösung ist es, den Algo-

⁵Der Begriff der heuristischen Effizienz wurde erstmals von Huckert u. a. (1980, S.59) verwendet.

rhythmus in einen guten Bereich, den Einzugsbereich der lokalen oder gar globalen Optima, zu führen. Dieses globale Lenken ist häufig primäre Aufgabe der Metaheuristik. Die Feinsuche wird anschließend zumeist speziellen lokalen Suchverfahren überlassen. Ansätze, die diese Zweiteilung von Metaheuristik und lokaler Suche aufweisen, werden auch als Hybride Ansätze bezeichnet. In diesem Zusammenhang sind folgende Fragen interessant:

Ist unter Beachtung dessen, was ein Nachbar im Lösungsraum ist, dieser auch ein Nachbar im Zielraum? Sind also zwei Sequenzen, die über einen sogenannten Nachbarschaftsoperator als Nachbarn im Lösungsraum definiert sind auch benachbarte Zielfunktionsvektoren und damit Nachbarn im Zielraum?⁶ Wie verhält sich die Lage der einzelnen lokalen Optima eines Ziels zueinander und zum globalen Optimum? Wie verhält sich die Lage der lokalen oder globalen Optima der verschiedenen verwendeten Ziele zueinander? Diese Fragen werden in unterschiedlichen Experimenten untersucht. Die Ergebnisse fließen anschließend in die zwei in Kapitel 6 entwickelten Ameisenalgorithmen ein.

Eine kurze Darstellung der ACO Metaheuristik sowie die Entwicklung von zwei Ameisenalgorithmen (SCMO- und MCMO- Ansatz) zur Lösung des vorliegenden Problems erfolgt in Kapitel 6. In einer Reihe von Experimenten werden unterschiedliche Strategien zum Absuchen des Zielraums sowie weitere Einstellungsmöglichkeiten untersucht.

⁶Zur Definition der Nachbarschaft im Zielraum vgl. Abschnitt 5.3.

Kapitel 2

Charakterisierung des Permutation Flow Shop Problems

2.1 Grundlagen und Prämissen

2.1.1 Klassifikation von Reihenfolgeproblemen

Ein Auftrag wird durch eine Folge von technologisch zusammenhängenden Arbeitsgängen beschrieben, die auszuführen sind, um eine einzelne Produkteinheit oder mehrere identische Produkteinheiten herzustellen. Jeder Auftrag besitzt eine Maschinenfolge, die bestimmt, in welcher zeitlichen Reihenfolge die notwendigen Arbeitsgänge auf den einzelnen Maschinen durchgeführt werden. Eine Maschine ist eine Ressource, welche einen bestimmten Arbeitsgang ausführt. Ist die Maschinenfolge bei mindestens zwei Aufträgen unterschiedlich, so liegt eine Werkstattfertigung vor, ist sie hingegen bei allen Aufträgen gleich, so liegt eine Reihenfertigung vor.¹ In der englischsprachigen Literatur wird für Werkstattfertigung der Begriff Job Shop und für Reihenfertigung der Begriff Flow Shop verwendet.² Dabei ist jedoch zu beachten, daß im Falle des Job Shops und Flow Shops als weitere Annahme gilt, daß sich auf jeder Fertigungsstufe nur eine Maschine gleicher Art befindet.³ Umfaßt eine Fertigungsstufe mehrere gleichartige Maschinen, wird das Problem als General Job Shop bzw. Flexible Flow

¹Vgl. zu den bisherigen Ausführungen Seelbach (1975, S.14f.).

²Vgl. Seelbach (1975, S.15).

³Vgl. Pinedo (1995, S.93f.).

Shop bezeichnet.⁴ Besteht die Maschinenfolge lediglich aus einer Maschine, so handelt es sich um einstufige Fertigung. Umfasst die Fertigungsstufe mehrere gleichartige Maschinen, so wird von einem Problem mit parallelen Maschinen gesprochen. Bei lediglich einer Maschine auf der Fertigungsstufe spricht man vom Single-Machine-Problem. Tabelle 2.1 fasst die Klassifikation der Reihenfolgeprobleme nochmals zusammen.

Fertigung	Maschinenfolge der Aufträge	Gleichartige Maschinen pro Fert.Stufe	
		mehrere	nur eine
einstufig	-	Parallel Machines	Single Machine
mehrstufig	unterschiedlich	General Job Shop	Job Shop
	gleich	Flexible Flow Shop	Flow Shop

Tabelle 2.1: Klassifikation von Reihenfolgeproblemen

Reihenfolgeprobleme können weiterhin nach folgenden Aspekten unterschieden werden:

Statisch / dynamisch: Liegen alle in der Planung zu berücksichtigenden Aufträge zum Planungszeitpunkt vor, so handelt es sich um ein statisches Problem. Treffen im Zeitablauf weitere zu berücksichtigende Aufträge ein, so ist das Problem dynamisch.⁵

Deterministisch / stochastisch: Als deterministisch wird ein Problem bezeichnet, falls zum Planungszeitpunkt alle entscheidungsrelevanten Informationen (Bearbeitungszeiten, Liefertermine usw.) über die Aufträge bekannt und sicher sind.⁶ Ist zum Planungszeitpunkt mindestens ein Datum nicht bekannt oder noch unsicher, so ist das Problem stochastisch.⁷

Planungsaufgabe ist es, bei gegebener Maschinenfolge die organisatorische Folge, also die Bearbeitungsreihenfolge der Aufträge auf den einzelnen Maschinen festzulegen.⁸ Im Fall des Flow Shops kann die Auftragsfolge auf den einzelnen Maschinen grundsätzlich unterschiedlich sein. Die Auftragsfolgen auf den einzelnen Maschinen bilden dann den Ablaufplan.⁹ Die Lö-

⁴Vgl. Pinedo (1995, S.94).

⁵Vgl. Seelbach (1975, S.16), Conway u. a. (1967, S.7).

⁶Die alleinige Kenntniss einer Wahrscheinlichkeitsverteilung z.B. für die Bearbeitungszeiten genügt nicht, damit das Problem als deterministisch bezeichnet werden kann. Es liegt zwar eine Information über die Bearbeitungszeiten vor, die genauen Ausprägungen sind aber weiterhin unsicher.

⁷Vgl. Conway u. a. (1967, S.7).

⁸Vgl. Seelbach (1975, S.15), Zäpfel (1982, S.247f.).

⁹Vgl. Seelbach (1975, S.16).

sung des Problems ist jedoch deutlich einfacher, wenn man die Annahme trifft, daß die Auftragsfolge auf allen Maschinen gleich ist.¹⁰ Sicherlich auch deswegen ist es eine in der Literatur übliche Annahme und ein in der Praxis anzutreffendes Vorgehen, auf allen Maschinen die gleiche Auftragsfolge zu verwenden.¹¹ Das daraus resultierende Planungsproblem wird als Permutation Flow Shop Problem (PFSP) bezeichnet.¹² Die Annahme der gleichen Auftragsfolge auf allen Maschinen schränkt den Lösungsraum für das Flow Shop Problem ein. Die Anzahl der Ablaufpläne beträgt beim Permutation Flow Shop Problem $N!$, wobei N die Anzahl der Aufträge ist. Beim Flow Shop Problem sind hingegen $N!^M$ unterschiedliche Ablaufpläne möglich, mit M als Anzahl der Maschinen im Produktionssystem. Aus der Einschränkung resultiert, daß eine optimale Lösung des Permutation Flow Shop Problems nicht zwingend eine optimale Lösung für das Flow Shop Problem ist.

Neben der Anzahl der Aufträge N und der Anzahl der Maschinen M sollen im Folgenden weitere, in dieser Arbeit verwendete Begriffe und Variablen definiert werden:

Ankunftszeitpunkt eines Auftrags n (r_n): Der Zeitpunkt, zu dem der Auftrag im Fertigungssystem bereitsteht und die Bearbeitung beginnen kann.

Bearbeitungszeit eines Auftrags n auf Maschine m (p_{nm}): Die Zeitspanne, die ein Auftrag eine Maschine beansprucht. Unter der gesamten Bearbeitungszeit eines Auftrags (P_n), wird die Summe der Bearbeitungszeiten eines Auftrags über alle Maschinen verstanden.

Liefertermin /Spätestzulässiger Fertigstellungstermin (d_n): Der Liefertermin ist der Zeitpunkt, zu dem der Auftrag abgeliefert werden muß. Kann ein Auftrag vor diesem Termin abgeliefert werden, muß er aber spätestens zu diesem Termin abgeliefert werden, so erscheint es angebracht, den Begriff des Liefertermins durch den Begriff des spätestzulässigen Fertigstellungstermins zu ersetzen.

¹⁰Vgl. Pinedo (1995, S.94).

¹¹Vgl. Rajendran u. Ziegler (1997, S.569).

¹²Vgl. Pinedo (1995, S.94).

2.1.2 Spezifikation des in der Arbeit betrachteten Permutation Flow Shop Problems

Im Folgenden soll das in dieser Arbeit betrachtete Problem spezifiziert werden. Um die Vergleichbarkeit mit den in der Literatur veröffentlichten Ansätzen und Ergebnissen zu gewährleisten, werden die dort üblichen Annahmen weitgehend übernommen.¹³

1. Ein Produktionsprogramm mit N Aufträgen wird auf M unabhängigen Maschinen gefertigt.
2. Auf jeder Fertigungsstufe gibt es nur eine Maschine und alle $n = 1(1)N$ Aufträge werden mit der gleichen Maschinenfolge $m = 1(1)M$ gefertigt (Flow Shop).
3. Die Auftragsfolge π ist auf allen $m = 1(1)M$ Maschinen gleich (Permutation Flow Shop).
4. Das Unterbrechen der Bearbeitung eines Auftrags ist nicht zulässig (kein Lossplitting).
5. Ein Auftrag kann zu einem Zeitpunkt immer nur auf einer Maschine bearbeitet werden (geschlossener Lostransport).
6. Eine Maschine kann zu einem Zeitpunkt immer nur einen Auftrag bearbeiten (keine Parallelproduktion).
7. Die Bearbeitungszeiten und spätestzulässigen Fertigstellungstermine aller Aufträge sind vorgegeben und bekannt und das Problem ist damit deterministisch (keine Schwankungen der Bearbeitungszeiten, keine intensitätsmäßige Anpassungen möglich).
8. Die Bearbeitungszeiten sind unabhängig von der Auftragsfolge.
9. Die Rüstzeiten sind reihenfolgeunabhängig, in den Bearbeitungszeiten enthalten und damit nicht entscheidungsrelevant.
10. Die Transportzeiten zwischen den Maschinen betragen Null Zeiteinheiten.
11. Die Zwischenlagerkapazitäten sind unbegrenzt.
12. Die Maschinen stehen ohne Unterbrechung während des gesamten Planungszeitraums zur Verfügung (keine wartungs- und störungsbedingten Unterbrechungen).

¹³Vgl. zu den Prämissen 1 bis 13 auch Seelbach (1975, S.16-18).

13. Die im Produktionsprogramm enthaltenen Aufträge sind alle zu Beginn des Planungszeitraums (d.h. zum Zeitpunkt Null) verfügbar. D.h. es gilt: $r_n = 0$, $n = 1(1)N$ (statisches Reihenfolgeproblem).
14. Nach Abschluß der Fertigung der N Aufträge wird ein neues Produktionsprogramm in das Fertigungssystem eingeplant.

Es ist die Menge aller Auftragsfolgen zu ermitteln, deren Zielfunktionsvektoren die effiziente Front bezüglich eines gegebenen Zielsystems im Zielraum bestimmen, diese Menge wird als die Menge aller effizienten Lösungen bezeichnet.¹⁴

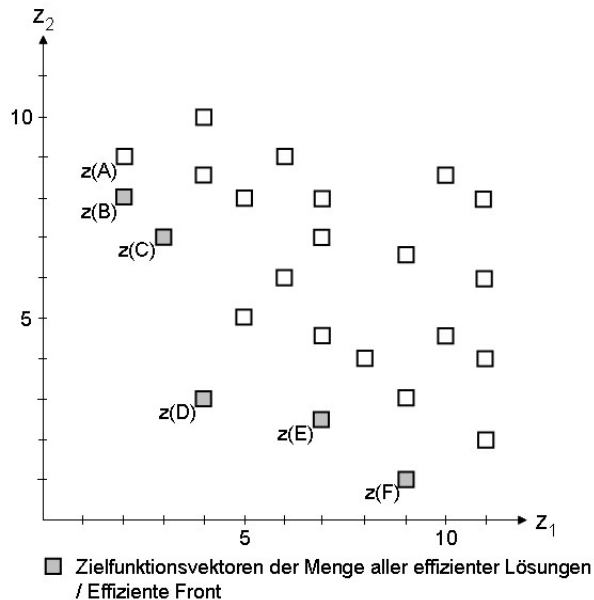


Abbildung 2.1: Menge aller effizienten Lösungen

Abbildung 2.1 zeigt die Zielfunktionsvektoren der $4! = 24$ möglichen Auftragsfolgen eines Permutation Flow Shop Problems mit $N = 4$ Aufträgen. Gesucht sind die Auftragsfolgen deren Zielfunktionsvektoren die effiziente Front bezüglich des aus zwei Zielen bestehenden Zielsystems (z_1, z_2) bilden. Es sind dies die Auftragsfolgen B, C, D, E und F.

Auch wenn der einzelne Planungslauf als statisches und deterministisches Reihenfolgeproblem betrachtet werden kann, so kann das gesamte Fertigungssystem aufgrund von Annahme 14 als dynamisch angesehen

¹⁴Vgl. zur Definition der Menge aller effizienter Lösungen und der effizienten Front Abschnitt 3.1.

werden. Während der Fertigung des aktuellen Produktionsprogramms treffen weitere Aufträge ein, die aber erst nach Abschluß der Fertigung des aktuellen Programms zur Fertigung freigegeben werden. Das insgesamt dynamische Problem wird dann in einzelne statische Probleme zerlegt.

Das betrachtete Permutation Flow Shop Problem gehört zur Klasse der NP-schweren Probleme. Ein Problem wird als NP-schwer bezeichnet, wenn kein Algorithmus zu seiner Lösung bekannt ist, dessen Rechenaufwand durch ein Polynom der Problemgröße nach oben beschränkt ist. Steigt der Rechenaufwand polynomial an, so gehört das Problem der Klasse P, der Klasse der effizient lösbaren Probleme an. Die Bestimmung einer Auftragsfolge, welche die Zykluszeit in einem Permutation Flow Shop minimiert, ist im Fall $M \geq 3$ NP-schwer.¹⁵ Da die Lösung dieses Problems Element der Menge aller effizienten Lösungen ist, ist auch das Problem der Ermittlung dieser Menge aller effizienten Lösungen NP-schwer. Wäre das Problem nicht NP-schwer, so gäbe es auch einen polynomialen Algorithmus zur Bestimmung von Auftragsfolgen mit minimaler Zykluszeit bzw. minimaler Durchlaufzeit.

Im folgenden Abschnitt, soll aufbauend auf die soeben getroffenen Annahmen das betrachtete Zielsystem spezifiziert werden.

2.2 Das betrachtete Zielsystem

2.2.1 Ziele für das Permutation Flow Shop Problem

2.2.1.1 Die Gewinnmaximierung als Beurteilungsmaßstab

Aus betriebswirtschaftlicher Sicht hat sich die Lösung des vorliegenden Planungsproblems am erwerbswirtschaftlichen Prinzip zu orientieren. Dieses fordert, daß der Gewinn zu maximieren ist. Auf die Erlöse hat die Reihenfolgeentscheidung kurzfristig keinen Einfluß, da diese durch die vorgelegte Absatz- und Produktionsprogrammplanung vorgegeben sind.¹⁶ Daher hat sich die Reihenfolgeentscheidung primär an der Minimierung der entscheidungsrelevanten Kosten zu orientieren. Als entscheidungsrelevante Kosten können Lager-, Rüst- sowie Terminabweichungskosten angesehen werden. Wie sich zeigen wird, können die häufig diskutierten Leerzeitkosten nicht den entscheidungsrelevanten Kosten zugeordnet werden. Die Erfassung der entscheidungsrelevanten Kosten ist in der Praxis mit

¹⁵Vgl. Pinedo (1995, S.101f.). Lediglich im Fall von nur 2 Maschinen kann das Problem mit Hilfe des Johnson-Algorithmus in polynomialer Zeit gelöst werden (vgl. Johnson (1954, S.64); Pinedo (1995, S.97f.)).

¹⁶Vgl. Seelbach (1975, S.32).

Schwierigkeiten behaftet,¹⁷ so daß als Ersatzziele häufig Zeitziele verwendet werden. Bei der Verfolgung dieser Ersatzziele wird unterstellt, daß sie die eigentlichen, monetären Ziele positiv beeinflussen.

Neben der Minimierung der entscheidungsrelevanten Kosten kann es laut Dinkelbach (1962) durchaus sinnvoll sein, weitere Ziele zu verfolgen, die kurzfristig keine, jedoch langfristig durchaus eine Wirkung auf den Gewinn entfalten können.¹⁸

Zunächst soll in den Abschnitten 2.2.1.2 und 2.2.1.3 die Erfassung der Lager- und Terminabweichungskosten dargestellt werden. Da die Rüstzeiten in dieser Arbeit reihenfolgeunabhängig sind, sind die dazugehörigen Rüstkosten ebenfalls reihenfolgeunabhängig und damit nicht entscheidungsrelevant. Aus diesem Grund werden sie in dieser Arbeit nicht weiter diskutiert. Im Anschluß an die entscheidungsrelevanten Kosten sollen dann die Leerzeiten und die in der Literatur häufig verwendete Zykluszeit als Zielkriterium diskutiert werden.¹⁹

2.2.1.2 Lagerkosten des Fertigungsbestandes

Die Lagerkosten der Aufträge des Produktionsprogramms setzen sich zusammen aus den Kapitalbindungskosten, den Lagerraumkosten sowie weiteren Lagerhaltungskosten, welche z.B. durch die Materialhandhabung entstehen. Als Kapitalbindungskosten werden hier die Zinsen auf das Kapital verstanden, welches in den Aufträgen des Produktionsprogramms gebunden ist.²⁰ Das gebundene Kapital ist wiederum abhängig von der wertmäßigen Höhe des Auftragsbestandes, dem Kapitalkostensatz sowie der Kapitalbindungsdauer. Nach Seelbach u. Zimmermann (1973) setzt sich in einem Produktionsbetrieb die Kapitalbindungsdauer aus folgenden Komponenten zusammen: "den [...] Zahlungsfristen, der Lagerdauer der Produktionsfaktoren, der Produktionsdauer einschließlich etwaiger Zwischenlagerungen und der Lagerdauer der Fertigerzeugnisse [...]."²¹ Die Zahlungsfristen²² sowie die Lagerdauer der Werkstoffe bis zum

¹⁷Neben der Ermittlung der Kostengrößen stellt auch das auszuwertende Datenvolumen einen Grund für die Verwendung von Ersatzmaßen da (vgl. Stefenelli (2000, S.7)).

¹⁸Vgl. Dinkelbach (1962, S.739). Vor diesem Hintergrund wird die Minimierung der Zykluszeit in Abschnitt 2.2.1.4 auf Seite 16 diskutiert und in das in dieser Arbeit betrachtete Zielsystem aufgenommen.

¹⁹Eine ausführliche Diskussion entscheidungsrelevanter Kosten der Reihenfolgeplanung findet sich in Stefenelli (2000) sowie der dort angegebenen Literatur.

²⁰Vgl. hierzu und den folgenden Satz Seelbach (1975, S.33).

²¹Seelbach u. Zimmermann (1973, S.338). Problematisch ist der Begriff des Produktionsfaktors, zu denen streng genommen z.B. auch Arbeitskräfte gehören. Diese werden jedoch nicht gelagert. Aus diesem Grund wird im Folgenden von der Lagerdauer der Werkstoffe gesprochen.

²²Zahlungsfristen meint hier sowohl die vom Unternehmen einzuhaltenden Zahlungsfristen für fremdbeschafte Einsatzmaterialien, als auch die dem Kunden gewährten Zahlungsfristen

Planungszeitpunkt sind durch die Reihenfolgeplanung nicht beeinflussbar, wie auch die in dem hier betrachteten Modell als reihenfolgeunabhängig angenommenen Bearbeitungszeiten.

Nach Günther (1971) kann sich die ablaufbedingte Wartezeit eines Auftrags aus der Vorlagerzeit, den Zwischenlagerzeiten sowie der Nachlagerzeit zusammensetzen.²³ Die Vorlagerzeit des Auftrags ist die Zeit, die vom Ankunftszeitpunkt in der Fertigung bis zum Beginn der Bearbeitung auf der ersten Maschine vergeht. Die Zwischenlagerzeiten sind die Zeiten, die ein Auftrag nach dem Abschluß der Bearbeitung auf einer Maschine bis zum Beginn der Bearbeitung auf der nächsten Maschine warten muß. Die Nachlagerzeit bezeichnet die Zeitspanne von der Fertigstellung des Auftrags bis zu seiner Auslieferung.²⁴ Seelbach (1975) zählt die Nachlagerzeiten nicht zu den Wartezeiten eines Auftrags²⁵. Werden die Aufträge bis zu einem Auslieferungstermin gelagert, so können die Kapitalbindungskosten der Nachlagerzeit als Terminunterschreitungskosten interpretiert werden. Die Kosten über die Terminunterschreitung zu erfassen macht auch deswegen Sinn, da die gefertigten Aufträge im Allgemeinen nicht mehr zum Fertigungsbestand gezählt werden. In dieser Arbeit betrachteten Problem beträgt die Nachlagerzeit immer Null, da alle Aufträge unmittelbar nach ihrer Fertigstellung ausgeliefert werden (vgl. Annahme 7). Somit bleiben die Wartezeiten der Aufträge vor den einzelnen Maschinen als einzige von der Reihenfolgeplanung zu beeinflussende Komponente der Lagerkosten des Fertigungsbestandes. Die Summe der Wartezeiten aller Aufträge ergibt die Gesamtwartezeit.²⁶

Der Erfassung der Lagerkosten stellen sich nun im wesentlichen zwei Erfassungsprobleme entgegen. Das erste Problem besteht in der "richtigen" Ermittlung des Kapitalkostensatzes.²⁷ Auch die Erfassung der Kostensätze je Zeiteinheit für die einzelnen Aufträge ist nicht unproblematisch.²⁸ Aus diesen Gründen wird, unter der Annahme gleicher, konstanter Kostensätze für alle Aufträge, in Lösungsansätzen für das Permutation Flowshop Problem häufig die Minimierung der Summe der Wartezeiten als Ziel verfolgt. Mit sinkenden Wartezeiten sind auch sinkende Lagerkosten zu erwarten.

für verkaufte Erzeugnisse und Materialien.

²³Vgl. Günther (1971, S.29). Die Aufnahme der unterschiedlichen Lagerzeiten in die ablaufbedingte Wartezeit wird bei Günther (1971, S.29f.) diskutiert.

²⁴Vgl. zu den Definitionen Seelbach (1975, S.14) und Günther (1971, S.29). Keine der beiden Arbeiten definiert die Vorlager- und Zwischenlagerzeit eindeutig. Die hier angegebene Definition erscheint jedoch aus dem Kontext heraus schlüssig.

²⁵Vgl. Seelbach (1975, S.14).

²⁶Vgl. Seelbach (1975, S.14).

²⁷Vgl. Seelbach (1975, S.34).

²⁸Vgl. Seelbach (1975, S.34).

Die Wartezeit eines Auftrags n (W_n) ergibt sich als die Summe seiner Wartezeiten vor den einzelnen Maschinen w_{nm} :

$$W_n = \sum_{m=1}^M w_{nm} \quad (2.1)$$

Die Summe der Wartezeiten aller N Aufträge (W_{sum}) beträgt damit:

$$W_{sum} = \sum_{n=1}^N W_n = \sum_{n=1}^N \sum_{m=1}^M w_{nm} \quad (2.2)$$

Aufgrund der reihenfolgeunabhängigen Bearbeitungszeiten eines Auftrags entspricht die Minimierung der Wartezeit eines Auftrags (W_n) der Minimierung seiner Durchlaufzeit (F_n):

$$F_n = \sum_{m=1}^M (p_{nm} + w_{nm}) = P_n + W_n \quad (2.3)$$

Die Durchlaufzeit eines Auftrags ist hier also definiert als die Zeitspanne, welche vom Eintreffen des Auftrags in der Produktion bis zur Beendigung des Arbeitsgangs auf der letzten Maschine vergeht.

Die Gesamtdurchlaufzeit (F_{sum}) ergibt sich somit zu:²⁹

$$F_{sum} = \sum_{n=1}^N F_n = \sum_{n=1}^N \sum_{m=1}^M (p_{nm} + w_{nm}) \quad (2.4)$$

Da die Anzahl der Aufträge N ebenfalls konstant ist, ist die Minimierung der mittleren Durchlaufzeit (F_{mid}) ebenfalls äquivalent zu den letztgenannten Zielen.

$$F_{mid} = \frac{1}{N} \sum_{n=1}^N F_n \quad (2.5)$$

Die Durchlaufzeit eines Auftrags ergibt sich alternativ aus dem Fertigstellungszeitpunkt des Auftrags (C_n) abzüglich seines Ankunftszeitpunktes: $F_n = C_n - r_n$. Da der Ankunftszeitpunkt für alle Aufträge annahmegemäß³⁰ Null beträgt, entsprechen sich der Fertigstellungszeitpunkt eines Auftrags und seine Durchlaufzeit. In dieser Arbeit wird die mittlere Durchlaufzeit in das Zielsystem aufgenommen werden.

²⁹Aufgrund der reihenfolgeunabhängigen Bearbeitungszeiten eines Auftrags entspricht die Minimierung der Gesamtdurchlaufzeit (F_{sum}) der Minimierung der Summe der Wartezeiten aller N Aufträge (W_{sum}).

³⁰Vgl. Annahme 13.

2.2.1.3 Terminabweichungskosten

Kosten der Terminunterschreitung können entstehen, falls ein Auftrag vor seinem Liefertermin fertiggestellt ist, d.h. es gilt: $C_n < d_n$. Damit wird das von den Wartezeiten nicht erfaßte Zeitintervall vom Abschluß des letzten Bearbeitungsschritts bis zur Auslieferung erfaßt. In dem hier betrachteten Problem wird angenommen, daß die Aufträge sofort nach ihrer Fertigstellung ausgeliefert werden,³¹ so daß die Aufnahme eines Ziels zur Berücksichtigung dieser Lagerkosten in das Zielsystem entfallen kann.

Kosten aufgrund einer Terminüberschreitung ($C_n > d_n$) können aufgrund von Konventionalstrafen, notwendig gewordenen Preisnachlässen oder Warterabatten entstehen.³² Auch Goodwill-Verluste z.B. in Form von entgangenen Folgeaufträgen können als Terminüberschreitungskosten angesehen werden.³³ Sind Konventionalstrafen, Preisnachlässe und Warterabatte durchaus kostenmäßig quantifizierbar,³⁴ so stellt sich bei den Goodwill-Verlusten erneut ein Erfassungs- und Bewertungsproblem. So ist es einfach nicht möglich exakt vorherzusagen, welche Auswirkungen eine verspätete Lieferung auf den Ruf des Unternehmens als zuverlässiger Lieferant hat und ob und wieviele zukünftige Aufträge daher für das Unternehmen verloren gehen. Bei den Terminüberschreitungen werden im wesentlichen folgende Zielgrößen verwendet, welche ebenfalls zu minimieren sind:

Summe der Terminüberschreitungen:

$$T_{sum} = \sum_{n=1}^N \max\{0; C_n - d_n\} \quad (2.6)$$

Mittlere Terminüberschreitung:³⁵

$$T_{mid} = \frac{1}{N} \sum_{n=1}^N \max\{0; C_n - d_n\} \quad (2.7)$$

Maximale Terminüberschreitung:

$$T_{max} = \max_{n=1(1)N} \{0; C_n - d_n\} \quad (2.8)$$

³¹Vgl. Annahme 7.

³²Vgl. Seelbach (1975, S.37), Blohm u. a. (1988, S.273f.).

³³Vgl. Blohm u. a. (1988, S.274).

³⁴So sind z.B. Konventionalstrafen in der Regel Bestandteil des Lieferantenvertrages mit dem Kunden (vgl. Stefenelli (2000, S.25).

³⁵Da in dem in dieser Arbeit betrachteten Permutation Flow Shop Problem die Anzahl der Aufträge N konstant ist, sind die Summe der Terminüberschreitungen und die Mittlere Terminüberschreitung äquivalente Zielgrößen.

Anzahl der terminüberschreitenden Aufträge:

$$N_T = |\{n \mid (C_n - d_n) > 0; n = 1(1)N\}| \quad (2.9)$$

Die Summe der Terminüberschreitungen sowie die mittlere Terminüberschreitung minimieren tendenziell die durch die Verspätung entstandenen Kosten, falls diese mit dem Ausmaß der Terminüberschreitung pro Zeiteinheit unverändert bleiben, also ein linearer Kostenverlauf vorliegt.³⁶ Die Minimierung der maximalen Terminüberschreitung wird als Ziel bevorzugt, falls die Kosten je Zeiteinheit Terminüberschreitung mit zunehmender Terminüberschreitung ansteigen.³⁷ In dieser Arbeit wird die maximale Terminüberschreitung verwendet. Entstehen für jeden verspäteten Auftrag fixe Verspätungskosten, welche unabhängig vom Ausmaß der Verspätung sind, so ist die Anzahl der terminüberschreitenden Aufträge eine geeignete Zielgröße.

2.2.1.4 Leerzeitkosten und Kapazitätsauslastung

Die Minimierung des maximalen Fertigstellungszeitpunktes (Zykluszeit) ist das wohl am häufigsten verfolgte Ziel in Arbeiten, welche das Permutation Flow Shop Problem betrachten.³⁸ Die Zykluszeit ist wie folgt definiert:

$$C_{max} = \max_{n=1(1)N} \{C_n\} \quad (2.10)$$

Als Begründung für ihre Wahl wird häufig angeführt, die Kapazitätsauslastung sei zu maximieren, bzw. die Leerzeiten der Maschinen zu minimieren und damit letztendlich die Leerzeitkosten.

Die Auslastung einer Maschine errechnet sich als Quotient aus der Summe der Bearbeitungszeiten und der Zeit, welche die Maschine zur Verfügung steht. Im deterministischen Fall ist die Summe der Bearbeitungszeiten eine Konstante. Im statischen Fall kann die Annahme getroffen werden, daß die Maschinen solange zur Verfügung stehen, bis der letzte Bearbeitungsschritt des letzten Auftrags abgeschlossen ist. Da alle Aufträge zum Planungszeitpunkt Null zur Verfügung stehen entspricht dieser Zeitraum der Zykluszeit. Die durchschnittliche Kapazitätsauslastung des Fertigungsbereichs ergibt sich im statischen Fall damit zu:³⁹

³⁶Vgl. Holthaus (1996, S.12), sowie Paulik (1984, S.141).

³⁷Vgl. Holthaus (1996, S.12).

³⁸Vgl. Pinedo (1995, S.104) sowie Zäpfel (1982, S.249).

³⁹Vgl. zu diesen Ausführungen und zu einer Diskussion der Kapazitätsauslastung im dynamischen Fall Conway u. a. (1967, S.14).

$$\bar{U} = \frac{\sum_{n=1}^N \sum_{m=1}^M p_{nm}}{M \cdot C_{max}} \quad (2.11)$$

Die Minimierung der Zykluszeit führt aufgrund der konstanten Bearbeitungszeiten zur einer Maximierung von \bar{U} . Ebenso führt sie zu einer Minimierung der Leerzeiten. Die Leerzeit auf Maschine m errechnet sich als:

$$I_m = C_{max} - \sum_{n=1}^N p_{nm} \quad (2.12)$$

Die durchschnittliche Maschinenleerzeit ergibt sich dann zu:

$$\bar{I} = \frac{1}{M} \left(\sum_{m=1}^M C_{max} - \sum_{m=1}^M \sum_{n=1}^N p_{nm} \right) = C_{max} - \frac{1}{M} \sum_{m=1}^M \sum_{n=1}^N p_{nm} \quad (2.13)$$

Ist das Problem in der Realität grundsätzlich dynamisch, so wird in dieser Arbeit ein daraus entwickeltes statisches Problem behandelt.⁴⁰ Die mit den Leerzeiten assoziierten Kosten stellen keine kurzfristigen entscheidungsrelevanten Kosten dar.⁴¹ Aus der Sicht der operativen Planungsaufgabe, eine Auftragsfolge zu bestimmen, stellen die Kosten der Kapazitätsbereitstellung fixe Kosten dar.⁴² Es ist fraglich, ob die Leerzeiten mit Opportunitätskosten bewertet werden können. Eine schlechte Ausnutzung der Kapazitäten kann zu einer verspäteten Fertigstellung von Aufträgen führen.⁴³ Diese Verspätungskosten sind jedoch bereits in den in Abschnitt 2.2.1.3 diskutierten Terminüberschreitungskosten erfaßt.

Dennoch kann es sinnvoll sein, die Zykluszeit als weiteres Ziel zu betrachten. Sie stellt sicher, daß möglichst früh ein neuer Block von Aufträgen in das Fertigungssystem eingeplant werden kann, was die Flexibilität in der Zukunft erhöht. Weiterhin kann ein zu großzügiger Umgang mit den Kapazitäten dazu führen, daß Aufträge in der Zukunft aus Kapazitätsmangel abgelehnt werden müssen. Diese eventuell entgangenen Gewinne können durchaus als Opportunitätskosten betrachtet werden. Jedoch sind sie weder sicher, noch betreffen sie die aktuelle Planungsperiode. Des weiteren wird das Risiko reduziert, daß zukünftige Aufträge aufgrund zu geringer Kapazität zu spät fertiggestellt werden und damit Terminüberschreitungskosten in zukünftigen Planungszeiträumen entstehen. Schließ-

⁴⁰Vgl. dazu auch Abschnitt 2.1.2.

⁴¹Vgl. Seelbach (1975, S.35).

⁴²Vgl. Seelbach (1975, S.35).

⁴³Vgl. Seelbach (1975, S.36).

lich liefert die Zykluszeit Informationen darüber, welche Zeit mindestens benötigt wird, um die Aufträge in einer gegebenen Auftragsfolge zu fertigen. Die Zykluszeit läßt sich daher nicht über eine Kostengröße begründen, ihre Aufnahme in das Zielsystem als zu minimierende Größe kann jedoch in ihrer Funktion als Steuer- und Kontrollparameter vertreten werden.⁴⁴

2.2.1.5 Zielsystem und Berechnung der Fertigstellungszeitpunkte der Aufträge

Das in dieser Arbeit betrachtete Zielsystem besteht aus der mittleren Durchlaufzeit, der maximalen Terminüberschreitung sowie der Zykluszeit. Alle drei Ziele sind zu minimieren, so daß sich mit π als Auftragsfolge und Π als Menge aller möglichen Auftragsfolgen (Lösungsraum), das Problem wie folgt formulieren läßt.

$$MIN! z(\pi) = \begin{pmatrix} F_{mid}(\pi) \\ C_{max}(\pi) \\ T_{max}(\pi) \end{pmatrix}, \pi \in \Pi \quad (2.14)$$

Damit liegt ein diskretes Vektorminimierungsproblem vor.⁴⁵

Ermittelt werden können die Fertigstellungszeitpunkte der Aufträge und damit auch die Zykluszeit mit Hilfe folgender rekursiver Gleichungen, wobei C_{nm} für den Fertigstellungszeitpunkt von Auftrag n auf Maschine m und $\pi_{[h]}$ für den Auftrag auf Position h in der Auftragsfolge π steht:⁴⁶

1. Bestimmen der Fertigstellungszeitpunkte des ersten Auftrags auf den einzelnen Maschinen:

$$C_{\pi_{[1]}m} = \sum_{u=1}^m p_{\pi_{[1]}u}; m = 1(1)M \quad (2.15)$$

2. Bestimmen der Fertigstellungszeitpunkte der Aufträge $h \geq 2$ auf der ersten Maschine:

$$C_{\pi_{[h]}1} = \sum_{v=1}^h p_{\pi_{[v]}1}; h = 2(1)N \quad (2.16)$$

⁴⁴Vgl. Seelbach (1975, S.36), Dinkelbach (1962, S.739).

⁴⁵Diskrete Vektoroptimierungsprobleme sowie Ihre Lösung mit Hilfe von Kompromißmodellen werden in Kapitel 3 näher betrachtet.

⁴⁶Vgl. Pinedo (1995, S.94).

3. Bestimmen der restlichen Fertigstellungszeitpunkte:

$$C_{\pi_{[h]}m} = \max \{C_{\pi_{[h-1]}m}; C_{\pi_{[h]},m-1}\} + p_{\pi_{[h]}m}; \quad (2.17)$$

$$h = 2(1)N; m = 2(1)M$$

2.2.2 Analyse der Zielbeziehungen

Bisher wurden die Ziele lediglich aufgrund ihres Einflusses auf bestimmte Komponenten der entscheidungsrelevanten Kosten bzw. hinsichtlich ihrer Steuer- und Kontrollfunktion beurteilt und in das Zielsystem aufgenommen. Eine Analyse der drei im Zielsystem enthaltenen Ziele dahingehend, ob sie sich komplementär, neutral oder konfliktionär zueinander verhalten ist bislang nicht erfolgt.

Komplementäre Ziele: Zwei Ziele verhalten sich zueinander komplementär⁴⁷, wenn beim Sinken des Zielfunktionswertes des einen Ziels auch der Zielfunktionswert des anderen Ziels sinkt. Die individuell optimalen Lösungen der beiden Ziele stimmen daher überein.⁴⁸

Konkurrierende Ziele: Zwei Ziele verhalten sich konkurrierend⁴⁹, wenn das Sinken des Zielfunktionswertes des einen Ziels einhergeht mit dem Steigen des anderen Zielfunktionswertes. Die individuell optimalen Lösungen der Ziele stimmen demnach nicht überein.⁵⁰

Unabhängige Ziele: Zwei Ziele sind unabhängig,⁵¹ wenn zwischen der Entwicklung der Zielfunktionswerte der Ziele kein Zusammenhang besteht.⁵²

Es ist weiterhin zu unterscheiden, ob die Zielbeziehung über den gesamten Lösungsraum vorliegt oder ob sie in verschiedenen Teilen des Lösungsraums unterschiedlich ausgeprägt ist. In diesem letzten Fall würden dann bereichsweise Zielbeziehungen vorliegen.

Bereits Gutenberg (1951) hat unter dem Begriff des "Dilemmas der Arbeitsablaufplanung" den Konflikt zwischen dem Ziel der Minimierung der

⁴⁷Diese Ziele werden mitunter auch als kumulative Ziele oder gleichgerichtete Ziele bezeichnet.

⁴⁸Vgl. Dinkelbach (1982, S.158f.). Diese und die folgenden beiden Definitionen beziehen sich jeweils auf Minimierungsprobleme.

⁴⁹Statt konkurrierend werden auch die Begriffe konfliktionär oder gegenläufig verwendet.

⁵⁰Vgl. Dinkelbach (1982, S.158f.).

⁵¹Statt unabhängig werden auch die Begriffe neutral oder indifferent verwendet.

⁵²Vgl. Dinkelbach (1982, S.158f.).

Wartezeiten der Aufträge und dem Ziel der Minimierung der Leerzeiten der Maschinen beschrieben.⁵³

Schweitzer (1967) hat das Dilemma zu einem Polylemma, bestehend aus fünf Zielen, erweitert.⁵⁴ Mensch (1972) hingegen beschreibt lediglich ein Trilemma der drei Ziele der Minimierung der Wartezeiten, der Minimierung der Leerzeiten sowie der Minimierung der Rüstzeiten. Das Ziel der Termineinhaltung ist für ihn lediglich eine Nebenbedingung.⁵⁵

Die Ersatzziele sind jedoch nicht vollständig konkurrierend. So gibt es für die in dieser Arbeit betrachteten drei Ziele durchaus Probleminstanzen, in denen die individuell optimalen Lösungen für alle drei Ziele gleich sind. Günther (1971) hat in seiner Arbeit das "Dilemma der Arbeitsablaufplanung" und die dazu erschienene Literatur einer ausführlichen Analyse unterzogen.⁵⁶ Eine anschließend durchgeführte Analyse der Beziehung zwischen unterschiedlichen Zielen, wobei erneut Beispiele und Diskussionen aus der Literatur aufgegriffen wurden, hat gezeigt, daß die jeweils betrachteten Ziele nicht vollständig konfliktionär sind, sondern sich in einem hohen Maß komplementär zueinander verhalten.⁵⁷ Die meisten in der von Günther (1971) gesichteten Literatur untersuchten Probleme sind jedoch sehr klein und so hat Günther (1971) selber darauf hingewiesen, daß es für eine genaue Aussage der Analyse mehrerer größerer Probleme bedarf.⁵⁸ Aus diesem Grund wurde in der vorliegenden Arbeit eine Korrelationsanalyse über 120 Testprobleme⁵⁹ mit 4, 6, 8 und 10 Aufträgen und 5, 10 und 20 Maschinen durchgeführt. Für jede der 12 Kombinationen aus Auftragszahl und Maschinenzahl wurden jeweils 10 Probleminstanzen gebildet. Jedes Problem wurde vollständig enumeriert und anschließend die Korrelation zwischen jeweils zwei Zielen berechnet. Zur Berechnung der Korrelation wurde der Korrelationskoeffizient nach PEARSON verwendet, der sich wie folgt berechnet:⁶⁰

⁵³Vgl. Gutenberg (1951, S.158-162). Die Darstellung des Dilemmas hat sich ab der zehnten Auflage geändert. Eine Diskussion dieser Veränderung kann in Günther (1971, S.10-12) nachgelesen werden.

⁵⁴Schweitzer (1967) betrachtet die folgenden fünf Ziele: Minimierung der Gesamtlagerzeiten(-kosten), Minimierung der Gesamtvorbereitungszeiten(-kosten), Minimierung der Terminüberschreitungen(-kosten), Minimierung der Durchlaufzeiten, Minimierung der Wartezeiten (vgl. dazu Schweitzer (1967), insbesondere die Seiten 291-293).

⁵⁵Vgl. Mensch (1972), insbesondere Seite 82.

⁵⁶Vgl. Günther (1971, S.45-78), sowie die dort angegebene Literatur.

⁵⁷Vgl. Günther (1971, S.46 u. 86-107).

⁵⁸Vgl. Günther (1971, S.74 u. 90).

⁵⁹Zur Generierung dieser Testprobleme vgl. Anhang A und B.

⁶⁰Vgl. Schaich u. a. (1993, S.92). Für die Implementation in C++ siehe Dietmar (1991, S. 148f.).

$$\rho = \frac{N \sum_{i=1}^N x_i y_i - \left(\sum_{i=1}^N x_i \right) \left(\sum_{i=1}^N y_i \right)}{\sqrt{\left[N \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2 \right] \left[N \sum_{i=1}^N y_i^2 - \left(\sum_{i=1}^N y_i \right)^2 \right]}} \quad (2.18)$$

Da für $N < 30$ der Korrelationskoeffizient unterschätzt wird, wurde die von Dietmar (1991) berücksichtigte und von OLKIN U. PRATT (1958) vorgeschlagene Korrektur verwendet:⁶¹

$$\rho_{korr} = \rho \left[1 + \frac{1 - \rho^2}{2(N - 3)} \right] \quad (2.19)$$

Tabelle 2.2 gibt die durchschnittlichen Korrelationen für die einzelnen Problemgrößen an

N	M	$C_{max} - F_{mid}$	$C_{max} - T_{max}$	$F_{mid} - T_{max}$	
	5	0,6405	0,5052	0,7247	
	10	0,5011	0,7131	0,6159	
	20	0,3635	0,5515	0,3856	
4		0,5017	0,5899	0,5754	
	5	0,5941	0,5533	0,7662	
	10	0,6106	0,8261	0,6530	
	20	0,3922	0,5947	0,5795	
	6		0,5323	0,6580	0,6662
		5	0,5941	0,7154	0,7174
10		0,4486	0,7776	0,5428	
	20	0,5056	0,7066	0,6336	
	8		0,5161	0,7332	0,6313
		5	0,5253	0,6860	0,6568
10		0,5936	0,7681	0,7024	
	20	0,5578	0,7284	0,6469	
	10		0,5589	0,7275	0,6687
		Total	0,5273	0,6772	0,6354

Tabelle 2.2: Korrelation der Ziele im betrachteten Zielsystem

Es zeigt sich eine mittlere bis starke Korrelation der Zielfunktionswerte. Die Ziele verhalten sich somit in einem mittleren bis starken Maße komplementär zueinander. Seelbach (1975) kommentierte bereits die gleichen Beobachtungen von Günther (1971) dahingehend, daß sich die festgestellte Gleichgerichtetheit der Ziele damit erklären läßt, daß die betrachteten Ziele der Gruppe der regulären Ziele angehören.⁶² Nach Conway u. a. (1967)

⁶¹Vgl. Olkin u. Pratt (1958, S.201) sowie Dietmar (1991, S. 148f).

⁶²Vgl. Seelbach (1975, S.39) und Günther (1971, S.86-107).

ist ein reguläres Ziel wie folgt definiert:⁶³

1. Ein reguläres Ziel ist eine Funktion der Fertigstellungszeitpunkte der einzelnen Aufträge.
2. Reguläre Ziele steigen nur dann an, falls mindestens ein Fertigstellungszeitpunkt ansteigt. Es seien zwei Auftragsfolgen π und π' gegeben:

$$z_k(\pi) = f(C_{[\pi_1]}; C_{[\pi_2]}; \dots; C_{[\pi_h]}; \dots; C_{[\pi_N]}) \quad (2.20)$$

$$z_k(\pi') = f(C_{[\pi'_1]}; C_{[\pi'_2]}; \dots; C_{[\pi'_h]}; \dots; C_{[\pi'_N]}) \quad (2.21)$$

Ziel k ist ein reguläres Ziel, d.h. es gilt: $z_k(\pi') \geq z_k(\pi)$, wenn und nur wenn $C_{[\pi'_h]} > C_{[\pi_h]}$ für mindestens ein h , $h = 1(1)N$, gilt.⁶⁴

Seelbach (1975) merkt an, "[...] daß für alle regulären Zielsetzungen gemeinsam Teile der Gesamtlösung aufgrund bestimmter Kriterien als nichtoptimal ausgesondert werden können. Erst bei der Wahl des bezüglich eines Zielkriteriums optimalen Planes aus der Restmenge können Abweichungen auftreten."⁶⁵ Diese Aussage von Seelbach (1975) kann dahingehend interpretiert werden, daß bereichsweise Zielbeziehungen vorliegen. Außerhalb des Teils des Lösungsraums, in dem die individuell optimalen Lösungen der einzelnen Ziele liegen, besteht tendenziell Zielkomplementarität. Innerhalb besteht ein Konflikt zwischen den einzelnen Zielen.⁶⁶ Für einen Ansatz, welcher die Menge aller effizienten Lösungen ermitteln möchte, würde dies bedeuten, daß er sich bei der Suche zum Auffinden der effizienten individuell optimalen Lösungen aller drei im Zielsystem berücksichtigten Ziele, auf einen Teil des Lösungsraums konzentrieren kann. Die Frage, ob die individuellen globalen Optima der Ziele in einem begrenzten Bereich des Lösungsraums liegen, wie dieser Bereich definiert ist und wie groß dieser ist, wird in Abschnitt 5.5 ausführlich diskutiert werden.

⁶³Vgl. Conway u. a. (1967, S.12).

⁶⁴Conway u. a. (1967, S.12) machen in ihrer Arbeit keine Aussagen darüber, was für die anderen Fertigstellungszeitpunkte gelten muß. Das Sinken eines anderen Fertigstellungszeitpunktes erscheint jedoch unzulässig. Ansonsten müßte ein Verhältnis zwischen dem Ausmaß des Steigendes und Sinkens der Fertigstellungszeitpunkte ermittelt werden, von dessen Wert dann das Steigen oder Sinken der Zielgröße abhängig gemacht wird.

⁶⁵Seelbach (1975, S.39).

⁶⁶Dieser Aspekt wird in Abschnitt 5.5.2.2 nochmals aufgegriffen.

Kapitel 3

Das multikriterielle PFSP als Vektorminimierungs- problem

3.1 Einführung

Formal läßt sich ein Vektorminimierungsmodell wie folgt darstellen:

$$\text{MIN } z(\pi) = (z_1(\pi), \dots, z_K(\pi)) \quad (3.1)$$

mit:

$$\pi \in \Pi \quad (3.2)$$

Es sind K Zielfunktionen gleichzeitig zu minimieren.¹ Dabei wird davon ausgegangen, daß der Entscheidungsträger kleine Zielfunktionswerte größeren vorzieht. Dies ist keine Einschränkung bezüglich der Präferenzen des Entscheidungsträgers. Zum einen kann jedes Maximierungsproblem durch Multiplikation der Zielfunktion mit minus Eins in ein Minimierungsproblem umgewandelt werden und umgekehrt. Zum anderen können Approximierungsziele des Entscheidungsträgers mittels geeigneter Abstandsfunktionen in Extremierungsziele transformiert werden. Satisfizierungs- und Fixierungsziele hingegen müssen über Nebenbedingungen berücksichtigt werden.² Eine mögliche Lösung wäre in diesem Fall auf ihre Zulässig-

¹Vgl. Dinkelbach (1982, S.156).

Bei dem in dieser Arbeit betrachteten Problem ist $K = 3$ und die Lösungsmenge Π wird durch alle $N!$ möglichen Auftragsfolgen gebildet.

²Vgl. Dinkelbach (1982, S.154f.).

keit hin zu überprüfen.

Die Minimierung oder Maximierung eines Skalars ist eindeutig definiert. Im Fall der Vektoroptimierung ist dies zunächst nicht gegeben, da der Zielraum $Z(\Pi)$ keine natürliche Ordnung aufweist, sobald eine vektorielle Betrachtung erfolgt.³ Somit ist es nicht immer möglich eine eindeutige Aussage darüber zu treffen, ob ein Vektor kleiner oder größer ist als ein anderer. Wenn es aber nicht immer einen eindeutig kleinsten Vektor gibt, wie kann dann eine Lösung für Gleichung 3.1 und damit eine Lösung des Vektorminimierungsproblems bestimmt werden? Eine sehr allgemeine Herangehensweise an dieses Problem ist, die Lösung des Problems als den vom Entscheidungsträger am stärksten präferierten Zielfunktionsvektor zu definieren.⁴ Wann immer Präferenzen des Entscheidungsträgers bei der Lösungsfindung oder -beurteilung eine Bedeutung zukommt, unterliegen diese stark subjektiven Einflüssen.⁵ Allgemein anerkannt ist jedoch, daß die Lösung des Problems aus einer oder allen effizienten Lösungen besteht. Dies entspricht der allgemeinen Akzeptanz einer, wenn auch sehr schwachen Annahme über die Präferenzen des Entscheidungsträgers, daß dieser effiziente Alternativen den nicht effizienten bzw. dominierten Alternativen vorzieht.⁶ Diese Annahme baut auf der Monotonie-Annahme auf, die besagt, daß die Präferenz eines Entscheidungsträgers für eine Lösung nicht abnimmt, falls ein Wert eines Ziels k c.p. sinkt.⁷

Im Folgenden sollen die hier verwendeten Relationsoperatoren definiert werden, bevor anschließend der Begriff der strengen sowie schwachen Effizienz eingeführt wird:⁸

$z(\pi') \leq z(\pi^\circ)$ gilt, wenn $z_k(\pi') \leq z_k(\pi^\circ) \forall k = 1(1)K$ und $z_{k'}(\pi') < z_{k'}(\pi^\circ)$ für mind. ein $k' \in 1, \dots, K$.

$z(\pi') \leq z(\pi^\circ)$ bedeutet, daß $z_k(\pi') < z_k(\pi^\circ)$ oder $z_k(\pi') = z_k(\pi^\circ) \forall k = 1(1)K$ gilt. Es kann somit auch $z_k(\pi') = z_k(\pi^\circ) \forall k = 1(1)K$ gelten.

$z(\pi') = z(\pi^\circ)$ gilt also, wenn $z_k(\pi') = z_k(\pi^\circ) \forall k = 1(1)K$.

Strenge Effizienz: Eine (zulässige) Lösung $\pi^\circ \in \Pi$ ist streng effizient (nicht dominiert), falls es keine andere (zulässige) Lösung $\pi' \in \Pi$ gibt, die bezüglich aller Ziele zumindest gleich gute und für mindestens

³Vgl. Rosenthal (1985, S.135).

⁴Vgl. Rosenthal (1985, S.135).

⁵Vgl. Rosenthal (1985, S.135).

⁶Vgl. Rosenthal (1985, S.138).

⁷Vgl. Rosenthal (1985, S.138).

⁸Vgl. Dinkelbach (1982, S.157).

ein Ziel einen besseren Zielfunktionswert aufweist. Also: $\pi^\circ \in \Pi$ ist effizient falls $\nexists \pi' \in \Pi$ mit $z(\pi') \leq z(\pi^\circ)$ und $\pi' \neq \pi^\circ$.⁹

Synonym zum Begriff der strengen Effizienz kann der Begriff des strengen Pareto-Optimums verwendet werden.¹⁰ Von der strengen Effizienz, bzw. dem strengen Pareto-Optimum ist die schwache Effizienz bzw. das schwache Pareto-Optimum zu unterscheiden.¹¹

Schwache Effizienz: Eine (zulässige) Lösung $\pi^\circ \in \Pi$ ist schwach effizient (ein schwaches Pareto-Optimum), falls es kein $\pi' \in \Pi$ gibt, das bezüglich aller Ziele einen besseren Zielfunktionswert aufweist. Also: $\pi^\circ \in \Pi$ ist schwach effizient falls $\nexists \pi' \in \Pi$ mit $z(\pi') < z(\pi^\circ)$ und $\pi' \neq \pi^\circ$.¹²

Effiziente Lösungen lassen sich weiterhin in die zwei Gruppen der "unterstützten" und "nicht-unterstützten" Lösungen einteilen.¹³ "Unterstützte" effiziente Lösungen sind solche, deren Zielfunktionsvektoren auf dem Rand der konvexen Hülle des Zielraums liegen. Sind sie nicht Element des Randes der konvexen Hülle, so handelt es sich um "nicht-unterstützte" Lösungen. "Nicht-unterstützte", effiziente Lösungen werden stets durch die Konvexkombination zweier anderer effizienter Lösungen dominiert. Diese Unterscheidung ist bei der multikriteriellen Linearen Optimierung ohne Bedeutung, da es dort keine "nicht-unterstützten" Lösungen geben kann.¹⁴

Ein Beispiel (vgl. auch Abb. 3.1 auf der nächsten Seite) soll die bisher diskutierten Begriffe verdeutlichen. Gegeben sei ein bikriterielles Permutation Flow Shop Problem mit $N = 4$ Aufträgen. Die Zielfunktionsvektoren der $4! = 24$ Auftragsfolgen sind in Abbildung 3.1 dargestellt. Die Auftragsfolgen A bis F erfüllen alle die Definition der schwachen Effizienz. Die Auftragsfolge A ist jedoch nicht streng effizient, da sie von Auftragsfolge B dominiert wird. Die Auftragsfolgen C und E stellen "nicht-unterstützte" effiziente Lösungen dar, da sie nicht Element des unteren Randes der konvexen Hülle sind.¹⁵ Dieser untere Rand wird von den "unterstützten" effizienten Lösungen A, B, D und F beschrieben.

⁹Vgl. z.B. Steuer (1986, S.347), Dinkelbach (1982, S.156) oder Soland (1979, S.29).

¹⁰Vgl. Rosenthal (1985, S.138).

¹¹Es ist notwendig die strenge und die schwache Effizienz zu unterscheiden, da die später in Abschnitt 3.2 diskutierten Kompromißmodelle nicht alle nur streng effiziente sondern zum Teil auch schwach effiziente Lösungen identifizieren.

¹²Vgl. Steuer (1986, S. 221 und 347).

¹³Vgl. Ulungu u. Teghem (1994, S.85).

¹⁴Vgl. zu diesen Ausführungen Steuer (1986, S.431f.).

¹⁵Dies erkennt man daran, daß, wenn man in Abbildung 3.1 die Punkte $z(B)$ und $z(D)$ bzw. $z(D)$ und $z(F)$ miteinander verbinden würde, die Punkte $z(C)$ bzw. $z(E)$ oberhalb dieser Verbindungslinien liegen würden.

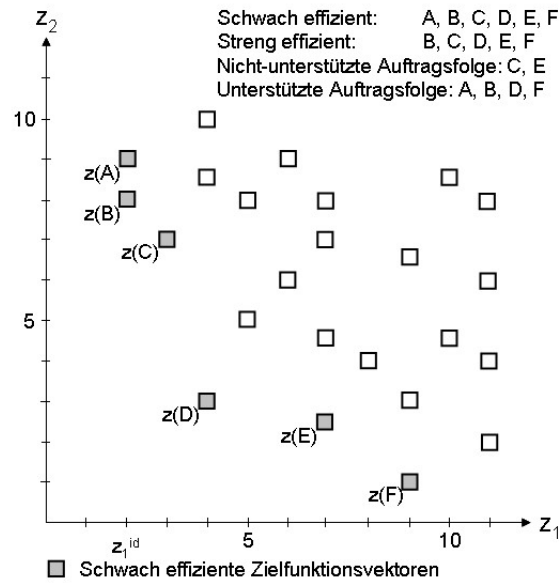


Abbildung 3.1: Arten effizienter Punkte

Aufbauend auf den Definitionen der strengen und schwachen Effizienz sollen im Folgenden die Menge aller effizienten Lösungen und die Menge aller effizienten Zielfunktionsvektoren definiert werden. Die als Lösung des in dieser Arbeit betrachteten Problems gesuchte Menge, ist die Menge aller effizienten Lösungen:

Menge aller effizienten Lösungen:

$$E(\Pi, z) := \{\pi^\circ \in \Pi \mid \nexists \pi' \in \Pi \text{ mit } z(\pi') \leq z(\pi^\circ)\} \quad (3.3)$$

Die Menge aller effizienten Lösungen ergibt sich aus den Auftragsfolgen, deren Zielfunktionsvektoren bezüglich den Zielfunktionsvektoren aller anderer Elemente aus Π effizient sind.

Menge aller effizienten Zielfunktionsvektoren (effiziente Front):

$$E(z) := \{z(\pi^\circ) \mid \pi^\circ \in E(\Pi, z)\} \quad (3.4)$$

Die Menge aller effizienten Zielfunktionsvektoren ergibt sich aus den Zielfunktionsvektoren der Elemente der Menge aller effizienten Lösungen.

Ebenso könnte die Menge aller schwach effizienten Lösungen definiert

werden. Die hier definierte Menge aller (streng) effizienten Lösungen ($E(\Pi, z)$) ist eine Teilmenge davon.¹⁶ Im Zusammenhang mit den im folgenden Abschnitt betrachteten Kompromißmodellen werden weitere Begriffe benötigt, die bereits an dieser Stelle definiert werden sollen. Dabei wird stets von einem Minimierungsproblem ausgegangen.

Menge individuell optimaler Lösungen: Die Menge der individuell optimalen Lösungen kann für jedes Ziel einzeln ermittelt werden und besteht aus den Lösungen, die den Zielfunktionswert von Ziel k minimieren.¹⁷ Also:

$$\Pi_k^* := \{\pi_k^* \in \Pi \mid z_k(\pi_k^*) = \min \{z_k(\pi) \mid \pi \in \Pi\}\} \quad (3.5)$$

Idealzielpunkt: Der Idealzielpunkt ist ein Vektor, welcher für jede Zielgröße k die kleinste bei allen Elementen des Lösungsraums vorkommende Ausprägung enthält.

$$z^{id} := (z_1(\pi_1^{id}), \dots, z_K(\pi_K^{id}))^T \in \mathbb{R}^K \quad (3.6)$$

mit:

$$\pi_k^{id} \in \operatorname{argmin} \{z_k(\pi) \mid \pi \in \Pi\} \quad (3.7)$$

Alternativ kann π_k^{id} wie folgt bestimmt werden:¹⁸

$$\pi_k^{id} \in \operatorname{argmin} \left\{ z_k(\pi) \mid \pi \in \bigcup_{k=1}^K \Pi_k^* \right\} \quad (3.8)$$

D.h. es genügt zu seiner Bestimmung die Elemente der K Mengen der individuell optimalen Lösungen zu betrachten.

Der Idealzielpunkt muß keiner existierenden (zulässigen) Lösung entsprechen. Eine Lösung, die den Idealzielpunkt als Zielfunktionsvektor besitzt, wird als perfekte Lösung bezeichnet. Diese existiert, falls gilt: $\bigcap_{k=1}^K \Pi_k^* \neq \emptyset$. Ein Zielkonflikt liegt somit offensichtlich genau dann vor, wenn gilt:

¹⁶Vgl. Steuer (1986, S.347).

¹⁷Vgl. Dinkelbach (1982, S.158).

¹⁸Vgl. die Definition des Idealzielpunkts bei Dinkelbach (1982, S.158) sowie Ehrgott u. Tenfelde-Podehl (2003, S.120). Ehrgott u. Tenfelde-Podehl (2003, S.120) verwenden zur Ermittlung alleine die Elemente von $E(\Pi, z)$. Die Bestimmung von π_k^{id} würde dann wie folgt notiert: $\pi_k^{id} \in \operatorname{argmin} \{z_k(\pi) \mid \pi \in E(\Pi, z)\}$.

$\bigcap_{k=1}^K \Pi_k^* = \emptyset$,¹⁹ d.h. der Idealzielpunkt keine zulässige Lösung ist.

Anti-Idealzielpunkt: Der Anti-Idealzielpunkt ist ein Vektor, welcher für jede Zielgröße k die größte bei allen Elementen des Lösungsraums vorkommende Ausprägung enthält.

$$z^{aid} := (z_1(\pi_1^{aid}), \dots, z_K(\pi_K^{aid}))^T \in \mathbb{R}^K \quad (3.9)$$

mit

$$\pi_k^{aid} \in \operatorname{argmax} \{z_k(\pi) \mid \pi \in \Pi\} \quad (3.10)$$

Auch der Anti-Idealzielpunkt muß keiner existierenden (zulässigen) Lösung entsprechen.

Utopiepunkt: Ein Utopiepunkt z^{ut} korrespondiert keinesfalls mit einer zulässigen Lösung, denn für ihn gilt bei Minimierungsproblemen:²⁰

$$z^{ut} = z^{id} - \epsilon \quad (3.11)$$

mit

$$\epsilon > 0 \quad (3.12)$$

D.h. es wird gefordert: $z^{ut} < z^{id}$.

Nadirpunkt:²¹ Der Nadirpunkt ist ein Vektor, welcher für jede Zielgröße k die größte, bei allen Elementen der Menge aller effizienten Lösungen vorkommende Ausprägung enthält.

$$z^{nad} := (z_1(\pi_1^{nad}), \dots, z_K(\pi_K^{nad}))^T \in \mathbb{R}^K \quad (3.13)$$

mit

$$\pi_k^{nad} \in \operatorname{argmax} \{z_k(\pi') \mid \pi' \in E(\Pi, z)\} \quad (3.14)$$

¹⁹Vgl. Dinkelbach (1982, S.158).

²⁰Vgl. Ehrgott u. Tenfelde-Podehl (2003, S.121). Steuer (1986, S.420f.) definiert $\epsilon \geq 0$. Steuer (1986) benennt jedoch zwei Fälle, in denen $\epsilon > 0$ gelten muß (vgl. Steuer (1986, S.420)). Mitunter finden sich abweichende Definitionen von der in dieser Arbeit verwendeten, vgl. T'Kindt u. Billaut (2002, S.67). Diese definieren grundsätzlich: $z^{ut} \leq z^{id}$.

z^{ut} wird bei der Diskussion der unten betrachteten Kompromißmodelle benötigt.

²¹Vgl. Ehrgott u. Tenfelde-Podehl (2003, S.120).

Wie schon der Ideal- und Anti-Idealzielpunkt muß der Nadirpunkt keiner existierenden (zulässigen) Lösung entsprechen. Zusammen mit dem Idealzielpunkt liefert der Nadirpunkt Informationen über das Wertintervall der einzelnen Zielgrößen, wenn nur die Elemente der effizienten Front betrachtet werden.²²

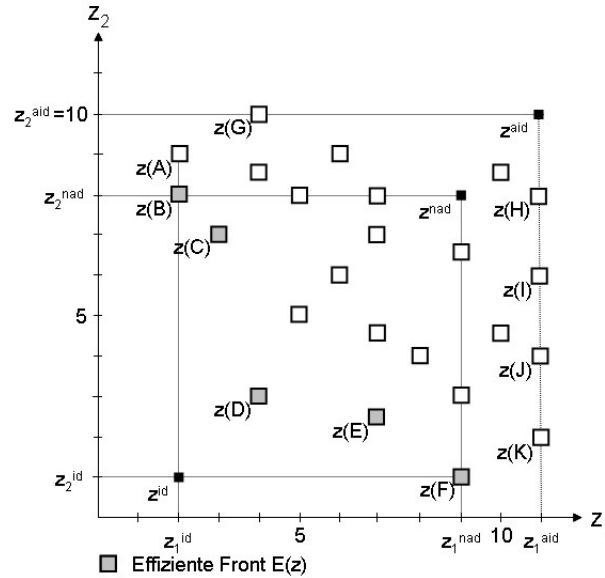


Abbildung 3.2: Besondere Punkte im Zielraum

In Abbildung 3.2 wird das Beispiel aus Abbildung 3.1 auf Seite 25 fortgeführt. Die Menge aller effizienten Auftragsfolgen ist $E(\Pi, z) = \{B; C; D; E; F\}$ ²³ und die effiziente Front ergibt sich daher zu: $E(z) = \left\{ \begin{pmatrix} 2 \\ 8 \end{pmatrix}; \begin{pmatrix} 3 \\ 7 \end{pmatrix}; \begin{pmatrix} 4 \\ 3 \end{pmatrix}; \begin{pmatrix} 7 \\ 2,5 \end{pmatrix}; \begin{pmatrix} 9 \\ 1 \end{pmatrix} \right\}$. Die Mengen der individuell optimalen Lösungen ergeben sich zu: $\Pi_1^* = \{A; B\}$ und $\Pi_2^* = \{F\}$. Der Idealzielpunkt ergibt sich zu $z^{id} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ mit $\pi_1^{id} \in \Pi_1^*$ und $\pi_2^{id} \in \Pi_2^*$. Als Anti-Idealzielpunkt ergibt sich $z^{aid} = \begin{pmatrix} 11 \\ 10 \end{pmatrix}$ mit $\pi_1^{aid} \in \{H; I; J; K\}$ und $\pi_2^{aid} = G$. Der Nadirpunkt errechnet sich zu: $z^{nad} = \begin{pmatrix} 9 \\ 8 \end{pmatrix}$ mit $\pi_1^{nad} = F$ und $\pi_2^{nad} = B$.

²²Vgl. Ehrgott u. Tenfelde-Podehl (2003, S.120). Im Unterschied zum Idealzielpunkt ist es, wie man sich leicht überlegen kann, nur bei bikriteriellen Problemstellungen möglich bei der Bestimmung von z^{nad} $E(\Pi, z)$ durch die Vereinigungsmenge der K Mengen der individuell optimalen Lösungen Π_k^* zu ersetzen.

²³Die Auftragsfolge A ist nur schwach effizient und daher nicht Element von $E(\Pi, z)$.

Um eine Lösung des Vektoroptimierungsproblems zu ermitteln, werden mit a-priori, interaktiven und a-posteriori Ansätzen drei Vorgehensweisen unterschieden. Insbesondere dann, wenn bereits vor der Problemlösung Informationen über die Präferenzstruktur des Entscheidungsträgers vorliegen, können diese bereits bei der Lösungsermittlung berücksichtigt werden. Methoden, welche diese Präferenzstruktur bereits vor Beginn der Lösungsermittlung durch eine feste Einstellung der Parameter des Kompromißmodells berücksichtigen und daraufhin nur eine einzelne Lösung ermitteln werden a-priori Ansätze genannt.²⁴ Interaktive Ansätze hingegen erfassen die Präferenzstruktur des Entscheidungsträgers während des Lösungsprozesses. Dabei werden diesem etwa unterschiedliche Lösungen präsentiert zu denen er sein Präferenzurteil abgibt oder aus denen er seine präferierten oder die nicht präferierten Lösungen auswählt. Durch Auswertung der Präferenzurteile bzw. der Auswahlentscheidungen konkretisiert das Verfahren die Präferenzstruktur des Entscheidungsträgers und ermittelt damit neue Lösungen, welche dem Entscheidungsträger wiederum präsentiert werden. Damit wechseln sich im Lösungsprozeß Präferenzeingaben und Ermittlung vorläufiger Lösungen ab.²⁵

Im Gegensatz dazu verzichten a-posteriori Ansätze auf die Berücksichtigung der Präferenzstruktur vor oder während des Lösungsprozesses. Als Konsequenz besteht die Lösung bei a-posteriori Ansätzen nicht aus einer einzelnen Lösung sondern aus einem Teil oder der vollständigen Menge aller effizienten Lösungen $E(\Pi, z)$.²⁶ Ihre Anwendung ist dann sinnvoll, wenn die Präferenzstruktur des Entscheidungsträgers nicht bekannt ist, bzw. es zu aufwendig wäre, sie im Voraus oder im Verlauf des Lösungsprozesses zu ermitteln. Die Ermittlung dieser Menge kann noch aus weiteren Gründen sinnvoll sein. So ist es sicherlich leichter, seine Präferenzen durch die Auswahl aus der konkreten Repräsentation der Alternativen mittels ihrer Zielerreichungsgrade zum Ausdruck zu bringen, als durch die abstrakte Festlegung von Zielgewichten oder Zielhierarchien. Das gleiche gilt, falls mehrere Entscheidungsträger unterschiedliche Präferenzstrukturen aufweisen und sich auf eine Alternative einigen müssen. Daneben kann die Entscheidung über eine zu realisierende Alternative besser von der Ermittlung der Lösung getrennt werden. Ebenso ermöglicht die Ermittlung der Menge aller effizienten Lösungen die existierenden Trade-Offs zwischen den Zielen aufzuzeigen. Zusätzlich ist die Ermittlung allein aus theoretischer Sicht von Bedeutung und letztlich ist sie auch für die Entwicklung von interaktiven

²⁴Vgl. Ehrgott u. Gandibleux (2000, S.430).

²⁵Vgl. Ehrgott u. Gandibleux (2000, S.430).

²⁶Vgl. Ehrgott u. Gandibleux (2000, S.430).

Ansätzen hilfreich. Bei diesen muß zumindest die sukzessive Ermittlung von $E(\Pi, z)$ möglich sein. Rosenthal (1985) merkt zur Ermittlung der Menge aller effizienten Lösungen als vorteilhaft an, daß die Lösung keinerlei subjektiven Einflüssen unterliegt. Seine Kritik richtet sich aber dagegen, daß die Menge sehr groß sein kann, so daß es dem Entscheidungsträger weiterhin nicht möglich sein wird, eine Lösung auszuwählen. In diesem Fall benötigt er weitere Methoden, welche ihn bei der Auswahlentscheidung unterstützen.²⁷ Diese Kritik stellt jedoch kein scharfes Gegenargument dar. So kann $E(\Pi, z)$ als Input für solche, z.B. interaktive Methoden verwendet werden. Für diese stellt dies eine Zeitersparnis im Prozeß der interaktiven Lösungsfindung dar, da nur noch aus (streng) effizienten Lösungen ausgewählt, die Suche nach zumindest einem Teil dieser Lösungen aber nicht mehr vorgenommen werden muß.²⁸

3.2 Methoden zur Ermittlung effizienter Lösungen

3.2.1 Definition Kompromißmodell

Bei Anwendung eines Kompromißmodells wird das Zielsystem durch eine skalare Zielfunktion $\psi(\pi)$ und gegebenenfalls zusätzliche Nebenbedingungen ersetzt. Die Präferenzen des Entscheidungsträgers werden dabei zum einen bei der Transformation der verschiedenen Ziele in einen Skalar und zum anderen in Form von Nebenbedingungen berücksichtigt. Kompromißmodelle sind grundsätzlich numerisch lösbar und mindestens eine dieser optimalen Lösungen ist kompromiß-optimal. Als kompromiß-optimal wird eine Lösung bezeichnet, wenn sie eine optimale Lösung des Kompromißmodells und gleichzeitig eine effiziente Lösung des ursprünglichen Vektoroptimierungsproblems ist.²⁹

Die Präferenzen des Entscheidungsträgers finden in der Regel in Form von Parametern Eingang ins Modell. Durch eine Variation der Parameter können Lösungen für unterschiedliche Präferenzen ermittelt werden. Eine geeignete Variation kann dabei die Bestimmung der Menge aller effizienten Lösungen $E(\Pi, z)$ ermöglichen. Beim vorliegenden PFSP ist dabei darauf zu achten, daß auch die "nicht-unterstützten" effizienten Auftragsfolgen ermittelt werden können. Im restlichen Teil dieses Abschnitts sollen

²⁷Vgl. Rosenthal (1985, S.140).

²⁸Dafür ist vor dem Start der interaktiven Lösungsfindung Rechenzeit zur Ermittlung der Menge aller effizienten Lösungen zu investieren.

²⁹Vgl. zu diesen Ausführungen Dinkelbach (1982, S.179).

unterschiedliche Kompromißmodelle insbesondere bezüglich dieser Möglichkeit betrachtet werden.

3.2.2 Diskussion verschiedener Kompromißmodelle

3.2.2.1 Lexikographische Optimierung

Eine erste Möglichkeit effiziente Lösungen zu bestimmen, ist die lexikographische Optimierung.³⁰ Bei dieser werden die K Ziele in einem ersten Schritt bezüglich ihrer relativen Wichtigkeit vom Entscheidungsträger geordnet. Anschließend wird für das am wichtigsten erachtete Ziele eine Menge optimaler Lösungen ermittelt. Bei dem ersten Ziel ist dies stets die Menge individuell optimaler Lösungen. Aus dieser wird die Menge von Lösungen bestimmt, die bezüglich des zweitwichtigsten Ziels den besten Zielfunktionswert aufweisen, usw.. Das Verfahren wird fortgesetzt, bis hinsichtlich aller K Ziele optimiert worden ist. Als frühes Abbruchkriterium kann auch die Überprüfung der aktuellen Lösungsmenge herangezogen werden. Ein Abbruch ist möglich, falls diese Menge nur noch ein Element enthält. Als Voraussetzung für das Finden einer effizienten Lösung mit Hilfe der lexikographischen Optimierung muß offensichtlich gelten, daß alle Zielfunktionswerte nach unten beschränkt sind.³¹ Da es sich in dieser Arbeit um ein diskretes Problem mit endlicher Lösungsmenge handelt ist diese Forderung erfüllt.

Die lexikographische Optimierung entspricht also dem sukzessiven Bestimmen der Mengen $\Pi_{[1]}, \dots, \Pi_{[e]}, \dots, \Pi_{[K]}$.

$$\Pi_{[1]} := \left\{ \pi^\circ \mid z_{[1]}(\pi^\circ) = \min_{\pi \in \Pi} \{z_{[1]}(\pi)\} \right\} \quad (3.15)$$

$$\Pi_{[2]} := \left\{ \pi^\circ \mid z_{[2]}(\pi^\circ) = \min_{\pi \in \Pi_{[1]}} \{z_{[2]}(\pi)\} \right\} \quad (3.16)$$

$$\vdots \quad \vdots \quad \vdots$$

$$\Pi_{[K]} := \left\{ \pi^\circ \mid z_{[K]}(\pi^\circ) = \min_{\pi \in \Pi_{[K-1]}} \{z_{[K]}(\pi)\} \right\} \quad (3.17)$$

Dabei ist $[e]$ der Index des in Rang e ($e = 1(1)K$) eingeordneten Ziels k und $\Pi_{[e]}$ die verbleibende Menge von zulässigen Lösungen, nachdem nach Ziel $[e]$ optimiert wurde.

³⁰Vgl. zu diesem Kompromißmodell Steuer (1986, S.228-231) sowie T'Kindt u. Billaut (2002, S.81).

³¹Vgl. Steuer (1986, S.231).

Die Elemente der Menge $\Pi_{[K]}$ sind streng effiziente Lösungen des ursprünglichen Vektorminimierungsproblems. Im Folgenden wird der Ziel-funktionsvektor der Elemente aus $\Pi_{[K]}$ mit z^{lex} bezeichnet. Obwohl die le-xikographische Optimierung streng effiziente Lösungen ermittelt, ist sie nicht zur Bestimmung der Menge aller effizienten Lösungen $E(\Pi, z)$ geeignet, da nicht alle Elemente dieser Menge erzeugt werden können. Durch unterschiedliche Rangordnung der Ziele können maximal $K!$ unterschiedliche Lösungen erzeugt werden. Die Menge aller effizienten Lösungen kann aber mehr als $K!$ Elemente enthalten.

3.2.2.2 Konvexkombination

Bei der Konvexkombination werden die einzelnen Ziele mit Gewichten bewertet und zu einem Skalar addiert.

Ψ_{λ}^{konvex} :

$$MIN \psi(\pi) = \sum_{k=1}^K \lambda_k z_k(\pi) \quad (3.18)$$

mit

$$\sum_{k=1}^K \lambda_k = 1 \quad (3.19)$$

$$0 < \lambda_k < 1 \quad (3.20)$$

$$\pi \in \Pi \quad (3.21)$$

Es läßt sich zeigen, daß wenn π° eine optimale Lösung bezüglich Ψ_{λ}^{konvex} ist, diese Lösung auch eine streng effiziente Lösung bezüglich des ursprünglichen Vektorminimierungsproblems ist. π° ist somit kompromiß-optimal.³²

Der Umkehrschluß ist ohne Einschränkungen hingegen nicht zulässig. Nur wenn die Menge aller effizienten Lösungen $E(\Pi, z)$ konvex und $z_k(\pi)$ konkav ist, existiert ein Vektor λ , so daß eine streng effiziente Lösung π° eine optimale Lösung des Problems Ψ_{λ}^{konvex} ist.

Aus den Aussagen folgt unmittelbar, daß nur bei Problemen, bei denen diese Bedingungen erfüllt sind, die Ermittlung der Menge aller effizienten

³²Diese und die folgende Aussage ergibt sich z.B. aus der Arbeit von Geoffrion (1968), insbesondere S.619-621. Dieser führt den Beweis für sogenannte eigentlich effiziente Lösungen, wobei diese auch stets effiziente Lösungen sind.

Lösungen $E(\Pi, z)$ durch Variation der Gewichte λ_k möglich ist.³³ Für das in dieser Arbeit betrachtete Problem scheidet dieses Kompromißmodell daher als ungeeignet aus.

3.2.2.3 Epsilon-Constraint Ansatz

Bei diesem Kompromißmodell wird lediglich bezüglich eines der K Ziele minimiert. Dabei werden die restlichen $K-1$ Ziele im Nebenbedingungssystem nach oben beschränkt.³⁴

$\Psi_\epsilon :$

$$\text{MIN} \psi(\pi) = z_k(\pi) \quad (3.22)$$

mit

$$z_i(\pi) \leq \epsilon_i^k, i \neq k, i = 1(1)K \quad (3.23)$$

$$\pi \in \Pi \quad (3.24)$$

Der Vektor $\epsilon^k = (\epsilon_1^k, \dots, \epsilon_{k-1}^k, \epsilon_{k+1}^k, \dots, \epsilon_K^k)^T \in \mathbb{R}^{K-1}$ ist der Beschränkungsvektor, falls in der Kompromißzielfunktion nach Ziel k minimiert wird. Bezüglich Ψ_ϵ kann folgende Aussage getroffen werden:

π° ist dann eine streng effiziente Lösung, wenn und nur wenn es für jedes Ziel k einen Beschränkungsvektor ϵ^k gibt, so daß π° die einzige optimale Lösung von Ψ_ϵ ist.³⁵

Bei diesem Kompromißmodell ist insbesondere die Forderung problematisch, daß π° die einzige optimale Lösung von Ψ_ϵ sein muß, was sich nur schwer nachweisen läßt.³⁶ Das im folgenden von Soland (1979) vorgestellte Kompromißmodell kommt ohne die Forderung nach Einzigartigkeit aus.

³³Vgl. z.B. Steuer (1986, S.432).

³⁴Vgl. Soland (1979, S.33).

³⁵Vgl. Soland (1979, S.33).

³⁶Vgl. Soland (1979, S.33).

3.2.2.4 Kompromißmodell von Soland (1979)

Dieses Kompromißmodell wird folgendermaßen formuliert:³⁷

$\Psi_{(g,b)}$:

$$\text{MIN } \psi(\pi) = g(z(\pi)) \quad (3.25)$$

mit

$$z(\pi) \leq b \quad (3.26)$$

$$\pi \in \Pi \quad (3.27)$$

Die Funktion g ist dabei Element der Menge G_Z . Diese Menge ist eine Menge streng monoton fallender Funktionen von $\mathbb{R}^K \rightarrow \mathbb{R}$.³⁸ Soland (1979) zeigt nun, daß folgende Aussagen gelten:³⁹

Ist $g \in G_Z$ und gibt es ein $b \in \mathbb{R}^K$, so daß π° eine optimale Lösung von $\Psi_{(g,b)}$ ist, so ist π° streng effizient und somit kompromiß-optimal.

Ebenso gilt im Umkehrschluß: Ist π° eine streng effiziente Lösung, so existiert für eine beliebige Funktion $g \in G_Z$ ein Vektor $b \in \mathbb{R}^K$, so daß π° optimale Lösung von $\Psi_{(g,b)}$ ist.

Durch Variation des Vektors b ist es somit möglich, alle Elemente von $E(\Pi, z)$ zu ermitteln.⁴⁰ Eine häufig für g verwendete Funktion ist die Konvexkombination der Ziele. In diesem Fall wird das Kompromißmodell $P_\lambda^{\text{konvex}}$ aus Abschnitt 3.2.2.2 derart modifiziert, daß aufgrund des Hinzufügens der Nebenbedingung $z(\pi) < b$ die Ermittlung der "nicht-unterstützten" Elemente aus $E(\Pi, z)$ möglich wird. Dabei ist zu beachten, daß b im Bereich $z^{id} \leq b \leq z^{nad}$ geeignet variiert wird. Ansonsten ist nicht sichergestellt, daß die nicht unterstützten Elemente identifiziert werden. Wie man sich leicht überlegen kann, würde z.B. eine Variation von b im Bereich $b \geq z^{nad}$ dazu führen, dass die gewünschte beschränkende Wirkung von b aufgehoben würde. Das Modell würde zwar nicht formal, jedoch faktisch, in eine reine Konvexkombination transformiert.

³⁷Vgl. Soland (1979, S.34).

³⁸Eine Funktion $f: \mathbb{R}^K \rightarrow \mathbb{R}$ wird als streng monoton fallend bezeichnet, wenn und nur wenn $\forall x, y \in \mathbb{R}^K, x \neq y, x < y \Rightarrow f(x) < f(y)$ gilt. D.h. ein Zielfunktionsvektor, der einen anderen Zielfunktionsvektor dominiert, hat einen kleineren Zielfunktionswert.

³⁹Vgl. Soland (1979, S.34).

⁴⁰Vgl. Soland (1979, S.34).

3.2.2.5 Abstandsfunktionen

Kompromißmodelle, die auf Abstandsfunktionen basieren, haben zum Ziel den Abstand zu einem Referenzpunkt (z^{ref}) zu minimieren. Bevor drei konkrete Kompromißmodelle vorgestellt werden, soll zunächst der Begriff des Maßes im Vektorraum allgemein betrachtet werden.

Ein Maß im Vektorraum \mathbb{R}^K kann allgemein definiert werden als Funktion, die jeweils zwei Vektoren $x, y \in \mathbb{R}^K$ eine skalare Größe $\|x - y\| \in \mathbb{R}$ zuordnet und dabei folgenden vier Anforderungen genügt:⁴¹

1. $\|x - y\| \geq 0$ und $\|x - x\| = 0$
2. $\|x - y\| = \|y - x\|$
3. $\|x - y\| \leq \|x - z\| + \|z - y\|$
4. Falls $x \neq y$, dann $\|x - y\| > 0$

Die wohl bekanntesten Maße stellen die l_p -Distanzen dar:

Gewichtete l_p -Distanz:

$$\|z^{ref} - z(\pi)\|_p^\lambda = \begin{cases} \sqrt[p]{\sum_{k=1}^K \lambda_k |z_k^{ref} - z_k(\pi)|^p} & \text{falls } 1 \leq p < \infty \\ \max_{k=1(1)K} \left\{ \lambda_k |z_k^{ref} - z_k(\pi)| \right\} & \text{falls } p = \infty \end{cases} \quad (3.28)$$

λ ist dabei ein Vektor nicht negativer Gewichte, es gilt also: $\lambda_k \geq 0 \forall k = 1(1)K$. Auf die Gewichtung mit λ kann auch verzichtet werden, in diesem Fall erhält man eine ungewichtete l_p -Distanz.

Ungewichtete l_p -Distanz:

$$\|z^{ref} - z(\pi)\|_p = \begin{cases} \sqrt[p]{\sum_{k=1}^K |z_k^{ref} - z_k(\pi)|^p} & \text{falls } 1 \leq p < \infty \\ \max_{k=1(1)K} \left\{ |z_k^{ref} - z_k(\pi)| \right\} & \text{falls } p = \infty \end{cases} \quad (3.29)$$

Für $p = 1$ resultiert eine (gewichtete) rechtwinklige Abstandsmessung (Manhattan-Distanz) zwischen den zwei Vektoren. Wird $p = 2$ gesetzt, so wird die (gewichtete) euklidische Distanz gemessen. Ist p unendlich,

⁴¹Vgl. zu diesen Ausführungen Steuer (1986, S.44).

so erhält man die sogenannte (gewichtete) Tchebycheff-Distanz.⁴² Im Folgenden sollen drei Kompromißmodelle, welche diese Tchebycheff-Distanz verwenden, vorgestellt werden: Das Kompromißmodell der ungewichteten Tchebycheff-Distanz, das Kompromißmodell der gewichteten sowie das der erweiterten gewichteten Tchebycheff-Distanz.

Ungewichtete Tchebycheff-Distanz: Ein Kompromißmodell, daß mit der ungewichteten Tchebycheff-Distanz arbeitet wurde von Bowman Jr. (1976) vorgestellt.⁴³

Ψ_{Θ} :

$$\text{MIN } \psi(\pi) = \|(z^{lex} - \Theta) - z(\pi)\|_{\infty} \quad (3.30)$$

mit

$$\pi \in \Pi \quad (3.31)$$

Als Referenzpunkt z^{ref} wird der Punkt $(z^{lex} - \Theta)$ verwendet. Θ ist ein Vektor mit K Elementen, von denen $K - 1$ Elemente nicht Null sind: $\Theta = (0, \theta_{[2]}, \dots, \theta_{[K]}) \in \mathbb{R}^K$. Dabei ist wichtig, daß das Null-Element dem Ziel zugeordnet ist, nach dem in der lexikographischen Optimierung zuerst optimiert wurde. Bowman Jr. (1976) hat nun gezeigt, daß folgende Aussage gilt:

Ist π° eine streng effiziente Lösung, $\exists \Theta^* \in \mathbb{R}^K$, so daß π° eine optimale Lösung von Ψ_{Θ} ist.⁴⁴

Die Aussage, daß eine optimale Lösung von Ψ_{θ} immer eine strenge effiziente Lösung ist, ist dagegen unzulässig.⁴⁵ Dennoch ist es möglich die Menge aller effizienten Lösungen $E(\Pi, z)$ durch Variation von Θ zu generieren, die erhaltenen Lösungen aus Ψ_{Θ} müssen jedoch auf ihre strenge Effizienz hin überprüft werden.

⁴²Vgl. Steuer (1986, S.45).

⁴³Vgl. zu den folgenden Ausführungen Bowman Jr. (1976) insbesondere die Seiten 79-81.

⁴⁴Vgl. Bowman Jr. (1976, S.76 u. 79).

⁴⁵Vgl. Bowman Jr. (1976, S.76 u. 81).

Gewichtete Tchebycheff-Distanz: Dieses Kompromißmodell wurde ebenfalls von Bowman Jr. (1976) vorgestellt.⁴⁶ Im Gegensatz zu Ψ_{Θ} sind alle Lösungen effizient, wenn auch nur schwach.

Ψ_{λ} :

$$\text{MIN } \psi(\pi) = \|z^{ut} - z(\pi)\|_{\infty}^{\lambda} \quad (3.32)$$

mit

$$\lambda_k > 0, k = 1(1)K \quad (3.33)$$

$$\pi \in \Pi \quad (3.34)$$

Die Gewichte sollten weiterhin noch normiert werden, indem z.B. gefordert wird:

$$\sum_{k=1}^K \lambda_k = 1 \quad (3.35)$$

Nun gilt:

Ist π° eine optimale Lösung von Ψ_{λ} , dann ist π° eine schwach effiziente Lösung.

Ist π° eine streng effiziente Lösung, dann $\exists \lambda \in \mathbb{R}_{+}^K$, so daß π° eine optimale Lösung von Ψ_{λ} ist.⁴⁷

Durch die Variation der Gewichte, ist es somit möglich, die Menge aller effizienten Lösungen $E(\Pi, z)$ zu generieren. Identifizierte schwache effiziente Lösungen bleiben einfach unberücksichtigt. Das Kompromißmodell der gewichtete Tchebycheff-Distanz wird in den in Kapitel 6 entwickelten ACO-Algorithmen verwendet werden.

Erweiterte gewichtete Tchebycheff-Distanz: Diese Form ist die allgemeinste der bisher vorgestellten Varianten. Im Gegensatz zur gewichteten Form, welche auch schwach effiziente Lösungen finden kann, identifiziert dieses Kompromißmodell nur streng effiziente Lösungen.⁴⁸

⁴⁶Vgl. Bowman Jr. (1976), insbesondere die Seiten 82-84 oder T*Kindt u. Billaut (2002, S.69-71).

⁴⁷Vgl. Bowman Jr. (1976, S.83).

⁴⁸Vgl. zu $P_{(\lambda, \delta)}$ Steuer (1986, S. 422-425 u. 440-443).

$\Psi_{(\lambda, \delta)}$:

$$\text{MIN } \psi(\pi) = \|z^{ut} - z(\pi)\|_{\infty}^{\lambda} + \delta \sum_{k=1}^K |z_k^{ut} - z_k(\pi)| \quad (3.36)$$

mit

$$\lambda_k \geq 0, k = 1(1)K \quad (3.37)$$

$$\sum_{k=1}^K \lambda_k = 1 \quad (3.38)$$

$$\pi \in \Pi \quad (3.39)$$

Gibt es einen Vektor λ sowie ein ausreichend kleines $\delta \in \mathbb{R}_+^K$, so daß π° eine optimale Lösung von $\Psi_{(\lambda, \delta)}$ ist, so ist π° auch eine streng effiziente Lösung.⁴⁹

Die Umkehraussage gilt ebenfalls: Ist π° eine streng effiziente Lösung, so gibt es einen Vektor der Gewichte λ und ein sehr kleines $\delta \in \mathbb{R}_+^K$, so daß π° auch eine optimale Lösung von $\Psi_{(\lambda, \delta)}$ ist.⁵⁰

Dieses Kompromißmodell ermöglicht die Generierung der Menge aller effizienten Lösungen. Der Term $\delta \sum_{k=1}^K |z_k^{ut} - z_k(\pi)|$ verhindert, daß schwach effiziente Lösungen als Ergebnis resultieren. Gerade die Bestimmung eines geeigneten Wertes für δ ist jedoch nicht unproblematisch. Um die größten Probleme zu umgehen, sollte der Wert ausreichend klein sein und gemäß Steuer (1986) im Bereich von 0,0001 bis 0,01 liegen.⁵¹ Steuer (1986) löst dieses Problem, indem er das Modell $\Psi_{(\lambda, \delta)}$ in das Modell Ψ_{λ}^{steuer} überführt. In diesem kann auf den Parameter δ verzichtet werden.⁵²

Ψ_{λ}^{steuer} :

$$\text{MIN } \psi(\pi) = \|z_k^{ut} - z_k(\pi)\|_1 \quad (3.40)$$

mit

$$\pi \in \Pi(\Psi_{\lambda}) \quad (3.41)$$

Zunächst wird die Lösungsmenge $\Pi(\Psi_{\lambda})$ des Kompromißmodells der gewichteten Tchebycheff-Distanz Ψ_{λ} bestimmt. Aus den Elementen von

⁴⁹Vgl. Steuer (1986, S.442), Theorem 14.18.

⁵⁰Vgl. Steuer (1986, S.443), Theorem 14.20.

⁵¹Vgl. Steuer (1986, S.429f.).

⁵²Vgl. Steuer (1986, S.444f.).

$\Pi(\Psi_\lambda)$ werden anschließend die Elemente mit der minimalen Summe der absoluten Abweichungen vom Utopiepunkt bestimmt und damit streng effiziente Lösungen identifiziert. Für dieses Modell gilt dann:

π° ist dann und nur dann streng effizient, falls es einen Gewichtungsvektor λ gibt, so daß π° eine optimale Lösung von Ψ_λ^{steuer} ist.⁵³

3.2.2.6 Goal-Programming

Goal-Programming ist mehr ein allgemeines Konzept bzw. eine Verallgemeinerung der Ansätze der Abstandsfunktionen, als ein konkretes Kompromißmodell. Auch wenn das Goal-Programming ursprünglich für die Betrachtung von Approximationszielen und weniger für Extremierungsziele konzipiert wurde, kann es auch für diese verwendet werden.⁵⁴ Dafür muß ein geeigneter Referenzpunkt in der Abstandsfunktion gewählt werden. Es ist offensichtlich, daß es sich bei diesem um einen Punkt handeln sollte, der nicht erreicht werden kann. Diese Eigenschaft wird regelmäßig vom Utopiepunkt erfüllt. Aber auch der Idealzielpunkt kann verwendet werden.⁵⁵ Auch dieser wird nicht erreicht, es sei denn, es gibt eine perfekte Lösung. Sollte es diese geben, so stellt das Erreichen des Idealzielpunktes kein Problem dar, da in diesem Fall das ursprüngliche Vektorminimierungsproblem eindeutig optimal gelöst ist.

Mit Hilfe eines gewichteten l_p -Maßes läßt sich ein Goal-Programming Kompromißmodell z.B. folgendermaßen formulieren:

Ψ_λ^g :

$$MIN \psi(\pi) = \begin{cases} \sqrt[p]{\sum_{k=1}^K \lambda_k |z_k^{ut} - z_k(\pi)|^p} & \text{falls } 1 \leq p < \infty \\ \max_{k=1(1)K} \{\lambda_k |z_k^{ut} - z_k(\pi)|\} & \text{falls } p = \infty \end{cases} \quad (3.42)$$

mit

$$\pi \in \Pi \quad (3.43)$$

Mit $p = \infty$ erhält man offensichtlich das in Abschnitt 3.2.2.5 vorgestellte Kompromißmodell der gewichteten Tchebycheff-Distanz Ψ_λ von Bowman

⁵³Vgl. Steuer (1986, S.445), sowie TKindt u. Billaut (2002, S.76).

⁵⁴Vgl. Dinkelbach (1982, S.192).

⁵⁵Vgl. Dinkelbach (1982, S.192).

Jr. (1976).⁵⁶ Setzt man hingegen $p = 1$ so geht Ψ_λ^g in Ψ_λ^{konvex} über. Da z^{ut} konstant ist, hat es keinen Einfluß auf die Lösung.

3.3 Lösungsansätze für das PFSP und Single Machine Problem bei mehrfacher Zielsetzung

3.3.1 Exakte und heuristische Verfahren

3.3.1.1 Exakte Verfahren

Exakte Verfahren bestimmen in einer endlichen Zahl von Schritten eine optimale Lösung des zu lösenden Problems. Für das in dieser Arbeit betrachtete multikriterielle Permutation Flow Shop Problem ist diese optimale Lösung die Menge aller effizienten Auftragsfolgen. Zu den exakten Verfahren gehören Branch-and-Bound Ansätze, Schnittebenenverfahren, Branch-and-Cut Ansätze sowie Verfahren der Dynamischen Optimierung.⁵⁷ Nelson u. a. (1986) haben a-posteriori Ansätze für das Single-Machine Problem vorgestellt.⁵⁸ Weiterhin haben Selen u. Hott (1986) einen Mixed Integer Goal Programming Ansatz für das bikriterielle Flow Shop Problem vorgeschlagen, wobei es sich um einen a-priori Ansatz handelt.⁵⁹ Für weitere Ansätze siehe auch die Literaturübersicht von T'Kindt u. Billaut (2001).⁶⁰

Jedoch kann selbst bei den häufig angewendeten Branch-and-Bound Ansätzen das Durchführen einer vollständigen Enumeration nicht ausgeschlossen werden, falls die Branching- und Boundingregeln für das Problem ungünstig gewählt werden oder in der konkreten Probleminstanz nicht effizient sind. Damit ist auch eine mit der Problemgröße exponentiell ansteigende Rechenzeit nicht grundsätzlich ausgeschlossen.

Anders als für polynomial lösbare Probleme, für die es stets ein exaktes Verfahren gibt, dessen Rechenaufwand höchstens polynomial mit der Problemgröße steigt, steigt der Rechenaufwand bei der Lösung NP-schwerer Probleme exponentiell mit der Problemgröße. Für praxisrelevante Problem-

⁵⁶Bowman Jr. (1976, S.84) weist in seiner Arbeit selber darauf hin, daß sein Ansatz auch als Goal-Programming Ansatz interpretiert werden kann.

⁵⁷Vgl. Domschke u. Drexl (1998, S.119f.).

⁵⁸Vgl. Nelson u. a. (1986, S.465-473).

Weiterhin haben Selen u. Hott (1986) einen Mixed Integer Goal Programming Ansatz für das bikriterielle Flow Shop Problem vorgeschlagen, wobei es sich um einen a-priori Ansatz handelt (vgl. Selen u. Hott (1986, S.1122f.)). Für weitere Ansätze siehe auch die Literaturübersicht von T'Kindt u. Billaut (2001, S.149-158).

⁵⁹Vgl. Selen u. Hott (1986, S.1122f.).

⁶⁰Vgl. T'Kindt u. Billaut (2001, S.149-158).

größen und Lösungsansätze scheidet exakte Verfahren zur Problemlösung daher in der Regel aus. Vielmehr findet eine Konzentration auf die Entwicklung heuristischer Lösungsansätze statt.

3.3.1.2 Heuristische Lösungsansätze

Heuristische Verfahren finden nicht zwingend die optimale Lösung des Problems. Vielmehr ist es ihr Bestreben, in einer angemessenen Rechenzeit, eine möglichst gute Lösung zu finden.⁶¹ D.h. anstatt der Menge aller effizienten Lösungen $E(\Pi, z)$ ermitteln Heuristiken eine Menge heuristisch effizienter Lösungen.

Heuristisch effiziente Lösung: Allgemein kann eine Lösung als heuristisch effizient bezeichnet werden, wenn sie von keiner anderen bekannten Lösung dominiert wird. Bezogen auf eine einzelne Heuristik ist eine Lösung somit dann heuristisch effizient, wenn es keine andere von der Heuristik erzeugte und auf Effizienz überprüfte Lösung gibt, die bezüglich aller Ziele keinen schlechteren und bei mindestens einem Ziel einen besseren Zielerreichungsgrad aufweist.⁶²

Heuristisch effiziente Menge (NDS): Die Menge heuristisch effizienter Lösungen⁶³ ergibt sich aus den von der Heuristik ermittelten heuristisch effizienten Lösungen.

Heuristisch effiziente Front (NDF): Die Menge der Zielfunktionsvektoren⁶⁴ einer heuristisch effizienten Menge.

Es ist somit zu beachten, daß es die (einzige) Menge heuristisch effizienter Lösungen nicht gibt. Vielmehr ist diese verfahrensabhängig. Daher kann eine Heuristik nur eine von sehr vielen möglichen heuristisch effizienten Mengen finden.

Neben der Unterscheidung in Konstruktions- und Verbesserungsheuristiken, sowie in deterministische und stochastische Verfahren können Heuristiken danach unterschieden werden, ob es sich um eine problemspezifische oder um eine sogenannte Metaheuristik handelt.

Problemspezifische Verfahren sind für ein konkretes Problem entwickelt und lassen sich, wenn überhaupt, nur schwer auf andere Probleme übertragen. Metaheuristiken hingegen beschreiben ein

⁶¹Die Beurteilung einer Lösung bezüglich ihrer Güte ist Thema des Kapitels 4.

⁶²Der Begriff der heuristischen Effizienz wurde erstmals von Huckert u. a. (1980, S.59) verwendet.

⁶³NDS: Non Dominated Set.

⁶⁴NDF: Non Dominated Front.

allgemeines Konzept zur Lösungsfindung. Dieses Konzept kann mehr oder weniger leicht auf unterschiedliche Probleme angepaßt werden. Zu den Metaheuristiken zählen unter anderem Simulated Annealing, Tabu Search, Genetische Algorithmen sowie die relativ junge Ant Colony Optimization (ACO).

In den folgenden beiden Abschnitten 3.3.2 und 3.3.3 sollen die bisher in der Literatur verfolgten Ansätze für das multikriterielle Permutation Flow Shop Problem mit mehr als zwei Maschinen vorgestellt werden.⁶⁵ Die ACO-Metaheuristik ist bisher nur für das multikriterielle Single Machine Problem sowie für einen bikriteriellen 2-Maschinen Permutation Flow Shop verwendet worden. Diese Ansätze werden in Abschnitt 3.3.4 vorgestellt.

3.3.2 A-priori Ansätze

In diesem Abschnitt sollen nur a-priori Ansätze für das Permutation Flow Shop Problem mit mehr als zwei Maschinen vorgestellt werden.

Rajendran (1994) hat die Ziele Minimierung der Zykluszeit und Minimierung der Gesamtdurchlaufzeit betrachtet.⁶⁶ Er stellt eine problemspezifische Verbesserungsheuristik vor. Als Kompromißmodell verwendet er die Konvexkombination der Ziele, wobei die beiden Ziele über die gesamte Laufzeit der Heuristik hinweg gleichgewichtet sind.

Ebenfalls 1994 haben Gangadharan u. Rajendran (1994) einen Genetischen Algorithmus vorgestellt.⁶⁷ Es wurden erneut die Zykluszeit und die Gesamtdurchlaufzeit als Ziele berücksichtigt. Auch hier findet eine Konvexkombination mit konstanter Gleichgewichtung der Ziele als Kompromißmodell Verwendung.

Rajendran (1995) stellt zwei Heuristiken vor. In der ersten wird ein Zielsystem aus den beiden Zielsgrößen Zykluszeit und Gesamtdurchlaufzeit betrachtet.⁶⁸ In der Zweiten modifiziert er diese Heuristik und behandelt ein Zielsystem mit den drei Zielgrößen Zykluszeit, Gesamtdurchlaufzeit und Maschinenleerzeiten.⁶⁹ Der Ansatz der Heuristiken ähnelt dem aus Rajen-

⁶⁵Eine Übersicht über Ansätze zur Lösung unterschiedlicher Reihenfolgeprobleme haben T'Kindt u. Billaut (2001) veröffentlicht (vgl. T'Kindt u. Billaut (2001, S.149-158)). Etwas älter ist die Übersicht von Nagar u. a. (1995) (vgl. Nagar u. a. (1995, S.93-101)).

Ehrgott u. Gandibleux (2000) sowie Ulungu u. Teghem (1994) fassen allgemein Lösungsansätze für multikriterielle kombinatorische Optimierungsprobleme zusammen (vgl. Ehrgott u. Gandibleux (2000, S.439-445) und Ulungu u. Teghem (1994, S.85-98)). Probleme des Scheduling werden von ihnen jedoch ausgeklammert. Van Veldhuizen u. Lamont (2000) geben einen guten Überblick über Lösungskonzepte mit evolutionären Ansätzen (vgl. Van Veldhuizen u. Lamont (2000, S.129-144)).

⁶⁶Vgl. Rajendran (1994), insbesondere die Seiten 2545-2551.

⁶⁷Vgl. Gangadharan u. Rajendran (1994), insbesondere die Seiten 473f.

⁶⁸Vgl. Rajendran (1995), insbesondere die Seiten 542-546.

⁶⁹Vgl. Rajendran (1995), insbesondere die Seiten 550-554.

dran (1994).⁷⁰ Auch hier handelt es sich um eine problemspezifische Verbesserungsheuristik, die eine konstante Gleichgewichtung der Ziele als Kompromißmodell benutzen.

3.3.3 A-posteriori Ansätze

Daniels u. Chambers (1990) haben für die beiden Ziele Minimierung der Zykluszeit und Minimierung der maximalen Terminüberschreitung eine problemspezifische konstruierende Heuristik vorgestellt, die mit einem Epsilon-Constraint Kompromißmodell arbeitet.⁷¹ Eine Beschreibung der Parametervariation, falls mehr als zwei Ziele betrachtet werden sollen, fehlt.⁷²

Ishibuchi u. Murata (1998) haben "Multi Objective Genetic Local Search" (MOGLS) vorgestellt.⁷³ Sie betrachten ein Zielsystem bestehend aus den drei Zielen Minimierung der Zykluszeit, Minimierung der maximalen Terminüberschreitung und Minimierung der Gesamtdurchlaufzeit. Der Algorithmus ist derart ausgestaltet, daß er theoretisch beliebig viele Ziele berücksichtigen kann. Als Kompromißmodell verwenden Ishibuchi u. Murata (1998) eine Konvexkombination der Ziele und zwar sowohl für die Ermittlung der Fitness der Populationsindividuen als auch für die lokale Suche.⁷⁴

Bagchi (1999) hat aufbauend auf den "Nondominated Sorting Genetic Algorithm" (NSGA) von Srinivas u. Deb (1994)⁷⁵ den "Elitist Nondominated Sorting Algorithm" (ENGA) vorgestellt.⁷⁶ Mit diesem betrachtet er ein Zielsystem mit den drei Zielen: Minimierung der Zykluszeit, Minimierung der mittleren Durchlaufzeit sowie der Minimierung der mittleren Terminüberschreitung. ENGA verzichtet auf die Verwendung eines Kompromißmodells. Stattdessen verwendet er eine sogenannte Dummy Fitness, welche mit Hilfe der Nondominated Sorting Prozedur berechnet wird. Wie auch MOGLS ist ENGA nicht in der Lage den Zielraum systematisch abzusuchen.

Framinan u. a. (2002) haben zwei Ansätze für ein Zielsystem bestehend aus der Minimierung der Zykluszeit und Minimierung der maximalen Ter-

⁷⁰Vgl. Rajendran (1994), insbesondere die Seiten 2545-2551.

⁷¹Vgl. Daniels u. Chambers (1990), insbesondere die Seiten 986-988.

⁷²Daniels u. Chambers (1990) haben in ihrer Arbeit ebenfalls eine Branch and Bound Ansatz sowie eine problemspezifische Heuristik für das Permutation Flow Shop Problem mit zwei Maschinen vorgestellt (vgl. Daniels u. Chambers (1990, S.982-986).

⁷³Vgl. Ishibuchi u. Murata (1998, S.392-398).

⁷⁴Vgl. Ishibuchi u. Murata (1998, S.392-398).

⁷⁵Vgl. Srinivas u. Deb (1994).

⁷⁶Vgl. Bagchi (1999), insbesondere die Seiten 216-244.

minüberschreitung vorgestellt.⁷⁷ Beide Heuristiken bauen auf dem Prinzip der Heuristik von Nawaz u. a. (1983)⁷⁸ auf. Ebenfalls ist ihnen gemeinsam, daß zu Beginn eine Menge von Zielgewichten festgelegt wird. Anschließend wird für jede der zuvor festgelegten Zielgewichte ein Kompromißmodell heuristisch gelöst. Nachdem für die unterschiedlichen Gewichtseinstellungen Auftragsfolgen konstruiert worden sind, wird aus diesen Auftragsfolge eine Menge heuristisch effizienter Lösungen ermittelt.⁷⁹ Die Autoren bezeichnen einen der beiden Ansätze als a-priori Ansatz. Weil er jedoch mehrere Auftragsfolgen als Lösung hervorbringt, wird er hier zu den a-posteriori Ansätzen gezählt.

Ebenfalls 2002 haben Chang u. a. (2002) den "Gradual-Priority Weighting Genetic Algorithm" (GPWGA) vorgestellt.⁸⁰ Als Ziele betrachten sie die Minimierung der Zykluszeit, die Minimierung der Gesamtdurchlaufzeit sowie die Minimierung der Summe der Terminüberschreitungen. Als Kompromißmodell wird auch hier eine Zielgewichtung eingesetzt, wobei die Autoren ein Verfahren beschreiben, mit dem die Gewichte über die Laufzeit hinweg systematisch variiert werden, um die Suche in alle Regionen des Zielraums zu führen.

Loukil u. a. (2005) verwenden das "Multi Objective Simulated Annealing" (MOSA).⁸¹ Dieses wird neben der Lösung von Permutation Flow Shop Problemen auch für die Lösung von Single Machine und Parallel Machine Problemen verwendet. Die experimentellen Analysen werden für unterschiedliche Zielsysteme, bestehend aus jeweils zwei Zielen, durchgeführt. Als Kompromißmodell wird eine Konvexkombination der jeweiligen Ziele verwendet.

Im übrigen ist es denkbar aus den Ansätzen von Rajendran (1994), Gangadharan u. Rajendran (1994) und Rajendran (1995) durch ein mehrmaliges Starten der Heuristik, verbunden mit einer geeigneten Variation der Zielgewichte, a-posteriori Ansätze zu entwickeln.⁸²

3.3.4 ACO Algorithmen für das multikriterielle PFSP und Single Machine Problem

Gravel u. a. (2002) betrachten ein multikriterielles Reihenfolgeproblem der

⁷⁷Vgl. Framinan u. a. (2002), insbesondere die Seiten 565-568.

⁷⁸Vgl. Nawaz u. a. (1983, S.92).

⁷⁹Dieser Schritt wird in der Beschreibung der Heuristiken nicht explizit dargestellt, sondern ergibt sich indirekt aus der Aussage auf S.564 (vgl. Framinan u. a. (2002, S.564)).

⁸⁰Vgl. Chang u. a. (2002, S.3-8).

⁸¹Vgl. Loukil u. a. (2005), insbesondere die Seiten 44-46.

⁸²Siehe zu den Heuristiken: Rajendran (1994, S.2545-2551), Gangadharan u. Rajendran (1994, S.473f.) und Rajendran (1995, S.542-546).

Aluminiumproduktion. Es kann als Single Machine Problem mit reihenfolgeabhängigen Rüstzeiten und den Zielen der Minimierung der Leerzeiten, der Minimierung der Gesamtverspätung der Aufträge, der Minimierung einer Transportkostenfunktion sowie einem weiteren Ziel⁸³ bezeichnet werden.⁸⁴ Als Kompromißmodell wird eine lexikographische Ordnung verwendet, wobei die Rangfolge der Ziele von den Autoren nicht festgelegt wird. Vielmehr wird sie dem einzelnen Entscheidungsträger überlassen.⁸⁵

Iredi u. a. (2001) betrachten zwei a-posteriori Ansätze für das Single Machine Problem.⁸⁶ Dabei werden die zwei Ziele der Minimierung der Summe der Terminüberschreitungen sowie die Minimierung der Rüstzeiten berücksichtigt. Die Ansätze werden in Abschnitt 6.1.4.3 vorgestellt.

T'kindt u. a. (2002) haben einen mit SACO bezeichneten Algorithmus entwickelt, welcher das Problem der zweistufigen Reihenfertigung bei zwei Zielen betrachtet.⁸⁷ Als Ziele werden die Minimierung der Zykluszeit und die Minimierung der Gesamtdurchlaufzeit berücksichtigt, wobei eine lexikographische Ordnung der Minimierung der Zykluszeit eine höhere Priorität zuteilt. Beim betrachteten Problem kann die optimale Zykluszeit mit Hilfe des Johnson-Algorithmus⁸⁸ bestimmt werden. Der Algorithmus verfolgt daher das Ziel, die Gesamtdurchlaufzeit zu minimieren, unter der Restriktion, daß die Zykluszeit minimal ist. Offensichtlich handelt es sich beim SACO-Algorithmus um einen a-priori Ansatz.

⁸³Dabei handelt es sich um ein spezifisches Problem der Produktion von Aluminium. Es soll die Anzahl der benötigten Drainagen beim Umrüsten minimiert werden.

⁸⁴Vgl. Gravel u. a. (2002, S.220 u. 223).

⁸⁵Vgl. Gravel u. a. (2002, S.222). In der experimentellen Analyse verwenden Sie folgende relative Ordnung der Ziele: 1. Minimierung der Leerzeiten, 2. Minimierung der Gesamtverspätung, 3. Minimierung der Anzahl benötigter Drainagen, 4. Minimierung der Transportstrafkosten (vgl. Gravel u. a. (2002, S.224)).

⁸⁶Vgl. Iredi u. a. (2001, S363-366). Die Ansätze sind auch in der Dissertation von Merkle (2002, S.111-128) dargestellt.

⁸⁷Vgl. T'kindt u. a. (2002, S.251-253).

⁸⁸Vgl. Johnson (1954, S.64).

Kapitel 4

Lösungsqualität heuristisch effizienter Mengen

4.1 Einführung

4.1.1 Grundsätzliche Überlegungen zur Lösungsqualität heuristisch effizienter Mengen

Bei Optimierungsproblemen mit einer einzelnen Zielgröße ist allgemein anerkannt, daß sich die Lösungsqualität durch den Wert des einzigen Zielfunktionswertes der Lösung beschreiben läßt. Ein Vergleich oder eine Bestimmung der Qualität von ermittelten heuristisch effiziente Mengen kann zum Beispiel durch die Berechnung einer relativen Abweichung von den Ergebnissen der Vergleichsverfahren oder von einem vorgegebenen, in der Regel des bisher besten bekannten Wertes, erfolgen. Diese Möglichkeit verschließt sich bei der Auswertung von Vektoren von Zielfunktionswerten und erst recht bei der Auswertung von Mengen von Zielfunktionsvektoren, wie es eine heuristisch effiziente Front ist.¹

Das Ziel einer Heuristik soll es sein, eine "gute Approximation" der Menge aller effizienten Lösungen $E(\Pi, z)$ zu finden.² Alternativ könnte auch

¹Bei dieser Argumentation ist zu beachten, daß eine heuristisch effiziente Menge über ihr Abbild im Zielraum, also die heuristisch effiziente Front beurteilt wird. Dabei ist zu beachten, daß die Zuordnung eines Elements der heuristisch effizienten Menge (Auftragsfolge) zu einem Element der heuristisch effizienten Front (Zielfunktionsvektor) nicht eindeutig ist, da mehrere Auftragsfolgen den gleichen Zielfunktionsvektor besitzen können (vgl. auch Van Veldhuizen (1999, S.6-13)).

²Vgl. die Spezifikation des in dieser Arbeit betrachteten Problems in Abschnitt 2.1.2.

eine "gute Approximation" der effiziente Front als Lösung angesehen werden. Dann wäre jedoch die Auftragsfolge, welche vom Entscheidungsträger zur Realisierung eines Zielfunktionsvektors benötigt wird, nicht bekannt. Desweiteren würden Informationen über Auftragsfolgen mit gleichem Zielfunktionsvektor verloren gehen.³ Was nun die Güte einer Lösung, also einer heuristisch effizienten Menge auszeichnet, ist bei weitem nicht so eindeutig und allgemein akzeptiert, wie im Ein-Ziel-Fall. Das Verständnis von "gut" ist subjektiv und hängt somit im wesentlichen von den Präferenzen des Entscheidungsträgers ab. Die einzige Annahme, welche über die Präferenzen bei einem a-posteriori Ansatz getroffen wird ist, daß der Entscheidungsträger effiziente gegenüber dominierten Lösungen vorzieht. Somit bietet sich zunächst das Effizienzkriterium als Beurteilungsmaßstab an, wobei die Bewertung über die Abbildung der heuristisch effizienten Menge im Zielraum, der heuristisch effizienten Front erfolgt.

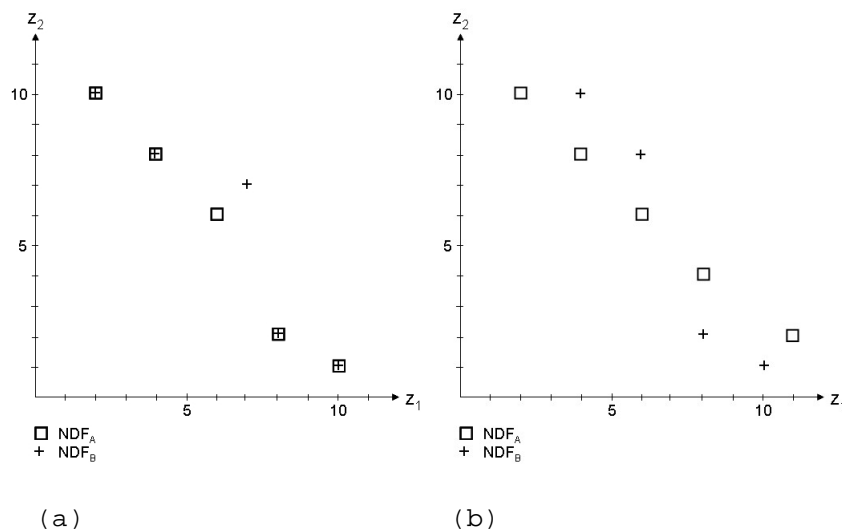


Abbildung 4.1: Das Effizienzkriterium bei der Beurteilung der Lösungsqualität

³Es mag argumentiert werden, daß falls alle entscheidungsrelevanten Aspekte im Zielsystem erfaßt sind, dieses also vollständig ist, es unerheblich ist ob man eine oder mehrere Lösungen kennt, die zum gleichen Zielfunktionsvektor führen. Im Falle mehrerer Lösungen wäre der Entscheidungsträger bzgl. dieser Lösungen indifferent. Dazu ist anzumerken, daß bei der Verwendung von a-posteriori Ansätzen keine Annahmen über die Präferenzen des Entscheidungsträgers getroffen werden. Das Zielsystem in dieser Arbeit ist überwiegend aus Überlegungen zu entscheidungsrelevanten Kosten abgeleitet worden. So kann es also durchaus sein, daß ein Entscheidungsträger die zunächst absichtlich nicht berücksichtigte Präferenz hat Auftragsfolgen zu realisieren, die ihm eine höhere Flexibilität bieten. Dies kann bei alternativen Auftragsfolgen, die zum gleichen Zielfunktionsvektor führen, der Fall sein. Annahmegemäß ist das in dieser Arbeit betrachtete Problem deterministisch, dennoch sollte der Flexibilitätsaspekt beachtet werden, da in der Realität stochastische Probleme existieren.

Eine heuristisch effiziente Menge A ist demnach besser als eine heuristisch effiziente Menge B , wenn jedes Element aus NDF_B gleich einem Element aus NDF_A ist oder von diesem dominiert wird und mindestens ein Element aus NDF_B von einem Element aus NDF_A dominiert wird (vgl. Abb. 4.1 (a)).⁴ Diese Situation wird allerdings nicht immer vorliegen, da sich die Mengen wechselseitig dominieren können (vgl. Abb. 4.1 (b)). Das Bilden einer Rangordnung bezüglich der Lösungsgüte mit Hilfe des Effizienzkriteriums alleine ist dann nicht möglich.⁵ Ebenso läßt sich nicht beurteilen, um wieviel eine Lösung besser ist als eine andere. Um diese Situation aufzulösen, werden in der Literatur bisher im wesentlichen zwei Ansätze verfolgt.

Der erste Ansatz trifft zusätzliche Annahmen über die Präferenzen des Entscheidungsträgers. Diese Technik wird jedoch bereits an dieser Stelle grundsätzlich abgelehnt. Eine auf dieses Vorgehen aufbauende Aussage über die Qualität verliert nicht nur den Anspruch auf Allgemeingültigkeit, sondern ist zugleich nur für Entscheidungsträger mit gleichen Präferenzen nützlich. Sollte der Qualitätsvergleich jedoch vom Entscheidungsträger selbst und zur Unterstützung der eigenen Entscheidung durchgeführt werden, so ist dieses Vorgehen sicherlich, sofern Präferenzen angegeben werden, zulässig. Maße, die die Präferenzen berücksichtigen, wurden z.B. von Daniels (1992), Hansen u. Jaszkiwicz (1998) oder Kim u. a. (2001) vorgestellt.⁶

Der zweite Ansatz betrachtet mehrere Qualitätsaspekte, wobei jeder Qualitätsaspekt mit eigenen Maßen erfaßt wird.⁷ Bei diesem Vorgehen wird es jedoch nicht mehr möglich sein, eine vollständige und eindeutige Rangordnung der Verfahren bezüglich der Lösungsgüte zu erstellen. Vielmehr ist zu erwarten, daß sich einzelne Stärken und Schwächen der Verfahren detailliert darstellen. Es fällt anschließend in den Aufgabenbereich des Entscheidungsträgers festzulegen, auf welche Aspekte bezüglich der Qualität er besonderen Wert legt und auf welche weniger. Aufgrund dieser individuellen Überlegungen ist eine Entscheidung für oder gegen ein Verfahren zu treffen.

Der Qualitätsaspekt, der am häufigsten betrachtet wird, ist die Nähe

⁴Gilt $NDF_A = NDF_B$, so haben beide Mengen die gleiche Qualität. Bei der Betrachtung spielt die Mächtigkeit der Mengen keine Rolle.

⁵Die soeben angestellten Überlegungen bezüglich des Effizienzkriteriums werden in Abschnitt 4.2.1 aufgegriffen.

⁶Vgl. Daniels (1992, S.501-512), Hansen u. Jaszkiwicz (1998, S.10-21), Kim u. a. (2001, 116-119).

⁷Beispiele für solche Qualitätsaspekte sind z.B. die später betrachtete "Distanz zur Menge aller effizienten Lösungen" (vgl. Abschnitt 4.2.1), die "Größe der aufgezeigten Variationsbreite der Zielfunktionswerte" (vgl. Abschnitt 4.2.2) oder die "Verteilung der Zielfunktionsvektoren im Zielraum" (vgl. Abschnitt 4.2.3).

der heuristisch effizienten Menge zur Menge aller effizienten Lösungen. Daher gibt es eine große Anzahl von Maßen, die für diesen Aspekt vorgestellt worden sind. Daneben wird häufig die Größe der Variationsbreite der Zielfunktionswerte in der NDF als weiterer Aspekt verwendet wie auch die Verteilung der Elemente der heuristisch effizienten Menge im Zielraum.

Die Beurteilung der Qualität durch den Entscheidungsträger kann allgemein über den visuellen Vergleich oder die Ermittlung von Qualitätsmaßen erfolgen. In der Literatur ist der visuelle Vergleich ein gängiges Verfahren, wie Van Veldhuizen u. Lamont (2000) feststellen.⁸ Sie merken an, daß eine Qualitätsbeurteilung bzw. ein Qualitätsvergleich wenig über die Effektivität und Effizienz, der in ihrer Arbeit betrachteten multikriteriellen Genetischen Algorithmen aussagt.⁹ Auch wenn der visuelle Vergleich, insbesondere im bikriteriellen Fall, einfach und schnell durchzuführen ist, so entsteht bei der Analyse mehrerer Probleminstanzen ein erheblicher Aufwand, da entsprechend viele Grafiken zu erstellen sind und jede einzelne vom Entscheidungsträger zu beurteilen ist. Eine Aggregation der Qualitätsbeurteilung über mehrere Probleminstanzen, z.B. gleicher Größe, erscheint schwierig.¹⁰ Ein weiterer Kritikpunkt am visuellen Vergleich ist, daß mit zunehmender Mächtigkeit der zu beurteilenden heuristisch effizienten Menge die Bewertung unmöglich oder zumindest sehr schwierig wird.¹¹ Bei der Bewertung von mehr als drei Zielen ist eine Projektion der heuristisch effizienten Menge theoretisch noch möglich, eine Interpretation und damit der visuelle Vergleich aber nahezu unmöglich.¹² Der visuelle Vergleich verschließt sich damit als allgemeiner Ansatz zur Qualitätsbeurteilung, so daß im Folgenden eine Konzentration auf Qualitätsmaße erfolgt.

⁸Vgl. Van Veldhuizen u. Lamont (2000, S.132).

⁹Vgl. Van Veldhuizen u. Lamont (2000, S.132).

¹⁰Hier kann der Einwand erhoben werden, daß ein Entscheidungsträger stets nur eine Probleminstanz zu beurteilen hat. Sollen jedoch, wie in dieser Arbeit, generelle Aussagen über die Güte von Lösungsverfahren getroffen werden, so müssen mehrere Probleminstanzen untersucht werden. Im übrigen gilt dies auch für Entscheidungsträger, welche vor einer grundsätzlichen Entscheidung für den Einsatz des einen oder anderen Lösungsverfahrens stehen.

¹¹Vgl. Farhang-Mehr u. Azarm (2002, S.723).

Steigt die Mächtigkeit der heuristisch effizienten Menge, so ist zu erwarten, daß auch die Mächtigkeit ihrer Abbildung im Zielraum, also die Mächtigkeit der heuristisch effizienten Front steigt.

¹²Vgl. Farhang-Mehr u. Azarm (2002, S.723).

4.1.2 Klassifikation von und Anforderungen an Maße zur Qualitätsanalyse

4.1.2.1 Klassifikation von Qualitätsmaßen

Maße, die zur Qualitätsbeurteilung entwickelt wurden, werden in der Literatur unterschiedlich klassifiziert. Ein erstes und einfaches Kriterium ist die Unterteilung in qualitative und quantitative Maße. Qualitative Maße ermöglichen eine Aussage darüber, ob eine Lösung besser ist als eine andere, jedoch nicht mehr. Soll zusätzlich eine Aussage darüber getroffen werden, um wieviel besser eine Lösung gegenüber einer anderen ist, müssen quantitative Maße verwendet werden.

Kim u. a. (2001) unterscheiden neben dem visuellen Vergleich in ihrer Arbeit drei Gruppen von Maßen für den Qualitätsvergleich.¹³ Die erste Gruppe von Maßen wertet geometrische Eigenschaften einer Lösung aus. Dazu gehören zum einen Längen- und Flächenmaße sowie Distanzmaße. Von diesen werden in einer zweiten Gruppe die Maße unterschieden, welche die Mächtigkeit der Menge der heuristisch effizienten Front als Berechnungsgrundlage heranziehen. In der dritten Gruppe werden von ihnen die Maße zusammengefaßt, die weitere Annahmen über die Präferenzen des Entscheidungsträgers zugrunde legen und bereits oben diskutiert wurden.

Knowles (2002) unterscheidet mit den unabhängigen Maßen, den direkten Maßen sowie den Referenzmaßen drei Klassen von Maßen, die zwei Lösungen miteinander vergleichen.¹⁴

Die unabhängigen Maße erfassen Aspekte, die weder von einer anderen Lösung noch von einer Referenz beeinflusst werden können, wie z.B. die Mächtigkeit einer Lösungsmenge.

Direkte Maße vergleichen zwei Lösungen A und B direkt miteinander und resultieren in einer skalaren Größe, die angibt, um wieviel Lösung A besser ist als Lösung B.

Referenzmaße vergleichen zunächst beide Lösungen gegen eine Referenzlösung. Der Qualitätsvergleich von A und B wird anschließend mit Hilfe der beiden daraus resultierenden Werte durchgeführt.

Direkte Maße können als Referenzmaße eingesetzt werden, Referenzmaße jedoch nicht grundsätzlich als Direkte Maße.¹⁵ Nach Knowles (2002) erzeugen direkte Maße nicht immer eine vollständige Ordnung der in den

¹³Vgl. dazu Kim u. a. (2001, S.115f.).

¹⁴Vgl. dazu Knowles (2002, S.72f.).

¹⁵Vgl. Knowles (2002, S.72). Daß Referenzmaße nicht immer als Direkte Maße verwendet werden können, liegt insbesondere an der bei einigen Maßen definierten Referenz. So ist die Error Ratio ursprünglich speziell auf die Effiziente Front $E(z)$ als Referenz definiert (vgl. die Darstellung der "Error Ratio" in Abschnitt 4.2.1.2 sowie Van Veldhuizen (1999, S.6-14)).

Vergleich aufgenommenen Verfahren,¹⁶ wohl aber die unabhängigen Maße und Referenzmaße.

4.1.2.2 Die TMPS als Referenzlösung

Mitunter findet sich in der Literatur das Vorgehen, eine heuristisch effiziente Menge um so besser zu beurteilen, je größer die für die Zielfunktionswerte aufgezeigten Variationsbreiten sind,¹⁷ oder je mehr Elemente in der heuristisch effizienten Menge enthalten sind. Ebenso wird häufig gefordert, daß eine gleichmäßige Verteilung der Elemente der heuristisch effizienten Front im Zielraum besser sei, als das Auftreten von Clustern.¹⁸ Dieses Vorgehen ist nicht dazu geeignet, die Qualität einer Lösung korrekt zu beurteilen, da es eine Beurteilung ohne Berücksichtigung der Eigenschaften der Menge aller effizienten Lösungen $E(\Pi, z)$ vornimmt.¹⁹ Angenommen die Menge aller effizienten Lösungen hat nur sehr wenige Elemente oder es liegt zufällig eine perfekte Lösung vor, dann ist die Forderung nach möglichst vielen Elementen in der zu beurteilenden heuristisch effizienten Menge nicht angebracht. Für die Größe der aufgezeigten Variationsbreiten der Zielfunktionswerte gilt das gleiche. Auch sie sollten bezüglich ihrer Relation zu den Variationsbreiten der Zielfunktionswerte der Elemente der Menge aller effizienten Lösungen beurteilt werden und nicht grundsätzlich maximal sein. Ebenso ist es durchaus denkbar, daß die effiziente Front $E(z)$ Cluster aufweist. Würde grundsätzlich gefordert, daß die Elemente der heuristisch effizienten Front im Zielraum gleichmäßig verteilt sein sollen, so würde eine heuristisch effiziente Front, welche ebenfalls Cluster aufweist, schlechter bewertet, als wenn sie eine gleichmäßige Verteilung aufweisen würde.

Daher konzentrieren sich Knowles (2002) und Knowles u. Corne (2002) auf Maße zur Erfassung der Distanz.²⁰ Die anderen Qualitätsaspekte bleiben unberücksichtigt, um der Gefahr der Verwendung irreführender Maße zu begegnen. Dabei wird übersehen, daß sich die Kritik nicht gegen den Sinn eines Qualitätsaspektes richtet, sondern gegen die Art und Weise, wie die Maße zu seiner Erfassung definiert und beurteilt werden.

Es sollten vielmehr die Eigenschaften der Menge aller effizienten Lösungen $E(\Pi, z)$ bei der Beurteilung berücksichtigt werden und somit eine

¹⁶Vgl. Knowles (2002, S.73). Z.B. gehört das später vorgestellte C-Maß zu den direkten Maßen, die keine vollständige Ordnung erzeugen können (vgl. Knowles (2002, S.86)).

¹⁷Vgl. Zitzler u. a. (2000, S.179).

¹⁸Vgl. Zitzler u. a. (2000, S.179).

¹⁹Auf dieses Problem haben auch Knowles (2002, S.70) und Knowles u. Corne (2002, S.711) hingewiesen.

²⁰Vgl. hierzu und den folgenden Satz Knowles (2002, S.70) bzw. Knowles u. Corne (2002, S.711).

heuristisch effiziente Menge direkt mit dieser als Referenzlösung beurteilt werden. Die Menge aller effizienten Lösungen ist in der Regel jedoch unbekannt. Ein Ausweg ist darin zu finden, daß die beste bekannte Approximation als Referenz verwendet wird. In Farhang-Mehr u. a. (2001) wird diese beste bekannte Approximation als "Total Meta Pareto Set" (TMPS) bezeichnet und ist wie folgt definiert:²¹

$$TMPS = \bigcup_{l=1(1)L} MPS_l \quad (4.1)$$

Dabei ist MPS_l die Menge der sogenannten "Meta Pareto Solutions" der heuristisch effizienten Menge $l(NDS_l)$, $l = 1(1)L$:

$$MPS_l = \left\{ \pi \mid \nexists \pi' \in \bigcup_{w=1(1)L} NDS_w; z(\pi') \leq z(\pi); \pi' \neq \pi; \pi \in NDS_l \right\} \quad (4.2)$$

Die MPS_l wird also von den Elementen der NDS_l gebildet, die von keinem anderen Element, der $l = 1(1)L$ zur Berechnung der $TMPS$ berücksichtigten heuristisch effizienten Mengen, dominiert werden. Es sind somit die Elemente, welche die heuristisch effiziente Menge $l(NDS_l)$ in die $TMPS$ einstellt.

Die "Total Meta Pareto Front" ($TMPF$) ist definiert als Abbildung der $TMPS$ im Zielraum:

$$TMPF = \{z(\pi) \mid \pi \in TMPS\} \quad (4.3)$$

Abbildung 4.2 zeigt ein kleines grafisches Beispiel zur Ermittlung der $TMPS$.

Es werden die $L = 3$ heuristisch effizienten Mengen $NDS_A = \{A; B; C\}$, $NDS_B = \{D; E; F\}$ und $NDS_C = \{G; H\}$ zur Ermittlung einer $TMPS$ verwendet. Die Mengen der Meta Pareto Solutions sowie die $TMPS$ ergeben sich wie in Abbildung 4.2 angegeben. Da die Auftragsfolgen A und D den gleichen Zielfunktionsvektor besitzen enthält die $TMPF$ ein Element weniger als die $TMPS$ und ergibt sich zu:

$$TMPF = \left\{ \begin{pmatrix} 1 \\ 10 \end{pmatrix}; \begin{pmatrix} 3 \\ 8 \end{pmatrix}; \begin{pmatrix} 7 \\ 3 \end{pmatrix}; \begin{pmatrix} 10 \\ 1 \end{pmatrix} \right\}.$$

Soll nun zum Beispiel die Anzahl der Elemente in einer heuristisch effizienten Menge als Qualitätsmaß zu deren Beurteilung verwendet werden, so erscheint es sinnvoll, die Mächtigkeit der Menge in Relation zur Mäch-

²¹Vgl. Farhang-Mehr u. a. (2001, S.5).

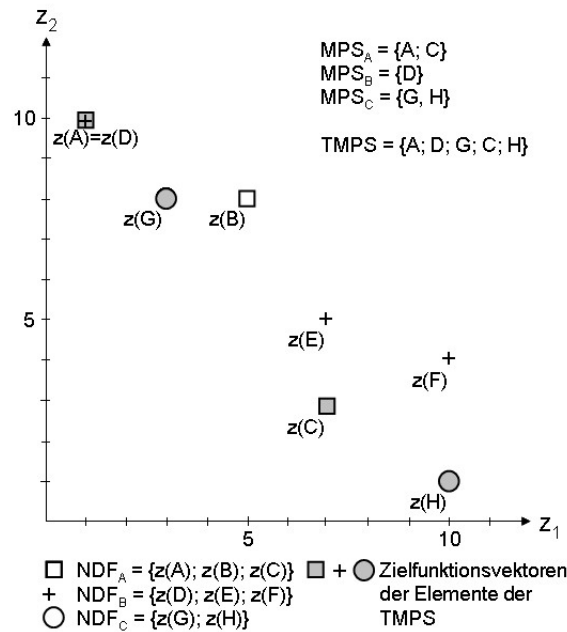


Abbildung 4.2: Ermittlung der TMPS

tigkeit der *TMPS* zu beurteilen und nicht nach einer grundsätzlichen Maximierung zu streben. Bei Verwendung der *TMPS* besteht keine Sicherheit, die Menge aller effizienten Lösungen $E(\Pi, z)$ zu betrachten. Damit kann nicht ausgeschlossen werden, daß diese Menge doch mehr Elemente enthält als die *TMPS*. Ebenso gut können es aber auch weniger sein. Gleiches gilt für die Größe der von der *TMPS* aufgezeigten Variationsbreiten der Zielfunktionswerte.

4.1.2.3 Anforderungen an geeignete Maße: Monotonie und Relativität

Auch ohne die explizite Ermittlung eines Qualitätsmaßes kann festgehalten werden, daß sich die Qualität einer heuristisch effizienten Menge immer verbessert, wenn dieser Menge ein weiteres Element hinzugefügt wird. Vor diesem Hintergrund sollen die Forderungen von Knowles (2002) diskutiert werden, daß Maße zur Qualitätsbeurteilung die Anforderungen der (schwachen) Monotonie und der (schwachen) Relativität erfüllen sollten.²² Obwohl er in seiner Arbeit nur Maße für die Erfassung der Distanz der NDS zur Menge aller effizienten Lösungen $E(\Pi, z)$ betrachtet, erscheint

²²Vgl. Knowles (2002, S.73). Die Definitionen finden sich auch in Knowles u. Corne (2002, S.712).

es sinnvoll, die Forderungen auch auf Maße für andere Qualitätsaspekte auszudehnen. Des Weiteren ist zu beachten, daß er in seiner Arbeit die heuristisch effiziente Front als Lösung betrachtet, während in dieser Arbeit heuristisch effiziente Mengen als Lösung betrachtet werden. Die von Knowles (2002) formulierten Anforderungen werden daher hier für die Anwendung auf die heuristisch effiziente Menge und die *TMPS* als Referenz angepaßt.²³

(Schwache) Monotonie: Das Hinzufügen eines Elements zur heuristisch effizienten Menge (verschlechtert nie) verbessert deren Bewertung.

Es ist absolut unerwünscht, daß ein Maß eine heuristisch effiziente Menge bezüglich eines Qualitätsaspekts besser beurteilt als die Menge aller effizienten Lösungen $E(\Pi, z)$, welche die optimale Lösung darstellt. Nach der Definition der *TMPS* (vgl. Seite 52) kann es kein Element einer heuristisch effizienten Menge geben, das ein Element der *TMPS* dominiert. Somit kann die von Knowles (2002) formulierte Relativitätsforderung hier wie folgt angepaßt werden:

(Schwache) Relativität: Die *TMPS* hat (nicht) als einzige die beste Bewertung bezüglich des verwendeten Qualitätsmaßes.

Qualitätsbeurteilungen, welche mit Hilfe von Maßen durchgeführt werden, welche diese Forderungen, zumindest in der schwachen Form, nicht erfüllen, führen zu Fehlentscheidungen bzw. zu Fehlentwicklungen, falls mit ihrer Hilfe Algorithmen entwickelt oder ihre Parameter eingestellt werden. Die in dem folgenden Abschnitt betrachteten Qualitätsmaße werden daher auf die Erfüllung dieser Eigenschaften hin überprüft. Dabei wird insbesondere bei der Überprüfung der (schwachen) Monotonie davon ausgegangen, daß sich nur eine heuristisch effiziente Menge ändert und die restlichen Mengen, welche zur Bestimmung der *TMPS* herangezogen werden, unverändert bleiben. Offensichtlich bedeutet ein Erfüllen der Forderungen nach Monotonie und Relativität durch ein Maß nicht automatisch, daß Fehlentscheidungen bei seiner Verwendung ausgeschlossen sind. Es ist zu beachten, daß ein Maß grundsätzlich Schwächen bei dem von ihm erfaßten Qualitätsaspekt haben kann. Als Beispiel soll ein Maß, welches die Verteilung der Elemente einer heuristisch effizienten Menge im Zielraum im Vergleich zu denen der *TMPS* erfasst, betrachtet werden. Angenommen

²³Vgl. für die Anforderungen Knowles (2002, S.73) oder auch Knowles u. Corne (2002, S.712).

Das Ersetzen der effizienten Front $E(z)$ durch die *TMPS* stellt keine Einschränkung dar. Die *TMPS* entspricht der Menge aller effizienten Lösungen $E(\Pi, z)$, falls diese bekannt ist und im Vergleich berücksichtigt wird. Damit ist dann auch die effiziente Front $E(z)$ bekannt.

sowohl die heuristisch effiziente Menge als auch die $TMPS$ weisen Cluster auf, so kann die Forderung der Relativität und Monotonie vom Maß durchaus erfüllt werden. Dennoch muss nicht sichergestellt sein, dass die Cluster der beiden Mengen im gleichen Bereich des Zielraums liegen.

4.2 Beurteilung von Qualitätsmaßen

4.2.1 Distanz zur Menge aller effizienten Lösungen

4.2.1.1 Kompatibilität mit den Outperformance Relationen als zusätzliche Anforderung

Den zentralen Qualitätsaspekt stellt zweifelsfrei die Nähe einer heuristisch effizienten Menge zur Menge aller effizienten Lösungen $E(\Pi, z)$ dar. Damit ein Maß überhaupt geeignet sein kann, die Entfernung korrekt zu erfassen, sollte es grundsätzlich in der Lage sein eine Lösung, die eine andere vollständig dominiert, besser oder zumindest nicht schlechter zu bewerten. Durch diese Überlegung kann allgemein gefordert werden, daß die Maße mit den von Hansen u. Jaszkiwicz (1998) formulierten drei Outperformance Relationen kompatibel sein müssen.²⁴

Schwache (weak) Outperformance Relation: NDF_A dominiert NDF_B schwach (AR_WB), falls jedes Element aus NDF_B von einem Element aus NDF_A dominiert wird oder gleich einem Element aus NDF_A ist. Weiterhin muß gelten, daß in NDF_A mindestens ein Element enthalten ist, das nicht in NDF_B enthalten ist. Abbildung 4.3 zeigt ein Beispiel für die Situation AR_wB .²⁵ Werden nur zwei heuristisch effiziente Mengen ($L = 2$) miteinander verglichen so gilt offenbar: $TMPF = NDF_A$.

Starke (strong) Outperformance Relation: Es gilt $AR_S B$, wenn es für jedes Element aus NDF_B ein Element in NDF_A gibt, das gleich diesem ist oder dieses dominiert. Mindestens ein Element aus NDF_B wird von einem Element aus NDF_A dominiert. In Abbildung 4.4 ist ein Beispiel für $AR_S B$ dargestellt. Werden nur zwei heuristisch effiziente Mengen ($L = 2$) miteinander verglichen so gilt offenbar: $TMPF = NDF_A$ und $NDF_B \setminus TMPF \neq \emptyset$.

²⁴Diesem Argument folgt auch Knowles (2002, S. 70f.), bzw. Knowles u. Corne (2002, S.712).

Die Forderung, nach Kompatibilität mit den drei Outperformance Relationen muss nur von Maßen, welche die Distanz zur Menge aller effizienten Lösungen messen wollen, erfüllt sein. Für die weiteren in dieser Arbeit betrachteten Qualitätsaspekte sind sie nicht relevant, da sie unabhängig vom Vorliegen (heuristischer) Effizienz sind.

²⁵Vgl. zu dieser und den folgenden Outperformance Relationen Hansen u. Jaszkiwicz (1998, S. 6-8).

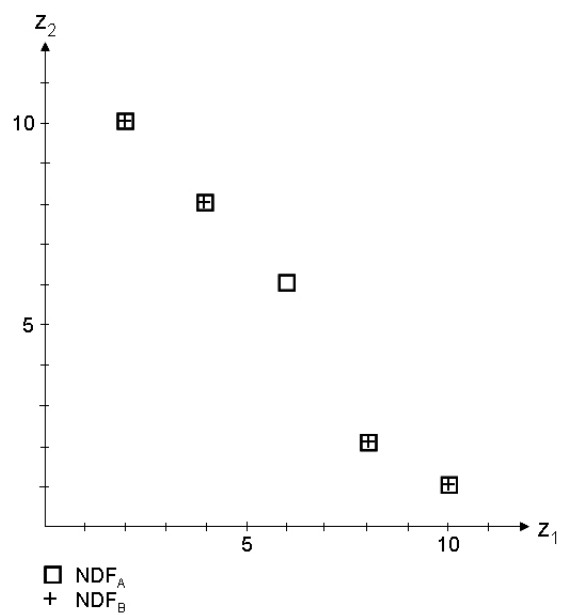


Abbildung 4.3: Schwache Outperformance

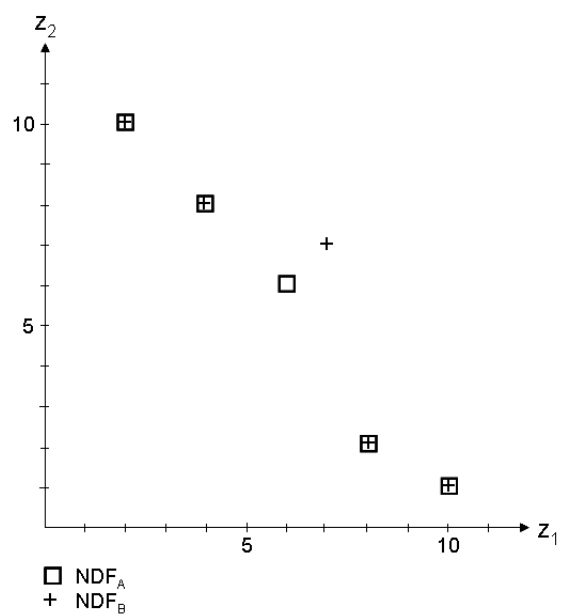


Abbildung 4.4: Starke Outperformance

Vollständige (complete) Outperformance Relation: Es gilt AR_{CB} , falls jedes Element aus NDF_B von mindestens einem Element aus NDF_A dominiert wird. Abbildung 4.5 gibt diese Situation wieder. Werden nur zwei heuristisch effiziente Mengen ($L = 2$) miteinander verglichen, so gilt offenbar: $TMPF = NDF_A$ sowie $NDF_B \cap TMPF = \emptyset$.

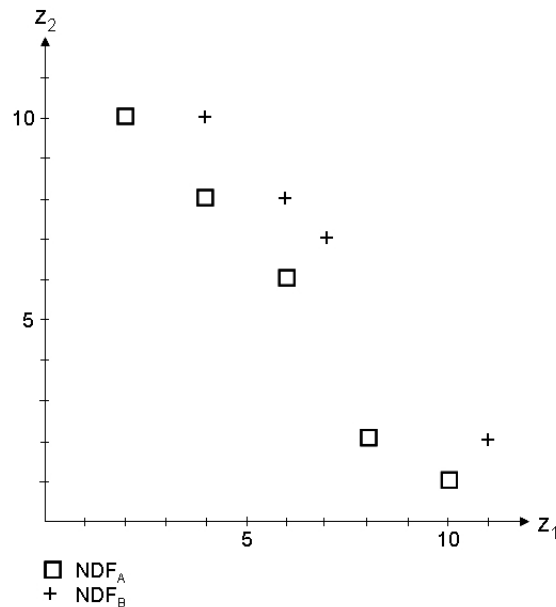


Abbildung 4.5: Vollständige Outperformance

Offensichtlich ist die vollständige Outperformance Relation ein Sonderfall der starken Outperformance Relation und diese wiederum ist ein Sonderfall der schwachen Outperformance Relation.²⁶

Die drei Relationen sind keine eigenständigen Maße und erfassen nicht alle Möglichkeiten, die bezüglich der Dominanz zwischen zwei heuristisch effizienten Fronten auftreten können. So wird insbesondere der Fall, in dem sich die Mengen in verschiedenen Bereichen des Zielraums wechselseitig dominieren, nicht erfaßt. Ein Beispiel für einen solchen Fall ist in Abbildung 4.6 dargestellt. Dort dominiert weder NDF_A die NDF_B noch umgekehrt. Dieser Fall wird von den Outperformance Relationen nicht erfaßt, in dem Sinne, daß die schwache Outperformance Relation nicht erfüllt ist und damit auch nicht die starke und vollständige Outperformance Relation.

²⁶Vgl. Hansen u. Jaszkiwicz (1998, S.8).

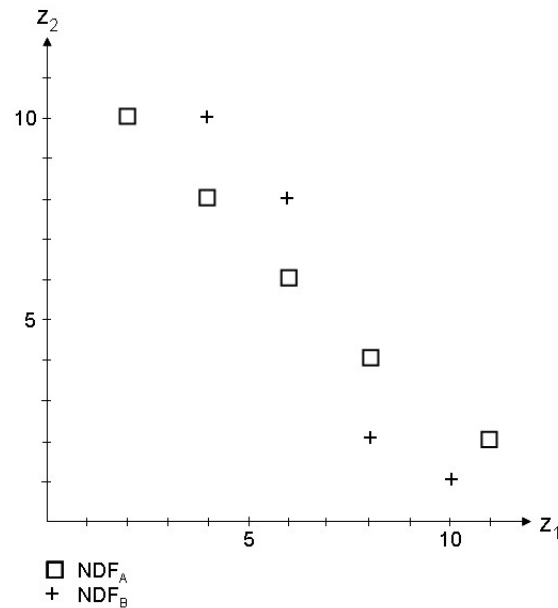


Abbildung 4.6: Gegenseitige Dominanz

Des Weiteren ist zu berücksichtigen, daß die Zuordnung zwischen den Elementen einer heuristisch effizienten Menge und denen der heuristisch effizienten Front nicht eineindeutig ist. Grundsätzlich ist es möglich, daß mehrere Auftragsfolgen den gleichen Zielfunktionsvektor aufweisen.²⁷ Dennoch kann der Argumentation von Knowles (2002) bzw. Knowles u. Corne (2002) gefolgt werden, daß Maße, die nicht in der Lage sind die drei Outperformance Relationen korrekt zu erfassen, nicht zur Messung der Distanz der heuristisch effizienten Menge zur Menge aller effizienten Lösungen und einen darauf aufbauenden Vergleich von heuristisch effizienten Lösungen geeignet sind.²⁸ Hansen u. Jaszkiwicz (1998) haben die Kompatibilität von Vergleichsmethoden mit den Outperformance Relationen definiert:²⁹

Schwache Kompatibilität: Ein Maß ist dann schwach kompatibel mit einer Outperformance Relation R_O , wenn bei Gültigkeit der Relation AR_OB , das Maß NDF_A nicht schlechter bewertet als NDF_B .³⁰

Kompatibilität: Ein Maß ist dann kompatibel mit einer Outperformance

²⁷Auf dieses Problem weist auch Van Veldhuizen (1999, S.6-13) hin.

²⁸Vgl. Knowles (2002, S.71) bzw. Knowles u. Corne (2002, S.712).

²⁹Vgl. Hansen u. Jaszkiwicz (1998, S.10).

³⁰ R_O steht hier allgemein für eine der drei Outperformance Relationen, daher: $R_O \in \{R_W, R_S, R_C\}$.

Relation, wenn bei Gültigkeit der Relation $AR_{\circ}B$, das Maß NDF_A besser bewertet als NDF_B .

Die Definition der schwachen Outperformance Relation umfaßt die Fälle der starken und die Definition der starken Outperformance Relation die Fälle der vollständigen Outperformance Relation. Die Kompatibilitätseigenschaften eines Maßes können aufgrund dieses Zusammenhangs von der schwachen über die starke hin zur vollständigen Outperformance Relation nicht abnehmen. Ist also z.B. ein Maß schwach kompatibel mit der schwachen Outperformance Relation, so ist es auch mindestens schwach kompatibel mit der starken und vollständigen Outperformance Relation. Es ist aber möglich, daß der Grad der Kompatibilität zunimmt.

Bei der in den folgenden Abschnitten durchgeführten Beurteilung von Qualitätsmaßen wird unterstellt, daß keine zwei oder mehrere Elemente der heuristisch effizienten Menge einen gleichen Zielfunktionsvektor besitzen. Sollte ein Maß bei Gültigkeit der Annahme nicht kompatibel sein, so wird es auch nach dem Aufheben der Annahme nicht kompatibel sein. Ebenso kann durch das Fallenlassen der Annahme aus einer schwachen Kompatibilität keine Kompatibilität werden. Es ist jedoch möglich, daß sich die Kompatibilitätseigenschaft eines Maßes abschwächt oder nicht mehr existiert, falls die Annahme aufgehoben wird. Somit wird ein Maß durch diese Annahme tendenziell besser beurteilt.

Die Kompatibilitätseigenschaften eines Maßes hängen von der Anzahl L der im Vergleich berücksichtigten heuristisch effizienten Mengen ab. Daher werden im folgenden Abschnitt 4.2.1.2 zwei Fälle unterschieden. Im ersten Fall ($L = 2$) werden nur zwei heuristisch effiziente Mengen im Vergleich und bei der Berechnung der $TMPS$ verwendet. Im zweiten Fall ($L \geq 3$) werden drei und mehr heuristisch effiziente Mengen im Vergleich und bei der Berechnung der $TMPS$ verwendet.

4.2.1.2 Beurteilung von Maßen

Number of non dominated solutions (Schaffer (1985)): In seinem Experiment hat Schaffer (1985) heuristisch effiziente Mengen mit Hilfe ihrer Mächtigkeit verglichen.³¹

$$NNDS_l = |TMPS \cap NDS_l| = |MPS_l| \quad (4.4)$$

Je größer der Wert von $NNDS_l$ ist, desto besser wird die ausgewertete Lösung beurteilt.

³¹Vgl. Schaffer (1985, S.96).

Im Fall $L = 2$ ist $NNDS_l$ mit allen drei Outperformance Relationen kompatibel.³² Bei nur zwei betrachteten heuristisch effizienten Mengen gilt aufgrund der Definition der schwachen Outperformance Relationen im Fall AR_wB immer $TMPS = MPS_A = NDS_A$ und somit: $NNDS_A = |TMPS| = |MPS_A| = |NDS_A|$. Aufgrund der Definition gilt offensichtlich ebenfalls immer: $|MPS_B| < |MPS_A|$. Daraus folgt dann, daß $NNDS_A > NNDS_B$ gilt. Da die starke und vollständige Outperformance Relation Sonderfälle der schwachen Form sind, muß auch bei Ihnen die Kompatibilitätseigenschaft erfüllt sein.

Im Fall $L \geq 3$ ist $NNDS_l$ mit allen drei Outperformance Relationen nur noch schwach kompatibel.³³

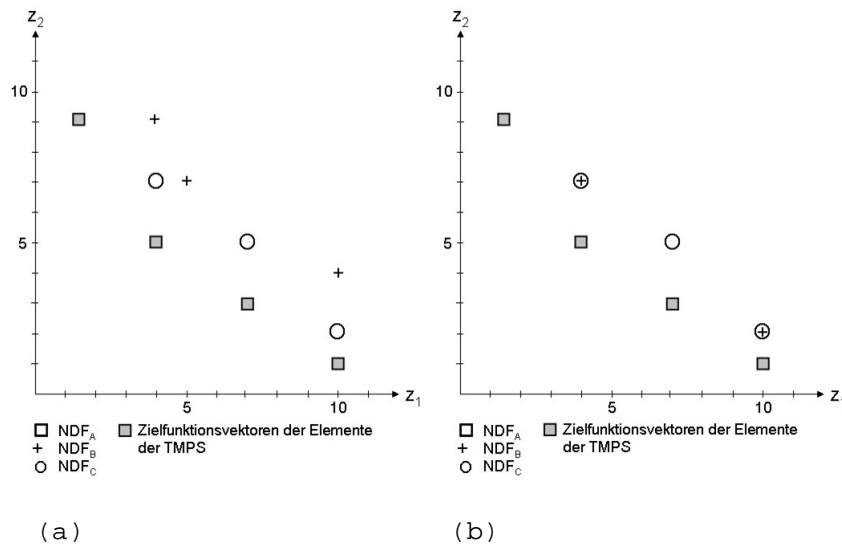


Abbildung 4.7: Number of non dominated Solutions, Fall: $L \geq 3$

Es soll zunächst die vollständige Outperformance Relation betrachtet werden. Gegeben sei der Fall AR_CB , CR_CB und AR_CC , wie in Abbildung 4.7 (a) dargestellt. Es ist grundsätzlich nicht möglich, daß NDS_C eine schlechtere Bewertung als NDS_B erhält. Da aber $TMPS = MPS_A = NDS_A$ und $MPS_B = MPS_C = \emptyset$ gilt, ergibt sich $NNDS_B = NNDS_C = 0$. Obwohl also CR_CB gilt, wird NDS_C damit nicht besser bewertet als NDS_B . Somit besteht schwache Kompatibilität. Da die vollständige Outperformance Relation ein Sonderfall der starken und schwachen Outperformance Relation ist, kann bei diesen höchstens

³² $L = 2$, d.h. es werden nur zwei heuristisch effiziente Mengen zur Berechnung der $TMPS$ herangezogen.

³³ $L = 3$, d.h. es werden 3 oder mehr heuristisch effiziente Mengen zur Berechnung der $TMPS$ herangezogen.

ebenfalls eine schwache Kompatibilität existieren. Im Fall der schwachen Outperformance Relation kann man sich leicht überlegen, daß es unabhängig von anderen bei der Berechnung der *TMPS* berücksichtigten Mengen nicht möglich ist, daß im Fall CR_{WB} NDS_B mehr Elemente in die *TMPS* einstellt als NDS_C . Damit besteht zumindest schwache Kompatibilität, da NDS_C nie schlechter bewertet werden kann als NDS_B . Gilt nun aber z.B. zusätzlich AR_{CB} und AR_{CC} , wie in Abbildung 4.7 (b) dargestellt, so stellen weder NDS_B noch NDS_C Elemente in die *TMPS* ein. Damit gilt $MPS_B = MPS_C = \emptyset$ und daher auch $NNDS_B = NNDS_C = 0$. Es ist also möglich das NDS_C zumindest gleich gut bewertet wird wie NDS_B . Es besteht daher nur schwache Kompatibilität. Da eine schwache Kompatibilität sowohl mit der schwachen als auch mit der vollständigen Outperformance Relation besteht, muß auch eine schwache Kompatibilität mit der starken Outperformance Relation bestehen. Es besteht daher eine schwache Kompatibilität mit allen drei Outperformance Relationen.

Das Maß erfüllt die Anforderung der Relativität in dem Sinne, daß es für keine heuristisch effiziente Menge $NDS_i \neq TMPS$ möglich ist, einen größeren oder gleichen Wert anzunehmen als die *TMPS*. Es ist prinzipiell möglich, daß der Wert von $NNDS_i$ durch das Einfügen eines Elements sinken kann, dann würde sich aber auch die *TMPS* verändern und in ihrer Mächtigkeit sinken. Da aber der Wert von $NNDS_i$ sinken kann und das Maß an sich nach einer Maximierung strebt, ohne den Wert der *TMPS* zu berücksichtigen, erfüllt das Maß nicht die Anforderung der Monotonie.

Error Ratio (Van Veldhuizen (1999)) / Inferiority Index (Farhang-Mehr u. a. (2001)): Die von Van Veldhuizen (1999) vorgestellte Variante der "Error Ratio" ermittelt den Anteil der Elemente der heuristisch effizienten Front *NDF*, die nicht in der effizienten Front $E(z)$ enthalten sind:³⁴

$$ER_i^{VEL} = \frac{|NDF_i| - |MPF_i|}{|NDF_i|} \quad (4.5)$$

Van Veldhuizen (1999) merkt selber an, daß anstelle der *NDF* und der effizienten Front auch die heuristisch effiziente Menge und die Menge aller effizienten Lösungen $E(\Pi, z)$ bei der Berechnung verwendet werden können.³⁵ Wird die Menge aller effizienten Lösungen, durch die *TMPS* ersetzt, dann ergibt sich die in dieser Arbeit betrachtete Variante der "Error Ratio" zu:

³⁴Vgl. Van Veldhuizen (1999, S.6-14).

³⁵Vgl. Van Veldhuizen (1999, S.6-20).

$$ER_l = \frac{|NDS_l| - |MPS_l|}{|NDS_l|} = 1 - \frac{|MPS_l|}{|NDS_l|} \quad (4.6)$$

In dieser Form entspricht das Maß dem von Farhang-Mehr u. a. (2001) vorgestellten "Inferiority Index" ($InfI_l$).³⁶

Ein Wert von Null ist die beste Bewertung und bedeutet, daß alle Elemente von NDS_l in der $TMPS$ enthalten sind.³⁷ Die schlechteste Bewertung ist Eins. In diesem Fall ist keines der Elemente aus NDS_l in der $TMPS$ enthalten.

Im Fall $L = 2$ ist ER_l bzw. $InfI_l$ mit der vollständigen und starken Outperformance Relation kompatibel.³⁸ Aufgrund der Definition der beiden Outperformance Relationen gilt bei $AR_C B$ und $AR_S B$ immer: $TMPS = MPS_A = NDS_A$. Daher ist ER_A immer Null, weil der Term $|NDS_A| - |MPS_A|$ immer den Wert Null annimmt. ER_B muß hingegen immer größer Null sein, da aufgrund der Definition der vollständigen und starken Outperformance Relation $NDS_B \setminus TMPS \neq \emptyset$ und daher $|NDS_B| > |MPS_B|$ und damit $\frac{|MPS_B|}{|NDS_B|} < 1$ gilt. Es gilt somit $ER_B > 0$. Damit wird NDS_A immer besser bewertet als NDS_B .

Bezüglich der schwachen Outperformance Relation ist ER_l schwach kompatibel, da hier im Fall $AR_W B$ $NDS_B \cap TMPS = NDS_B$ gelten (vgl. dazu Abbildung 4.3 auf Seite 56) und damit $MPS_B = NDS_B$ gelten würde. Damit ergibt sich $ER_B = 0$. Da weiterhin $ER_A = 0$ gilt, ist eine bessere Bewertung von NDS_B als NDS_A ausgeschlossen.

Im Fall $L \geq 3$ ist ER_l mit der vollständigen Outperformance Relation schwach kompatibel.

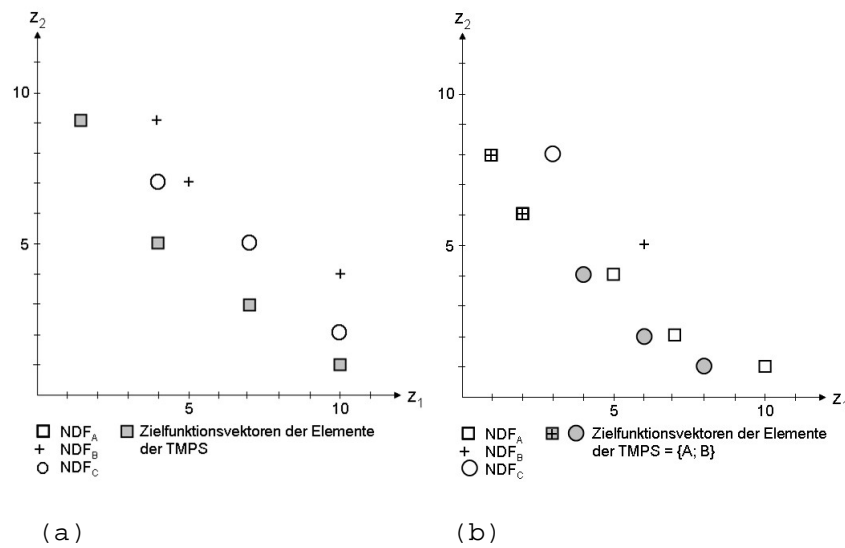
Gilt z.B. $AR_C C$, $CR_C B$, $AR_C B$ (vgl. Abbildung 4.8 (a)), dann ist $|MPS_B| = |MPS_C| = \emptyset$ und daher $ER_B = ER_C = 1$. Daß bei Gültigkeit von $CR_C B$ ein Element aus NDS_B in die $TMPS$ eingestellt wird, ist, unabhängig von den andern bei der Berechnung der $TMPS$ berücksichtigten heuristisch effizienten Mengen, nicht möglich. Daher kann NDS_B grundsätzlich keine bessere Bewertung erhalten als NDS_C .

Für die starke Outperformance Relation besteht keine Kompatibilität

³⁶Vgl. Farhang-Mehr u. a. (2001, S.7).

³⁷Vgl. zu dieser und zur folgenden Aussage die Interpretation in Van Veldhuizen (1999, S.6-15) oder Knowles (2002, S.75).

³⁸Knowles (2002, S.75-77 u. 92) betrachtet ebenfalls die "Error Ratio", wobei er die Version, in welcher die NDF und die effiziente Front $E(z)$ zur Berechnung verwendet werden betrachtet. Er stellt in seiner Arbeit fest, daß die "Error Ratio" schwach kompatibel mit der vollständigen Outperformance Relation ist. Bezüglich der schwachen und starken Outperformance Relation besteht keine Kompatibilität. (vgl. Knowles (2002, S.75)). In seiner Tabelle 3.2 auf S.92 ist jedoch eine fehlende Kompatibilität für alle drei Outperformance Relationen eingetragen. Auch in Tabelle 1 in der Arbeit von Knowles u. Corne (2002, S.716) wird eine fehlende Kompatibilität mit allen drei Outperformance Relationen angegeben. Ein stichhaltiger Beweis für diese Aussagen fehlt jedoch in beiden Arbeiten.

Abbildung 4.8: Error Ratio, Fall: $L \geq 3$

mehr, auch keine schwache. Hier ist es möglich, daß $ER_B < ER_A$ resultiert, obwohl $AR_S B$ gilt. Ein Beispiel für einen solchen Fall ist in Abbildung 4.8 (b) angegeben. Dort gilt: $ER_B = 1 - \frac{2}{3} = \frac{1}{3}$ und $ER_A = 1 - \frac{2}{5} = \frac{3}{5}$. Da keine Kompatibilität mit der starken Outperformance Relation besteht, kann auch keine mit der schwachen Form bestehen.

Das Maß erfüllt in beiden Fällen die Anforderung der Relativität, da nur falls $NDS_l = TMPS$ gilt $ER_l = 1$ ist. Die Eigenschaft der Monotonie ist hingegen nicht erfüllt, da es grundsätzlich möglich ist, daß ein Element eingefügt wird, welches kein bis dahin in der Lösung enthaltenes Element verdrängt und auch nicht in die MPS_l geht. Dieser Einfügevorgang hat jedoch zu Folge, daß der Wert der "Error Ratio" ER_l steigt. Würde z.B. in Abbildung 4.8 (b) in NDS_B eine Element mit dem Zielfunktionsvektor $\begin{pmatrix} 8 \\ 3 \end{pmatrix}$ eingefügt, so würde ER_B von $\frac{1}{3}$ auf $1 - \frac{2}{4} = \frac{1}{2}$ steigen.

Proportions of efficient Solutions (Tuyttens u. a. (2000)): Dieses Maß ermittelt den Anteil der Elemente der heuristisch effizienten Menge, die in der Menge aller effizienten Lösungen enthalten sind.³⁹ Für die Zwecke dieser Arbeit wird anstelle der heuristisch effizienten Menge $E(\Pi, z)$ erneut die $TMPS$ verwendet.

³⁹Vgl. Tuyttens u. a. (2000, S.305).

$$PES_l = \frac{|MPS_l|}{|TMPS|} \quad (4.7)$$

Ein Wert von Eins bedeutet, daß alle Elemente aus NDS_l in der $TMPS$ enthalten sind. Ein Wert von Null zeigt an, daß NDS_l von der $TMPS$ vollständig dominiert wird.⁴⁰

Im Fall $L = 2$ ist das Maß mit allen drei Outperformance Relationen kompatibel. Es genügt zu zeigen das PES_l mit der schwachen Outperformance Relation kompatibel ist. Aufgrund der Definition der schwachen Outperformance Relation gilt im Fall AR_WB : $TMPS = MPS_A = NDS_A$ und damit $PES_A = 1$. Weiterhin muß aufgrund der Definition von R_W immer $|MPS_B| < |TMPS|$ gelten. Damit gilt immer: $PES_B < 1$.

Im Fall $L \geq 3$ ist das Maß mit allen drei Outperformance Relationen nur noch schwach kompatibel.

Betrachtet man die schwache Form, so ist im Fall CR_WB aufgrund der Definition von R_W sofort ersichtlich, daß unabhängig von anderen Lösungen die NDS_B nie mehr Elemente in die $TMPS$ einstellen kann als NDS_C und somit nie besser bewertet werden kann als NDS_C . Jedoch wird NDS_C nicht zwingend besser bewertet. Dies ist z.B. dann der Fall, wenn zusätzlich zu CR_WB AR_CB und AR_CC gilt (vgl. Abbildung 4.9 (a)). In diesem Fall gilt $|MPS_B| = |MPS_C| = 0$ und daher $PES_B = PES_C = 0$.

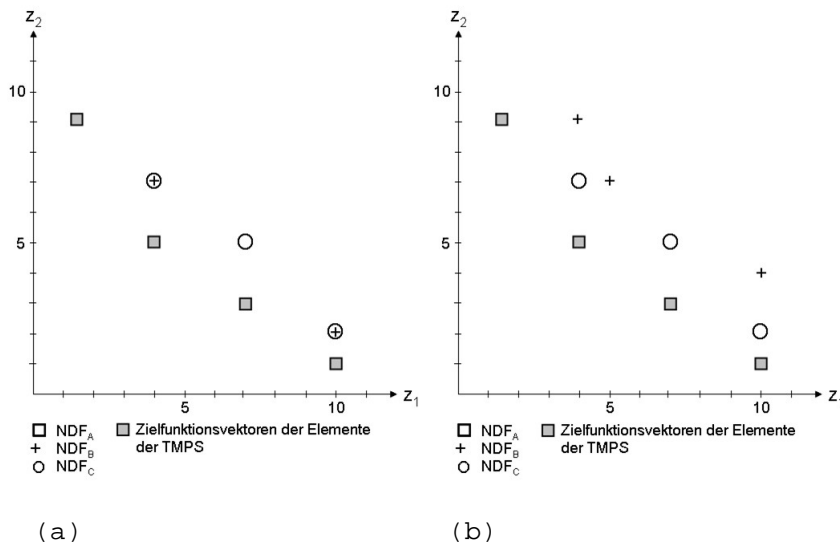


Abbildung 4.9: Proportions of efficient Solutions, Fall: $L \geq 3$

⁴⁰In der Arbeit von Tuyttens u. a. (2000, S.305) wird nicht direkt angegeben, wie dieses Maß zu interpretieren ist. Die hier angegebene Interpretation erscheint aus der Argumentation von Tuyttens u. a. (2000) heraus jedoch sinnvoll.

Im Fall der vollständigen Outperformance Relation kann im Fall $CR_C B$ aufgrund der Definition von $R_C NDS_B$ keine Elemente in die $TMPS$ einstellen und somit nie besser bewertet werden als NDS_C . Jedoch wird NDS_C nicht zwingend besser bewertet. Dies ist z.B. dann der Fall, wenn zusätzlich zu $CR_C B$ $AR_C B$ und $AR_C C$ gilt (vgl. Abbildung 4.9 (b)). In diesem Fall gilt $|MPS_B| = |MPS_C| = 0$ und daher $PES_C = PES_C = 0$. Da bezüglich der schwachen und vollständigen Outperformance Relation schwache Kompatibilität besteht, muß sie auch bezüglich der starken Outperformance Relation bestehen.

Das Maß erfüllt die Anforderung der Relativität. Nur wenn $NDS_l = TMPS$ gilt, ist $PES_l = 1$. Die Monotonieeigenschaft wird dagegen nicht erfüllt. Würde in Abbildung 4.4 auf Seite 56 in NDS_B ein Element eingefügt, dessen Zielfunktionsvektor dem Punkt (1/8) entspricht, so würde sich die Bewertung von NDS_B von vorher 4/5 auf 3/4 verschlechtern.

Durchschnittliche und maximale Distanz (Czyzak u. Jaszkiwicz (1998)): Czyzak u. Jaszkiwicz (1998) haben mit $Dist1_l$ ein Maß vorgestellt, daß die durchschnittliche Distanz einer heuristisch effizienten Menge von einer Referenzmenge (Ref)⁴¹ ermittelt. $Dist2_l$ gibt die größte Distanz eines Elements der Referenzmenge zum jeweils über $c(x, y)$ definierten nächsten Element aus NDS_l an.⁴² Die beiden Maße bestimmen sich wie folgt:⁴³

$$Dist1_l = \frac{1}{|Ref|} \sum_{y \in Ref} \left\{ \min_{x \in NDS_l} \{c(x, y)\} \right\} \quad (4.8)$$

$$Dist2_l = \max_{y \in Ref} \left\{ \min_{x \in NDS_l} \{c(x, y)\} \right\} \quad (4.9)$$

mit

$$c(x, y) = \max_{k=1(1)K} \left\{ 0; \frac{1}{\Delta_k} (z_k(x) - z_k(y)) \right\} \quad (4.10)$$

wobei Δ_k die Variationsbreite des Ziels k bezüglich der Referenzmenge Ref ist, also:

⁴¹Czyzak u. Jaszkiwicz (1998) nennen als Beispiele für Ref die Menge aller effizienten Lösungen $E(\Pi, z)$ oder ihre beste bekannt Approximation (vgl. Czyzak u. Jaszkiwicz (1998, S.41).

⁴²Vgl. Czyzak u. Jaszkiwicz (1998, S.41).

⁴³Vgl. Czyzak u. Jaszkiwicz (1998, S.41).

In den folgenden Darstellungen werden wegen der Problemunabhängigkeit der Darstellung nicht π und π' (Symbolik für Auftragsfolgen) sondern allgemein x und y als Elemente von Lösungs- oder Referenzmengen verwendet.

$$\Delta_k = \left\{ \max_{x \in Ref} \{z_k(x)\} - \min_{x \in Ref} \{z_k(x)\} \right\} \quad (4.11)$$

Czyzak u. Jaszkiwicz (1998) beurteilen eine Lösung um so besser, je kleiner die Werte von $Dist1_l$ bzw. $Dist2_l$ sind.⁴⁴ Des weiteren ist zu beachten, daß Czyzak u. Jaszkiwicz (1998) in ihrer Arbeit ein Maximierungsproblem betrachten, so daß Gleichung 4.10 für die Betrachtung von Minimierungsproblemen hier entsprechend angepaßt wurde.⁴⁵

Das Problem bei der vorgenommenen Normierung ist, daß in den zu beurteilenden heuristisch effizienten Mengen, für eine Zielgröße größere Werte als der Größte in der Referenz und kleinere Werte als der Kleinste in der Referenz auftreten können. Des weiteren haftet den Maßen in dieser Form das Problem an, daß sie im Falle eines $\Delta_k = 0$ nicht mehr definiert sind. Dieser Fall ist jedoch nicht grundsätzlich ausgeschlossen.

In der folgenden Analyse soll die $TMPS$ als Referenz benutzt werden. Somit ergeben sich:

$$D1_l = \frac{1}{|TMPS|} \sum_{y \in TMPS} \left\{ \min_{x \in NDS_l} \{c(x, y)\} \right\} \quad (4.12)$$

und

$$D2_l = \max_{y \in TMPS} \left\{ \min_{x \in NDS_l} \{c(x, y)\} \right\} \quad (4.13)$$

Der kleinste und beste mögliche Wert von $D1_l$ und $D2_l$ ist Null. Wie man sich leicht überlegen kann, gibt es in diesem Fall kein Element in der $TMPS$, das nicht auch in der NDS_l enthalten ist. Es gilt offenbar: $TMPS = MPS_l = NDS_l$.

Im Fall $L = 2$ ist $D1_l$ mit allen drei Versionen der Outperformance Relationen kompatibel. In allen Fällen beträgt $D1_A = 0$, da $TMPS = NDS_A$. Gilt AR_WB so muß aufgrund der Definition der schwachen Outperformance Relation in NDS_A mindestens ein Element enthalten sein, daß nicht in NDS_B enthalten ist. Damit gibt es auch in der $TMPS$ mindestens ein Element, welches nicht in NDS_B enthalten ist. Für diese Elemente der $TMPS$ sind die Werte $c(x, y) > 0$, mit $x \in NDS_B$. Damit ist dann $D1_B$ größer Null.

⁴⁴Vgl. Czyzak u. Jaszkiwicz (1998, S.41).

⁴⁵Bei Maximierungsproblemen ist $c(x, y)$ wie folgt zu ermitteln: $c(x, y) = \max_{k=1(1)K} \left\{ 0; \frac{1}{\Delta_k} (z_k(y) - z_k(x)) \right\}$ (vgl. Czyzak u. Jaszkiwicz (1998, S.41)). Die bei der Minimierung zu verwendende Form kann auch aus folgender Überlegung heraus hergeleitet werden. Maximierungsprobleme können durch Multiplikation mit -1 in Minimierungsprobleme überführt werden. Somit ergibt sich: $c(x, y) = \max_{k=1(1)K} \left\{ 0; \frac{1}{\Delta_k} (-z_k(y) - (-z_k(x))) \right\} = \max_{k=1(1)K} \left\{ 0; \frac{1}{\Delta_k} (z_k(x) - z_k(y)) \right\}$.

Für $D2_l$ gelten mit analoger Begründung die gleichen Aussagen wie für $D1_l$.

Im Fall $L \geq 3$ sind $D1_l$ und $D2_l$ mit der schwachen Outperformance Relationen schwach kompatibel. Mit der starken und vollständigen Outperformance Relation sind die beiden Maße kompatibel.

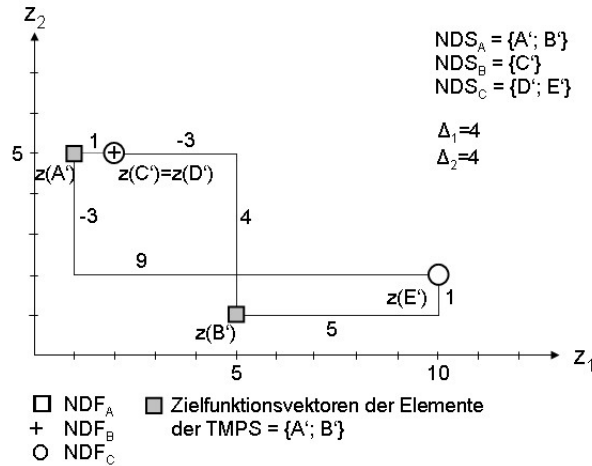


Abbildung 4.10: $D1_l$ und $D2_l$, Fall: $L \geq 3$

In Abbildung 4.10 ist ein Beispiel für die Situation $CR_w B$ dargestellt.⁴⁶ Dort gilt $D1_B = D1_C = \frac{5}{8}$ und $D2_B = D2_C = 1$.

Für die Werte von Δ_1 und Δ_2 ergeben sich:

$$\Delta_1 = 5 - 1 = 4 \text{ und } \Delta_2 = 5 - 1 = 4.$$

Berechnung von $D1_B$ und $D2_B$:

$$c(C', A') = \max \left\{ 0; \frac{2-1}{4}; \frac{5-5}{4} \right\} = \frac{1}{4}$$

$$c(C', B') = \max \left\{ 0; \frac{2-5}{4}; \frac{5-1}{4} \right\} = 1$$

$$D1_B = \left(\frac{1}{4} + 1 \right) \cdot \frac{1}{2} = \frac{5}{8}$$

$$D2_B = \max \left\{ \frac{1}{4}; 1 \right\} = 1$$

Berechnung von $D1_C$ und $D2_C$:

$$c(D', A') = \max \left\{ 0; \frac{2-1}{4}; \frac{5-5}{4} \right\} = \frac{1}{4}$$

$$c(E', A') = \max \left\{ 0; \frac{10-1}{4}; \frac{2-5}{4} \right\} = \frac{9}{4}$$

$$c(D', B') = \max \left\{ 0; \frac{2-5}{4}; \frac{5-1}{4} \right\} = 1$$

$$c(E', B') = \max \left\{ 0; \frac{10-5}{4}; \frac{2-1}{4} \right\} = \frac{5}{4}$$

⁴⁶An den grauen Linien sind die Werte für $c(x, y)$ angegeben.

$$D1_C = (\min \{ \frac{1}{4}; \frac{9}{4} \} + \min \{ 1; \frac{5}{4} \}) \cdot \frac{1}{2} = (\frac{1}{4} + 1) \cdot \frac{1}{2} = \frac{5}{8}$$

$$D2_C = \max \{ \min \{ \frac{1}{4}; \frac{9}{4} \}; \min \{ 1; \frac{5}{4} \} \} = 1$$

Daß bei Gültigkeit von CR_wB NDS_B besser bewertet wird als NDS_C , ist nicht möglich. Dazu müßte ein Element aus NDS_B näher an einem Element der $TMPS$ liegen als ein Element aus NDS_C . Dies ist bei Gültigkeit von CR_wB nicht möglich, da ein solches Element aus NDS_B das Element aus NDS_C dominieren würde.

Bezüglich der starken und vollständigen Outperformance Relation sind beide Maße kompatibel. Man kann sich leicht überlegen, daß es nicht möglich ist, daß es bei CR_sB bzw. CR_CB für ein Element der $TMPS$ ein Element aus NDS_B mit einen kleineren Wert für $c(x, y)$ gibt, als für ein Element aus NDS_C . Würde es ein solches Element geben, so würde dieses ein Element aus NDS_C dominieren.

Das Maß erfüllt sowohl für $D1_l$ und $D2_l$ die Anforderung der Relativität. Die Maße können keinen Wert kleiner Null annehmen. Null ist die beste mögliche Bewertung. Sofern für eine heuristisch effiziente Menge $NDS_l \neq TMPS$ gilt, gibt es mindestens ein Element in der $TMPS$ mit einem positiven Wert für $c(x, y)$.

Die Maße sind weiterhin schwach monoton. Dies läßt sich in Abbildung 4.10 leicht nachvollziehen. Würde dort in NDS_C ein Element mit dem Ziel-funktionsvektor $\begin{pmatrix} 15 \\ 1 \end{pmatrix}$ eingefügt, so würde sich die Bewertung von NDS_C nicht verbessern. Eine Verschlechterung der Bewertung ist nicht möglich. Ein Element, das von einem Element der $TMPS$ bzgl. $c(x, y)$ weiter entfernt liegt als das gegenwärtig am nächsten liegende Element, wird aufgrund des min Operators in der Berechnung von $D1_l$ und $D2_l$ nicht berücksichtigt. Eingefügte Elemente, die bisherige Elemente dominieren, können keinen größeren Wert für $c(x, y)$ haben, sonst würden sie das alte Element nicht dominieren.

Generational Distance (Van Veldhuizen (1999)): Dieses Distanzmaß errechnet sich wie folgt:⁴⁷

$$GD_l = \frac{\sqrt{\sum_{x \in NDS_l} d(x)^2}}{|NDS_l|} \quad (4.14)$$

⁴⁷Vgl. Van Veldhuizen (1999, S.6-15 und 6-20). Dabei berechnet Van Veldhuizen (1999) GD_l auf S. 6-15 zunächst auf Basis der heuristisch effizienten Front und der effizienten Front $E(z)$. Auf S.6-20 zeigt er dann die Variation, welche die Berechnung auf Basis der heuristisch effizienten Menge und der Menge aller effizienten Lösungen $E(\Pi, z)$ vornimmt auf. In dieser Arbeit wird $E(\Pi, z)$ durch die $TMPS$ ersetzt.

Dabei ist $d(x)$ die euklidische Distanz im Zielraum von Element $x \in NDS_l$ zum im Zielraum am nächsten liegenden Element der Menge aller effizienten Lösungen $E(\Pi, z)$. Wie dieses nächste Element ermittelt wird, wird in der Arbeit von Van Veldhuizen (1999) nicht weiter ausgeführt. Des weiteren ist die Menge aller effizienten Lösungen erneut durch die *TMPS* zu ersetzen. $d(x)$ wird für die in dieser Arbeit angestellten Überlegungen wie folgt berechnet:

$$d(x) = \min_{y \in TMPS} \left\{ \|z(x) - z(y)\|^2 \right\} \quad (4.15)$$

Der beste mögliche Wert für GD_l ist offenbar Null, falls $NDS_l = TMPS$ gilt.⁴⁸

In Abbildung 4.11 ist ein kleines Beispiel zur Berechnung der Generational Distance gegeben.

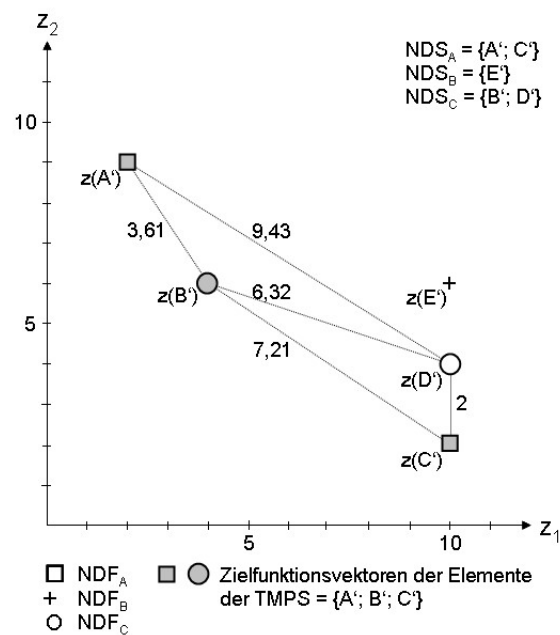


Abbildung 4.11: Berechnung der Generational Distance GD_l

Es soll die GD_C berechnet werden. Dazu sind zunächst die Werte von $d(x)$ für die Elemente aus NDS_C zu bestimmen, hier die Auftragsfolgen B und D:

$$\begin{aligned} d(B) &= \min \left\{ \sqrt{(4-2)^2 + (6-9)^2}; \sqrt{(4-4)^2 + (6-6)^2}; \sqrt{(4-10)^2 + (6-2)^2} \right\} \\ &= \min \{3,61; 0; 7,21\} \\ &= 0 \end{aligned}$$

⁴⁸Vgl. Van Veldhuizen (1999, S.6-15).

$$\begin{aligned}
d(D) &= \min \left\{ \sqrt{(10-2)^2 + (4-9)^2}; \sqrt{(10-4)^2 + (4-6)^2}; \sqrt{(10-10)^2 + (4-2)^2} \right\} \\
&= \min \{9, 43; 6, 32; 2\} \\
&= 2
\end{aligned}$$

Anschließend ergibt sich GD_C zu: $GD_C = \frac{\sqrt{0^2+2^2}}{2} = 1$.

Im Fall $L = 2$ ist GD_l mit der vollständigen und starken Outperformance Relation kompatibel.⁴⁹ In beiden Fällen ($AR_S B$ u. $AR_C B$) gilt immer $GD_A = 0$, da aufgrund der Definition von R_S und R_C alle Elemente aus NDS_A auch Element der $TMPS$ sind, so daß für alle Elemente x der NDS_A $d(x) = 0$ gilt. Nicht alle Elemente der NDS_B sind Element der $TMPS$. Für die Elemente in NDS_B , die nicht in der $TMPS$ sind, resultiert $d(x) > 0$. Daher ist $GD_B > 0$. NDS_A ist damit sowohl bei $AR_S B$ als auch bei $AR_C B$ immer besser bewertet als NDS_B .

Bezüglich der schwachen Outperformance Relation ist GD_l nur noch schwach kompatibel. Gilt $AR_W B$ so ist, falls $NDS_B \subset NDS_A$ gilt, $GD_B = 0$. Somit wird NDS_B nicht schlechter bewertet als NDS_A .

Im Fall $L \geq 3$ ist GD_l mit keiner der drei Outperformance Relationen kompatibel, auch nicht schwach. In Abbildung 4.12 ist ein Beispiel angegeben, in dem $AR_C B$ und $GD_A > GD_B$ gilt.

GD_A berechnet sich wie folgt:

$$\begin{aligned}
d(D) &= \min \left\{ \sqrt{(3-2)^2 + (6-7)^2}; \sqrt{(3-3)^2 + (6-3)^2} \right\} \\
&= \min \left\{ \sqrt{1^2 + 1^2}; \sqrt{0 + 3^2} \right\} \\
&= \sqrt{2} \\
d(E) &= \min \left\{ \sqrt{(11-2)^2 + (3-7)^2}; \sqrt{(11-3)^2 + (3-3)^2} \right\} \\
&= \min \left\{ \sqrt{9^2 + 4^2}; \sqrt{8^2 + 0} \right\} \\
&= \sqrt{64}
\end{aligned}$$

Somit ergibt sich $GD_A = \frac{\sqrt{\sqrt{2}^2 + \sqrt{64}^2}}{2} = 4,06$.

GD_B berechnet sich wie folgt:

$$\begin{aligned}
d(C) &= \min \left\{ \sqrt{(3-2)^2 + (7-7)^2}; \sqrt{(3-3)^2 + (7-3)^2} \right\} \\
&= \min \left\{ \sqrt{1^2 + 0}; \sqrt{0 + 4^2} \right\} \\
&= \sqrt{1}
\end{aligned}$$

Somit ergibt sich $GD_B = \frac{\sqrt{\sqrt{1}^2}}{1} = 1$.

⁴⁹Knowles (2002) hat in seiner Arbeit ebenfalls die General Distance untersucht. Er kommt zu dem Ergebnis, daß sie nicht kompatibel mit der schwachen Outperformance Relation und kompatibel mit der starken sowie vollständigen Outperformance Relation ist (vgl. Knowles (2002, S.78f. u. 92) oder auch Knowles u. Corne (2002, S.713 u. 716)). Eine Begründung für diese Ergebnisse wird nicht angegeben. Beide Arbeiten betrachten die Variante der Generational Distance, bei der die heuristisch effiziente Front und die effiziente Front $E(z)$ zur Berechnung verwendet werden.

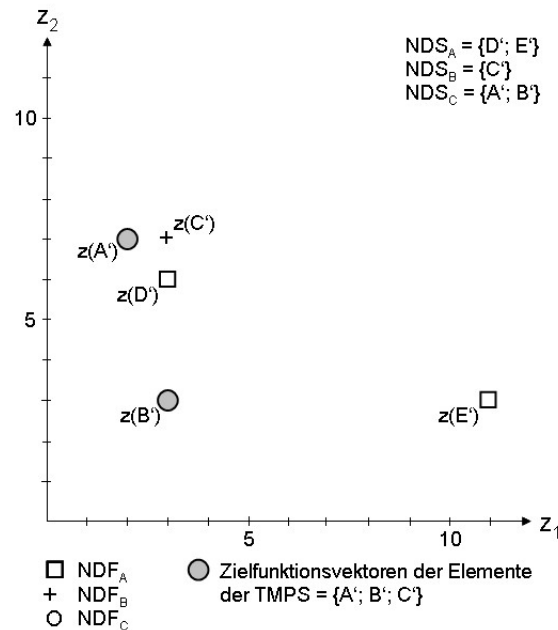


Abbildung 4.12: Inkompatibilität der Generational Distance mit R_O

Die "Generational Distance" erfüllt in beiden Fällen die Anforderung der Relativität, nicht jedoch der Monotonie. So steigt der Wert von GD_B , falls in Abbildung 4.12 ein Element mit den Werten (9 ; 3) in NDS_B eingefügt wird. Dann evaluiert GD_B zu: $GD_B = \frac{\sqrt{\sqrt{1}^2 + \sqrt{36}^2}}{2} = 3,04$. Das Einfügen eines Elements läßt den Wert von GD_B von 1 auf 3,04 ansteigen. Um eine Verbesserung der Qualität anzuzeigen, müßte der Wert jedoch sinken.

Maximum Pareto Front Error (Van Veldhuizen (1999)): Dieses Maß bestimmt die maximale minimale Distanz zwischen den Elementen der heuristisch effizienten Front NDF und dem zu diesen jeweils nächsten Element der effizienten Front $E(z)$.⁵⁰ In der folgenden Formel ist die von Van Veldhuizen (1999) verwendete effiziente Front $E(z)$ durch die $TMPE$ ersetzt.

$$MPFE_l = \max_{y \in TMPE} \left\{ \min_{x \in NDF_l} \sqrt{\sum_{k=1}^K |x_k - y_k|^2} \right\} \quad (4.16)$$

Das Maß hat den Wert Null, falls $NDF_l = TMPE$ ist. Ein Wert ungleich

⁵⁰Vgl. Van Veldhuizen (1999, S.6-15f.). "... this is the largest minimum distance between each vector in PF_{known} and the corresponding closest vector in PF_{true} ." (Van Veldhuizen (1999, S.6-16)).

Null zeigt an, daß mindestens ein Element aus NDF_l nicht Element der $TMPF$ ist.⁵¹

In Abbildung 4.13 (a) ist ein kleines Beispiel zur Berechnung des Maximum Pareto Front Error gegeben.

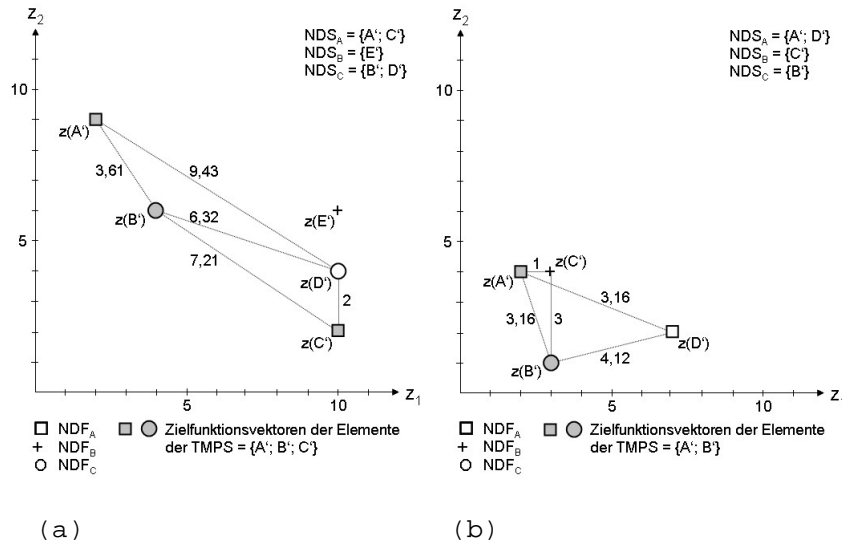


Abbildung 4.13: Inkompatibilität des Maximum Pareto Front Errors mit R_O

Es soll $MPFE_C$ berechnet werden. Dazu ist zunächst für jedes Element y aus der $TMPF$ die Distanz zu dem Element aus NDF_C mit der kleinsten euklidischen Distanz zu ermitteln.⁵² Für Element $z(A')$ ist Element $z(B')$ mit $3,61$ das nächste Element aus der NDF_B . Für $z(B')$ ist die Distanz 0 da $z(B')$ Element in der $TMPF$ und in NDF_C ist. Für $z(C')$ ist mit einer euklidischen Distanz von 2 Element $z(D')$ aus NDF_C das nächstgelegene Element. Um den Maximum Pareto Front Error für NDF_C zu bestimmen, ist der größte dieser Werte zu wählen, also: $MPFE_C = \max \{ \min \{ 3,61 ; 9,43 \} ; \min \{ 0 ; 6,32 \} ; \min \{ 7,21 ; 2 \} \} = \max \{ 3,61 ; 0 ; 2 \} = 3,61$

Im Fall $L = 2$ ist $MPFE_l$ mit allen drei Outperformance Relationen (AR_OB) kompatibel.⁵³ $MPFE_A$ ist immer Null, da $TMPF = MPF_A$ gilt.

⁵¹Vgl. zur Interpretation des Maßes die Interpretation von Van Veldhuizen (1999, S.6-16).

⁵²Es ist also der Term $\min_{x \in NDF_l} \sqrt{\sum_{k=1}^K |x_k - y_k|^2}$ für jedes $y \in TMPF$ zu bestimmen.

⁵³Knowles (2002) hat in seiner Arbeit ebenfalls den "Maximum Pareto Front Error" untersucht. Er kommt zu dem Ergebnis, daß er mit keiner der Outperformance Relationen kompatibel ist (vgl. Knowles (2002, S.79f. u. 92). Das gleiche Ergebnis wird auch in Knowles u. Corne (2002, S.713f. u. 716) angegeben. Eine Begründung für die Aussagen fehlt in beiden Arbeiten. Ihre Aussagen beziehen sich jedoch nicht auf die hier betrachtete Variante, in welcher die $TMPF$ als Referenz verwendet wird.

Daher hat jedes Element aus $TMPF$ ein nächstes Element aus MPF_A mit einer Distanz von Null. In der NDF_B gibt es Elemente, die nicht in der $TMPF$ sind und in der $TMPF$ Elemente, die nicht Element von NDF_B sind. Somit muß es mindestens ein Element in der $TMPF$ geben, dessen nächstes Element der NDF_B eine positive Distanz hat. Daher ist $NPFE_B$ immer größer Null.

Falls $L \geq 3$ gilt, so besteht keine Kompatibilität, auch keine schwache, mit allen drei Outperformance Relationen.⁵⁴ Ein Beispiel, in dem NDF_B besser bewertet wird als NDF_A , obwohl AR_{CB} gilt, ist in Abbildung 4.13 (b) gegeben. Offensichtlich gilt hier: $MPFE_B = \max\{1; 3\} = 3$ und $MPFE_A = \max\{0; 3, 16\} = 3, 16$.

Der "Maximum Pareto Front Error" erfüllt die Anforderung der Relativität. Die Anforderung der Monotonie wird schwach erfüllt, da ein neu eingefügtes Element nicht zwingend näher zu einem Element der $TMPF$ als eines der bereits enthaltenen Elemente liegt.

Coverage of two sets / C-Maß (Zitzler u. Thiele (1998)): In seiner Urform vergleicht das C-Maß zwei Lösungen ohne Verwendung einer Referenz direkt miteinander. Es gibt den Anteil der Elemente in NDS_B an, die von einem Element in NDS_A dominiert werden oder gleich einem Element aus NDS_A sind.⁵⁵

$$C(NDS_A, NDS_B) = \frac{|\{b \in NDS_B \mid \exists a \in NDS_A; a \leq b\}|}{|NDS_B|} \quad (4.17)$$

Ein Wert von 1 bedeutet demnach, daß alle Elemente von NDS_B gleich einem Element aus NDS_A sind oder von diesem dominiert werden.⁵⁶ Dabei ist zu beachten, daß nicht grundsätzlich $C(NDS_A, NDS_B) = 1 - C(NDS_B, NDS_A)$ gilt.⁵⁷ Daß Maß ist also nicht symmetrisch. Aus diesem Grund müssen stets beide Richtungen ausgewertet werden. In Abbildung 4.14 (a) ist ein Beispiel für diese Situation angegeben.

In der Abbildung ergibt sich $C(NDS_A; NDS_B) = \frac{2}{2} = 1$ und $C(NDS_B; NDS_A) = \frac{2}{4}$. Damit gilt $C(NDS_A, NDS_B) \neq 1 - C(NDS_B; NDS_A)$.

⁵⁴Siehe dazu auch die Ausführungen von Knowles (2002, S.79f. u. 92).

⁵⁵Vgl. Zitzler u. Thiele (1998, S.297) für eine verbale Definition sowie Zitzler (1999, S.43f.) und Zitzler u. a. (2000, S.180) für eine formale Darstellung.

⁵⁶Vgl. Zitzler u. a. (2000, S.180) sowie Zitzler (1999, S.43).

⁵⁷Vgl. Zitzler u. a. (2000, S.180), Zitzler (1999, S.43) oder Knowles (2002, S.85).

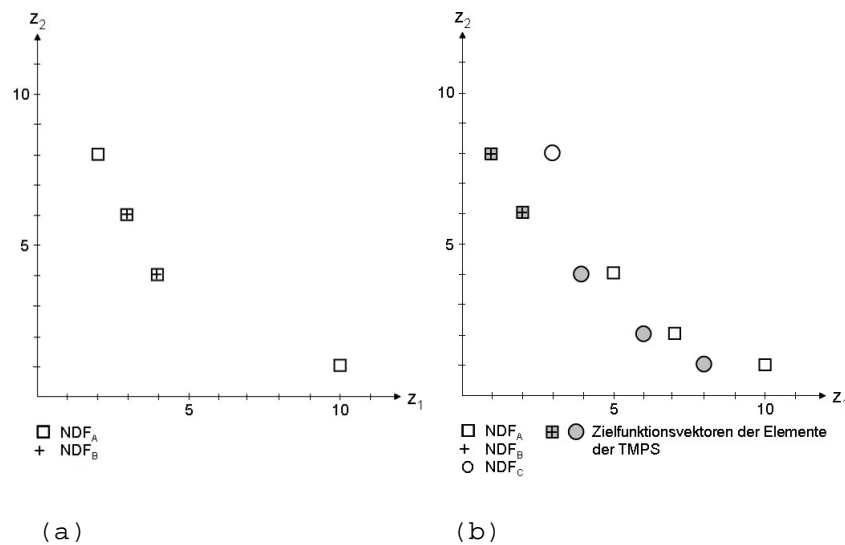


Abbildung 4.14: Coverage of two sets

Akzeptiert man das von Knowles (2002) diskutierte Vorgehen, daß falls $C(NDS_A, NDS_B) = 1$ und $C(NDS_B, NDS_A) < 1$ bedeutet, daß Lösung B schlechter bewertet wird als Lösung A, so ist das C-Maß kompatibel mit der schwachen Outperformance Relation.⁵⁸ Da bei $AR_{O}B$ grundsätzlich alle Elemente von NDF_B gleich einem Element von NDF_A sind oder von einem Element aus NDF_A dominiert werden, gilt immer $C(NDS_A; NDS_B) = 1$. Um die Kompatibilität mit der schwachen Outperformance Relation nachzuweisen, muß gezeigt werden, daß wenn $AR_{W}B$ gilt, grundsätzlich nicht $C(NDS_B, NDS_A) = C(NDS_A, NDS_B) = 1$ gelten kann. Aufgrund der Definition der schwachen Outperformance Relation gibt es bei $AR_{W}B$ mindestens ein Element in NDS_A , das von keinem Element aus NDS_B dominiert wird oder gleich diesem ist. Daher gilt bei $AR_{W}B$ immer $C(NDS_B, NDS_A) < 1$. Es besteht somit Kompatibilität mit der schwachen und somit auch mit der starken und vollständigen Outperformance Relation.

Bei dem Maß besteht grundsätzlich ebenfalls die Möglichkeit die Lösungen gegen eine Referenzlösung zu vergleichen und das C-Maß von einem Direktmaß in ein Referenzmaß umzuwandeln.⁵⁹ Diese Möglichkeit, soll im Folgenden mit der *TMPS* als Referenz weiter untersucht werden.⁶⁰

⁵⁸Vgl. Knowles (2002, S.85).

⁵⁹Vgl. Knowles (2002, S.85).

⁶⁰Vgl. dazu auch die Analysen von Knowles (2002, S.85f.). Dieser macht jedoch nur Aussagen über die schwache Outperformance Relation. Des weiteren betrachtet er auch den Vergleich mit Hilfe von beliebigen Referenzmengen. Dort stellt er fest, daß keine Kompatibilität mit der

Im Fall $L = 2$ gilt aufgrund der Definition der *TMPS* offensichtlich immer $C(TMPS, NDS_A) = 1$ und $C(TMPS, NDS_B) = 1$. Damit muß nur noch eine Richtung bei der Auswertung explizit untersucht werden. Es soll nun gezeigt werden, daß das C-Maß mit allen drei Outperformance Relationen kompatibel ist. Aufgrund der Definition der *TMPS* gilt im Fall $AR_W B$ $TMPS = NDS_A$ und damit $C(NDS_A, TMPS) = 1$. Bei $AR_W B$ gibt es in NDF_A und damit in der *TMPS* mindestens ein Element, welches von keinem Element aus NDF_B dominiert wird oder gleich diesem ist. Damit gilt immer $C(NDS_B, TMPS) < 1$. Es besteht somit Kompatibilität mit der schwachen Outperformance Relation und damit auch mit der starken und vollständigen.

Im Fall $L \geq 3$ gilt weiterhin grundsätzlich $C(TMPS, NDS_A) = 1$ und $C(TMPS, NDS_B) = 1$. Es soll nun gezeigt werden, daß das C-Maß mit der schwachen, starken und vollständigen Outperformance Relationen schwach kompatibel ist. Abbildung 4.14 (b) zeigt eine Situation, in der $AR_W B$ gilt, NDS_B aber nicht schlechter bewertet wird als NDS_A . Dort gilt offenbar $C(NDS_A, TMPS) = \frac{2}{5}$ und $C(NDS_B, TMPS) = \frac{2}{5}$. Es ist jedoch bei keiner der Outperformance Relationen möglich, daß bei $AR_O B$ die NDS_B mehr Elemente in die *TMPS* einstellt als NDS_A . Damit ist eine bessere Bewertung von NDS_B als NDS_A nicht möglich. Gleiche Überlegungen lassen sich für die starke Outperformance Relation anstellen. Bezüglich der vollständigen Outperformance Relation besteht ebenfalls schwache Kompatibilität. Im Fall $AR_C B$ wird immer gelten $C(NDS_B, TMPS) = 0$. Jedoch kann auch $C(NDS_A = TMPS) = 0$ gelten. Dies geschieht, falls die Situation $AR_C B$, $CR_C A$ und $CR_C B$ eintritt.

Das C-Maß erfüllt die Anforderung nach Relativität. Nur für die *TMPS* oder ihr entsprechende Mengen kann gelten: $C(NDS_i, TMPS) = 1$. Die Anforderung der Monotonie wird offensichtlich schwach erfüllt.

Size of Dominated Space / S-Maß (Zitzler u. Thiele (1998)): Das S-Maß wurde erstmals in Zitzler u. Thiele (1998) vorgestellt.⁶¹ In Zitzler (1999) wird die Definition für mehr als zwei Ziele ausführlich dargestellt.⁶² Das Maß wird auch von Fleischer (2002, 2003) betrachtet und dort als "Hypervolumen" bezeichnet.⁶³

Grundlegende Idee ist, daß jedes Element in der heuristisch effizienten Front einen Bereich des Zielraums dominiert. Bei Minimierungsproblemen liegt dieser Bereich oberhalb der heuristisch effizienten Front, weshalb ei-

schwachen Outperformance Relation mehr vorliegt.

⁶¹Vgl. Zitzler u. Thiele (1998, S.297).

⁶²Vgl. Zitzler (1999, S.43).

⁶³Vgl. Fleischer (2002, S.5-8), Fleischer (2003, S.523-526).

ne Beschränkung des Zielraums von oben durch den Punkt $z^{max} \in \mathbb{R}^K$ notwendig ist.⁶⁴ Die Definition von z^{max} erfolgt in Zitzler (1999) dabei derart, daß $z^{max} = z^{aid}$ gilt.⁶⁵

Der von dem Element $z(\pi)$ der heuristisch effizienten Front dominierte Bereich entspricht dem durch diesem Vektor und z^{max} beschriebenen (Hyper)Rechteck q . Das S-Maß ergibt sich als Volumen der Vereinigung der von allen Elementen der NDF gebildeten (Hyper)Rechtecke.⁶⁶

Der Antiidealzielpunkt z^{aid} kann jedoch nur schwer bestimmt werden. Offensichtlich genügt es jedoch, wenn bei der Berechnung des S-Maßes der Beschränkungsvektor z^{max} wie folgt berechnet wird:

$$z^{max} = (z_1^{max}, z_2^{max}, \dots, z_k^{max}, \dots, z_K^{max}) \quad (4.18)$$

mit

$$z_k^{max} = \max \left\{ z_k(\pi) \mid \pi \in \bigcup_{l=1}^L NDS_l \right\} + \epsilon \text{ mit } \epsilon > 0 \quad (4.19)$$

Damit ist sichergestellt, daß $z_k^{max} - z_k(\pi) > 0 \forall \pi \in \bigcup_{l=1}^L NDS_l$ gilt. Für alle Ziel k weist z^{max} einen größeren Wert auf als der größte Wert, der in den, in den Vergleich aufgenommenen heuristisch effizienten Mengen auftritt.⁶⁷

Für den bikriteriellen Fall ist die Berechnung des S-Maß in Abbildung 4.15 für NDS_A und $\epsilon = 1$ grafisch dargestellt.⁶⁸ Der Wert des S-Maßes von NDS_A (S_A) ergibt sich aus der Größe der Vereinigung der schraffierten Flächen.

Ein wesentlicher Unterschied zu den bisher betrachteten Maßen ist, daß es nicht die heuristisch effiziente Menge ausgewertet, sondern die heuristisch effiziente Front.

Eine Lösung ist offensichtlich umso besser, je größer ihr S-Maß ist. Das Maß ist ein unabhängiges Maß und mit allen drei Outperformance Relationen kompatibel.⁶⁹

Würde die Auswertung auf die heuristisch effiziente Front beschränkt, so würde das S-Maß offensichtlich die Anforderung der Relativität und Monotonie erfüllen. Wird es auf die heuristisch effiziente Menge bezogen und

⁶⁴Vgl. zu dieser und zur folgenden Aussage Zitzler (1999, S.43f.), insbesondere Fußnote 1.

⁶⁵Vgl. Zitzler (1999, S.43).

⁶⁶Vgl. Zitzler (1999, S.43).

⁶⁷Knowles (2002) weist darauf hin, daß die Wahl von z^{max} Auswirkungen auf die Rangfolge der verglichenen heuristisch effizienten Mengen haben kann. Er zeigt dies auch anhand eines grafischen Beispiels (vgl. Knowles (2002, S.75f.)).

⁶⁸Das S-Maß wird bei der experimentellen Analyse in Kapitel 6 verwendet werden. Bei den Berechnungen wurde $\epsilon = 1$ gesetzt.

⁶⁹Vgl. Knowles (2002, S.74f. u. 92).

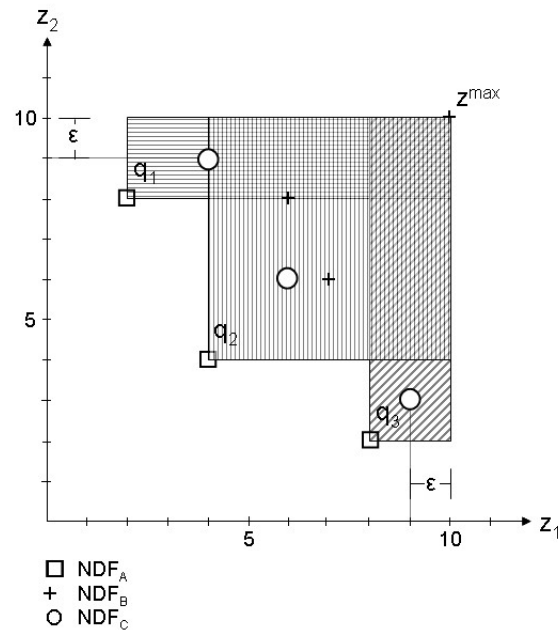


Abbildung 4.15: S-Maß

die vorne aufgestellte Annahme, daß mehrerer Elemente des Lösungsraum keinen gleichen Zielfunktionsvektor besitzen können, fallengelassen, so wird die Monotonieanforderung nur schwach erfüllt, da mehrere Lösungen den gleichen Zielfunktionsvektor besitzen können. Würde in die heuristisch effiziente Menge ein Element eingefügt, das einen Zielfunktionsvektor hat, den bereits ein anderes in der NDS enthaltenes Element hat, so würde sich das Maß nicht ändern, da nur die Vereinigung der (Hyper)Rechtecke q betrachtet wird. Auch die Relativitätsforderung würde nur schwach erfüllt. Da in diesem Fall $NDF_l = TMPPF$ aber $NDS_l \neq TMPS$ gelten kann.

Die Berechnung des Maßes ist sehr aufwendig. Prozeduren zu seiner Ermittlung finden sich z.B. in Fleischer (2002) und Knowles (2002).⁷⁰

Hyperareadifference (Wu u. Azarm (2001)): Dieses Maß baut auf das S-Maß von Zitzler u. Thiele (1998) auf. Es errechnet sich als die Differenz des Volumens des von der heuristisch effizienten Front dominierten Bereichs und dem Volumen des von der effizienten Front dominierten Bereichs. Da die effiziente Front nicht bekannt ist, wird sie von Wu u. Azarm (2001) durch den Idealzielpunkt approximiert.⁷¹

⁷⁰Vgl. Fleischer (2002, S.9) und Knowles (2002, S.97-99).

⁷¹Vgl. Wu u. Azarm (2001, S.20f.).

Des weiteren verwenden sie im Gegensatz zu Zitzler u. Thiele (1998) und Zitzler (1999) normierte Zielfunktionswerte. Die Normierung eines Elements der NDF_l wird dabei wie folgt vorgenommen:⁷²

$$\tilde{z}_k(x) = \frac{z_k(x) - z_k^g}{z_k^b - z_k^g}, \quad k = 1(1)K \quad (4.20)$$

Dabei ist z^g der sogenannte Good Point. Dieser kann unter Einhalten der Bedingung $z^g \leq z^{id}$ beliebig festgelegt werden.⁷³ Der Bad Point z^b kann unter Einhalten der Bedingung $z^b > z^{aid}$ beliebig festgelegt werden.⁷⁴ Der Idealzielpunkt ist, wie auch der Antiidealzielpunkt, nicht bekannt, so daß der Good Point und Bad Point anders festgelegt werden müssen. Der Good Point kann auf den Ursprung des Koordinatensystems gesetzt werden, also $z^g = 0$.

Für die Elemente des Bad Points ist es ausreichend folgende Bedingung einzuhalten: $z_k^b > \max_{l=1(1)L} \{\max \{z_k \mid z_k \in NDF_l\}\}$. Durch die Normierung gilt dann: $\tilde{z}^g = 0$ und $\tilde{z}^b = 1$.⁷⁵

Wird mit \tilde{S}_l das mit den normierten Zielfunktionswerten berechnete S-Maß bezeichnet, so ergibt sich die "Hyperarreadifference" zu:

$$HD_l = 1 - \tilde{S}_l \quad (4.21)$$

Je kleiner der Wert desto besser wird die Lösung beurteilt.⁷⁶

Das Maß ist, wie auch das S-Maß, mit allen drei Outperformance Relationen kompatibel, des weiteren erfüllt es die Anforderungen der Relativität und Monotonie in der schwachen Form.⁷⁷

Zusammenfassung der Maße und ihrer Eigenschaften

Die folgende Tabelle faßt die diskutierten Eigenschaften der Maße nochmals zusammen:

⁷²Vgl. Wu u. Azarm (2001, S.19).

⁷³Vgl. zur Definition des Good Points und Bad Points auch Wu u. Azarm (2001, S.19).

⁷⁴Würde $z^b \geq z^{aid}$ gelten, so könnte nicht sichergestellt werden, daß $z_k^b - z_k^g > 0$, $k = 1(1)K$ gilt.

⁷⁵Vgl. Wu u. Azarm (2001, S.19).

⁷⁶Vgl. Wu u. Azarm (2001, S.21).

⁷⁷Da bei diesem Maß die Werte zur Ermittlung des S-Maßes lediglich normiert werden, ergibt sich diese Aussage analog aus den entsprechenden Aussagen des S-Maßes.

	$L = 2$			$L \geq 3$			M	R	Sonstiges
	R_W	R_S	R_C	R_W	R_S	R_C			
$NNDS_l$	++	++	++	+	+	+	-	++	
$ER_l; InfI_l$	+	++	++	-	-	+	-	++	
PES_l	++	++	++	+	+	+	-	++	
$D1_l$	++	++	++	+	++	++	+	++	bei $\Delta_k = 0$ nicht def.
$D2_l$	++	++	++	+	++	++	+	++	bei $\Delta_k = 0$ nicht def.
GD_l	+	++	++	-	-	-	-	++	
$MPFE_l$	++	++	++	-	-	-	+	++	Für NDF def.
$C\text{-Ma\ss}$	++	++	++	+	+	+	+	++	
$S\text{-Ma\ss} (S_l)$	++	++	++	++	++	++	++/+	++/+	
HD_l	++	++	++	++	++	++	+	+	

++ Kompatibilität, Monotonie bzw. Relativität erfüllt

+ Kompatibilität, Monotonie bzw. Relativität schwach erfüllt

- Kompatibilität, Monotonie bzw. Relativität nicht erfüllt

M Monotonie

R Relativität

Tabelle 4.1: Eigenschaften der Qualitätsmaße zur Erfassung der Distanz zur Menge aller effizienten Lösungen

4.2.2 Größe der aufgezeigten Variationsbreiten der Zielfunktionswerte

Um einen ergänzenden bzw. vertiefenden Einblick in die Lösungsgüte zu erhalten, wird in der Literatur zusätzlich zur Distanz häufig die Variationsbreite der Zielfunktionswerte in einer heuristisch effizienten Menge bzw. in der aus ihr ermittelten heuristisch effizienten Front ausgewertet. Die Beurteilung sollte aus den bereits diskutierten Gründen ebenfalls mit der *TMPS* bzw. *TMPF* als Referenzlösung erfolgen.

Kth-Objective Pareto Spread, Overall Pareto Spread (Wu u. Azarm (2001)): Der "Kth-Objective Pareto Spread" gibt die normierte Variationsbreite eines Ziels k in einer Lösung an und errechnet sich wie folgt:⁷⁸

$$OS_l^k = \max_{x \in NDS_l} \{\tilde{z}_k(x)\} - \min_{x \in NDS_l} \{\tilde{z}_k(x)\} \quad (4.22)$$

Dabei wird eine Normierung der Zielfunktionswerte wie bei der "Hypereadifference" vorgenommen (vgl. Gl.4.20).⁷⁹

⁷⁸Vgl. Wu u. Azarm (2001, S.21).

⁷⁹Vgl. Wu u. Azarm (2001, S.19).

Im Gegensatz zum "Kth-Objective Pareto Spread" wird im "Overall Pareto Spread" die Variationsbreite aller Ziele gleichzeitig berücksichtigt:⁸⁰

$$OS_l = \prod_{k=1}^K \left| \max_{x \in NDS_l} \{\tilde{z}(x)\} - \min_{x \in NDS_l} \{\tilde{z}(x)\} \right| \quad (4.23)$$

Bezüglich beider Maße beurteilen Wu u. Azarm (2001) eine Lösung l um so besser, je größer der Wert von OS_l bzw. OS_l^k ist.⁸¹

Eine Schwierigkeit bei beiden Maßen besteht darin, daß der Idealziel-punkt sowie der Antiidealziel-punkt nicht bekannt sind und es somit schwer ist, den Good Point und Bad Point mit ihrer Hilfe geeignet zu bestimmen. Es ist jedoch möglich, deren Ermittlung wie bei der Hyperareadifference vorzunehmen.

Bei OS_l tritt als weiteres Problem hinzu, daß, falls für mindestens eines der Ziele eine Variationsbreite von Null auftritt, das gesamte Maß den Wert Null annimmt. Ein Wert von Null ist jedoch nur bei Vorliegen einer perfekten Lösung sinnvoll.

Des weiteren fehlt den Maßen bei der Beurteilung ein Bezug zur *TMPS*. Daher erfüllen sie auch nicht die Anforderung der Relativität, da nicht bekannt ist, was der beste Wert ist. Die Monotonieeigenschaft kann nicht beurteilt werden. Es ist möglich, daß die NDS_l eine kleinere, eine gleiche oder eine größere Variationsbreite der Zielgrößen aufweist als die *TMPS*. Somit zeigt eine Vergrößerung der Variationsbreite bei OS_l nur dann eine Verbesserung der Qualität an, falls vor dem Zeitpunkt des Einfügens eines neuen Elements in die NDS_l , $OS_l < OS_{TMPS}$ galt. Würde vor dem Einfügen hingegen $OS_l > OS_{TMPS}$ gelten, so würde ein Ansteigen von OS_l eine Verschlechterung der Qualität bedeuten. Diese Aussage gilt ebenso für OS_l^k . In Abbildung 4.16 ist ein Beispiel für diese Situationen dargestellt.

In dem Beispiel gilt: $OS_{TMPS}^1 = \frac{8}{11} - \frac{1}{11} = \frac{7}{11}$ und $OS_{TMPS}^2 = \frac{8}{11} - \frac{1}{11} = \frac{7}{11}$. NDS_B weist offensichtlich die gleichen Werte für die Variationsbreite der beiden Zielgrößen auf. Bei NDS_A ist die Variationsbreite für Ziel 1 größer und für Ziel 2 gleich groß, denn es ergibt sich: $OS_A^1 = \frac{10}{11} - \frac{1}{11} = \frac{9}{11}$ und $OS_A^2 = \frac{8}{11} - \frac{1}{11} = \frac{7}{11}$. Für NDS_C sich: $OS_C^1 = \frac{8}{11} - \frac{3}{11} = \frac{5}{11}$ und $OS_C^2 = \frac{10}{11} - \frac{1}{11} = \frac{9}{11}$.

Modified Kth-Objective Pareto Spread, Modified Overall Pareto Spread (Farhang-Mehr u. a. (2001)): Diese Maße stellen eine Modifikation von OS_l^k und OS_l da. Anstelle der Variationsbreite der Zielfunktionswerte der heuristisch effizienten Menge wird hier die

⁸⁰Vgl. Wu u. Azarm (2001, S.21).

⁸¹Vgl. Wu u. Azarm (2001, S.21).

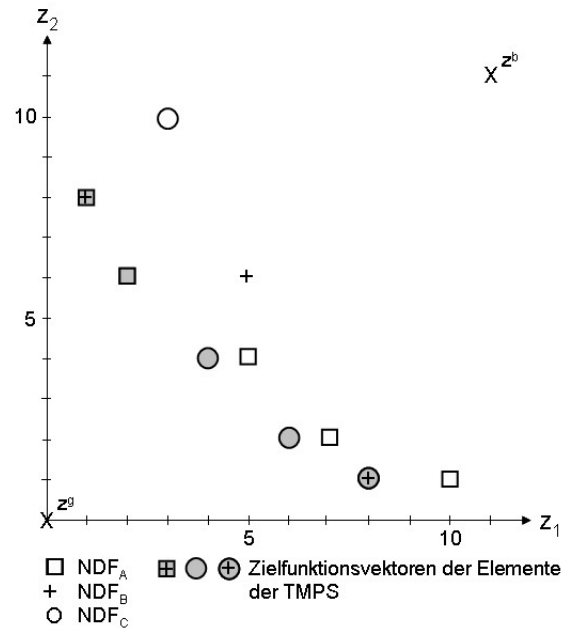


Abbildung 4.16: Kth-Objective Pareto Spread

Variationsbreite der Zielfunktionswerte der Elemente der Menge MPS_l ausgewertet. Als Begründung für dieses Vorgehen argumentieren Farhang-Mehr u. a. (2001), daß Elemente, die sich als nicht effizient erwiesen haben, bei der Beurteilung nicht berücksichtigt werden sollten, und somit eine korrekte Erfassung der Variationsbreite allein unter Berücksichtigung der Meta Pareto Lösungen möglich sei.⁸² Ihre Maße ergeben sich daher zu:

$$MOS_l^k = \max_{x \in MPS_l} \{\tilde{z}_k(x)\} - \min_{x \in MPS_l} \{\tilde{z}(x)\} \quad (4.24)$$

$$MOS_l = \prod_{k=1}^K \left| \max_{x \in MPS_l} \{\tilde{z}_k(x)\} - \min_{x \in MPS_l} \{\tilde{z}(x)\} \right| \quad (4.25)$$

Auch bei diesen Maßen gilt, daß ein größerer Wert besser beurteilt wird als ein kleinerer.⁸³

Zur Bestimmung von z^g und z^b machen die Autoren keine weiteren Ausführungen, so daß hier angenommen wird, daß sie die Definition von Wu u. Azarm (2001) übernehmen. Grundsätzlich besitzt die Kritik an OS_l und OS_l^k auch für diese Maße ihre Gültigkeit.

⁸²Vgl. Farhang-Mehr u. a. (2001, S.6).

⁸³Vgl. Farhang-Mehr u. a. (2001, S.8).

Die Autoren stellen mit diesem Maß einen Bezug zur *TMPS* her. Ein wesentliches Problem bei der Verwendung dieses Maßes ist jedoch, daß es nicht in allen Fällen definiert ist. Bereits bei $AR_C B$ ergibt sich $MPS_B = \emptyset$. MOS_B^k und MOS_B können somit nicht berechnet werden. Auch bei diesem Maß ist der beste Wert nicht bekannt, so daß die Anforderung der Relativität nicht erfüllt ist. Aus dem gleichen Grund wie bei OS_i und OS_i^k beschrieben, kann die Monotonieeigenschaft auch hier nicht beurteilt werden.

Binary Coverage Metric (Farhang-Mehr u. Azarm (2003)): Kann man die bisher betrachteten Maße als Flächen- bzw. Volumenmaße bezeichnen, so stellen Farhang-Mehr u. Azarm (2003) mit der "Binary Coverage Metric" ein Maß vor, das als Winkelmaß klassifiziert werden könnte.⁸⁴

Allgemein gilt, daß das Skalarprodukt $a \cdot b$ zweier Vektoren a und b Null ist, falls a und b orthogonal ($a \perp b$) zueinander sind. Wird nun ausgehend von ($a \perp b$) der Vektor b in Richtung a gedreht, so steigt das Skalarprodukt an, ohne daß sich die Länge der Vektoren $\|a\|$ bzw. $\|b\|$ ändert. Sind a und b parallel ($a \parallel b$), so erreicht das Skalarprodukt sein Maximum. Somit kann der Winkel φ zwischen zwei Vektoren, bzw. sein Cosinus, über ihr Skalarprodukt erfaßt werden: $\cos(\varphi) = \frac{a \cdot b}{\|a\| \cdot \|b\|}$. Dabei darf weder $a = 0$ noch $b = 0$ gelten.

Die "Binary Coverage Metric" errechnet sich nun wie folgt:⁸⁵

$$BC(NDS_A, NDS_B) = \min_{x,y \in NDS_B} \left\{ \frac{z(x) \cdot z(y)}{\|z(x)\| \cdot \|z(y)\|} \right\} - \min_{x,y \in NDS_A} \left\{ \frac{z(x) \cdot z(y)}{\|z(x)\| \cdot \|z(y)\|} \right\} \quad (4.26)$$

Ist $BC(NDS_A, NDS_B) < 0$, so hat NDS_A einen größeren Winkel als NDS_B , was einer größeren Spannweite im Zielraum entspricht. Es handelt sich hierbei um ein direktes Maß, das eine Beurteilung ohne Berücksichtigung der *TMPS* vornimmt. Ein Wert von Null für $BC(TMPS, NDS_i)$ würde anzeigen, daß die *TMPS* und die NDS_i die gleiche Spannweite haben. Jedoch bedeutet ein Wert von Null nicht, daß $TMPS = NDS_i$ gilt. Wird ein Wert von Null daher als bester Wert angesehen, so erfüllt das Maß die Anforderung der Relativität nur schwach. Es ist möglich, daß die NDS_i eine kleinere, eine gleiche oder eine größere Spannweite aufweist als die *TMPS*. Aus dem gleichen Grund wie bei OS_i und OS_i^k beschrieben, kann die Mono-

⁸⁴Vgl. Farhang-Mehr u. Azarm (2003, S.415).

⁸⁵Vgl. Farhang-Mehr u. Azarm (2003, S.415).

tonieeigenschaft auch hier nicht beurteilt werden.

Als weitere Einschränkung ist festzuhalten, daß sich die "Binary Coverage Metric" nur auf Zielsysteme mit zwei Zielen anwenden läßt.

Zusammenfassung der Maße und ihrer Eigenschaften

Die folgende Tabelle faßt die diskutierten Eigenschaften der Maße nochmals zusammen:

	Relativität	Monotonie	Sonstiges
OS_l^k	-	-	Null obwohl keine perfekte Lösung
OS_l	-	-	Null obwohl keine perfekte Lösung
MOS_l^k	-	-	Nicht für alle Fälle definiert
MOS_l	-	-	Nicht für alle Fälle definiert
BC	+	-	Nur für $K=2$ definiert

++ Monotonie bzw. Relativität erfüllt

+ Monotonie bzw. Relativität schwach erfüllt

- Monotonie bzw. Relativität nicht erfüllt

Tabelle 4.2: Eigenschaften der Qualitätsmaße zur Erfassung der Variationsbreite der Zielfunktionswerte

4.2.3 Verteilung der Zielfunktionsvektoren im Zielraum

Als weiterer Qualitätsaspekt kann die Verteilung der Zielfunktionsvektoren der Elemente der heuristisch effizienten Menge im Zielraum betrachtet werden. Häufig wird dabei eine gleichmäßige Verteilung als erstrebenswert angesehen.⁸⁶ Dies vernachlässigt den Umstand, daß die Elemente der Menge aller effizienten Lösungen, bzw. der *TMPS* als ihrer Approximation durchaus eine ungleichmäßige Verteilung aufweisen können. Es sei bereits an dieser Stelle darauf hingewiesen, daß keines der hier vorgestellten Maße die Anzahl der Cluster sowie deren Lage im Zielraum zu identifizieren in der Lage ist.

Spacing (Deb u. a. (2000)): Ist $d(z(x))$ die euklidische Distanz zwischen dem Element $z(x)$ und dem direkt darauf folgenden Element in der NDF_l und \bar{d} der Durchschnitt der Werte $d(z(x))$, so ergibt sich das Maß zu:⁸⁷

⁸⁶Vgl. zu dieser Aussage auch Zitzler u. a. (2000, S.179). Auch Deb u. a. (2000, S.856) sehen eine gleichmäßige Verteilung als ideal an.

⁸⁷Vgl. Deb u. a. (2000, S.855).

$$SD_l = \sum_{z(x) \in NDF_l} \frac{|d(z(x)) - \bar{d}|}{|NDF_l|} \quad (4.27)$$

Eine gleichmäßige Verteilung der Elemente im Zielraum wird als erstrebenswert betrachtet.⁸⁸ Diese ist offensichtlich gegeben, falls $SD_l = 0$ gilt.⁸⁹ Diese Interpretation vernachlässigt jedoch den Bezug zur *TMPPF*. Wie das direkt auf $z(x)$ folgende Element in der NDF_l bestimmt wird, ist in der Arbeit von Deb u. a. (2000) ebenfalls nicht explizit formuliert.⁹⁰ Auch Knowles (2002), welcher in seiner Arbeit das Maß betrachtet, macht keine genaueren Aussagen. Er merkt lediglich an, daß im Fall von mehr als zwei Zielen, das direkt auf $z(x)$ folgende Element nicht mehr genau definiert ist⁹¹

Damit sichergestellt ist, daß das Maß auch die Variationsbreite der effizienten Front $E(z)$ berücksichtigt, nehmen Deb u. a. (2000) zusätzlich deren Endpunkte, welche die kleinsten und größten Werte der Ziele enthalten, bei der Berechnung von SD_l in NDF_l mit auf. Jedoch wird dieser Schritt nur bei Problemen ausgeführt, bei denen die effiziente Front $E(z)$ von den getesteten Algorithmen nicht gefunden wurde⁹². Problematisch dabei ist, daß $E(z)$ nicht bekannt ist. Ebenfalls ist offen, was mit Elementen der NDF_l passiert, die von den in NDF_l zusätzlich eingefügten Endpunkten dominiert werden.

Eine wesentliche Einschränkung dieses Maßes ist, daß es nur auf Zielsysteme mit zwei Zielen anwendbar ist. Ansonsten ist nicht eindeutig festgelegt, welches das $z(x)$ unmittelbar folgende Element in der NDF_l ist.⁹³ Des Weiteren fallen bei mehr als zwei Zielen die größten und kleinsten Werte der Ziele nicht länger in zwei Elementen der heuristisch effizienten Front zusammen, wie dies bei lediglich zwei Zielen der Fall ist.

Schließlich wird gemäß Knowles (2002) auch die Relativitäts- und Monotonieforderung von diesem Maß nicht erfüllt.⁹⁴

Distanzquotient (Czyzak u. Jaszkiwicz (1998)): Aufbauend auf die zwei Distanzmaße $Dist1$ und $Dist2$ errechnet sich das Maß als deren Quotient:⁹⁵

⁸⁸Vgl. Deb u. a. (2000, S.856).

⁸⁹Sind die Elemente der heuristisch effizienten Front gleichmäßig verteilt, so ist $d(z(x))$ für alle $z(x) \in NDF_l$ gleich groß. Ist dies der Fall, ist der Wert von $d(z(x))$ gleich dem Durchschnittswert \bar{d} aller $d(z(x))$. Der Zähler des Terms ist damit Null.

⁹⁰Vgl. Deb u. a. (2000, S.855f.).

⁹¹Vgl. Knowles (2002, S.83).

⁹²Vgl. Deb u. a. (2000, S.855).

⁹³Vgl. Knowles (2002, S.83). Wobei Knowles (2002, S.82f.) keine Aussage macht, wie das unmittelbar auf $z(x)$ folgende Element definiert ist.

⁹⁴Vgl. Knowles (2002, S.83).

⁹⁵Vgl. Czyzak u. Jaszkiwicz (1998, S.41).

$$SCZY_l = \frac{Dist1_l}{Dist2_l} \quad (4.28)$$

Czyzak u. Jaskiewicz (1998) interpretieren das Maß derart, daß ein kleiner Wert von $SCZY_l$ eine gleichmäßige Verteilung der Elemente der NDS_l über die Referenzlösung bedeutet. Je kleiner der Wert ist, desto gleichmäßiger sind die Elemente folglich verteilt.⁹⁶ Diese Beurteilung berücksichtigt die Eigenschaften der $TMPS$, falls diese als Referenz verwendet wird und stellt damit nicht auf eine grundsätzliche Gleichverteilung der Lösung ab. Problematisch ist jedoch, daß das Maß nicht immer definiert ist, wie Abbildung 4.10 zeigt. Hier hätte $Dist2_A$ den Wert Null. Für die optimale Lösung ist das Maß wegen $Dist2_{TMPS} = 0$ nicht definiert. Somit ist es nicht möglich den Wert der besten Lösung zu bestimmen. Aus diesem Grund sind die Relativitäts- und Monotonieforderung nicht erfüllt.

Cluster-Maß (Wu u. Azarm (2001)): Das "Cluster Maß" (CL_l^μ) errechnet sich als Quotient aus der Mächtigkeit der heuristisch effizienten Menge und der "Number of Distinct Choices" (NDC_l^μ).⁹⁷ Diese ist die Anzahl der Elemente aus NDS_l , die im Zielraum ausreichend unterschiedlich sind. Der Grad der verlangten Unterschiedlichkeit wird durch den Parameter μ ($0 < \mu < 1$) eingestellt.⁹⁸

$$CL_l^\mu = \frac{|NDS_l|}{NDC_l^\mu} \quad (4.29)$$

Im Folgenden soll die Berechnung der "Number of Distinct Choices" dargestellt werden.⁹⁹ Zunächst werden im skalierten Zielraum¹⁰⁰ $\left(\frac{1}{\mu}\right)^K$ (Hyper)Würfeln definiert. Dabei muß μ aus dem Bereich $0 < \mu < 1$ derart gewählt werden, daß der Term $\left(\frac{1}{\mu}\right)$ zu einer ganzen Zahl evaluiert.¹⁰¹ Ein einzelner (Hyper)Würfel wird auch als Indifferenzregion $T_\mu(c)$ bezeichnet. Dabei definiert der Vektor $c = (c_1, c_2, \dots, c_k, \dots, c_K)$ eine Ecke des (Hyper)Würfels. Die Elemente des Vektors c ergeben sich aus:

$$c_k = l_k \cdot \mu, \quad k = 1(1)K \quad (4.30)$$

Die "Number of Distict Choices" ermittelt nun die Anzahl an Indiffe-

⁹⁶Vgl. Czyzak u. Jaskiewicz (1998, S.41).

⁹⁷Vgl. Wu u. Azarm (2001, S.22f.).

⁹⁸Vgl. Wu u. Azarm (2001, S.22).

⁹⁹Vgl. zur im Folgenden dargestellten Berechnung auch Wu u. Azarm (2001, S.22f.).

¹⁰⁰Die Skalierung wird mit Hilfe von Gleichung 4.20 vorgenommen.

¹⁰¹Nur so ist sichergestellt, daß eine ganz Zahl an (Hyper)Würfeln betrachtet wird, da K als Anzahl der Ziele ebenfalls ganzzahlig ist. Wu u. Azarm (2001) treffen dazu einfach die entsprechende Annahme, daß $\frac{1}{\mu}$ ganzzahlig sei (vgl. Wu u. Azarm (2001, S.22)).

renzregionen $T_\mu(c)$, in denen mindestens ein normierter Zielfunktionsvektor $\tilde{z}(x)$ eines Elements π aus der NDS_l bei gegebenem μ liegt.

Ob bei gegebenem μ in der Indifferenzregion $T_\mu(c)$ mindestens ein normierter Zielfunktionsvektor eines Element aus NDS_l liegt wird durch die Binärvariable $NT_l^\mu(c)$ erfaßt:

$$NT_l^\mu(c) = \begin{cases} 1 & \text{falls } \exists x \in NDS_l \text{ mit } \tilde{z}(x) \in T_\mu(c) \\ 0 & \text{sonst} \end{cases} \quad (4.31)$$

Damit $\tilde{z}(x) \in T_\mu(c)$ ist, muß gelten: $c_k \leq \tilde{z}_k(x) < c_k + \mu \forall k$.

NDC_l^μ errechnet sich dann wie folgt:

$$NDC_l^\mu = \sum_{l_K=0}^{\frac{1}{\mu}-1} \cdots \sum_{l_k=0}^{\frac{1}{\mu}-1} \cdots \sum_{l_2=0}^{\frac{1}{\mu}-1} \sum_{l_1=0}^{\frac{1}{\mu}-1} NT_l^\mu(c) \quad (4.32)$$

Ist das "Cluster-Maß" Eins, dann sind alle Elemente in der heuristisch effizienten Menge ausreichend verschieden und aufgrund der in Gleichung 4.32 beschriebenen Berechnungsmethode von NDC_l^μ im skalierten Zielraum gleichmäßig verteilt. Ein Wert größte als Eins zeigt an, daß die Elemente der heuristisch effizienten Menge dichter beieinander liegen, also Cluster bilden.¹⁰² Wu u. Azarm (2001) streben ebenfalls eine gleichmäßige Verteilung an, so daß $CL_l^\mu = 1$ der beste mögliche Wert ist.¹⁰³

Abbildung 4.17 verdeutlicht die Bestimmung von NDC_l^μ und CL_l^μ anhand eines Beispiels.

Im Beispiel wurde $\mu = \frac{1}{6}$ gesetzt. Mit $K = 2$ ergeben sich somit 36 Indifferenzregionen. Diese werden nacheinander daraufhin überprüft, ob $NT_l^\mu(c)$ den Wert 1 oder 0 hat. Im Beispiel werden die Regionen zeilenweise von links nach rechts sowie von unten nach oben systematisch überprüft.¹⁰⁴ Im Folgenden sollen beispielhaft drei Indifferenzregionen genauer betrachtet werden.

Indifferenzregion 10 wird durch den Eckpunkt bzw. den Vektor $c^{10} = (3 \cdot \frac{1}{6}; 1 \cdot \frac{1}{6}) = (\frac{3}{6}; \frac{1}{6})$ bestimmt.¹⁰⁵ In dieser Indifferenzregion $T_\mu(c^{10})$ liegt $z(C')$, weshalb $NT_A^\mu(c^{10}) = 1$ ist. In $T_\mu(c^{22})$ liegt kein Element von NDF_A , weshalb $NT_A^\mu(c^{22}) = 0$ ist. Schließlich soll noch $T_\mu(c^{26})$ genauer betrachtet werden. In dieser Region liegen gleich zwei Elemente von NDF_A . Es genügt jedoch, daß ein Element in der Region liegt, damit $NT_A^\mu(c^{26})$ den Wert 1 annimmt.

¹⁰²Wu u. Azarm (2001, S.23).

¹⁰³Wu u. Azarm (2001, S.23).

¹⁰⁴Also in aufsteigender Reihenfolge der Numerierung.

¹⁰⁵Die Werte der Summenindizes sind also: $l_1 = 3$ und $l_2 = 1$.

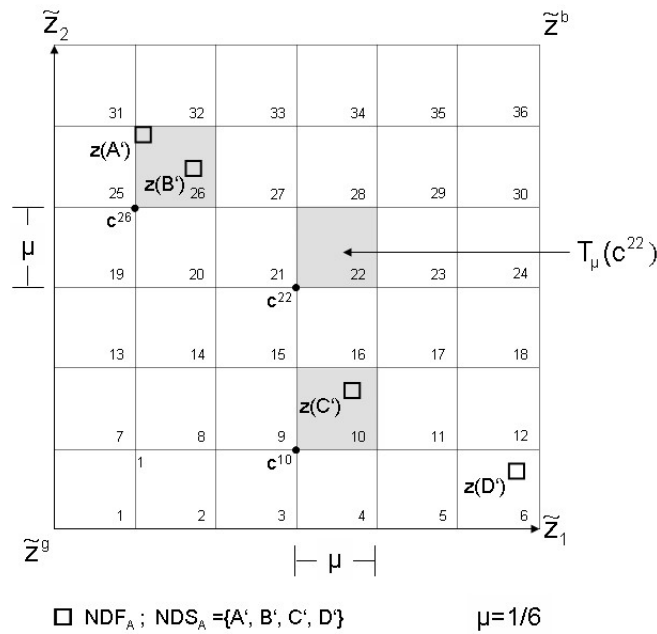


Abbildung 4.17: Cluster-Maß

Insgesamt gibt es 3 Indifferenzregionen, in denen mindestens ein Zielfunktionsvektor der Elemente der NDS_A liegt. NDC_A^μ ist daher 3. Das Clustermaß evaluiert damit zu $CL_A^\mu = \frac{4}{3}$. Die Elemente der heuristisch effizienten Menge sind daher nicht alle ausreichend verschieden.

In der beschriebenen, von Wu u. Azarm (2001) vorgestellten Form hat das Maß den Nachteil, daß es eine Bewertung ohne Bezug zur Menge aller effizienten Lösungen $E(\Pi, z)$ bzw. der $TMPS$ vornimmt. Bei der Beurteilung wird grundsätzlich eine gleichmäßige Verteilung bevorzugt. Des weiteren bestehen Freiheitsgrade bei der Bestimmung des Parameters μ ($0 < \mu < 1$). Der anzustrebende Wert ist nicht bekannt. Daher kann die Eigenschaft der Relativität und Monotonie nicht beurteilt werden.

Um den notwendigen Bezug zur $TMPS$ herzustellen, wird an dieser Stelle vorgeschlagen, den Quotienten aus dem "Cluster Maß" der $TMPS$ und der zu bewertenden heuristisch effizienten Menge zu bilden. Die "Cluster Ratio" errechnet sich also zu:

$$CLR_i^\mu = \frac{CL_i^\mu}{CL_{TMPS}^\mu} \quad (4.33)$$

Ein Wert von Eins ist der angestrebte Wert. In diesem Fall kann gelten,

daß die Elemente der heuristisch effizienten Menge in dem von ihr abgedeckten Bereich des Zielraums im gleichen Ausmaß verschieden sind wie die Elemente der *TMPS* im von ihr abgedeckten Bereich. Ein Wert größer Eins zeigt an, daß die NDS_l im von ihr abgedeckten Bereich des Zielraums eine stärkere Clusterbildung aufweist als die *TMPS* in dem von ihr abgedeckten Bereich. Ein Wert kleiner Eins zeigt, daß die heuristisch effiziente Menge entsprechend gleichmäßiger verteilt ist als die *TMPS*.

Das Maß erfüllt die Anforderung der schwachen Relativität, wenn Eins als bester Wert angesehen wird. Die Monotonieeigenschaft wird hingegen auch von diesem Maß nicht erfüllt. Dies soll an Abbildung 4.18 verdeutlicht werden.

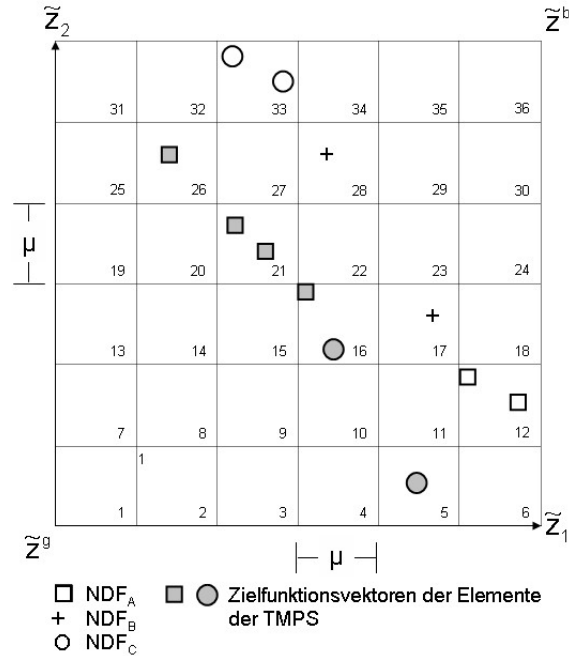


Abbildung 4.18: Cluster Ratio

Offensichtlich hat die *TMPS* nicht alleine die beste mögliche Bewertung von 1 sondern auch NDS_A . Denn es ergibt sich: $CLR_{TMPS}^\mu = \frac{6/4}{6/4} = 1$ und $CLR_A^\mu = \frac{6/4}{6/4} = 1$. Damit besteht nur schwache Relativität. Würde in NDS_C ein Element eingefügt, dessen Zielfunktionsvektor z.B. in Indifferenzregion 28 liegt und kein bisheriges Element aus NDS_C dominiert noch von einem solchen dominiert wird, so würde sich die Bewertung von NDS_C von $CLR_C^\mu = \frac{4/3}{6/4} = \frac{8}{9} = 0,888$ auf $CLR_C^\mu = \frac{5/4}{6/4} = \frac{5}{6} = 0,833$ verschlechtern. Damit besteht keine Monotonie.

Zusammenfassung der Maße und ihrer Eigenschaften

Die folgende Tabelle faßt die diskutierten Eigenschaften der Maße nochmals zusammen:

	Relativität	Monotonie	Sonstiges
SD_i	-	-	Nur für $K=2$
$SCZY_i$	-	-	Nicht immer def.
CL_i^μ	-	-	
CLR_i^μ	+	-	

++ Monotonie bzw. Relativität erfüllt

+ Monotonie bzw. Relativität schwach erfüllt

- Monotonie bzw. Relativität nicht erfüllt

Tabelle 4.3: Eigenschaften der Qualitätsmaße zur Erfassung der Verteilung der Elemente der NDS im Zielraum

4.2.4 Optimierungseigenschaft bezüglich der einzelnen Ziele

Dieser Qualitätsaspekt ist durch den Gedanken motiviert, daß ein multi-kriterielles Verfahren auch eine gute Optimierungseigenschaft bezüglich der einzelnen Ziele des Zielsystems haben sollte.

Relative Abweichung vom Minimum der TMPF: Dieses Maß ermittelt für jedes Ziel die Abweichung des kleinsten Wertes in der heuristisch effizienten Menge, vom kleinsten Wert in der TMPF.

$$DTMPF_i^k = \frac{z_i^{k*}}{z_{TMPF}^{k*}} \quad (4.34)$$

mit:

$$z_i^{k*} = \min \{z_k(x) \mid x \in NDS_i\} \quad (4.35)$$

$$z_{TMPF}^{k*} = \min \{z_k(x) \mid x \in TMPS\} \quad (4.36)$$

Aufgrund der Definition der $TMPS$ muß offensichtlich gelten: $z_{TMPS}^{k*} \leq z_i^{k*}$. Der beste und kleinstmögliche Wert ist somit Eins. Das Maß erfüllt die schwache Relativitätsanforderung. Die $TMPS$ hat für alle Zielgrößen k bzgl. $DTMPF_i^k$ einen Wert von 1. Eine bessere Bewertung als die der $TMPS$ ist nicht möglich, da es ausgeschlossen ist, daß $DTMPF_i^k$ einen

Wert kleiner als Eins annimmt, da immer $z_{TMPS}^{k*} \leq z_l^{k*}$ gilt. Ebenso ist das Maß schwach monoton. Das Hinzufügen eines Elements muß den Wert z_l^{k*} der NDF_l bzgl. eines Ziels k nicht verbessern, er kann ihn aber auch nicht verschlechtern.

Relative Abweichung von der besten bekannten kleinsten oberen

Schranke: Die relative Abweichung vom Minimum der $TMPF$ $DTMPF_l^k$ stellt keine Beziehung zur bislang besten bekannten kleinsten oberen Schranke der Zielgröße k her. Diese bisher beste bekannte kleinste obere Schranke kann z.B. von speziellen auf die Minimierung einer einzelnen Zielgröße ausgestalteten Verfahren ermittelt worden sein. Um einen Vergleich mit solchen auf die Minimierung einzelner Zielgrößen spezialisierten Verfahren durchzuführen, ist es daher erforderlich, den besten von diesen ermittelten Werten bei der Beurteilung zu berücksichtigen. Das folgende Maß berücksichtigt bei der Beurteilung diese kleinste obere Schranke (z_{UB}^k). Des weiteren hat $DTMPF_l^k$ den Nachteil, daß, falls der Nenner den Wert Null annimmt, das Maß nicht definiert ist. Daher wird sowohl im Zähler als auch im Nenner ein Wert ζ^k addiert. Werden Zielgrößen betrachtet, welche nur positive Werte annehmen können oder aber Zielgrößen, welche den Wert Null annehmen können, dieser aber in den zu beurteilten Lösungen nicht vorkommt und gleichzeitig $z_{UB}^k > 0$ gilt, so kann ζ^k gleich Null gesetzt werden. Ansonsten ist ζ^k als sehr kleine Zahl zu wählen.

$$DUB_l^k = \frac{z_l^{k*} + \zeta^k}{\min \{z_{UB}^k; z_{TMPF}^k\} + \zeta^k} \quad (4.37)$$

Es ist durchaus möglich, daß eine Lösung die exogen vorgegebenen kleinsten oberen Schranken z_{UB}^k verbessert. Diese Situation wird im Nenner berücksichtigt.

Der beste mögliche und kleinste Wert ist erneut Eins. Jedoch muß dieser nicht zwingend von der $TMPF$ erreicht werden. Dennoch kann keine heuristisch effiziente Menge bzw. ihre heuristisch effiziente Front eine bessere Beurteilung aufweisen als die $TMPF$. Das Maß ist daher schwach relativ. Ebenso ist es schwach monoton.

Anteil der gefundenen Extrempunkte der $TMPF$: Eine Abweichung von Null für $DTMPF_l^k$ bedeutet nicht, daß in der entsprechenden heuristisch effizienten Menge eine Lösung existiert, deren Zielfunktionsvektor den Extrempunkten der "Total Meta Pareto Front" bezüglich des betrachteten Ziels entspricht. Zwar sind die Werte für Ziel k identisch, dennoch ist

es möglich, daß der Zielfunktionsvektor der Lösung vom Extrempunkt der "Total Meta Pareto Front" dominiert wird. Das folgende Beispiel soll diese Situation verdeutlichen:

Für das erste Problem der Taillard-Benchmarks beträgt die beste bekannte kleinste obere Schranke für die Zykluszeit 1278.¹⁰⁶ Wird das vollständige Zielsystem betrachtet, so existieren mehrere Zielfunktionsvektoren, die den Wert 1278 für die Zykluszeit aufweisen.¹⁰⁷

#	z_{Cmax}	z_{Fmid}	z_{Tmax}
1	1278	701,7	528
2	1278	702,05	529
3	1278	702,1	528
4	1278	703,6	581
5	1278	703,0	530
⋮	⋮	⋮	⋮

Tabelle 4.4: Multiple Optima der Zykluszeit in TAIL001

Zielfunktionsvektor 1 sei nun ein Element der "Total Meta Pareto Front". Die Zielfunktionsvektoren 2 und 3 seien Elemente der heuristisch effizienten Front NDF_A der heuristisch effizienten Menge NDS_A . Obwohl die Maße $DTMPF_A^{Cmax}$ und DUB_A^{Cmax} einen Wert von Eins aufweisen, enthält NDS_A kein effizientes Element.

Daher sollte zusätzlich ein Maß existieren, das anzeigt, ob die Zielfunktionsvektoren der $TMPF$, die für Ziel k das Minimum besitzen, in der zu beurteilenden heuristisch effizienten Front enthalten sind.

$$AEX_l^k = \frac{|ETMPF^k \cap ENDF_l^k|}{|ETMPF^k|} \quad (4.38)$$

mit:

$$ETMPF^k = \{z(x) \mid z_k(x) = \min \{z(x) \in TMPF\}\} \quad (4.39)$$

$$ENDF_l^k = \{z(x) \mid z_k(x) = \min \{z(x) \in NDF_l\}\} \quad (4.40)$$

$ETMPF^k$ ist die Menge der Zielfunktionsvektoren, die für Ziel k den kleinsten Wert innerhalb der $TMPF$ aufweisen.

$ENDF_l^k$ ist die Menge der Zielfunktionsvektoren, die für Ziel k den kleinsten Wert innerhalb von NDF_l aufweisen.

¹⁰⁶Vgl. Taillard (1993, S.280). Dem Verfasser ist keine Arbeit bekannt, die einen besseren Wert ermittelt.

¹⁰⁷Die Tabelle ist keine abschließende Aufzählung der Zielfunktionsvektoren.

Der beste und größte mögliche Wert ist Eins. Daß Maß AEX_l^k ist schwach relativ und schwach monoton.

4.3 Aufbau des verwendeten Beurteilungssystems

Die Diskussion unterschiedlicher Qualitätsmaße in diesem Kapitel hat gezeigt, daß ein großer Teil der existierenden Maße allein aus theoretischen Überlegungen heraus nicht für die Verwendung in einer Qualitätsanalyse geeignet ist.

Zur Erfassung der Distanz der NDS zur Menge aller effizienten Lösungen empfiehlt sich das S-Maß, auch wenn es nur die heuristisch effiziente Front als Beurteilungsgrundlage verwendet. In dieser Arbeit wird ein Quotient des S-Maßes der zu beurteilenden heuristisch effizienten Front und der $TMPF$ verwendet werden:

$$SQ_l = \frac{S_l}{S_{TMPF}} \quad (4.41)$$

Die Verwendung dieses Quotienten ermöglicht eine Aussage darüber, zu welchem Anteil der von der $TMPF$ dominierte Bereich von der zu beurteilenden Lösung abgedeckt wird. Es gilt immer: $0 < SQ_l \leq 1$.

Zur Bewertung der Qualitätsaspekte, Variationsbreite der Zielfunktionswerte und Clusterung der heuristisch effizienten Front, ist keines der in diesem Kapitel vorgestellten Maße geeignet. Auch wenn diese Qualitätsaspekte häufig betrachtet werden und von Interesse sind, werden sie in dieser Arbeit durch kein Maß berücksichtigt.

Die Maße $DTMPF_l^k$ und DUB_l^k erfüllen die Anforderungen der Relativität und Monotonie zumindest in der schwachen Form. DUB_l^k hat jedoch einen höheren Aussagegehalt und ist im Gegensatz zu $DTMPF_l^k$ immer definiert. Daher wird ihm in dieser Arbeit der Vorzug gegeben und neben AEX_l^k sowie dem SQ-Maß in das System von Qualitätsmaßen aufgenommen.

Kapitel 5

Zielraum- und Lösungsraumanalyse

5.1 Grundlegende Aspekte

5.1.1 Stand der Forschung

Ein gängiges Vorgehen bei der Problemlösung mit Hilfe von Metaheuristiken ist es, die Suche zunächst in einen vielversprechenden Bereich des Lösungsraums zu lenken (Grobsuche) und diesen anschließend genauer zu untersuchen (Feinsuche). Das Identifizieren eines vielversprechenden Bereichs soll durch Anwenden der Suchstrategie der verwendeten Metaheuristik erreicht werden. Die so erzeugten Lösungen dienen anschließend alle oder teilweise als Ausgangslösung für ein einfaches lokales Suchverfahren. Durch dessen Anwendung soll der angesteuerte Bereich des Lösungsraums genauer untersucht werden. Lösungsansätze, die die Suchstrategie einer Metaheuristik und eines einfachen lokalen Suchverfahrens miteinander verbinden, werden auch als "Hybride Ansätze" bezeichnet. Dieses Vorgehensweise ist motiviert durch die Vorstellung, daß jedes globale und lokale Optimum einen Einzugsbereich im Lösungsraum besitzt. Es soll die Suche zunächst in den Einzugsbereich eines oder mehrere Optima gelenkt werden, um anschließend mit einer lokalen Suche das lokale oder gar globale Optimum zu finden. Die Analyse der Einzugsbereiche von globalen und lokalen Optima ist in den letzten Jahren Thema einiger wissenschaftlicher Veröffentlichungen gewesen.¹ Von besonderem Interesse war dabei die Frage, ob die globalen und lokalen Optima in einem Bereich des Lösungsraums

¹Siehe z.B. Boese u. a. (1993, S.1-12); Boese u. a. (1994, S.101-110); Borges u. Hansen (1998, S.1-14); Reeves (1999, S.479-489).

nahe beieinander liegen oder aber zusammenhanglos verstreut über den gesamten Lösungsraum angeordnet sind. Boese u. a. (1993, 1994) haben in ihrer Arbeit die Existenz des sogenannten "Großen Tals"² für das symmetrische TSP bei Verwendung einer Lin-2-opt-Nachbarschaft festgestellt.³ Darauf aufbauend hat Reeves (1999) eine entsprechende Analyse für das Permutation Flow Shop Problem durchgeführt und in seinen Experimenten für das Ziel der Zykluszeit und unterschiedliche Nachbarschaftsoperatoren die Existenz des "Großen Tals" ebenfalls beobachtet.⁴ Das "Große Tal" beschreibt in den durchgeführten Versuchen die Möglichkeit, mit nur wenigen Zügen eines Nachbarschaftsoperators von einem lokalen Optimum in ein anderes zu gelangen.⁵ Es liegt dann vor, wenn die lokalen Optima in einem kleinen Bereich des Lösungsraums nahe beieinander liegen.⁶ Die Distanz wird als die Anzahl der mindestens notwendigen Züge mit einem Nachbarschaftsoperator gemessen.⁷ Es ist wichtig zu erkennen, daß das Vorliegen eines "Großen Tals" keine grundsätzliche Eigenschaft eines Optimierungsproblems ist. Vielmehr ist die Distanz zwischen den lokalen Optima abhängig von dem bei der lokalen Suche verwendeten Nachbarschaftsoperators.⁸ In Abhängigkeit vom verwendeten Nachbarschaftsoperator können Auftragsfolgen direkte Nachbarn⁹ im Lösungsraum sein oder aber mehrere Züge voneinander entfernt liegen. Die beschriebenen Ergebnisse, insbesondere die Beobachtung des "Großen Tals", lassen das Vorgehen der "Hybriden Ansätze" als gerechtfertigt erscheinen. Die Arbeit von Reeves (1999) betrachtet nur die Zykluszeit als Zielgröße.¹⁰ Dies ist für den in dieser Arbeit betrachteten, multikriteriellen Fall insofern von Bedeutung, als ein Transfer der Aussagen auf die zwei zusätzlich betrachteten Zielgrößen der mittleren Durchlaufzeit und der maximalen Terminüberschreitung nicht vorgenommen werden kann. Für sie müssen eigene Analysen durchgeführt werden.

Für das multikriterielle TSP gibt es bereits eine Arbeit von Borges u. Hansen (1998).¹¹ Diese betrachten ein Zielsystem, welches dreimal die glei-

²Alternativ zum "Großen Tal" wird auch der Begriff der "Global Convexity" verwendet (vgl. Borges u. Hansen (1998, S.1)).

³Vgl. Boese u. a. (1993, S.12); Boese u. a. (1994, S.110).

⁴Vgl. Reeves (1999, S.483 u. 489).

⁵Vgl. Boese u. a. (1993, S.4), Boese u. a. (1994, S.103f.).

⁶Dabei ergibt sich die Größe des Bereichs über die Anzahl der notwendigen Züge, bzw. ihrer Approximation zwischen den lokalen Optima.

⁷Vgl. Boese u. a. (1993, S.3), Boese u. a. (1994, S.103), sowie Reeves (1999, S.478).

⁸Vgl. Borges u. Hansen (1998, S.1).

⁹Zwei Auftragsfolgen gelten als direkt benachbart, wenn nur ein einziger Zug mit einem Nachbarschaftsoperator nötig ist, um von einer Auftragsfolge zur anderen Auftragsfolge zu gelangen.

¹⁰Vgl. Reeves (1999, S.477).

¹¹Vgl. Borges u. Hansen (1998, S.4-14).

che Zielgröße enthält. Dabei wird für die Berechnung jeder der drei Zielgrößen eine andere Entfernungsmatrix verwendet, so daß sich jeweils unterschiedliche Zielwerte ergeben und die Zielwerte als voneinander unabhängig betrachtet werden können.¹² Im Mittelpunkt der Betrachtungen steht unter anderem die heuristisch nichtdominierte Menge und die Frage, ob sich auch hier eine "Global Convexity" bei Verwendung von unterschiedlichen Kompromißzielfunktionen feststellen läßt, ob also die Elemente der heuristisch effizienten Menge nahe beieinander liegen. Als Kompromißzielfunktion verwenden sie zum einen eine Konvexkombination und zum anderen die erweiterte Tchebycheff-Distanz. Bei Verwendung einer 2-opt-Nachbarschaft und Untersuchung einer einzigen Probleminstanz beobachten sie in ihren Experimenten, daß die Elemente der heuristisch effizienten Menge nahe beieinander liegen.¹³ Für echt multikriterielle Permutation Flow Shop Probleme existieren bisher keine vergleichbaren Untersuchungen.

Alle betrachteten Arbeiten haben gemeinsam, daß sie die echte Distanz zwischen den einzelnen Permutationen (von Auftragsfolgen beim PFSP, Städte beim TSP) nicht messen können, sondern vielmehr mit Approximationen der echten Distanz arbeiten.¹⁴

5.1.2 Weiterführende Experimente

Die bisherigen Versuche zur Existenz des "Großen Tals" sind ohne eine explizite Betrachtung der Nachbarschaft im Zielraum ausgekommen. Zwei Auftragsfolgen gelten in dieser Vorstellung als im Lösungsraum nahe beieinander oder benachbart, wenn nur wenige oder ein Zug mit einem Nachbarschaftsoperator durchgeführt werden muß, um eine Auftragsfolge in die andere zu überführen. Dies bedeutet daher nicht, daß die Nachbarn im Lösungsraum auch Nachbarn im Zielraum sind. Die Beantwortung der Frage, ob oder wie häufig ein Nachbar im Lösungsraum auch ein Nachbar im Zielraum ist, ist insbesondere beim vorliegenden Problem nicht ohne Bedeutung,¹⁵ wie in den weiteren Ausführungen gezeigt wird.

Viele Ansätze, die eine heuristisch effiziente Menge oder heuristisch effi-

¹²Vgl. Borges u. Hansen (1998, S.5f.).

¹³Wobei es eigentlich keinen vernünftigen Grund für das Auftreten der "Global Convexity" gibt. Es werden für jede der drei Zielgrößen drei unabhängige Kostenmatrizen verwendet, und auch die Funktionen zur Berechnung der Zielgrößen stellen keine Verbindung zwischen den Matrizen her. Die drei Ziele sind daher völlig unabhängig voneinander.

¹⁴Dieses Problem der Distanzmessung wird in Abschnitt 5.4 betrachtet.

Als Distanz zwischen zwei Auftragsfolgen wird die Anzahl der mindestens notwendigen Züge mit einem Nachbarschaftsoperator im Lösungsraum verstanden, um die Auftragsfolge π in π' zu überführen (vgl. Reeves (1999, S.478)). Zur Unterscheidung zu der über Ersatzmaße ermittelten Distanz wird diese Distanz im Folgenden auch als echte Distanz bezeichnet.

¹⁵Die Definition der Nachbarschaft im Zielraum erfolgt in Abschnitt 5.3.

ziente Front bestimmen wollen, nutzen eine Kompromißzielfunktion und variieren deren Parameter.¹⁶ Die Parametervariation soll die Suche in unterschiedliche Bereiche des Zielraums lenken.¹⁷ Ein systematisches Vorgehen bei der Parametervariation ermöglicht das systematische Absuchen des Zielraums nach Elementen der Lösungsmenge. Zu jeder Parameter-einstellung der Kompromißzielfunktion gibt es ein oder mehrere globale Optima¹⁸, welche gemäß der Vorstellung des "Großen Tals" jeweils einen Einzugsbereich im Lösungsraum besitzen. Ein hybrider Ansatz versucht zunächst den Einzugsbereich zu identifizieren und anschließend ein globales Optimum zu finden. Das Ergebnis des Suchprozesses für eine gegebene Parametervariation liegt mit hoher Wahrscheinlichkeit im angesteuerten Bereich des Zielraums. Damit wird das grundlegende Ziel einer systematischen Parametervariation erreicht.

Viele Ansätze¹⁹ überprüfen nur das Ergebnis einer lokalen Suche, nicht jedoch die Ausgangslösung oder die während der lokalen Suche generierten Lösungen auf heuristische Effizienz. Ein Verfahren sollte alle während der lokalen Suche erzeugten Lösungen überprüfen. Ansonsten werden unter Umständen heuristisch effiziente Lösungen generiert, aber nicht als solche erkannt. Die so identifizierten heuristisch effizienten Lösungen werden jedoch im Zielraum verstreut und i.d.R. nicht in dem durch die Parameter-einstellung des Kompromißmodells angesteuerten Bereich des Zielraums liegen. Insbesondere so identifizierte heuristisch effiziente Lösungen, deren Kompromißzielfunktionswert größer ist als der der Ausgangslösung, liegen vermutlich außerhalb dieses angesteuerten Bereichs. Anhand von Abbildung 5.1 soll dieser Gedanke nochmals verdeutlicht werden.

Die aktuelle Lösung der lokalen Suche sei $z(T)$. Die Parameter des Kompromißmodells sind derart eingestellt, daß Punkt $z(D)$ identifiziert werden soll. Werden nun die Nachbarn im Lösungsraum von $z(T)$ auf heuristische Effizienz untersucht, so kann es sein, daß $z(F)$ als Element der aktuellen heuristisch effizienten Menge gefunden wird. Dieser Punkt liegt jedoch nicht im angesteuerten Bereich des Zielraums um Punkt $z(D)$.

Erweitert man die Idee des systematischen Absuchens des Zielraums zur Identifikation aller effizienten Lösungen auf den Prozeß der lokalen

¹⁶Zu diesen Ansätzen zählen z.B. die in Abschnitt 3.3.3 angesprochenen Ansätze von Daniels u. Chambers (1990), Ishibuchi u. Murata (1998), Framinan u. a. (2002), Chang u. a. (2002) und Loukil u. a. (2005).

¹⁷Vgl. auch Abschnitt 3.2.

¹⁸Siehe zu multiplen globalen Optima auch den Abschnitt 5.5.2.

¹⁹Zu diesen Ansätzen gehören z.B. die in Abschnitt 3.3.3 angesprochenen Ansätze von Ishibuchi u. Murata (1998) und Chang u. a. (2002). Der Ansatz von Loukil u. a. (2005) verwendet keine lokale Suche, überprüft aber ansonsten jede generierte Auftragsfolge auf heuristische Effizienz.

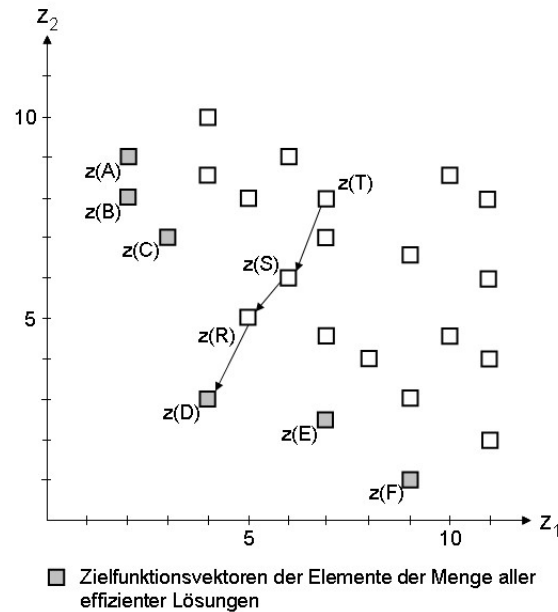


Abbildung 5.1: Motivation der Zielraum- und Lösungsraumanalyse

Suche, so wäre es wünschenswert, gegen Ende des lokalen Suchprozesses gezielt die Nachbarn der Ausgangslösung im Zielraum zu suchen bzw. die Streuwirkung zu reduzieren. Insbesondere die Ausgangslösung der lokalen Suche kann weit von der effizienten Front entfernt liegen. Eine Auswertung ihrer unmittelbaren Nachbarschaft im Zielraum macht daher noch keinen Sinn, da hier keine effizienten Lösungen zu erwarten sind. Daher ist in einer ersten Phase der lokalen Suche zunächst eine Annäherung an die effiziente Front erwünscht. Erst nachdem diese Annäherung erfolgt ist, erscheint es sinnvoll, die unmittelbare Nachbarschaft im Zielraum nach effizienten Elementen abzusuchen. Damit dies möglich ist, wird eine Nachbarschaft im Lösungsraum benötigt, deren Elemente auch Nachbarn im Zielraum sind oder zumindest eine geringe Streuung um diese aufweisen. Die beschriebenen Überlegungen sollen mit Hilfe von Abbildung 5.1 verdeutlicht werden:

Die lokale Suche beginnt in $z(T)$. Dieser ist noch weit von der effizienten Front entfernt, so daß eine Auswertung seiner unmittelbaren Nachbarschaft im Zielraum noch keinen Sinn macht. Daher ist zunächst eine schnelle Annäherung an die effiziente Front erwünscht und die Parametereinstellung der Kompromißzielfunktion lenkt die Suche zunächst über Punkt $z(S)$ und $z(R)$ zu $z(D)$. $z(D)$ ist Element der effizienten Front und

eine Auswertung seiner Nachbarschaft im Zielraum ist wünschenswert, um die Punkte $z(C)$ und $z(E)$ zu finden. Offensichtlich wird dazu eine Nachbarschaft im Lösungsraum benötigt, die eine hohe Chance bietet, auch einen Nachbarn im Zielraum, insbesondere die Punkte $z(C)$ und $z(E)$, zu enthalten.

Bislang existiert keine solche Analyse der Nachbarschaften im Ziel- und Lösungsraum. Insbesondere ist noch kein Konzept zur Bestimmung der Nachbarschaft im Zielraum bekannt. Es läßt sich argumentieren, daß die Elemente, die innerhalb eines Abschnitts der effizienten Front liegen, im Zielraum benachbart sind oder zumindest im Zielraum nahe beieinander liegen. In Abschnitt 5.2 werden die in dieser Arbeit betrachteten Nachbarschaftsoperatoren im Lösungsraum definiert. Anschließend wird in Abschnitt 5.3 die Nachbarschaft im Zielraum eingeführt und die angesprochene Analyse durchgeführt.

In Abschnitt 5.4 wird die Qualität verschiedener Ersatzmaße zur Approximation der (echten) Distanz zwischen zwei Auftragsfolgen analysiert. Die betrachteten vier Maße wurden in der Untersuchung zum "Großen Tal" von Reeves (1999) verwendet.²⁰ In Abschnitt 5.5 soll die Topographie des Lösungsraums des in dieser Arbeit betrachteten multikriteriellen PFSP untersucht werden. Die Verwendung der von Reeves (1999) verwendeten Ersatzmaße erscheint dabei nur dann sinnvoll, wenn sie in der Lage sind, die echte Distanz angemessen gut zu approximieren. Die Analyse der Approximationsqualität ist daher Inhalt des Abschnitts 5.4.3. Wie sich zeigen wird, rechtfertigen die Ergebnisse die Verwendung der Ersatzmaße für weitere Untersuchungen zum "Großen Tal", wie sie in Abschnitt 5.5 durchgeführt werden, nicht. Es wird daher ein zu der Arbeit von Reeves (1999) abweichender Aufbau des Experiments gewählt.²¹

In 5.5 soll unter Verwendung der echten Distanz in einer Reihe weiterer Experimente untersucht werden, ob sich die Topographie des "Großen Tales" bzw. die Existenz einer "Global Convexity" für die Zykluszeit bestätigen und für die Ziele der mittleren Durchlaufzeit und der maximalen Terminüberschreitung nachweisen läßt. Ebenso soll analysiert werden, wie sich bei multiplen globalen Optima deren Lage zueinander verhält. Zunächst werden diese Analysen für jedes Ziel gesondert durchgeführt. Anschließend setzt ein weiteres Experiment die globalen Optima der Ziele zueinander in Beziehung. In einem abschließenden Experiment wird die Distanz der Elemente der effizienten Front im Lösungsraum analysiert.

²⁰Vgl. Reeves (1999, S.478-489).

²¹Vgl. Reeves (1999, S.478-489).

5.2 Nachbarschaften im Lösungsraum

5.2.1 Nachbarschaftsoperatoren

Als Nachbarschaft im Lösungsraum $NSLR(\pi^\circ)$ einer Auftragsfolge π° werden alle Auftragsfolgen verstanden, die durch einmaliges Anwenden eines sogenannten Nachbarschaftsoperators auf π° erzeugt werden können.²² Dieses einmalige Anwenden wird auch als Zug bezeichnet.²³ Ein Nachbarschaftsoperator ist ein deterministischer Transformationsmechanismus, der eine Ausgangsfolge modifiziert, um sie in eine andere Folge zu überführen.²⁴ Im Folgenden werden die in dieser Arbeit verwendeten Nachbarschaftsoperatoren und die darauf aufbauenden Nachbarschaften vorgestellt.²⁵

Exchange (EX) Der Exchange-Operator vertauscht einen Auftrag der Auftragsfolge mit einem beliebigen anderen Auftrag. Sollen z.B. die Aufträge auf den Positionen g und h vertauscht werden und ist die Ausgangssequenz

$\pi^\circ = (\pi_1, \dots, \pi_{h-1}, \pi_h, \pi_{h+1}, \dots, \pi_{g-1}, \pi_g, \pi_{g+1}, \dots, \pi_N)$, so ergibt sich als neue Auftragsfolge: $\pi' = (\pi_1, \dots, \pi_{h-1}, \pi_g, \pi_{h+1}, \dots, \pi_{g-1}, \pi_h, \pi_{g+1}, \dots, \pi_N)$.

Die Exchange-Nachbarschaft ergibt sich, indem jeder Auftrag mit jedem anderen Auftrag der Auftragsfolge vertauscht wird. Die Nachbarschaft hat damit eine Größe von:

$$|EX(\pi^\circ)| = \frac{N(N-1)}{2} \quad (5.1)$$

Adjacent Pairwise Interchange (API) Der API-Operator vertauscht einen Auftrag mit einem direkt benachbarten Auftrag. Bezogen auf das Beispiel des EX-Operators gilt somit $g = h + 1$.

Die API-Nachbarschaft ergibt sich, indem jeder Auftrag mit seinen benachbarten Aufträgen vertauscht wird. Diese Nachbarschaft ist somit eine Teilmenge der EX-Nachbarschaft und hat eine Größe von:

$$|API(\pi^\circ)| = N - 1 \quad (5.2)$$

Forward Shift (FSH) Der FSH-Operator verschiebt einen Auftrag von seiner gegenwärtigen Position auf eine Position, die größer ist als seine

²²Vgl. Pinedo (1995, S.148).

²³Vgl. Domschke u. Drexl (1998, S.121).

²⁴Vgl. Domschke u. Drexl (1998, S.121).

²⁵Dabei handelt es sich um die sechs in der Untersuchung von Reeves (1999) berücksichtigten Nachbarschaftsoperatoren (vgl. Reeves (1999, S.478f.)).

aktuelle. Soll demnach der Auftrag auf Position h an die Position g verschoben werden, so muß $g > h$ gelten. Ausgehend von der Ausgangssequenz $\pi^\circ = (\pi_1, \dots, \pi_{h-1}, \pi_h, \pi_{h+1}, \dots, \pi_g, \pi_{g+1}, \dots, \pi_N)$ ergibt sich als neue Auftragsfolge: $\pi' = (\pi_1, \dots, \pi_{h-1}, \pi_{h+1}, \dots, \pi_g, \pi_h, \pi_{g+1}, \dots, \pi_N)$.

Die FSH-Nachbarschaft ergibt sich, indem jeder Auftrag von seiner gegenwärtigen Position auf alle Positionen, die größer sind als seine aktuelle Position verschoben wird. Die Größe der Nachbarschaft ergibt sich somit zu:

$$|FSH(\pi^\circ)| = \frac{N(N-1)}{2} \quad (5.3)$$

Backward Shift (BSH) Der BSH-Operator verschiebt einen Auftrag von seiner gegenwärtigen Position auf eine Position, die kleiner ist als seine aktuelle. Soll demnach der Auftrag auf Position h an die Position g verschoben werden, so muß $g < h$ gelten. Ausgehend von der Ausgangssequenz $\pi^\circ = (\pi_1, \dots, \pi_{g-1}, \pi_g, \dots, \pi_{h-1}, \pi_h, \pi_{h+1}, \dots, \pi_N)$ ergibt sich $\pi' = (\pi_1, \dots, \pi_{g-1}, \pi_h, \pi_g, \dots, \pi_{h-1}, \pi_{h+1}, \dots, \pi_N)$.

Die BSH-Nachbarschaft ergibt sich, indem jeder Auftrag von seiner gegenwärtigen Position auf alle Positionen, die kleiner sind als seine aktuelle Position, verschoben wird. Die Größe der Nachbarschaft entspricht der des FSH-Operators:

$$|BSH(\pi^\circ)| = \frac{N(N-1)}{2} \quad (5.4)$$

Double Shift (DSH) Der DSH-Operator verschiebt einen Auftrag auf eine beliebige andere Position.

Die DSH-Nachbarschaft ergibt sich durch das aufeinanderfolgende Generieren der FSH- und BSH-Nachbarschaft. Die Größe der DSH-Nachbarschaft ist somit:²⁶

$$|DSH(\pi^\circ)| = N(N-1) \quad (5.5)$$

Inversion (INV) Der INV-Operator invertiert eine Teilauftragsfolge innerhalb der Ausgangsfolge. Soll der Operator auf die Auftragsfolge $\pi^\circ = (\pi_1, \dots, \pi_{e-1}, \pi_e, \pi_f, \pi_g, \pi_h, \pi_{h+1}, \dots, \pi_N)$ angewandt werden und ist als Teilauftragsfolge $(\pi_e, \pi_f, \pi_g, \pi_h)$ selektiert, so ergibt sich nach der Durchführung des Zugs: $\pi' = (\pi_1, \dots, \pi_{e-1}, \pi_h, \pi_g, \pi_f, \pi_e, \pi_{h+1}, \dots, \pi_N)$.

Die INV-Nachbarschaft wird durch Inversion aller möglichen Teilauftragsfolgen in der Permutation erzeugt. Die Größe der Nachbarschaft ist

²⁶Elemente, die auch in der API-Nachbarschaft enthalten sind, sind in dieser Nachbarschaft doppelt enthalten. Sie werden einmal vom FSH- und ein zweites Mal vom BSH-Operator erzeugt.

somit:

$$|INV(\pi^\circ)| = \frac{N(N-1)}{2} \quad (5.6)$$

5.2.2 Beziehungen zwischen Nachbarschaftsoperatoren

Von den vorgestellten Nachbarschaftsoperatoren erzeugt jeder eine eigene Topographie im Lösungsraum.²⁷ Die lokalen Optima, welche die Operatoren erreichen, können sich daher unterscheiden. Als lokales Optimum eines Nachbarschaftsoperators wird hierbei eine Lösung verstanden, die bei Anwendung des Nachbarschaftsoperators nicht mehr verlassen werden kann, da alle Nachbarn keinen besseren Zielfunktionswert haben als die Ausgangslösung.²⁸ Beim Versuch, diese lokalen Optima durch einen Wechsel des Nachbarschaftsoperators zu verlassen, sind einige Zusammenhänge zwischen den einzelnen Operatoren zu beachten. So gilt offensichtlich:

$$\begin{aligned} FSH(\pi^\circ) &\subset DSH(\pi^\circ), BSH(\pi^\circ) \subset DSH(\pi^\circ), \\ API(\pi^\circ) &\subset FSH(\pi^\circ), API(\pi^\circ) \subset BSH(\pi^\circ) \\ API(\pi^\circ) &\subset EX \text{ und} \\ API(\pi^\circ) &\subset INV(\pi^\circ) \end{aligned}$$

Dies bedeutet, daß die vom DSH-Operator erreichten lokalen Optima durch die Anwendung eines FSH- oder BSH-Operators nicht verlassen werden können. Ebenso können die lokalen Optima der FSH- und BSH-Operatoren mit Hilfe des API-Nachbarschaftsoperators nicht verlassen werden, also auch keine DSH-Optima. Schließlich ist auch ein Verlassen der lokalen Optima der INV-Nachbarschaft und EX-Nachbarschaft durch Anwenden des API-Operators nicht möglich.

²⁷Vgl. zu diesem Abschnitt Reeves (1999, S.479).

²⁸Diese Definition eines lokalen Optimums setzt die Verwendung eines lokalen Suchverfahrens voraus, bei dem ein Zug zu einem Nachbarn nur ausgeführt wird, wenn dieser einen besseren Zielfunktionswert hat. Für Suchverfahren, bei denen auch zu einem Nachbarn gezogen werden kann, der einen nicht besseren Zielfunktionswert hat (z.B. Simulated Annealing), muß das lokale Optimum anders definiert werden. Die hier verwendete Definition erscheint jedoch für die Zwecke dieser Arbeit zweckmäßig.

5.3 Nachbarschaft im Zielraum

5.3.1 Nachbarschaftskonzepte

5.3.1.1 Allgemeine Überlegungen

Ist die Diskussion der Nachbarschaft im Lösungsraum häufig Bestandteil von Arbeiten, die sich mit lokalen Suchverfahren für das Permutation Flow Shop Problem beschäftigen, so ist eine Auseinandersetzung mit der Nachbarschaft im Zielraum bislang nicht bekannt. Daher ist zunächst die Frage zu klären, wie ein Nachbar im Zielraum definiert ist. Anders als bei der Nachbarschaft im Lösungsraum, wo die Nachbarschaft über einen Transformationsmechanismus definiert ist, ist es möglich, im Zielraum auf die "Lage" der Zielfunktionsvektoren im K -dimensionalen Zielraum abzustellen. Ein Zielfunktionsvektor kann durch eine oder mehrere Auftragsfolgen realisiert werden. In Abhängigkeit davon, wie die Nachbarschaft im Zielraum definiert ist, gehören unterschiedliche Auftragsfolgen aufgrund ihrer Zielfunktionsvektoren der Nachbarschaft im Zielraum an. Weiterhin ist sie im Gegensatz zu den Nachbarschaften im Lösungsraum abhängig von einer konkreten Problemausprägung. In den hier betrachteten Permutation Flow Shop Problemen ist die Nachbarschaft im Zielraum somit abhängig von den Bearbeitungszeiten sowie den vorgegebenen spätestzulässigen Fertigstellungsterminen.

Zur Einführung soll der Fall lediglich einer Zielgröße betrachtet werden. In diesem Fall ergibt sich die Nachbarschaft im Zielraum einer Auftragsfolge intuitiv. Sie besteht aus allen Auftragsfolgen, deren Zielfunktionswerte die kleinste nichtpositive sowie kleinste nichtnegative Abweichung vom Zielfunktionswert der Ausgangslösung aufweisen. Gibt es Auftragsfolgen, die den gleichen Zielfunktionswert aufweisen, so wird die Nachbarschaft im Zielraum von diesen gebildet. Die Nachbarschaft im Zielraum der Ausgangslösung ergibt sich somit zu:

$NS_1^-(z(\pi^\circ))$: Menge aller Auftragsfolgen, deren Zielfunktionswert die kleinste nichtpositive Abweichung vom Zielfunktionswert $z(\pi^\circ)$ der Ausgangsfolge π° aufweist.

$$NS_1^-(z(\pi^\circ)) : = \{ \pi \mid z(\pi) = \max \{ z(\pi') \mid z(\pi') \leq z(\pi^\circ) \} \\ \pi^\circ, \pi' \in \Pi; \pi' \neq \pi^\circ \} \quad (5.7)$$

$NS_1^+(z(\pi^\circ))$: Menge aller Auftragsfolgen, deren Zielfunktionswert die kleinste nichtnegative Abweichung vom Zielfunktionswert $z(\pi^\circ)$ der

Ausgangsfolge π° aufweist.

$$NS_1^+(z(\pi^\circ)) : = \{ \pi \mid z(\pi) = \min \{ z(\pi') \mid z(\pi') \geq z(\pi^\circ) \} \\ \pi^\circ, \pi' \in \Pi; \pi' \neq \pi^\circ \} \quad (5.8)$$

Die gesamte Nachbarschaft ergibt als:

$$NS_1(z(\pi^\circ)) := NS_1^-(z(\pi^\circ)) \cup NS_1^+(z(\pi^\circ)) \quad (5.9)$$

Es könnte auch die Ansicht vertreten werden, daß lediglich die Auftragsfolgen als Nachbarn im Zielraum betrachtet werden, deren Zielfunktionswerte den kleinsten absoluten Abstand zum Zielfunktionswert der Ausgangsfolge π° aufweisen. Diese Ansicht erscheint jedoch wenig zweckmäßig, da dann nicht sichergestellt ist, daß ein "linker" (im Sinne von kleinerer) und ein "rechter" (im Sinne von größerer) Nachbar in die Nachbarschaft aufgenommen wird.

Wird die Anzahl der Ziele erhöht, so entstehen weitere Freiheitsgrade. Im Folgenden sollen aufbauend auf die Ausführungen zum Ein-Ziel-Fall verschiedene denkbare Nachbarschaftskonzepte vorgestellt werden.

5.3.1.2 Nachbarschaft im Zielraum: Konzept 1 (NSZR1)

Grundlegende Idee dieses Konzepts ist es, zunächst eine Nachbarschaft für jede einzelne der K Zielgrößen zu ermitteln und diese anschließend zu vereinigen. Nachbarn im Zielraum, bezogen auf Zielgröße k , sind zunächst alle Auftragsfolgen, die bezüglich Ziel k den gleichen Zielfunktionswert wie die Ausgangsfolge π° aufweisen. Existieren keine solchen Auftragsfolgen, so wird die Nachbarschaft durch diejenigen Auftragsfolgen gebildet, die bezüglich des Ziels k die kleinsten positiven und negativen Abweichungen aufweisen. Der oben betrachtete Fall nur einer Zielgröße ist somit ein Spezialfall ($K=1$) dieses Konzepts.

$NS_k^-(z(\pi^\circ))$: Menge aller Auftragsfolgen, deren Zielfunktionswerte die kleinste nichtpositive Abweichung vom Zielfunktionsvektor $z(\pi^\circ)$ der Ausgangsfolge π° bezüglich Ziel k aufweisen.

$$NS_k^-(z(\pi^\circ)) : = \{ \pi \mid z_k(\pi) = \max \{ z_k(\pi') \mid z_k(\pi') \leq z_k(\pi^\circ) \}; \\ \pi^\circ, \pi' \in \Pi; \pi' \neq \pi^\circ \} \quad (5.10)$$

$NS_k^+(z(\pi^\circ))$: Menge aller Auftragsfolgen, deren Zielfunktionswerte die

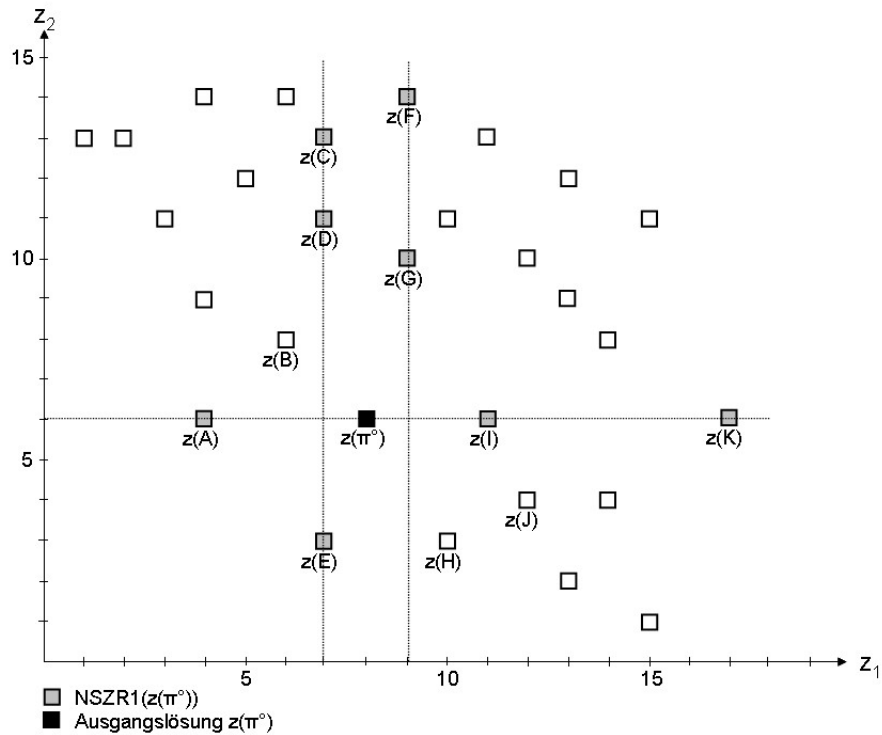


Abbildung 5.2: Nachbarschaft im Zielraum - Konzept 1

kleinste nichtnegative Abweichung vom Zielfunktionsvektor $z(\pi^\circ)$ der Ausgangsfolge π° bezüglich Ziel k aufweisen.

$$NS_k^+(z(\pi^\circ)) := \{\pi \mid z_k(\pi) = \min \{z_k(\pi') \mid z_k(\pi') \geq z_k(\pi^\circ)\}; \quad (5.11)$$

$$\pi^\circ, \pi' \in \Pi; \pi' \neq \pi^\circ\}$$

Die Nachbarschaft bezüglich Ziel k ergibt sich anschließend aus der Vereinigung von $NS_k^+(z(\pi^\circ))$ und $NS_k^-(z(\pi^\circ))$:

$$NS_k(z(\pi^\circ)) := NS_k^+(z(\pi^\circ)) \cup NS_k^-(z(\pi^\circ)) \quad (5.12)$$

Die vollständige Nachbarschaft im Zielraum ergibt sich dann wie folgt:

$$NSZR1(z(\pi^\circ)) := \bigcup_{k=1}^K NS_k(z(\pi^\circ)) \quad (5.13)$$

Abbildung 5.2 verdeutlicht das Konzept grafisch für den bikriteriellen Fall. Die Menge $NS_1^-(z(\pi^\circ))$ wird von den Auftragsfolgen C, D und E gebildet. Die Menge $NS_1^+(z(\pi^\circ))$ besteht aus den Auftragsfolgen F und G. Bei Ziel 2

gilt: $NS_2^-(z(\pi^\circ)) = NS_2^+(z(\pi^\circ)) = \{A, I, K\}$.

Das vorgestellte Konzept resultiert in einer sehr großen Nachbarschaft. Insbesondere ist nicht überzeugend, daß die Auftragsfolgen C, F und K als Nachbarn im Zielraum von π° aufgefaßt werden. Des weiteren ist diese Nachbarschaft nicht grundsätzlich symmetrisch. In Abbildung 5.2 ist z.B. F Nachbar von π° , nicht aber umgekehrt.

5.3.1.3 Nachbarschaft im Zielraum: Konzept 2 (NSZR2)

Zur Bestimmung der NSZR2 wird zunächst für jede Auftragsfolge ein Vektor der absoluten Abstände der Zielfunktionswerte zu den Zielfunktionswerten der Ausgangsfolge π° bestimmt. Die Nachbarschaft wird anschließend von den Auftragsfolgen gebildet, deren Vektor der absoluten Abstände nicht dominiert wird.

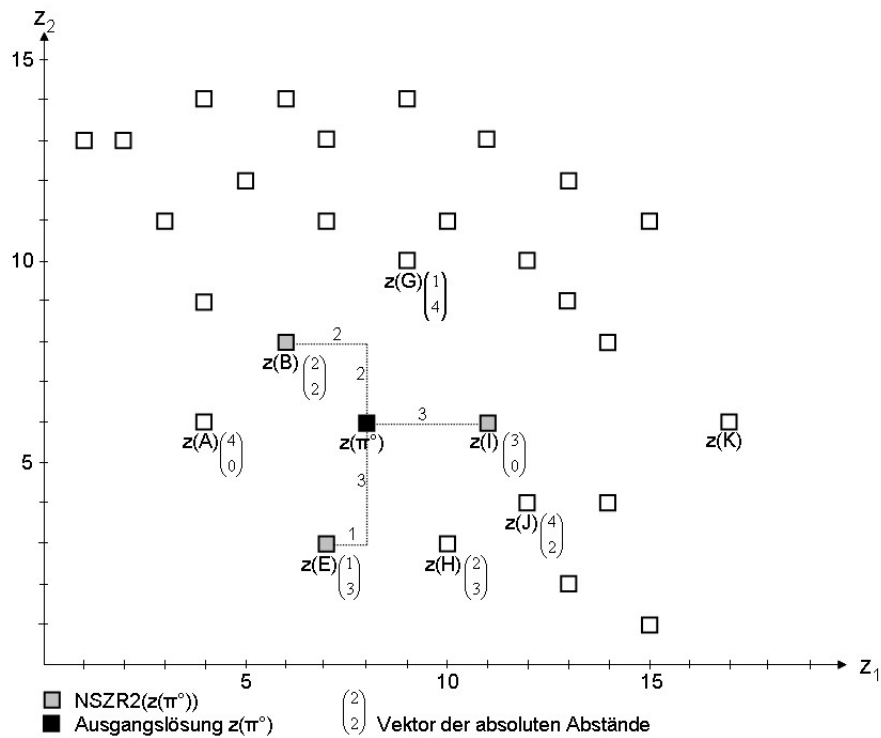


Abbildung 5.3: Nachbarschaft im Zielraum - Konzept 2

Der Vektor der absoluten Abstände ermittelt sich als:

$$z^a(z(\pi^\circ), z(\pi')) = |z(\pi^\circ) - z(\pi')| \quad (5.14)$$

Die Nachbarschaft im Zielraum von π° ergibt sich anschließend zu:

$$NSZR2(z(\pi^\circ)) : = \{ \pi \mid \nexists \pi' \in \Pi; \pi' \neq \pi \text{ mit } z^a(z(\pi^\circ), z(\pi')) \leq z^a(z(\pi^\circ), z(\pi)) \} \quad (5.15)$$

Abbildung 5.3 zeigt ein grafisches Beispiel. Die Vektoren an den Punkten geben den jeweiligen Vektor der absoluten Abstände an. Es ist sofort offensichtlich, daß die Vektoren der absoluten Abstände der Auftragsfolgen A, G, H, J und K sowie die der nicht beschrifteten Punkte dominiert werden und daher nicht in die Nachbarschaft aufgenommen werden.

Konzept 2 stellt ebenfalls auf die räumliche Nachbarschaft ab und ist wie bereits Konzept 1 nicht grundsätzlich symmetrisch. Das folgende Beispiel soll dies verdeutlichen:

Der Lösungsraum bestehe aus den 3 Elementen π^1 , π^2 und π^3 mit den dazugehörigen Zielfunktionsvektoren:

$$z(\pi^1) = \begin{pmatrix} 8 \\ 8 \end{pmatrix}; z(\pi^2) = \begin{pmatrix} 10 \\ 10 \end{pmatrix}; z(\pi^3) = \begin{pmatrix} 11 \\ 10 \end{pmatrix}$$

Als Nachbarschaften im Zielraum ergeben sich somit:

$$\begin{aligned} NSZR2(z(\pi^1)) &= \{\pi^2\} \rightarrow \left\{ \begin{pmatrix} 10 \\ 10 \end{pmatrix} \right\} = \{z(\pi^2)\} \\ NSZR2(z(\pi^2)) &= \{\pi^3\} \rightarrow \left\{ \begin{pmatrix} 11 \\ 10 \end{pmatrix} \right\} = \{z(\pi^3)\} \\ NSZR2(z(\pi^3)) &= \{\pi^2\} \rightarrow \left\{ \begin{pmatrix} 10 \\ 10 \end{pmatrix} \right\} = \{z(\pi^2)\} \end{aligned}$$

Die Auftragsfolge π^2 ist also Nachbar von π^1 , nicht jedoch umgekehrt. Die Nachbarschaft ist somit nicht symmetrisch. Sie verbindet auch nicht grundsätzlich alle Auftragsfolgen, da π^1 offensichtlich nie Nachbar von π^2 oder π^3 ist.

5.3.1.4 Nachbarschaft im Zielraum: Konzept 3 (NSZR3)

In diesem Konzept werden Konzept 1 und Konzept 2 sukzessive angewendet. Zunächst werden die Mengen $NS_k^+(z(\pi^\circ))$ und $NS_k^-(z(\pi^\circ))$ analog zu Konzept 1 ermittelt. Anschließend werden für die Elemente dieser Mengen die in Konzept 2 verwendeten Vektoren der absoluten Abstände bezogen auf π° ermittelt. Nur Elemente der Mengen $NS_k^+(z(\pi^\circ))$ bzw. $NS_k^-(z(\pi^\circ))$, die bezüglich dieses Abstandsvektors gegenüber den anderen Elementen ihrer Menge effizient sind, werden in die anschließend ermittelten Mengen $NS_k^{+E}(z(\pi^\circ))$ bzw. $NS_k^{-E}(z(\pi^\circ))$ aufgenommen. Diese Mengen werden ent-

sprechend dem Vorgehen in Konzept 1 zur Nachbarschaft $NSZR3(z(\pi^\circ))$ zusammengefügt.

$NS_k^{+E}(z(\pi^\circ))$: Menge aller bezogen auf den Vektor der absoluten Abstände (z^a) effizienten Elemente aus $NS_k^+(z(\pi^\circ))$:

$$NS_k^{+E}(z(\pi^\circ)) := \{\pi \mid \nexists \pi' \neq \pi; \pi', \pi \in NS_k^+(z(\pi^\circ)) \text{ mit } z^a(z(\pi^\circ), z(\pi')) \leq z^a(z(\pi^\circ), z(\pi))\} \quad (5.16)$$

$NS_k^{-E}(z(\pi^\circ))$: Menge aller bezogen auf den Vektor der absoluten Abstände (z^a) effizienten Elemente aus $NS_k^-(z(\pi^\circ))$:

$$NS_k^{-E}(z(\pi^\circ)) := \{\pi \mid \nexists \pi' \neq \pi; \pi', \pi \in NS_k^-(z(\pi^\circ)) \text{ mit } z^a(z(\pi^\circ), z(\pi')) \leq z^a(z(\pi^\circ), z(\pi))\} \quad (5.17)$$

Die Nachbarschaft bezüglich Ziel k ergibt sich anschließend aus der Vereinigung von $NS_k^{+E}(z(\pi^\circ))$ und $NS_k^{-E}(z(\pi^\circ))$:

$$NS_k^E(z(\pi^\circ)) := NS_k^{+E}(z(\pi^\circ)) \cup NS_k^{-E}(z(\pi^\circ)) \quad (5.18)$$

Die gesamte Nachbarschaft ergibt sich anschließend wie folgt:

$$NSZR3(z(\pi^\circ)) := \bigcup_{k=1}^K NS_k^E(z(\pi^\circ)) \quad (5.19)$$

Abbildung 5.4 verdeutlicht das Konzept grafisch. Die Mengen $NS_k^+(z(\pi^\circ))$ und $NS_k^-(z(\pi^\circ))$ ergeben sich analog zu Konzept 1. Also: $NS_1^-(z(\pi^\circ)) = \{C, D, E\}$, $NS_1^+(z(\pi^\circ)) = \{F, G\}$ sowie $NS_2^-(z(\pi^\circ)) = NS_2^+(z(\pi^\circ)) = \{A, I, K\}$. Für die Elemente dieser Mengen werden die Vektoren der absoluten Abstände ermittelt. Die bezüglich dieses Vektors effizienten Elemente werden in die Mengen $NS_k^{-E}(z(\pi^\circ))$ und $NS_k^{+E}(z(\pi^\circ))$ aufgenommen. Also: $NS_1^{-E}(z(\pi^\circ)) = \{E\}$, $NS_1^{+E}(z(\pi^\circ)) = \{G\}$, $NS_2^{+E}(z(\pi^\circ)) = NS_2^{-E}(z(\pi^\circ)) = \{I\}$. Die Nachbarschaft ergibt sich dann zu $NSZR3(z(\pi^\circ)) = \{E, G, I\}$.

Da diese Nachbarschaft auf dem Nachbarschaftskonzept 1 aufbaut, besteht auch hier keine grundsätzliche Symmetrie. Auch hier ist z.B. G Nachbar von π° , nicht aber umgekehrt.

5.3.1.5 Nachbarschaft im Zielraum: Konzept 4 (NSZR4)

Zunächst wird der Zielraum in $R = 2^K$ Regionen eingeteilt. Für jede Zielgröße k ergeben sich ausgehend vom Zielfunktionswert der Ausgangslö-

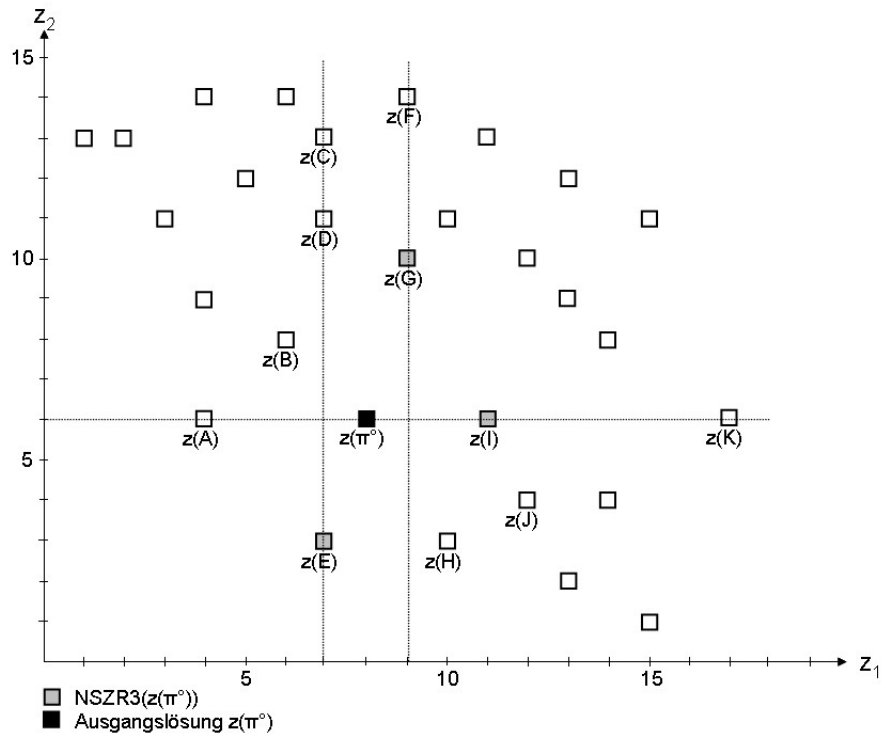


Abbildung 5.4: Nachbarschaft im Zielraum - Konzept 3

sung $z_k(\pi^\circ)$ zwei Bereiche. Im ersten Bereich liegen Auftragsfolgen, die bezüglich Zielgröße k einen kleineren oder gleichen Zielfunktionswert besitzen, also gilt: $z_k(\pi) \leq z_k(\pi^\circ)$. Im zweiten Bereich liegen Auftragsfolgen, für die der Zielfunktionswert von Zielgröße k größer oder gleich dem der Ausgangsfolge ist, also gilt: $z_k(\pi) \geq z_k(\pi^\circ)$. Die Anzahl der Regionen R ergibt sich durch die Anzahl an möglichen Kombinationen der größer-gleich und kleiner-gleich Relationen für die K Ziele.

Anschließend wird für jede Region r , $r = 1(1)R$, festgestellt, welche Auftragsfolgen aufgrund ihres Zielfunktionsvektors Mitglied der Region r sind. Aus den Mitgliedern einer Region r wird die Menge $NS^r(z(\pi^\circ))$ bestimmt, deren Elemente Nachbarn der Ausgangsfolge π° in der Region r sind. Dies sind alle Auftragsfolgen der Region, deren Vektor der absoluten Abstände $z^a(z(\pi^\circ), z(\pi))$ (vgl. auch Gleichung 5.14) effizient bezüglich aller Mitglieder der Region r sind.

Die Menge der Auftragsfolgen in Region r , welche Nachbarn im Zielraum sind ($NS^r(z(\pi^\circ))$), wird wie folgt ermittelt:

$$NS^r(z(\pi^\circ)) := \{\pi \mid \exists \pi' \neq \pi; z(\pi), z(\pi') \in \text{Region } r \quad (5.20)$$

$$\text{mit } z^a(z(\pi^\circ), z(\pi')) \leq z^a(z(\pi^\circ), z(\pi))\}$$

Die Vereinigung der Mengen $NS^r(z(\pi^\circ))$, $r = 1(1)R$, ergibt die Nachbarschaft im Zielraum.

$$NSZR4(z(\pi^\circ)) := \bigcup_{r=1}^R NS^r(z(\pi^\circ)) \quad (5.21)$$

Ein grafisches Beispiel für das Vorgehen ist in Abbildung 5.5 dargestellt. Dort soll die Nachbarschaft für Ausgangslösung π° berechnet werden.

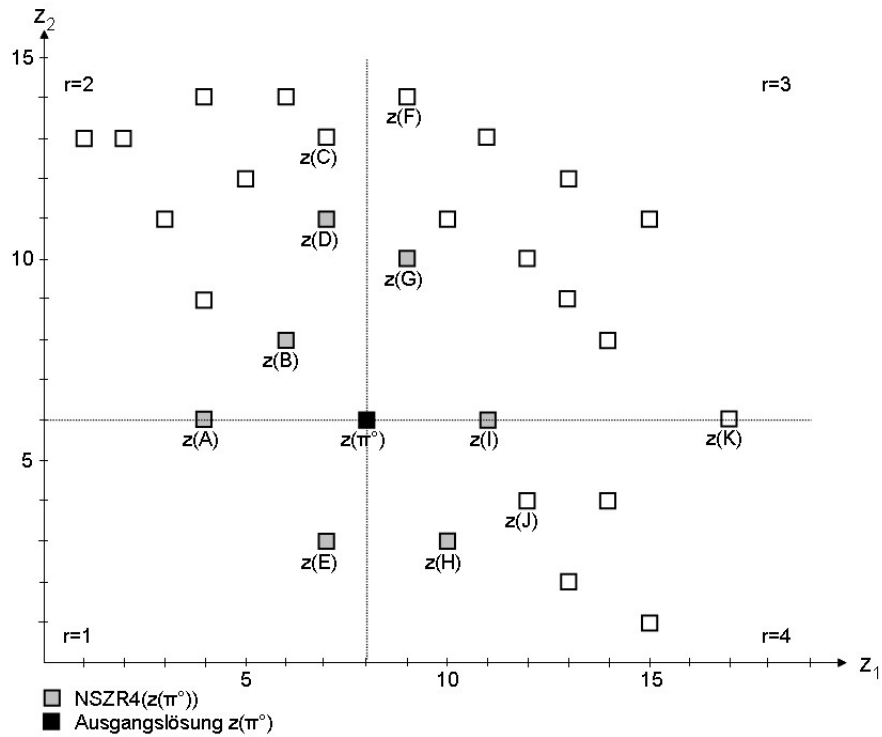


Abbildung 5.5: Nachbarschaft im Zielraum - Konzept 4

Die $R = 4$ Regionen ergeben sich wie in Tabelle 5.1 angegebene.

Region r	Ziel k	
	1	2
1	\leq	\leq
2	\leq	\geq
3	\geq	\geq
4	\geq	\leq

Tabelle 5.1: NSZR4: Definition der Regionen bei K=2

In Region $r = 1$ fallen alle Auftragsfolgen, deren Zielfunktionsvektoren bezüglich Ziel 1 und Ziel 2 einen nichtgrößeren Zielfunktionswert als π° aufweisen. Dies sind im Beispiel die Auftragsfolgen A und E. Beide Auftragsfolgen gehören auch der Menge $NS^1(z(\pi^\circ))$ an.

In Region $r = 2$ fallen alle Auftragsfolgen, deren Zielfunktionsvektoren bezüglich Ziel 1 einen nichtgrößeren und Ziel 2 einen nichtkleineren Zielfunktionswert als π° aufweisen. Es ergibt sich: $NS^2(z(\pi^\circ)) = \{A, B, D\}$. C ist hingegen kein Element von $NS^2(z(\pi^\circ))$, da sein Vektor der absoluten Abstände dominiert wird, es gilt: $z^a(z(\pi^\circ), z(D)) \leq z^a(z(\pi^\circ), z(C))$.

In Region $r = 3$ fallen Auftragsfolgen, die für beide Ziele einen nichtkleineren Zielfunktionswert als die Ausgangsfolge π° aufweist. Es ergibt sich: $NS^3(z(\pi^\circ)) = \{G, I\}$.

In Region $r = 4$ schließlich fallen Auftragsfolgen, die bezüglich Ziel 1 einen nichtkleineren und Ziel 2 einen nichtgrößeren Zielfunktionswert als π° aufweisen. Somit ergibt sich: $NS^4(z(\pi^\circ)) = \{I, H\}$.

Die gesamte Nachbarschaft ergibt sich durch Vereinigung der Mengen $NS^1(z(\pi^\circ))$, $NS^2(z(\pi^\circ))$, $NS^3(z(\pi^\circ))$ und $NS^4(z(\pi^\circ))$ zu: $NSZR4(z(\pi^\circ)) = \{A, B, D, E, G, I, H\}$.

Im Fall $K = 3$ sind $2^3 = 8$ Regionen zu unterscheiden, die in folgender Tabelle 5.2 aufgeführt sind:

Region r	Ziel k		
	1	2	3
1	\leq	\leq	\leq
2	\leq	\leq	\geq
3	\leq	\geq	\geq
4	\leq	\geq	\leq
5	\geq	\leq	\leq
6	\geq	\leq	\geq
7	\geq	\geq	\geq
8	\geq	\geq	\leq

Tabelle 5.2: NSZR4: Definition der Regionen bei K=3

Entsprechend der Beispiele für $K = 2$ und $K = 3$ lassen sich Regionen für eine beliebige Anzahl an Zielen bilden. Für die Zwecke dieser Arbeit reicht das Bilden der Regionen für drei Ziele aus, weshalb auf eine weiterführende allgemeine Darstellung verzichtet wird.

Im Folgenden soll gezeigt werden, daß die Nachbarschaft NSZR4 symmetrisch ist, d.h., wenn gilt: $\pi' \in NSZR4(z(\pi^\circ))$, dann gilt auch $\pi^\circ \in NSZR4(z(\pi'))$.

Zu jeder Region r sei die Region r' definiert, indem die \leq und \geq Operatoren bezüglich der einzelnen Zielgrößen im Vergleich zu r umgekehrt werden. Tabelle 5.3 zeigt die Definition der zu Region r gehörenden Region r' im Fall $K = 3$.

	Region r				Region r'		
	Ziel k				Ziel k		
	1	2	3		1	2	3
1	\leq	\leq	\leq	\Leftrightarrow	\geq	\geq	\geq
2	\leq	\leq	\geq	\Leftrightarrow	\geq	\geq	\leq
3	\leq	\geq	\geq	\Leftrightarrow	\geq	\leq	\leq
4	\leq	\geq	\leq	\Leftrightarrow	\geq	\leq	\geq
5	\geq	\leq	\leq	\Leftrightarrow	\leq	\geq	\geq
6	\geq	\leq	\geq	\Leftrightarrow	\leq	\geq	\leq
7	\geq	\geq	\geq	\Leftrightarrow	\leq	\leq	\leq
8	\geq	\geq	\leq	\Leftrightarrow	\leq	\leq	\geq

Tabelle 5.3: Definition Region r'

NSZR4 ist symmetrisch, wenn $\pi' \in NS^r(z(\pi^\circ))$ gilt, dann auch $\pi^\circ \in NS^{r'}(z(\pi'))$ für ein r' ist. Da die Elemente aus NS^r und $NS^{r'}$ aufgrund Gleichung 5.21 Elemente von NSZR4 sind, genügt es, daß diese Forderung erfüllt ist, damit die Nachbarschaft symmetrisch ist.

Damit $\pi^\circ \notin NS^{r'}(z(\pi'))$ gilt, obwohl $\pi' \in NS^r(z(\pi^\circ))$ ist, müßte es wegen Gleichung 5.20 ein Element π'' geben, so daß gilt: $z^a(\pi', \pi'') \leq z^a(\pi', \pi^\circ)$. Wie man sich leicht überlegen kann, ist dies nur möglich, falls für die Ziel-funktionswerte der Lösungen gilt (vgl. auch Abbildung 5.6):

$$z_k(\pi') \leq z_k(\pi'') \leq z_k(\pi^\circ), \text{ falls in } r \text{ für Ziel } k \text{ der } \leq\text{-Operator gilt} \quad (5.22)$$

Für mindestens ein k , für das in r der \leq - Operator gilt, muß in Gleichung 5.22 mindestens einmal die $<$ - Relation erfüllt sein.

$$z_k(\pi') \geq z_k(\pi'') \geq z_k(\pi^\circ), \text{ falls in } r \text{ f\u00fcr Ziel } k \text{ der } \geq\text{-Operator gilt} \quad (5.23)$$

F\u00fcr mindestens ein k , f\u00fcr das in r der \geq - Operator gilt, mu\u00df in Gleichung 5.23 mindestens einmal die $>$ - Relation erf\u00fcllt sein.

Dies w\u00fcrde aber wiederum bedeuten, da\u00df auch gilt: $z^a(\pi^\circ, \pi'') \leq z^a(\pi^\circ, \pi')$, was wiederum zur Folge hat, da\u00df auch gelten w\u00fcrde: $\pi' \notin NS^r(z(\pi^\circ))$. Es w\u00fcrde vielmehr gelten: $\pi'' \in NS^r(z(\pi^\circ))$, aber auch $\pi^\circ \in NS^{r'}(z(\pi''))$. NSZR4 ist folglich symmetrisch.

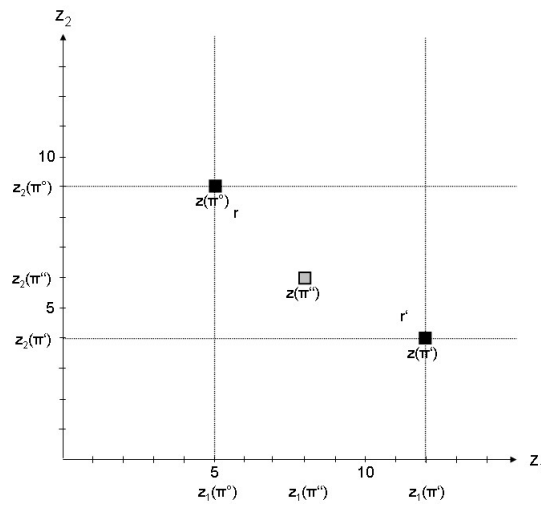


Abbildung 5.6: Symmetrie von NSZR4

Die Nachbarschaft verbindet jedoch nicht grunds\u00e4tzlich alle Elemente des Zielraums miteinander. Falls zwei Auftragsfolgen π° und π' den gleichen Zielfunktionsvektor besitzen, sind diese jeweils die einzigen gegenseitigen Nachbarn, es gilt also: $NSZR4(z(\pi^\circ)) = \{\pi'\}$ und $NSZR4(z(\pi')) = \{\pi^\circ\}$. Mit diesem Beispiel kann auch f\u00fcr NSZR1, NSZR2 und NSZR3 gezeigt werden, da\u00df diese nicht grunds\u00e4tzlich alle Elemente des Zielraums verbinden.

Alle vier vorgestellten Nachbarschaften im Zielraum verbinden nicht grunds\u00e4tzlich alle Elemente des Zielraums miteinander. Mit Ausnahme von NSZR4 sind sie auch nicht symmetrisch. Es bleibt jedoch zu diskutieren, ob das Erf\u00fcllen dieser Eigenschaften zwingend vorliegen mu\u00df.

Die Suche nach Elementen der Lösungsmenge wird im Lösungsraum, hier also der Menge aller (zulässigen) Auftragsfolgen, durchgeführt. Aus diesem Grund ist es bei den Nachbarschaften im Lösungsraum eine wesentliche Forderung, daß sie alle Elemente des Lösungsraums miteinander verbinden. Diesen Suchprozeß auf den Zielraum zu verlagern, erscheint nicht möglich. Im Gegensatz zum Lösungsraum sind die Elemente des Zielraums nicht ex ante bekannt, und damit ist es nicht möglich, eine der hier vorgestellten Nachbarschaften im Zielraum einfach zu ermitteln. Da die Suche nach Elementen der Lösungsmenge somit nicht durch Züge von einem zum anderen Element des Zielraum gesteuert werden kann, erscheint es auch nicht zwingend erforderlich, daß die Nachbarschaften im Zielraum alle Elemente des Zielraums miteinander verbinden. Die Symmetrie der Nachbarschaft hingegen erscheint wesentlich, wobei jedoch anzumerken ist, daß es mit FSH und BSH Nachbarschaften im Lösungsraum gibt, die ebenfalls nicht symmetrisch sind. Da die NSZR4 das einzige der hier vorgestellten Nachbarschaftskonzepte ist, welches symmetrisch ist, werden die in Abschnitt 5.3.2 folgenden Analysen alleine mit diesem Nachbarschaftskonzept im Zielraum durchgeführt.

5.3.2 Experimentelle Analyse

5.3.2.1 Experiment 1: Nachbarschaft im Lösungsraum gleich Nachbarschaft im Zielraum?

In der folgenden experimentellen Analyse wird überprüft, ob und wie häufig ein Nachbar im Lösungsraum auch ein Nachbar im Zielraum ist und umgekehrt. Als Nachbarschaften im Lösungsraum werden die in Abschnitt 5.2 beschriebenen Nachbarschaften verwendet. Als Nachbarschaften im Zielraum wird alleine NSZR4 in die Analyse einbezogen. Da die Nachbarschaften im Zielraum, anders als die Nachbarschaften im Lösungsraum, abhängig von einem konkreten Problem sind, wird die Analyse für 90 Testprobleme mit unterschiedlichen Problemgrößen durchgeführt. Es werden Probleminstanzen mit $N \in \{4, 6, 8\}$ Aufträgen und $M \in \{5, 10, 20\}$ Maschinen betrachtet. Durch die Kombination von Auftrags- und Maschinenzahl ergeben sich 9 Testfelder. Für jedes Testfeld werden 10 Probleminstanzen erzeugt.²⁹ Zur Generierung der Nachbarschaft im Zielraum ist es notwendig, die Zielfunktionsvektoren aller Auftragsfolgen zu kennen. Daher werden lediglich Probleme mit 4, 6 und 8 Aufträgen betrachtet. Für diese Größen ist es möglich, in angemessener Rechenzeit zunächst eine vollständige Enumeration

²⁹Die Generierung dieser Testprobleme wird in den Anhängen A und B beschrieben.

durchzuführen sowie die Nachbarschaften im Zielraum zu berechnen. Der betrachtete Zielvektor besteht aus den in dieser Arbeit betrachteten drei Zielen Minimierung der Zykluszeit, Minimierung der mittleren Durchlaufzeit und Minimierung der maximalen Terminüberschreitung. Jedes der 90 Probleme wurde zunächst vollständig enumeriert. Anschließend wurden für alle $N!$ Auftragsfolgen eines Problems die Nachbarschaften im Lösungs- und Zielraum ermittelt und miteinander verglichen. Zur Auswertung wurden folgende Größen bestimmt:

1. Größe der Nachbarschaft im Lösungsraum (NSLR): $|NSLR|$.
2. Größe der Nachbarschaft im Zielraum (NSZR): $|NSZR|$.
3. Anzahl der Auftragsfolgen, die in beiden Nachbarschaften enthalten sind: $Match = |NSLR \cap NSZR|$.
4. Anteil der Auftragsfolgen der Nachbarschaft im Lösungsraum, die auch Element von NSZR sind:

$$\%LR = \frac{|NSLR \cap NSZR|}{|NSLR|} \quad (5.24)$$

5. Anteil der Auftragsfolgen der Nachbarschaft im Zielraum, die auch Element von NSLR sind:

$$\%ZR = \frac{|NSLR \cap NSZR|}{|NSZR|} \quad (5.25)$$

Die fünf Werte wurden für jede einzelne der $N!$ Auftragsfolge eines Problems ermittelt und anschließend für das jeweilige Problem der Durchschnittswert über diese $N!$ Werte berechnet.

Die Tabelle 5.4 auf der nächsten Seite und Tabelle 5.5 auf Seite 116 geben die durchschnittlichen Auswertungsgrößen für die einzelnen Nachbarschaftskombinationen an³⁰. Die fett gedruckten Zeilen geben jeweils den Durchschnittswert über die Probleme mit gleicher Auftragszahl bzw. über alle betrachteten Probleminstanzen für die jeweilige Nachbarschaft im Lösungsraum an. In den restlichen Zeilen darüber sind jeweils die Mittelwerte über die betrachteten Probleminstanzen mit gleicher Auftrags- und Maschinenzahl für ein Nachbarschaft im Lösungsraum wiedergegeben.

Zunächst zeigt sich, daß die Größe der NSZR₄ mit steigender Anzahl Aufträge und damit mit zunehmender Größe des Lösungsraums zunimmt.

³⁰Aus der Analyse ausgenommen war Problem 75. Dieses Problem hat sehr viele globale Optima für die Zielgröße T_{max} und damit eine Struktur, die es nicht ermöglicht, die Nachbarschaften im Zielraum in angemessener Rechenzeit und mit dem verfügbaren Arbeitsspeicher zu berechnen.

<i>NSLR</i>	<i>N</i>	<i>M</i>	<i>NSLR</i>	<i>NSZR</i>	<i>Match</i>	<i>%LR</i>	<i>%ZR</i>
		5	3,0000	8,0167	1,9333	0,6444	0,2914
		10	3,0000	7,6833	1,7167	0,5722	0,2652
		20	3,0000	8,1333	1,9833	0,6611	0,2679
	4	Mittelwert	3,0000	7,9444	1,8778	0,6259	0,2749
		5	5,0000	25,9253	1,6349	0,3270	0,0982
		10	5,0000	23,4163	1,2133	0,2427	0,0799
		20	5,0000	41,7181	1,5708	0,3142	0,0583
	6	Mittelwert	5,0000	30,3532	1,4730	0,2946	0,0788
		5	7,0000	35,7945	0,6612	0,0945	0,0521
		10	7,0000	32,9854	0,5004	0,0715	0,0303
		20	7,0000	65,0385	0,6698	0,0957	0,0172
	8	Mittelwert	7,0000	45,0068	0,6143	0,0878	0,0333
API			4,9775	27,5745	1,3296	0,3389	0,1301
		5	6,0000	8,0167	2,9000	0,4833	0,4064
		10	6,0000	7,6833	2,7500	0,4583	0,3922
		20	6,0000	8,1333	3,0417	0,5069	0,3943
	4	Mittelwert	6,0000	7,9444	2,8972	0,4829	0,3976
		5	15,0000	25,9253	2,6096	0,1740	0,1401
		10	15,0000	23,4163	1,9575	0,1305	0,1147
		20	15,0000	41,7181	2,7775	0,1852	0,0918
	6	Mittelwert	15,0000	30,3532	2,4482	0,1632	0,1156
		5	28,0000	35,7945	0,9462	0,0338	0,0776
		10	28,0000	32,9854	0,7051	0,0252	0,0374
		20	28,0000	65,0385	1,0715	0,0383	0,0240
	8	Mittelwert	28,0000	45,0068	0,9146	0,0327	0,0466
EX			16,2022	27,5745	2,0998	0,2284	0,1882
		5	6,0000	8,0167	2,8833	0,4806	0,4036
		10	6,0000	7,6833	2,7000	0,4500	0,3848
		20	6,0000	8,1333	2,9667	0,4944	0,3853
	4	Mittelwert	6,0000	7,9444	2,8500	0,4750	0,3912
		5	15,0000	25,9253	2,4815	0,1654	0,1355
		10	15,0000	23,4163	1,8762	0,1251	0,1116
		20	15,0000	41,7181	2,5685	0,1712	0,0857
	6	Mittelwert	15,0000	30,3532	2,3087	0,1539	0,1109
		5	28,0000	35,7945	0,8869	0,0317	0,0708
		10	28,0000	32,9854	0,6600	0,0236	0,0364
		20	28,0000	65,0385	0,9592	0,0343	0,0224
	8	Mittelwert	28,0000	45,0068	0,8414	0,0301	0,0434
INV			16,2022	27,5745	2,0131	0,2218	0,1834

Tabelle 5.4: Gemeinsame Elemente in NSZR4 und API, EX und INV

<i>NSLR</i>	<i>N</i>	<i>M</i>	<i> NSLR </i>	<i> NSZR </i>	<i>Match</i>	<i>%LR</i>	<i>%ZR</i>
		5	6,0000	8,0167	2,9708	0,4951	0,4188
		10	6,0000	7,6833	2,7958	0,4660	0,4052
		20	6,0000	8,1333	3,1000	0,5167	0,4059
	4	Mittelwert	6,0000	7,9444	2,9556	0,4926	0,4100
		5	15,0000	25,9253	2,5915	0,1728	0,1418
		10	15,0000	23,4163	2,0438	0,1362	0,1220
		20	15,0000	41,7181	2,9435	0,1962	0,0964
	6	Mittelwert	15,0000	30,3532	2,5262	0,1684	0,1201
		5	28,0000	35,7945	0,9257	0,0331	0,0665
		10	28,0000	32,9854	0,6959	0,0249	0,0376
		20	28,0000	65,0385	1,0634	0,0380	0,0246
	8	Mittelwert	28,0000	45,0068	0,9018	0,0322	0,0431
FSH			16,2022	27,5745	2,1417	0,2333	0,1927
		5	6,0000	8,0167	2,9708	0,4951	0,4239
		10	6,0000	7,6833	2,7958	0,4660	0,4098
		20	6,0000	8,1333	3,1000	0,5167	0,4056
	4	Mittelwert	6,0000	7,9444	2,9556	0,4926	0,4131
		5	15,0000	25,9253	2,5910	0,1727	0,1417
		10	15,0000	23,4163	2,0426	0,1362	0,1226
		20	15,0000	41,7181	2,9474	0,1965	0,0964
	6	Mittelwert	15,0000	30,3532	2,5270	0,1685	0,1202
		5	28,0000	35,7945	0,9250	0,0330	0,0663
		10	28,0000	32,9854	0,6957	0,0248	0,0375
		20	28,0000	65,0385	1,0634	0,0380	0,0246
	8	Mittelwert	28,0000	45,0068	0,9016	0,0322	0,0430
BSH			16,2022	27,5745	2,1418	0,2333	0,1938
		5	12,0000	8,0167	5,9417	0,4951	0,8427
		10	12,0000	7,6833	5,5917	0,4660	0,8150
		20	12,0000	8,1333	6,2000	0,5167	0,8115
	4	Mittelwert	12,0000	7,9444	5,9111	0,4926	0,8231
		5	30,0000	25,9253	5,1825	0,1727	0,2835
		10	30,0000	23,4163	4,0864	0,1362	0,2446
		20	30,0000	41,7181	5,8908	0,1964	0,1928
	6	Mittelwert	30,0000	30,3532	5,0532	0,1684	0,2403
		5	56,0000	35,7945	1,8507	0,0330	0,1328
		10	56,0000	32,9854	1,3916	0,0248	0,0751
		20	56,0000	65,0385	2,1268	0,0380	0,0492
	8	Mittelwert	56,0000	45,0068	1,8034	0,0322	0,0861
DSH			32,4045	27,5745	4,2835	0,2333	0,3865

Tabelle 5.5: Gemeinsame Elemente NSZR4 und FSH, BSH und DSH

Bei c.p. ansteigender Maschinenzahl kann hingegen keine systematische Entwicklung der Größe der Nachbarschaft beobachtet werden.

Die Auswertung des Anteils der Auftragsfolgen der Nachbarschaft im Lösungsraum, die auch Element der Nachbarschaft im Zielraum sind ($%LR$), zeigt, daß für den API-Operator die Wahrscheinlichkeit dafür, daß ein Nachbar im Zielraum auch ein Nachbar im Lösungsraum ist, am höchsten ist. Am niedrigsten ist diese Wahrscheinlichkeit für den INV-Operator. Obwohl also der API-Operator die kleinste Nachbarschaft im Lösungsraum aufweist, hat er die höchste Wahrscheinlichkeit auch einen Nachbarn im Zielraum zu treffen.

Die Werte des Anteils der Auftragsfolgen der Nachbarschaften im Zielraum, die auch Element der Nachbarschaft im Lösungsraum sind ($%ZR$), zeigen, daß der Wert für die API-Nachbarschaft hier am niedrigsten ist. Die höchsten Werte können nun bei der DSH-Nachbarschaft beobachtet werden.

Eine Analyse der Entwicklungen der Werte von $%LR$ und $%ZR$ bei unterschiedlicher Anzahl an Aufträgen zeigt, daß sowohl $%LR$ als auch $%ZR$ bei allen Nachbarschaften im Lösungsraum mit steigender Auftragszahl sinken. Insofern ist zu erwarten, daß die Werte für Probleme mit 20 und mehr Aufträgen, wie sie in den Benchmarkproblemen von Taillard (1993) auftreten,³¹ deutlich niedriger sein werden. Eine systematisches Steigen oder Sinken von $%LR$ bei c.p. steigender Maschinenzahl kann hingegen nicht beobachtet werden.

Es läßt sich somit feststellen, daß ein Nachbar im Lösungsraum in der Regel kein Nachbar im Zielraum ist und umgekehrt. Bei Anwendung des API-Operators ist die Wahrscheinlichkeit dafür, einen Nachbarn im Zielraum zu erhalten, noch am höchsten, wenn auch auf einem niedrigen Niveau. Es wird somit nur eingeschränkt möglich sein, ein Verfahren zu entwickeln, welches unter Verwendung der hier diskutierten Nachbarschaftsoperatoren im Lösungsraum den Zielraum systematisch nach Elementen der effizienten Front absuchen kann.

5.3.2.2 Experiment 2: Abweichung NSLR von NSZR4

Die Ergebnisse der bisherigen Analyse lassen offen, wo die Zielfunktionsvektoren der Auftragsfolgen aus der Nachbarschaft im Lösungsraum, die nicht in der Nachbarschaft im Zielraum vertreten sind, im Zielraum angeordnet sind. Damit fehlt die Information darüber, wie weit diese Zielfunktionsvektoren von der Nachbarschaft im Zielraum entfernt sind bzw. von

³¹Vgl. Taillard (1993, S.280).

dieser abweichen.

Eine Möglichkeit, diese Abweichung zu erfassen, könnte die Ermittlung des Varianzkoeffizienten der Zielfunktionswerte der Elemente der Nachbarschaft im Lösungsraum sein. Dieses Vorgehen ist jedoch nicht unproblematisch. So werden die Eigenschaften der Nachbarschaft im Zielraum nicht berücksichtigt. Weiterhin fehlt ein Referenzwert, bis zu welchem eine Abweichung als "unbedenklich" anzusehen ist. Schließlich fehlt ein Bezug zum Zielfunktionswertniveau der Ausgangsfolge. Der letzte Einwand könnte damit behoben werden, daß die Varianz bezogen auf den Zielfunktionswert der Ausgangsfolge berechnet wird. Ebenso würde dieser Wert dann als Quotient bei der Bestimmung des Varianzkoeffizienten verwendet. Insbesondere beim Ziel der maximalen Terminüberschreitung entsteht dabei häufig die Situation, daß dieser Ausgangswert Null beträgt und daher eine Division durch Null durchzuführen wäre. Dieses Problem kann auch bei der Ermittlung der normalen Varianz auftreten, wenn nämlich alle Nachbarn eine maximale Terminüberschreitung von Null aufweisen. Auch diese Situation ist nicht ausgeschlossen, so daß im Folgenden eine Alternative zur Bestimmung der Abweichung vorgestellt werden soll.

In einem ersten Schritt werden aus der jeweils betrachteten Nachbarschaft einer Ausgangslösung π° im Lösungsraum ($NSLR(\pi^\circ)$) alle Elemente entfernt, die von der Ausgangsfolge (π°) dominiert werden. Diese so generierte Nachbarschaft wird als reduzierte Nachbarschaft im Lösungsraum von Ausgangslösung π° ($NSLR^r(\pi^\circ)$) bezeichnet:

$$NSLR^r(\pi^\circ) = NSLR(\pi^\circ) \setminus \{\pi \mid z(\pi^\circ) \leq z(\pi); \pi \in NSLR(\pi^\circ)\} \quad (5.26)$$

Die Reduzierung wird vorgenommen, da es nicht zweckmäßig erscheint, einen Zug zu einem von der Ausgangsfolge π° dominierten Nachbarn durchzuführen.³² Angenommen, Auftragsfolge I in Abbildung 5.7 sei Element von $NSLR(\pi^\circ)$, so erscheint es nicht sinnvoll, einen Zug zu dieser Auftragsfolge auszuführen, da man sich von der effizienten Front entfernt bzw. dieses Element mit Sicherheit kein Element der Menge aller effizienten Lösungen sein wird.

Im zweiten Schritt wird für jede Zielgröße k der kleinste in der $NSZRA(z(\pi^\circ))$ auftretende Wert z_k^{grenze} berechnet:

$$z_k^{grenze} = \min \{z_k(\pi) \mid \pi \in NSZRA(z(\pi^\circ))\} \quad (5.27)$$

³²Sollte es keine nicht dominierte Auftragsfolge geben, also $NSLR^r(\pi^\circ) = \emptyset$ gelten, so kann dies als das Vorliegen eines lokalen Optimums interpretiert werden.

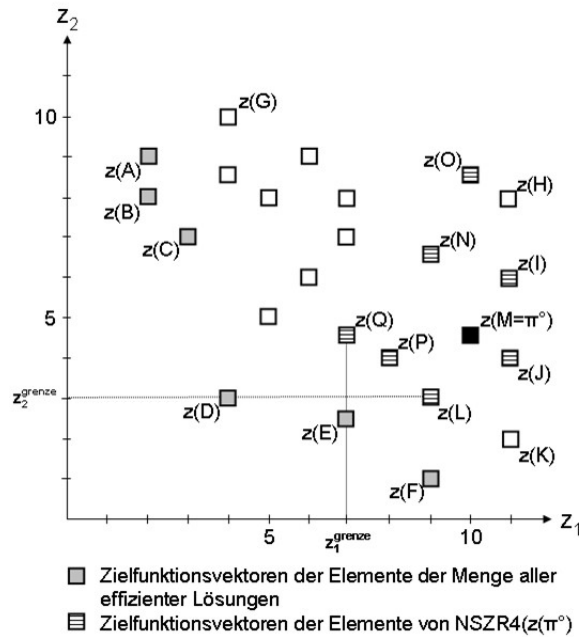


Abbildung 5.7: Abweichungsanalyse von NSZR4

Vergleiche zur Ermittlung der Werte von z_k^{grenze} auch Abbildung 5.7. Dort wird Auftragsfolge M als Ausgangsfolge betrachtet ($\pi^\circ = M$). Die schraffierten Elemente geben die Zielfunktionsvektoren der $NSZR4(z(\pi^\circ = M)) = \{I, J, L, N, O, P, Q\}$ wieder. Es ergibt sich offensichtlich: $z_1^{grenze} = z_1(Q) = 7$ und $z_2^{grenze} = z_2(L) = 3$.

Im dritten Schritt wird für jedes Element π' der reduzierten Nachbarschaft im Lösungsraum ($NSLR^r$) die Abweichung zur Nachbarschaft im Zielraum von Ausgangslösung π° bzgl. Ziel k berechnet:

$$DEV_k(\pi', \pi^\circ) = \begin{cases} 1 - \frac{z_k(\pi')}{z_k^{grenze}} & \text{falls } z_k(\pi') < z_k^{grenze} \\ 0 & \text{sonst} \end{cases} \quad (5.28)$$

Hat ein Element π' der Nachbarschaft im Lösungsraum für Ziel k einen kleineren Wert als z_k^{grenze} , so wird diesem eine positive Abweichung zugewiesen, und es hat daher einen positiven Wert für $DEV_k(\pi', \pi^\circ)$. Würde ein Zug zu einem Element mit positiver Abweichung ausgeführt, so würde die daraus resultierende Bewegung im Zielraum bedeuten, daß der Bereich der Nachbarschaft im Zielraum auf jeden Fall verlassen wird. Die Abweichung bzgl. Ziel k beträgt Null, falls $z_k(\pi') \geq z_k^{grenze}$ gilt. Ein Zug zu diesen Ele-

menten würde nicht grundsätzlich ein Abweichen von $NSZR4(z(\pi^\circ))$ bedeuten. Für den Zielfunktionsvektor $z(P)$ in Abbildung 5.7 beträgt die Abweichung bezüglich Ziel 1 und 2 Null. Angenommen, Auftragsfolge P wäre Element von $NSLR(\pi^\circ)$, so dürfte sie auch keine positive Abweichung von der $NSZR4(z(\pi^\circ))$ aufweisen, da sie Element der $NSZR4(z(\pi^\circ))$ ist.

Für eine Nachbarschaft im Lösungsraum und eine Ausgangsfolge π° können auf $DEV_k(\pi', \pi^\circ)$ aufbauend drei Größen für jedes der K Ziele ermittelt werden:

1. Durchschnittliche Abweichung von $NSZR4$ bzgl. Ziel k bezogen auf die gesamte reduzierte Nachbarschaft im Lösungsraum. Hier wird (implizit) der Teil der Auftragsfolgen an der Nachbarschaft berücksichtigt, welche den Grenzwert $z_k^{grenze}(\pi)$ unterschreiten:

$$AVG_k^1 = \frac{\sum_{\pi' \in NSLR^r} DEV_k(\pi', \pi^\circ)}{|NSLR^r|} \quad (5.29)$$

2. Durchschnittliche Abweichung von $NSZR4$ bzgl. Ziel k bezogen auf die Anzahl der Auftragsfolgen, die eine positive Abweichung aufweisen:

$$AVG_k^2 = \frac{\sum_{\pi' \in NSLR^r} DEV_k(\pi', \pi^\circ)}{y_k} \quad (5.30)$$

mit: y_k Anzahl an Auftragsfolgen in $NSLR^r$ mit $z_k(\pi') < z_k^{grenze}$.

3. Größte Abweichung von $NSZR4$ bzgl. Ziel k in der reduzierten Nachbarschaft im Lösungsraum:

$$WC_k = \max \{DEV_k(\pi', \pi^\circ) \mid \pi' \in NSLR^r\} \quad (5.31)$$

Offensichtlich bedeutet bei allen drei Größen ein kleinerer Wert eine geringere Abweichung.

Tabelle 5.6 gibt zunächst eine Übersicht über die Anzahl der Auftragsfolgen, die sowohl in der um die dominierten Auftragsfolgen reduzierten Nachbarschaft im Lösungsraum ($NSLR^r$) als auch in $NSZR4$ enthalten sind.³³

Die Ergebnisse zeigen, daß für den API-Operator die Wahrscheinlichkeit, einen Nachbarn im Zielraum zu enthalten ($\%LR$), weiterhin am größten ist.

³³Aus Gründen der Übersichtlichkeit wird hier nur noch der Mittelwert für Probleme mit gleicher Auftragszahl angegeben.

$NSLR^r$	N	$ NSLR^r $	$ NSZR4 $	$Match$	$\%LR$	$\%ZR$
API	4	1,4875	7,9444	0,9000	0,5394	0,1162
	6	1,7803	30,3532	0,4089	0,1806	0,0144
	8	1,9280	45,0068	0,0885	0,0330	0,0015
		1,7297	27,5745	0,4700	0,2534	0,0445
EX	4	2,0194	7,9444	0,8694	0,3709	0,1064
	6	2,6474	30,3532	0,2509	0,0673	0,0087
	8	3,0412	45,0068	0,0300	0,0060	0,0006
		2,5640	27,5745	0,3874	0,1497	0,0390
FSH	4	2,1486	7,9444	1,0056	0,4253	0,1219
	6	2,7700	30,3532	0,3552	0,0973	0,0121
	8	3,2756	45,0068	0,0468	0,0095	0,0008
		2,7253	27,5745	0,4740	0,1792	0,0454
BSH	4	1,7444	7,9444	0,7264	0,3472	0,0901
	6	2,4675	30,3532	0,2435	0,0711	0,0085
	8	2,7078	45,0068	0,0279	0,0064	0,0005
		2,3021	27,5745	0,3360	0,1431	0,0334
DSH	4	2,2931	7,9444	0,8583	0,3233	0,1032
	6	3,0618	30,3532	0,2534	0,0581	0,0089
	8	3,5030	45,0068	0,0248	0,0043	0,0005
		2,9464	27,5745	0,3828	0,1300	0,0380
INV	4	2,0361	7,9444	0,8458	0,3668	0,1025
	6	2,7549	30,3532	0,2756	0,0760	0,0095
	8	3,2086	45,0068	0,0395	0,0082	0,0007
		2,6604	27,5745	0,3909	0,1519	0,0380

Tabelle 5.6: Gemeinsame Elemente in der reduzierten NSLR und NSZR4.

Die eigentliche Analyse der Abweichungen wurde für die bereits im ersten Experiment verwendeten Testprobleme durchgeführt. Für jedes Problem wurden die Abweichungen für alle $N!$ Auftragsfolgen und ihre Nachbarschaften bestimmt und anschließend der Durchschnitt für jede Probleminstanz ermittelt. Tabelle 5.7 gibt die durchschnittlichen Abweichungen für die jeweiligen Nachbarschaften im Lösungsraum wieder. Tabelle 5.8 gibt die entsprechenden Werte für die WC_k an.

Eine Betrachtung der durchschnittlichen Abweichungen zeigt zunächst, daß die Abweichungen bezüglich der maximalen Terminüberschreitung bei allen betrachteten Nachbarschaften im Lösungsraum am größten und für die Zykluszeit am kleinsten sind. Eine systematische Entwicklung der Abweichungen bei steigender Auftragszahl N kann hingegen nicht festgestellt werden. Schließlich kann beobachtet werden, daß die API-Nachbarschaft sowohl bei AVG_k^1 als auch bei AVG_k^2 von allen betrachteten Nachbarschaften im Lösungsraum die kleinsten Abweichungen aufweist. Die größten Abweichungen treten in der Regel bei der INV-Nachbarschaft auf. Nur bei

<i>NSLR</i>	<i>N</i>	AVG_{Cmax}^1	AVG_{Fmid}^1	AVG_{Tmax}^1	AVG_{Cmax}^2	AVG_{Fmid}^2	AVG_{Tmax}^2
	4	0,0098	0,0114	0,0531	0,0103	0,0125	0,0581
	6	0,0089	0,0155	0,0448	0,0099	0,0173	0,0524
	8	0,0132	0,0210	0,0427	0,0148	0,0227	0,0490
API		0,0106	0,0159	0,0469	0,0116	0,0175	0,0532
	4	0,0231	0,0268	0,1086	0,0261	0,0315	0,1255
	6	0,0279	0,0403	0,1218	0,0328	0,0451	0,1537
	8	0,0367	0,0551	0,1108	0,0413	0,0591	0,1280
EX		0,0292	0,0406	0,1138	0,0333	0,0451	0,1358
	4	0,0163	0,0183	0,0834	0,0192	0,0210	0,0995
	6	0,0197	0,0276	0,0907	0,0230	0,0297	0,1087
	8	0,0270	0,0388	0,0824	0,0306	0,0412	0,0963
FSH		0,0209	0,0281	0,0855	0,0242	0,0305	0,1016
	4	0,0204	0,0237	0,0938	0,0225	0,0267	0,1074
	6	0,0194	0,0338	0,0937	0,0232	0,0378	0,1184
	8	0,0258	0,0455	0,0859	0,0291	0,0488	0,0969
BSH		0,0218	0,0342	0,0912	0,0249	0,0376	0,1077
	4	0,0236	0,0266	0,1113	0,0272	0,0309	0,1347
	6	0,0255	0,0384	0,1171	0,0306	0,0428	0,1488
	8	0,0315	0,0503	0,1021	0,0358	0,0537	0,1190
DSH		0,0268	0,0383	0,1103	0,0312	0,0423	0,1343
	4	0,0232	0,0270	0,1076	0,0262	0,0316	0,1285
	6	0,0280	0,0405	0,1234	0,0327	0,0454	0,1559
	8	0,0380	0,0566	0,1168	0,0425	0,0608	0,1362
INV		0,0296	0,0412	0,1159	0,0337	0,0458	0,1403

Tabelle 5.7: Durchschnittliche Abweichungen AVG_k^1 u. AVG_k^2

$N=4$ hat die DSH-Nachbarschaft für AVG_{Cmax}^1 , AVG_{Tmax}^1 , AVG_{Cmax}^2 und AVG_{Tmax}^2 die größten Abweichungen.

Die Auswertung der größten Abweichung (WC_k) zeigt, daß auch hier die API-Nachbarschaft stets die geringste Abweichung aufweist. Außer bei $N=4$ und WC_{Fmid} und WC_{Tmax} , wo die DSH-Nachbarschaft die größten Werte hat, weist die INV-Nachbarschaft die größten Werte für WC_k auf.

Daraus folgt, daß, falls mit dem API-Operator ein "zu großer" Zug³⁴ im Zielraum durchgeführt würde, dieser sowohl im Durchschnitt als auch im ungünstigsten Fall kleiner wäre als mit den anderen untersuchten Operatoren. Da AVG_k^1 die Größe der gesamten reduzierten Nachbarschaft berücksichtigt, erscheint beim API-Operator auch die Wahrscheinlichkeit für einen solchen Zug am geringsten.

³⁴Als "zu groß" werden hier Züge bezeichnet, wenn für den Zielfunktionswert k des Elements π' zu dem gezogen wird, gilt: $z_k(\pi') < z_k^{grenze}$. Vergleiche auch die Ausführungen zur Berechnung von $DEV_k(\pi', \pi^\circ)$ auf Seite 119.

<i>NSLR</i>	<i>N</i>	WC_{Cmax}	WC_{Fmid}	WC_{Tmax}
	4	0,0106	0,0127	0,0589
	6	0,0103	0,0184	0,0544
	8	0,0161	0,0256	0,0534
API		0,0123	0,0188	0,0556
	4	0,0278	0,0335	0,1328
	6	0,0367	0,0518	0,1727
	8	0,0493	0,0726	0,1533
EX		0,0378	0,0524	0,1529
	4	0,0209	0,0231	0,1068
	6	0,0259	0,0330	0,1208
	8	0,0364	0,0489	0,1155
FSH		0,0276	0,0348	0,1143
	4	0,0234	0,0280	0,1123
	6	0,0258	0,0434	0,1325
	8	0,0347	0,0588	0,1146
BSH		0,0279	0,0432	0,1199
	4	0,0299	0,0350	0,1479
	6	0,0353	0,0501	0,1715
	8	0,0441	0,0669	0,1472
DSH		0,0364	0,0505	0,1556
	4	0,0279	0,0339	0,1352
	6	0,0373	0,0526	0,1770
	8	0,0509	0,0752	0,1648
INV		0,0385	0,0537	0,1589

Tabelle 5.8: Größte Abweichung WC_k

5.3.2.3 Zusammenfassung

Nach der Darstellung verschiedener Nachbarschaften im Lösungsraum wurden verschiedene Nachbarschaftskonzepte für den Zielraum vorgestellt. Anschließend wurde in einer zweistufigen Versuchsreihe überprüft, ob oder wie häufig ein Nachbar im Zielraum auch ein Nachbar im Lösungsraum ist. Es hat sich gezeigt, daß für die API-Nachbarschaft die Chance, einen Nachbarn im Zielraum zu enthalten, am größten ist. Es hat sich weiter gezeigt, daß die API-Nachbarschaft die geringsten Abweichungen von *NSZR4* für alle drei berücksichtigten Ziele aufweist. Somit erscheint es angebracht, die API-Nachbarschaft in einem Algorithmus zur intensivierten Untersuchung der näheren Nachbarschaft einer Ausgangsfolge im Zielraum zu verwenden. Es bleibt jedoch anzumerken, daß diese Resultate lediglich empirisch sind. Ein Austauschen der Testprobleme, insbesondere durch solche mit speziellen Strukturen, wie z.B. korrelierten Bearbeitungszeiten oder einer anderen Verteilung der

Bearbeitungszeiten, kann zu abweichenden Ergebnissen führen.³⁵

5.4 Analyse der Qualität der Distanzapproximationen

5.4.1 Zur Notwendigkeit der Approximation

Reeves (1999) hat aufgrund einer signifikanten Korrelation von der approximierten Distanz zwischen lokalen Optima und deren Zykluszeit in einem Randomization Test auf die Existenz des "Großen Tals" bei Permutation Flow Shop Problemen geschlossen.³⁶ Dabei hat er vier unterschiedliche Maße als Ersatz für die echte Distanz zwischen jeweils zwei Auftragsfolgen verwendet, da sich die echte Distanz nicht in polynomialer Zeit bestimmen läßt. Reeves (1999) weist darauf hin, daß es schwer ist zu beurteilen, welches Distanzmaß die echte Distanz am besten approximiert. Es wird lediglich die Vermutung geäußert, daß zu erwarten sei, daß das Ersatzmaß POSN am besten in Lage ist, die einzelnen lokalen Optima voneinander abzugrenzen. Das Maß ADJ1 sei dazu im Falle des Permutation Flow Shop Problems am wenigsten in der Lage.³⁷

Grundsätzlich kann die Distanz zwischen zwei Auftragsfolgen als die Anzahl der mindestens notwendigen Züge mit einem Nachbarschaftsoperator, um die Auftragsfolge π in π' zu überführen, gemessen werden.³⁸ Bisher ist jedoch kein Algorithmus bekannt, der diese Distanz in polynomialer Zeit zu bestimmen in der Lage ist.³⁹ Aus diesem Grund verwendet Reeves (1999) in seiner Untersuchung die vier im folgenden Abschnitt 5.4.2 vorgestellten Ersatzmaße.⁴⁰ Offen ist jedoch, wie gut die Ersatzmaße in der Lage sind, die echte Distanz zu approximieren. In Abschnitt 5.5 soll die Topographie des Lösungsraums untersucht werden. Die Verwendung der Ersatzmaße erscheint dabei nur dann sinnvoll möglich, wenn diese in der Lage sind, die echte Distanz angemessen gut zu approximieren. Die Analyse der Approximationsqualität ist daher Inhalt des Abschnitts 5.4.3.

³⁵Auf die gleiche Einschränkung weist auch Reeves (1999) hin, indem er anmerkt, daß alle Untersuchungen zu Nachbarschaftsoperatoren lediglich empirisch sein können. Vgl. Reeves (1999, S. 479).

³⁶Vgl. Reeves (1999, S.483).

³⁷Vgl. Reeves (1999, S.478).

³⁸Vgl. Reeves (1999, S.478). Um im Folgenden eine bessere Unterscheidung zu den Distanzen, welche mit Hilfe der Ersatzmaße bestimmt werden, zu erreichen, wird diese Distanz im Folgenden auch als echte Distanz bezeichnet.

³⁹Vgl. Reeves (1999, S.478).

⁴⁰Vgl. zu den in Abschnitt 5.4.2 Vorgestellten Auftragsfolgen Reeves (1999, S.478).

5.4.2 Maße zur Distanzapproximation zwischen zwei Auftragsfolgen

Unidirectional Adjacency Metric (ADJ1) Dieses Maß wertet die Anzahl gleicher geordneter Auftragspaare in den Auftragsfolgen π und π' aus und subtrahiert diese von der bei N Aufträgen auftretenden Anzahl geordneter Auftragspaare $(N - 1)$.⁴¹ ADJ1 errechnet sich somit wie folgt, wobei $\pi_{[h]}$ für den Auftrag auf Position h in Auftragsfolge π steht:

$$ADJ1(\pi, \pi') = N - 1 - \sum_{h=1}^{N-1} x_{\pi_{[h]}, \pi'_{[h+1]}} \quad (5.32)$$

mit:

$$x_{ij} = \begin{cases} 1 & \text{falls Auftrag } j \text{ direkt auf Auftrag } i \text{ in } \pi \text{ und } \pi' \text{ folgt} \\ 0 & \text{sonst} \end{cases} \quad (5.33)$$

$N - 1$ ist die Anzahl der möglichen geordneten Auftragspaare bei N Aufträgen. Gilt $ADJ(\pi, \pi') = 0$, so gilt offensichtlich $\pi = \pi'$.

Bidirectional Adjacency Metric (ADJ2) Das Maß wertet die Anzahl gleicher ungeordneter Auftragspaare in den Auftragsfolgen π und π' aus und subtrahiert diese von der bei N Aufträgen auftretenden Anzahl von Auftragspaaren $(N - 1)$.⁴² Im Unterschied zu ADJ1 macht es hier keinen Unterschied, in welcher Reihenfolge die Aufträge in den Auftragsfolgen auftreten. Daher errechnet sich ADJ2 wie folgt:

$$ADJ2(\pi, \pi') = N - 1 - \sum_{h=1}^{N-1} (x_{\pi_{[h]}, \pi'_{[h+1]}} + x_{\pi'_{[h+1]}, \pi_{[h]}}) \quad (5.34)$$

mit:

$$x_{ij} = \begin{cases} 1 & \text{falls Auftrag } j \text{ direkt auf Auftrag } i \text{ in } \pi \text{ und } \pi' \text{ folgt} \\ 0 & \text{sonst} \end{cases} \quad (5.35)$$

Hier gilt offenbar, daß, wenn $\pi = \pi'$ gilt, auch $ADJ2(\pi, \pi') = 0$ gilt. Die Umkehrung dieser Aussage gilt hingegen nicht.

⁴¹Das ADJ1 Maß wird in der Arbeit von Reeves (1999) verbal beschrieben (vgl. Reeves (1999, S.478)). Das Maß errechnet sich dort aus $N - 1 - n_{adj}$, wobei n_{adj} die Anzahl gleicher geordneter Auftragspaare ist. Eine formale Darstellung, wie n_{adj} berechnet werden kann, ist dort nicht erfolgt.

⁴²Das ADJ2 Maß wird in der Arbeit von Reeves (1999) verbal beschrieben (vgl. Reeves (1999, S.478)). Eine formale Darstellung wie in dieser Arbeit ist dort nicht erfolgt.

Precedence Metric (PREC) Dieses Maß läßt die Forderung der unmittelbaren Nachbarschaft von Aufträgen in den Auftragsfolgen fallen. Es wertet lediglich aus, ob Auftrag i in π und π' (irgendwo) vor Auftrag j steht.⁴³ Es errechnet sich aus:⁴⁴

$$PREC(\pi, \pi') = \frac{N(N-1)}{2} - \sum_{g=1}^{N-1} \sum_{h=g+1}^N y_{\pi_{[g]}, \pi'_{[h]}} \quad (5.36)$$

mit:

$$y_{ij} = \begin{cases} 1 & \text{falls Auftrag } i \text{ sowohl in } \pi \text{ und } \pi' \text{ vor Auftrag } j \text{ steht} \\ 0 & \text{sonst} \end{cases} \quad (5.37)$$

Es gibt somit an, wieviele Aufträge i in Auftragsfolge π' hinter Auftrag j stehen, obwohl sie in Auftragsfolge π vor Auftrag j stehen.

Position Based Metric (POSN) Dieses Ersatzmaß verzichtet auf die Analyse der Reihenfolge von Aufträgen. Vielmehr zielt es direkt auf die Position eines Auftrags in den zwei Auftragsfolgen ab und ermittelt die Differenz der Positionen.⁴⁵ Ist $[i]_{\pi}$ die Position von Auftrag i in Auftragsfolge π und $[i]_{\pi'}$ die Position von Auftrag i in π' , so ergibt sich das Distanzmaß aus:

$$POSN(\pi, \pi') = \sum_{i=1}^N |[i]_{\pi} - [i]_{\pi'}| \quad (5.38)$$

Das POSN-Maß kann nur den Wert Null oder gerade Werte annehmen. Den Wert Null nimmt es an, wenn gilt: $\pi = \pi'$. Weicht die Position eines Auftrags in π und π' um eine ungerade Anzahl an Positionen ab, so muß es dafür einen weiteren Auftrag geben, der ebenfalls um eine ungerade Anzahl an Positionen verschoben ist. D.h., es gibt stets keinen oder eine gerade Anzahl an Aufträgen, die um eine ungerade Anzahl an Positionen verschoben sind. Die Summe zweier ungerader Zahlen ergibt jedoch stets eine gerade Zahl. Somit kann das POSN-Maß nur gerade Werte oder aber den Wert Null annehmen.

⁴³Das PREC Maß wird in der Arbeit von Reeves (1999) verbal beschrieben (vgl. Reeves (1999, S.478)). Eine formale Darstellung wie in dieser Arbeit ist dort nicht erfolgt.

⁴⁴Im Gegensatz zu x_{ij} von *ADJ1* und *ADJ2* ist es bei y_{ij} nun nicht mehr notwendig, daß Auftrag i unmittelbar vor Auftrag j steht, diese also in der Auftragsfolge benachbart sind.

⁴⁵In dieser Arbeit ist eine exaktere, der von Reeves (1999) abweichende formale Darstellung gewählt worden (vgl. Reeves (1999, S.478)).

5.4.3 Analyse der Approximationsqualität

5.4.3.1 Allgemeine Anforderungen an ein Maß

In diesem und im folgenden Abschnitt 5.4.3.2 soll eine Analyse der Approximationsqualität der Ersatzmaße durchgeführt werden. In Abschnitt 5.5 soll die Topographie des Lösungsraums untersucht werden.

Damit die von Reeves (1999) verwendeten Größen als Maße verwendet werden können, erscheint es sinnvoll, sie zunächst daraufhin zu überprüfen, ob sie die folgenden vier allgemeinen Anforderungen erfüllen:⁴⁶

1. Nicht-Negativität: $\delta(\pi, \pi') \geq 0$
2. Separation: $\delta(\pi, \pi') = 0 \Rightarrow \pi = \pi'$
3. Symmetrie: $\delta(\pi, \pi') = \delta(\pi', \pi)$
4. Dreiecks-Ungleichung: $\delta(\pi, \pi^\circ) + \delta(\pi^\circ, \pi') \geq \delta(\pi, \pi')$

Um die vier Ersatzmaße auf die Erfüllung dieser Eigenschaften hin zu überprüfen, wurden für $N = 2$ bis $N = 7$ jeweils alle $N!$ Auftragsfolgen durch vollständige Enumeration erzeugt und für jedes der vier Ersatzmaße die Distanzmatrix errechnet. Diese enthält die Distanz von allen Auftragsfolgen zu allen anderen Auftragsfolgen. Anschließend wurde mit ihrer Hilfe überprüft, ob die vier Forderungen erfüllt werden. Die folgenden Tabellen 5.9 und 5.10 geben das Ergebnis dieser Untersuchung wieder.

N=2	ADJ1	ADJ2	PREC	POSN
Nicht-Negativität:	+	+	+	+
Separation:	+	-	+	+
Symmetrie:	+	+	+	+
Dreiecks-Ungleichung:	+	+	+	+

+ Anforderung wird erfüllt; - Anforderung wird nicht erfüllt

Tabelle 5.9: Anforderungen an ein Distanzmaß N=2

Die vier Maße erfüllen in allen überprüften Fällen die Anforderung der Nicht-Negativität. Aufgrund ihrer Definition ist es bei allen vier Maßen nicht möglich, daß sie einen negativen Wert annehmen, so daß die Anforderung der Nicht-Negativität für beliebige Werte von N erfüllt ist. Bei ADJ1 und ADJ2 kann der Summenterm maximal den Wert $N-1$ annehmen, so daß Null der kleinste mögliche Wert ist. Bei PREC ist der maximal mögliche Wert des Summenterms $\frac{N(N-1)}{2}$. Das POSN-Maß ermittelt sich aus den

⁴⁶Vgl. dazu auch die Anforderungen, die in Steuer (1986) an l_p -Maße gestellt werden, (vgl. Steuer (1986, S.44) sowie Abschnitt 3.2.2.5, S.35).

N=3,4,5,6,7	ADJ1	ADJ2	PREC	POSN
Nicht-Negativität:	+	+	+	+
Separation:	+	-	+	+
Symmetrie:	+	+	-	+
Dreiecks-Ungleichung:	+	+	+	+

+ Anforderung wird erfüllt; - Anforderung wird nicht erfüllt

Tabelle 5.10: Anforderungen an ein Distanzmaß $N=3,4,5,6,7$

absoluten Differenzen der Auftragspositionen, so daß ebenfalls kein negativer Wert möglich ist.

Die Anforderung der Separation wird von ADJ1, PREC und POSN in allen überprüften Fällen erfüllt. Sobald sich in einer Auftragsfolge die Position eines Auftrags ändert, ist mindestens ein x_{ij} bzw. y_{ij} gleich Null und ADJ1 bzw. PREC nehmen einen Wert größer als Null an.⁴⁷ Bei POSN gibt es im Fall $\pi \neq \pi'$ mind. zwei Aufträge i mit einem positiven Wert für $|[i]_{\pi} - [i]_{\pi'}|$. ADJ2 hingegen verletzt die Separationseigenschaft. Eine Analyse der Distanzmatrix zeigt, daß unterschiedliche Auftragsfolgen eine Distanz von Null haben können. Dies geschieht, da ADJ2 auf die direkte Nachbarschaft zweier Aufträge in den Auftragsfolgen abstellt, jedoch unabhängig von ihrer Reihenfolge. Damit wird der Inversion der Auftragsfolge eine Distanz von Null zugewiesen. So haben z.B. $\pi = (1, 2, 3, 4)$ und $\pi' = (4, 3, 2, 1)$ eine ADJ2-Distanz von Null. Die Verletzung dieser Eigenschaft ist unerwünscht und läßt ADJ2 als Ersatzmaß zur Distanzmessung ungeeignet erscheinen. Insbesondere, wenn man bedenkt, daß die echte Distanz zwischen diesen beiden Auftragsfolgen für den API-Operator 7 Züge beträgt. Das ist die größte Distanz, die bei vier Aufträgen unter Verwendung dieses Operators zwischen zwei Auftragsfolgen auftreten kann.

Die Symmetrieeigenschaft wird von ADJ1, ADJ2 und POSN in allen untersuchten Fällen erfüllt. Bei ADJ1 und ADJ2 wird die Anzahl der geordneten bzw. ungeordneten Auftragspaare in zwei Auftragsfolgen erfaßt. Daher ist es egal, welche der beiden Auftragsfolgen als Referenz bzw. Ausgangsfolge verwendet wird. Da POSN die absoluten Differenzen der Auftragspositionen berücksichtigt, ist es auch dort egal, welche der beiden Auftragsfolgen als Ausgangslösung verwendet wird. Die drei Maße erfüllen daher für beliebige Werte von N die Symmetrieeigenschaft. Das PREC-Maß erfüllt in den überprüften Fällen nur im Fall $N=2$ die Symmetrieeigenschaft. Ansonsten wird die Symmetrieeigenschaft verletzt, so daß auch PREC als Er-

⁴⁷In diese Fällen ist der Wert des Summenterms bei ADJ1 und kleiner $N-1$ und bei PREC kleiner als $\frac{N(N-1)}{2}$.

satzmaß zur Distanzmessung ungeeignet scheint. Jedoch muß angemerkt werden, daß auch einige Nachbarschaften im Lösungsraum nicht symmetrisch sind. Dazu gehören die in dieser Arbeit betrachteten Nachbarschaften FSH und BSH. Unter diesem Blickwinkel kann die Verletzung der Symmetrieeigenschaft unter Umständen als erwünscht gelten. Offen ist, ob sich die Asymmetrie des Maßes mit der Asymmetrie der Nachbarschaft im Lösungsraum deckt. Aus diesem Grund wird PREC weiterhin in den Analysen berücksichtigt.

Die Dreiecks-Ungleichung wird in allen überprüften Fällen von allen vier Maßen erfüllt. Es ist im Rahmen der Arbeit nicht möglich, einen Beweis für das Erfüllen der Dreiecksungleichung für beliebige Werten von N zu führen. Aufgrund der überprüften Fälle liegt jedoch die Vermutung nahe, daß sie immer erfüllt ist.

5.4.3.2 Differenzierungstiefe und Korrelation mit der echten Distanz

Für ein Ersatzmaß genügt es nicht, die untersuchten allgemeinen Anforderungen an ein Maß zu erfüllen. Zusätzlich muß es eine gute Abschätzung der echten Distanz, gemessen in Zügen, zwischen den Auftragsfolgen geben.

Um die Approximationsqualität der Ersatzmaße beurteilen zu können, ist die Kenntnis der echten Distanz zwischen den Auftragsfolgen unverzichtbar. Die Untersuchung muß somit auf kleine Problemgrößen mit vier, sechs sowie acht Aufträgen beschränkt werden, da es bei diesen Problemgrößen noch möglich ist, die echten Distanzen in angemessener Rechenzeit zu ermitteln. Dies geschieht mit Hilfe eines rekursiven Algorithmus. Dieser ermittelt die echte Distanz von einer Auftragsfolge zu allen anderen $N! - 1$ Auftragsfolgen für einen gegebenen Nachbarschaftsoperator. Im Algorithmus 1 ist dieses rekursive Verfahren wiedergegeben.

Algorithmus 1 Berechnung der "echten Distanz" zwischen zwei Auftragsfolgen

Beschreibung der verwendeten Datenstrukturen:

- DISTANZLISTE: Globale Datenstruktur, die für jede der $N!$ möglichen Auftragsfolgen die Anzahl an Zügen speichert, die mindestens benötigt werden, um sie von der Ausgangsfolge π° aus zu erreichen.
- KNOTENLISTE: Datenstruktur, welche eine Liste von Auftragsfolgen speichert.

Start der Berechnung

- 1: Initialisierung der Ausgangsfolge π° also:
 $\pi^\circ = (1, 2, \dots, N)$;
- 2: Setze die Anzahl der Züge für alle Elemente in DISTANZLISTE auf ∞ ;
- 3: Setze die Anzahl der Züge in DISTANZLISTE für π° gleich 0;
- 4: Def. START als Datenstruktur vom Typ KNOTENLISTE und initialisiere sie als leere Liste.
 KNOTENLISTE START = {};
- 5: Füge π° in START ein: START= $\langle \pi^\circ \rangle$;
- 6: Starte Rekursion
 get_distance(START, 1);
- 7: STOP;

-
- 8: Funktion **get_distance(KNOTENLISTE LETZTE_LAGE, int Zug)** {
 - 9: Def. LAGE als Datenstruktur vom Typ KNOTENLISTE und initialisiere sie als leere Liste.
 KNOTENLISTE LAGE = {};
 - 10: Ermittle für jedes Element der übergebenen Liste von Auftragsfolgen LETZTE_LAGE die Nachbarn im Lösungsraum und füge diese in die Liste von Auftragsfolgen LAGE ein.
 - 11: Führe für jede Auftragsfolge y aus LAGE Schritt 12 aus:
 - 12: Falls Zug < aktuell in DISTANZLISTE gespeicherte Zugzahl für y :
 dann: aktualisiere Zugzahl für y in DISTANZLISTE mit dem Wert von Zug
 sonst: lösche y aus LAGE, da von dieser Auftragsfolge die Nachbarschaft bereits früher untersucht wurde.
 - 13: Falls LAGE $\neq \{\}$ rufe get_distance(LAGE, Zug+1) auf (Rekursion).
 - 14: } // Ende der Funktion get_distance
-

Die echte Distanz wurde mit Hilfe dieses Algorithmus für vier, sechs und acht Aufträge für jeden der vorgestellten Nachbarschaftsoperatoren im Lösungsraum berechnet. Zum Vergleich wurden die Werte der vier Ersatzmaße ebenfalls ermittelt. Um die Qualität der Approximation durch die Ersatzmaße zu beurteilen, wurden folgende Fragen untersucht:

1. Hat das Ersatzmaß eine ausreichende Differenzierungstiefe? Das Ersatzmaß sollte annähernd soviele Ausprägungsstufen annehmen können wie die Höhe der maximalen echten Distanz. Bewegt sich die echte Distanz z.B. im Intervall $[0; 29]$, so sollte ein gutes Ersatzmaß ca. 30 unterschiedliche Ausprägungsstufen annehmen können. Je größer die Abweichung, desto schwieriger ist es, von der Approximation auf die echte Distanz zu schließen.
2. Besteht eine starke Korrelation zwischen dem Ersatzmaß und der echten Distanz? Es muß gewährleistet sein, daß von einem hohen Wert des Ersatzmaßes auf eine große echte Distanz (positive Korrelation), bzw. von einem hohen Wert auf eine niedrige echte Distanz (negative Korrelation) geschlossen werden kann. Besteht keine solche starke Korrelation, so ist eine sinnvolle Interpretation des Ersatzmaßes nicht möglich.

Zunächst soll die Differenzierungstiefe analysiert werden. Dazu werden für die zu analysierenden Distanzmaße sowie für die echten Distanzen sowohl der maximale und minimale Wert sowie die Anzahl unterschiedlicher Ausprägungsstufen ermittelt. Die Werte sind in den Tabellen 5.11, 5.12 und 5.13 dargestellt. Der minimale Wert ist immer Null. Die Berücksichtigung von Null ist erforderlich, da ADJ2 als einziges Maß den Wert Null für unterschiedliche Auftragsfolgen annimmt.⁴⁸ Die Differenzierungstiefe wurde durch die Division der Anzahl der Ausprägungsstufen der Ersatzmaße durch die Anzahl der Ausprägungsstufen der echten Distanz bestimmt.

Für POSN ist die Anzahl der Ausprägungsstufen kleiner als seine maximale Ausprägung, was sich damit begründen läßt, daß POSN aufgrund seiner Definition nur den Wert Null und gerade Werte annehmen kann. Außer bei der API-Nachbarschaft, wo ADJ1, ADJ2 sowie POSN Werte kleiner 1 annehmen, wird die notwendige Differenzierungstiefe erreicht oder überschritten. Bei ADJ1 und ADJ2 bleibt mit Ausnahme der API-Nachbarschaft die Differenzierungstiefe konstant bei 100%.

⁴⁸Dies folgt aus der Verletzung der Separationseigenschaft.

N=4				ADJ1	ADJ2	PREC	POSN
			MIN	0	0	0	0
			MAX	3	3	6	8
	echte Distanz						
	MIN	MAX	Stufen	4	4	7	5
API	0	6	7	0,571	0,571	1	0,714
EX	0	3	4	1	1	1,75	1,25
FSH	0	3	4	1	1	1,75	1,25
BSH	0	3	4	1	1	1,75	1,25
DSH	0	3	4	1	1	1,75	1,25
INV	0	3	4	1	1	1,75	1,25

Tabelle 5.11: Differenzierungstiefe Ersatzmaße N=4

N=6				ADJ1	ADJ2	PREC	POSN
			MIN	0	0	0	0
			MAX	5	5	15	18
	echte Distanz						
	MIN	MAX	Stufen	6	6	16	10
API	0	15	16	0,375	0,375	1	0,625
EX	0	5	6	1	1	2,667	1,667
FSH	0	5	6	1	1	2,667	1,667
BSH	0	5	6	1	1	2,667	1,667
DSH	0	5	6	1	1	2,667	1,667
INV	0	5	6	1	1	2,667	1,667

Tabelle 5.12: Differenzierungstiefe Ersatzmaße N=6

N=8				ADJ1	ADJ2	PREC	POSN
			MIN	0	0	0	0
			MAX	7	7	28	32
	echte Distanz						
	MIN	MAX	Stufen	8	8	29	17
API	0	28	29	0,276	0,276	1	0,5862
EX	0	7	8	1	1	3,625	2,125
FSH	0	7	8	1	1	3,625	2,125
BSH	0	7	8	1	1	3,625	2,125
DSH	0	7	8	1	1	3,625	2,125
INV	0	7	8	1	1	3,625	2,125

Tabelle 5.13: Differenzierungstiefe Ersatzmaße N=8

Als zweites Kriterium soll die Korrelation zwischen echter Distanz und den Werten der Ersatzmaße ermittelt werden. Die Tabellen 5.14, 5.15 sowie 5.16 geben die ermittelten Korrelationskoeffizienten für N=4, 6 bzw. 8 wieder. Zunächst fällt auf, daß die Korrelationskoeffizienten

N=4	ADJ1	ADJ2	PREC	POSN
API	0,314	0,000	0,637	0,928
EX	0,095	0,343	0,754	0,747
FSH	0,410	0,137	0,327	0,747
BSH	0,410	0,137	0,754	0,747
DSH	0,676	0,087	0,550	0,758
INV	0,272	0,750	0,620	0,614

Tabelle 5.14: Korrelationen zwischen echter Distanz und den Ersatzmaßen, N=4

N=6	ADJ1	ADJ2	PREC	POSN
API	0,174	0,000	0,521	0,925
EX	0,032	0,157	0,598	0,567
FSH	0,275	0,104	0,044	0,567
BSH	0,275	0,104	0,598	0,567
DSH	0,545	0,104	0,423	0,698
INV	0,368	0,707	0,286	0,269

Tabelle 5.15: Korrelationen zwischen echter Distanz und den Ersatzmaßen, N=6

N=8	ADJ1	ADJ2	PREC	POSN
API	0,117	0,000	0,468	0,929
EX	0,015	0,098	0,505	0,463
FSH	0,212	0,086	-0,057	0,463
BSH	0,212	0,086	0,505	0,463
DSH	0,473	0,105	0,368	0,669
INV	0,430	0,730	0,152	0,138

Tabelle 5.16: Korrelationen zwischen echter Distanz und den Ersatzmaßen, N=8

mit steigender Zahl der Aufträge zumeist sinken. Für die Kombination FSH/PREC wird sie bei N=8 sogar sehr schwach negativ. Eine Ausnahme dieser Beobachtung bilden hier die Kombinationen API/POSN, API/ADJ2 sowie DSH/ADJ2. Für Probleme mit acht Aufträgen kann lediglich POSN für die API-Nachbarschaft noch eine sehr stark positive Korrelation aufweisen und für die DSH-Nachbarschaft zumindest noch eine starke.

Aus den ermittelten Werten sticht API/POSN aufgrund der sehr starken Korrelation heraus, gleichzeitig erfüllt es die allgemeinen Anforderungen an Maße.⁴⁹ Die Schwäche dieses Maßes liegt in der fehlenden Differenzierungstiefe. Einer sinkenden Differenzierungstiefe steht ein sehr hoher Korrelationskoeffizient gegenüber. Eine mögliche Erklärung für die beschriebene Entwicklung ist, daß große echte Distanzen, welche von der Differenzierungstiefe des Maßes nicht erfaßt werden, zumindest einen hohen POSN-Wert erhalten. Damit bleibt eine hohe Kovarianz der beiden Zahlenreihen erhalten.

Bezüglich PREC bleibt anzumerken, daß es die Symmetrieeigenschaften nicht erfüllt. Dies kann, wie bereits erwähnt, bei der Erfassung von nicht symmetrischen Nachbarschaften wie FSH oder BSH eine wünschenswerte Eigenschaft sein. Jedoch zeigt sich auch hier eine schwache bis mittlere Korrelation. Die Asymmetrie der Nachbarschaft wird damit von PREC nicht "richtig" nachvollzogen.

Hat Reeves (1999) vermutet, daß ADJ1 am schlechtesten und POSN am besten zur Diskriminierung der lokalen Optima in der Lage sind,⁵⁰ so bleibt abschließend festzuhalten, daß beide Maße zwar unter Umständen diese Diskriminierungseigenschaften in seinen Experimenten aufweisen, sie aufgrund der hier vorgestellten Ergebnisse aber nicht in der Lage sind, eine gute Approximation der echten Distanzen zu geben. Ihre Verwendung zur Distanzmessung zwischen Auftragsfolgen scheint damit nicht zweckmäßig, da dies zu nicht sehr aussagekräftigen Ergebnissen führen würde. In den folgenden Analysen wird daher auf die Verwendung der Ersatzmaße verzichtet.

5.5 Analyse der Topographie im Lösungsraum

5.5.1 Die Korrelation von Distanz und Zielfunktionswert

Wie bereits in der Einführung zu diesem Kapitel ausgeführt, beschränken sich die bisherigen Arbeiten zur Existenz eines "Großen Tals" beim Permutation Flow Shop Problem auf die Zielgröße der Zykluszeit. Vergleichbare

⁴⁹Die Erfüllung der Dreiecks-Ungleichung wird in der Analyse nur für $N=4$ gezeigt. Es ist im Rahmen der Arbeit nicht möglich, einen Beweis für das Erfüllen der Dreiecksungleichung für beliebige Werten von N zu führen. Aufgrund der überprüften Fälle liegt jedoch die Vermutung nahe, daß sie immer erfüllt ist.

⁵⁰Vgl. Reeves (1999, S.478).

Analysen für die Zielgrößen der mittleren Durchlaufzeit sowie der maximalen Terminüberschreitung sind nicht bekannt.

Das experimentelle Vorgehen zur Untersuchung der Topographie des Lösungsraums in dieser Arbeit unterscheidet sich deutlich von den erwähnten Arbeiten von Boese u. a. (1993), Boese u. a. (1994) und Reeves (1999),⁵¹ da aufgrund der im letzten Abschnitt erhaltenen Ergebnisse auf die Verwendung von Ersatzmaßen zur Distanzmessung verzichtet wird. Die Analyse wird auf kleine Probleminstanzen mit bis zu acht Aufträgen beschränkt. Dies hat den Vorteil, daß die echte Distanz ermittelt und somit auf die Verwendung von Ersatzmaßen verzichtet werden kann. Ein weiterer wesentlicher Unterschied zu vorangegangenen Untersuchungen besteht darin, daß auf die Bestimmung lokaler Optima mit Hilfe einer Heuristik verzichtet wird. Damit ist der Einfluß der Heuristik auf die Qualität der lokalen Optima als Einflußfaktor auf das Analyseergebnis ebenfalls ausgeschlossen. Als Ausgangspunkt für die Analyse werden allein die globalen Optima verwendet. Damit die Existenz eines "Großen Tals" für ein Ziel bestätigt werden kann, muß die Korrelation zwischen der Distanz der Auftragsfolgen vom globalen Optimum mit dem Zielfunktionswert stark ausgeprägt sein. Existieren mehrere globale Optima für das betrachtete Ziel, so müssen diese im Lösungsraum nahe beieinander liegen.⁵² Im Zielraum sind globale Optima eines Ziels immer Nachbarn.

Aufgrund der Ergebnisse der Experimente zur Nachbarschaft im Zielraum in Abschnitt 5.3 wird die Distanz unter Verwendung eines API-Nachbarschaftsoperators berechnet. Zum einen ist bei diesem Nachbarschaftsoperator die Wahrscheinlichkeit, daß ein Nachbar im Lösungsraum auch ein Nachbar im Zielraum ist, am höchsten, was der Existenz eines "Großen Tales" zumindest förderlich ist. Zum anderen ist bei diesem Operator die maximale Anzahl der notwendigen Züge, um den gesamten Lösungsraum abzudecken, am größten (vgl. Tabellen: 5.11, 5.12, 5.13). Dies ermöglicht eine gute Differenzierung zwischen Distanz und Zielfunktionswert.

Im ersten Experiment werden für jedes Problem und jedes Ziel die globalen Optima bestimmt. Anschließend wird ausgehend von jedem dieser globalen Optima die Distanz zu allen anderen Auftragsfolgen berechnet, um anschließend die Korrelation zwischen Distanz und Zielfunktionswert zu ermitteln. Zusätzlich zur Korrelation mit der echten Distanz werden die Korrelationen mit den in Abschnitt 5.4 diskutierten Ersatzmaßen ermit-

⁵¹Vgl. Boese u. a. (1993, S.3-12), Boese u. a. (1994, S.102-110) und Reeves (1999, S.479-489).

⁵²Im Idealfall liegen diese direkt nebeneinander, d.h die Distanz beträgt einen Zug.

telt. Damit wird sichtbar, welcher Unterschied in der Beurteilung resultieren würde, falls die oben diskutierten Ersatzmaße verwendet würden. Für jedes der Testprobleme⁵³ werden die Durchschnittswerte der Korrelationen der globalen Optima der einzelnen Ziele berechnet. Die Ergebnisse sind in den Tabellen 5.17, 5.18 und 5.19 angegeben. In den fett gedruckten Zeilen sind die Durchschnittswerte alle untersuchten Probleminstanzen mit gleicher Auftragszahl angegeben. In den drei Zeilen darüber sind jeweils die Durchschnittswerte über die 10 Probleminstanzen mit gleicher Maschinenzahl bei gegebener Auftragszahl angegeben.

N	M	#O	ρ^{real}	ρ^{ADJ1}	ρ^{ADJ2}	ρ^{PREC}	ρ^{POSN}
4	5	1,3000	0,6874	0,2233	0,0473	0,5632	0,6987
4	10	1,3000	0,5893	0,2539	0,1148	0,5218	0,5991
4	20	1,7000	0,6136	0,2692	0,0815	0,5496	0,6320
4	\emptyset	1,4333	0,6301	0,2488	0,0812	0,5449	0,6433
6	5	3,3000	0,5100	0,1425	0,0760	0,3390	0,5365
6	10	1,3000	0,5580	0,1786	0,0411	0,3723	0,5392
6	20	4,1000	0,4312	0,1617	0,0483	0,2521	0,4376
6	\emptyset	2,9000	0,4997	0,1609	0,0551	0,3211	0,5045
8	5	19,5000	0,5926	0,0928	0,0286	0,2602	0,5514
8	10	4,0000	0,4354	0,1175	0,0526	0,2736	0,4315
8	20	4,0000	0,3860	0,0985	0,0500	0,2340	0,4123
8	\emptyset	9,3448	0,4725	0,1024	0,0435	0,2553	0,4662

#O Durchschnittliche Anzahl globaler Optima ,

ρ^{real} : Korrelation mit realer Distanz

ρ^{ADJ1} : Korrelation Zielfunktionswerte und ADJ1

ρ^{ADJ2} : Korrelation Zielfunktionswerte und ADJ2

ρ^{PREC} : Korrelation Zielfunktionswerte und PREC

ρ^{POSN} : Korrelation Zielfunktionswerte und POSN

Tabelle 5.17: Korrelation zwischen echter Distanz, Ersatzmaßen und Zykluszeit

Zunächst ist zu beobachten, daß bei der maximalen Terminüberschreitung in der Regel am häufigsten mehrere globale Optima existieren (vgl. die Werte für #O), lediglich bei N=8 gibt es bei der Zykluszeit im Durchschnitt mehr. Am seltensten treten multiple globale Optima bei der mittleren Durchlaufzeit auf.

Weiterhin ist zu erkennen, daß bei der Zykluszeit und der maximalen Terminüberschreitung die Korrelation mit der echten Distanz ρ^{real} mit steigender Anzahl von Aufträgen im Durchschnitt sinkt (vgl. die \emptyset -Werte bei gegebenem N).

⁵³Es werden Probleminstanzen mit $N \in \{4, 6, 8\}$ Aufträgen und $M \in \{5, 10, 20\}$ Maschinen betrachtet. Durch die Kombination von Auftrags- und Maschinenzahl ergeben sich 9 Testfelder. Für jedes Testfeld werden 10 Probleminstanzen erzeugt. Die Generierung dieser Testprobleme wird in den Anhängen A und B beschrieben.

N	M	#O	ρ^{real}	ρ^{ADJ1}	ρ^{ADJ2}	ρ^{PREC}	ρ^{POSN}
4	5	1,0000	0,7180	0,2085	0,0507	0,3863	0,7122
4	10	1,0000	0,6817	0,2011	0,1343	0,3860	0,6806
4	20	1,0000	0,8006	0,1948	0,0750	0,4566	0,7931
4	\emptyset	1,0000	0,7334	0,2015	0,0867	0,4096	0,7286
6	5	1,0000	0,5970	0,1337	0,0557	0,2158	0,5653
6	10	1,0000	0,5188	0,1839	0,0605	0,1562	0,4883
6	20	1,1000	0,6845	0,1693	0,0847	0,3024	0,6744
6	\emptyset	1,0333	0,6001	0,1623	0,0670	0,2248	0,5760
8	5	1,3000	0,5913	0,1138	0,0352	0,2124	0,5689
8	10	1,0000	0,6467	0,1197	0,0701	0,1928	0,6115
8	20	1,0000	0,5708	0,0965	0,0515	0,1618	0,5511
8	\emptyset	1,1034	0,6014	0,1097	0,0516	0,1889	0,5759

Tabelle 5.18: Korrelation zwischen echter Distanz, Ersatzmaßen und mittlerer Durchlaufzeit

N	M	#O	ρ^{real}	ρ^{ADJ1}	ρ^{ADJ2}	ρ^{PREC}	ρ^{POSN}
4	5	4,6000	0,4777	0,1932	0,0848	0,4362	0,5161
4	10	3,2000	0,4946	0,1593	0,1327	0,4400	0,5397
4	20	4,5556	0,4004	0,1869	0,1538	0,4453	0,4560
4	\emptyset	4,1034	0,4595	0,1796	0,1227	0,4403	0,5056
6	5	4,5000	0,4594	0,0955	0,0655	0,3102	0,5018
6	10	55,2000	0,5107	0,1230	0,0270	0,3192	0,5025
6	20	63,2000	0,2593	0,1177	0,0493	0,2387	0,3022
6	\emptyset	40,9667	0,4098	0,1121	0,0473	0,2894	0,4355
8	5	10,6000	0,4293	0,0761	0,0145	0,2335	0,4426
8	10	4,2222	0,3860	0,0960	0,0528	0,2266	0,4007
8	20	5,5000	0,3357	0,0629	0,0477	0,2307	0,3906
8	\emptyset	6,8621	0,3836	0,0777	0,0378	0,2304	0,4116

Tabelle 5.19: Korrelation zwischen echter Distanz, Ersatzmaßen und maximaler Terminüberschreitung

Auffällig ist, daß bei Verwendung der Ersatzmaße ADJ1, ADJ2 sowie PREC⁵⁴ die echte Korrelation ρ^{real} unterschätzt wird. Die Approximation mit Hilfe des POSN-Maßes führt bei dem Ziel der Zykluszeit bei Problemen mit N=4 und N=6 Aufträgen zu einer Überschätzung der echten Korrelation.⁵⁵ Wird bei der mittleren Durchlaufzeit die echte Korrelation immer unterschätzt, wird sie bei der maximalen Terminüberschreitung, bis auf den Fall N=6, M=10, überschätzt.

Stärke der Korrelation	Bereich	$C_{max}[\%]$	$F_{mid}[\%]$	$T_{max}[\%]$
mittel negativ	[-0,6;-0,4]	0,00	0,00	0,06
schwach negativ	[-0,4;-0,2]	0,00	0,00	6,01
sehr schwach negativ	[-0,2;0]	0,00	1,07	18,42
keine	0	0,00	0,00	0,00
sehr schwach positiv	[0;0,2]	1,50	3,23	26,89
schwach positiv	[0,2;0,4]	11,22	7,53	25,08
mittel positiv	[0,4;0,6]	38,90	21,50	18,49
stark positiv	[0,6;0,8]	46,63	45,16	4,72
sehr stark positiv	[0,8;0,99]	1,75	21,51	0,32

Tabelle 5.20: Häufigkeitsverteilung der Korrelationen

In Tabelle 5.20 ist für jedes Ziel eine Häufigkeitsverteilung der für die einzelnen globalen Optima ermittelten Korrelationen ρ^{real} angegeben. Auf eine Differenzierung nach Problemgrößen wird dabei verzichtet. Es wurde also für jedes der 90 Testprobleme für ein Ziel die globalen Optima bestimmt und ρ^{real} errechnet und einer der gebildeten Klassen "Stärke der Korrelation" zugeordnet. Es zeigt sich, daß sich beim Ziel der Zykluszeit bei 85,53 Prozent der globalen Optima eine mittlere bis starke Korrelation zwischen Distanz und Zielfunktionswert ergibt. Lediglich bei 1,75 Prozent der globalen Optima ergibt sich eine sehr starke Korrelation. Bezüglich der mittleren Durchlaufzeit ist in 66,66 Prozent der Fälle eine mittlere bis starke Korrelation festzustellen. Im Gegensatz zur Zykluszeit tritt in 21,51 Prozent eine sehr stark positive Korrelation auf. Aber auch sehr schwach negative Korrelationen können beobachtet werden. Dieses Resultat läuft der Vorstellung der Existenz eines "Großen Tals" entgegen.

Bei der maximalen Terminüberschreitung tritt sogar bei 24,43 Prozent der globalen Optima eine negative Korrelation auf. Lediglich in 23,21 Prozent der Fälle ist eine mittlere bis starke Korrelation zu beobachten. Sehr stark ist sie lediglich zu 0,32 Prozent ausgeprägt.

In den folgenden Experimenten soll die Lage multipler globaler Optima

⁵⁴Lediglich bei der maximalen Terminüberschreitung, 4 Aufträgen und 20 Maschinen überschätzt PREC die echte Korrelation.

⁵⁵Lediglich bei N=6 und M=10 wird die echte Korrelation ebenfalls unterschätzt.

bezüglich eines Problems und eines Ziel zueinander analysiert werden, sofern es diese gibt. Bereits Reeves (1999) hat das Thema multipler globaler Optima in seiner Arbeit diskutiert.⁵⁶ Nicht nahe im Lösungsraum beeinanderliegende multiple globale Optima könnten ein Grund für die Resultate der Korrelationsanalyse dieses Abschnitts sein. Im folgenden Abschnitt soll daher untersucht werden, wie weit multiple globale Optima voneinander entfernt liegen.

5.5.2 Analyse der Distanz globaler Optima

5.5.2.1 Isolierte Analyse multipler globaler Optima der einzelnen Ziele

In diesem Experiment werden die drei Ziele und die zu ihnen gehörenden globalen Optima zunächst unabhängig voneinander betrachtet. Für jedes Problem, welches für das betrachtete Ziel multiple globale Optima besitzt, wird eine Distanzmatrix berechnet. Diese enthält die echten Distanzen bei Anwendung des API-Operators von jedem globalen Optimum zu jedem anderen.⁵⁷ Zu jedem Element dieser Distanzmatrix wird außerdem bestimmt, wieviele Auftragsfolgen innerhalb dieser Distanz von der Ausgangsfolge aus erreichbar sind und in einer Reichweitenmatrix gespeichert. *MaxDist* ist das größte Element der Distanzmatrix, also die maximale Distanz zwischen zwei globalen Optima. *InMax* gibt die innerhalb vom *MaxDist* erreichbare Anzahl von Auftragsfolgen an. Schließlich wurde mit *%InMax* der Anteil des durch *MaxDist* abgedeckten Lösungsraums errechnet. Er errechnet sich wie folgt:

$$\%InMax = \frac{InMax}{N!} \quad (5.39)$$

Die Tabelle 5.21, 5.22 und 5.23 geben die Ergebnisse der Analyse, aggregiert nach Problemgröße (Anzahl der Aufträge N)⁵⁸, wieder.⁵⁹ Die Spalte *#P* gibt an, bei wievielen Probleminstanzen des jeweiligen Testfelds für das jeweils betrachtete Ziel multiple globale Optima aufgetreten sind. Die

⁵⁶Vgl. Reeves (1999, S.485-489).

⁵⁷Auch bei allen anderen in Abschnitt 5.5.2 durchgeführten Analysen wird die echte Distanz bei Anwendung des API-Operators zugrunde gelegt.

⁵⁸Die Problemgröße wird hier allein von der Anzahl der Aufträge N abhängig gemacht, da alleine N Einfluß auf die Größe des Lösungsraums ($|\Pi| = N!$) hat, welche in die Berechnung von *%InMax* eingeht.

⁵⁹Die Berechnungen dieser Experimente sind sehr rechenintensiv. Problem 75 weist bei der maximalen Terminüberschreitung eine sehr große Anzahl an globalen Optima auf. Aus diesem Grund wurde bei allen Experimenten dieses Abschnitts 4.5.2 das Problem 75 von der Analyse ausgeschlossen.

Spalte $\#O$ gibt an, wieviele globale Optima durchschnittlich aufgetreten sind. Die Werte weichen von denen aus den Tabellen 5.17, 5.18 und 5.19 ab, da in der hier vorgenommenen Auswertung nur der Durchschnitt über die Probleminstanzen mit multiplen globalen Optima ermittelt wird.

N	M	$\#P$	$\#O$	$MaxDist$	$\%InMax$
4	5	2	2,5000	1,5000	0,2708
4	10	3	2,0000	1,0000	0,1667
4	20	3	3,3333	1,6667	0,3194
4	\emptyset		2,6250	1,3750	0,2500
6	5	7	4,2857	2,5714	0,1115
6	10	3	2,0000	1,6667	0,0213
6	20	2	16,5000	6,0000	0,4458
6	\emptyset		5,7500	2,9167	0,1447
8	5	9	21,5556	5,5556	0,1032
8	10	7	4,8571	5,5714	0,0987
8	20	4	8,5000	4,5000	0,0576
8	\emptyset		13,1000	5,3500	0,0925

Tabelle 5.21: Isolierte Analyse globaler Optima - Zykluszeit

Bei der Zykluszeit (vgl. Tabelle 5.21) steigt die Anzahl der globalen Optima im Durchschnitt bei zunehmender Anzahl Aufträge an. Ebenso steigen die Werte von $MaxDist$ und damit die Distanz zwischen den globalen Optima bei steigendem N an. Gleichzeitig sinkt $\%InMax$. Insbesondere das Sinken von $\%InMax$ bedeutet, daß der Anteil am Lösungsraum, in dem die globalen Optima beeinander liegen, kleiner wird. Die Größe des Lösungsraums ($|\Pi| = N!$) steigt also mit zunehmendem N relativ stärker an als $InMax$.

N	M	$\#P$	$\#O$	$MaxDist$	$\%InMax$
4	5	-	-	-	-
4	10	-	-	-	-
4	20	-	-	-	-
4	\emptyset		-	-	-
6	5	-	-	-	-
6	10	-	-	-	-
6	20	1	2,00000	2,0000	0,0278
6	\emptyset		2,00000	2,0000	0,0278
8	5	3	2,00000	1,6667	0,0010
8	10	-	-	-	-
8	20	-	-	-	-
8	\emptyset		2,0000	1,6667	0,0010

- : keine multiplen globalen Optima aufgetreten

Tabelle 5.22: Isolierte Analyse globaler Optima - Mittlere Durchlaufzeit

Bei der mittleren Durchlaufzeit (vgl. Tabelle 5.22) treten nur bei einem Problem mit $N=6$ Aufträgen und $M=20$ Maschinen sowie bei drei Problemen mit $N=8$ Aufträgen und $M=5$ Maschinen multiple globale Optima auf. Es gibt in allen 4 Probleminstanzen jeweils zwei globale Optima. Im Gegensatz zur Zykluszeit nimmt die Distanz zwischen diesen globalen Optima mit steigender Auftragszahl ab (vgl. die Werte $MaxDist$). Der Anteil am Lösungsraum, in dem die zwei globalen Optima liegen ($%InMax$), wird daher auch kleiner, da der Zähler des Terms zu seiner Berechnung sinkt, während der Nenner steigt.

N	M	$\#P$	$\#O$	$MaxDist$	$%InMax$
4	5	5	8,2000	3,4000	0,5917
4	10	6	4,6667	2,3333	0,4167
4	20	6	10,1667	3,6667	0,6181
4	\emptyset		7,6471	3,1176	0,5392
6	5	8	5,3750	2,8750	0,0931
6	10	8	68,7500	4,6250	0,2519
6	20	6	104,6667	8,3333	0,5500
6	\emptyset		55,5000	5,0000	0,2754
8	5	9	11,6667	4,8889	0,0884
8	10	8	4,6250	5,3750	0,0885
8	20	7	7,4286	7,5714	0,1941
8	\emptyset		8,0833	5,8333	0,1193

Tabelle 5.23: Isolierte Analyse globaler Optima - Maximale Terminüberschreitung

Bei der maximalen Terminüberschreitung (vgl. Tabelle 5.23) ist insbesondere auffällig, daß die Anzahl der globalen Optima im Durchschnitt bei Problemen mit $N=6$ Aufträgen am größten ist. Der Anteil am Lösungsraum, in dem die globalen Optima liegen ($%MaxDist$), sinkt auch bei der maximalen Terminüberschreitung systematisch mit steigender Auftragszahl. Zusätzlich fällt auf, daß es beim Betrachten des Ziels der maximalen Terminüberschreitung am häufigsten Probleminstanzen mit multiplen globalen Optima gibt (vgl. die Werte für $\#P$). Am seltensten treten Probleminstanzen mit multiplen globalen Optima bei der mittleren Durchlaufzeit auf. Am weitesten auseinander liegen die globalen Optima bei der Zielgröße der maximalen Terminüberschreitung. Am dichtesten beieinander liegen sie beim Ziel der mittleren Durchlaufzeit.

Es erscheint wahrscheinlich, daß die Werte von $%InMax$ mit zunehmender Auftragszahl weiter sinken. Die globalen Optima eines Ziels liegen daher häufig in einem begrenzten Bereich des Lösungsraum beisammen.

Bisherige Untersuchungen haben mit lokalen Optima gearbeitet.⁶⁰ Hier wurde hingegen eine Analyse mit globalen Optima durchgeführt. Die Definition des "Großen Tals" ist daher hier wie folgt anzupassen: Ein "Großes Tal" liegt vor, wenn die globalen Optima eines Ziels in einem begrenzten Bereich des Lösungsraums beieinander liegen. Wie begrenzt der Bereich sein muß, ist weiterhin offen. Die hier beobachteten Werte unterstützen jedoch die Vermutung der Existenz eines "Großen Tals" für die in der Analyse verwendete API-Nachbarschaft.

5.5.2.2 Integrierte Analyse aller globalen und der effizienten globalen Optima aller Ziele

In der folgenden Analyse werden zunächst die Distanzen zwischen den globalen Optima aller Ziele für jede Probleminstanz untersucht. Dazu wurden die Mengen der globalen Optima der einzelnen Ziele vereinigt.⁶¹ Die Berechnung der Auswertungsgrößen wurde analog zur isolierten Analyse vorgenommen. Die Auswertungsgrößen beziehen sich nun also auf die globalen Optima aller drei Ziele. Dabei werden auch globale Optima berücksichtigt, die bei Betrachtung des gesamten Zielfunktionsvektors nicht effizient sind und damit nicht der Menge aller effizienten Lösungen angehören. Die Ergebnisse sind in Tabelle 5.24 angegeben.

<i>N</i>	<i>M</i>	<i>#O</i>	<i>MaxDist</i>	<i>%InMax</i>
4	5	5,9000	3,6000	0,6833
4	10	4,1000	3,4000	0,6458
4	20	7,8000	4,3000	0,7875
4	\emptyset	5,9333	3,7667	0,7056
6	5	8,0000	7,6000	0,5656
6	10	56,3000	7,7000	0,5343
6	20	67,7000	9,9000	0,7222
6	\emptyset	44,0000	8,4000	0,6074
8	5	29,5000	13,4000	0,4924
8	10	7,7778	12,1111	0,3969
8	20	8,6000	12,2000	0,4329
8	\emptyset	15,5517	12,5862	0,4422

Tabelle 5.24: Integrierte Analyse - Alle globalen Optima

Die hohe Anzahl globaler Optima bei $N = 6$ läßt sich mit der hohen Anzahl globaler Optima bei der maximalen Terminüberschreitung bei $N = 6$ begründen (vgl. Tabelle 5.23). Insgesamt läßt sich beobachten, daß

⁶⁰Vgl. z.B. Reeves (1999, S.479-485).

⁶¹Damit werden globale Optima, die ein Optimum bezüglich mehrerer Ziele darstellen, nur einmal berücksichtigt.

der Anteil am Lösungsraum größer ist als die einzelnen Anteile bei der isolierten Analyse. Da die Werte in Tabelle 5.24 größer sind als die Werte bei der isolierten Analyse, liegen die globalen Optima der Ziele nicht alle im gleichen Bereich des Lösungsraums, sondern in unterschiedlichen Bereichen. Diese können sich aber durchaus überschneiden.

In Abschnitt 2.2.2 wurde eine Korrelationsanalyse der drei betrachteten Ziele durchgeführt (vgl. Tabelle 2.2 auf Seite 20) und bereits die Möglichkeit bereichsweiser Zielbeziehungen diskutiert. Die hier erhaltenen Ergebnisse lassen in Verbindung mit den Ergebnissen aus Abschnitt 2.2.2 die Vermutung zu, daß bereichsweise Zielbeziehungen vorliegen. Außerhalb des Bereichs im Lösungsraum, in dem die globalen Optima aller Ziele liegen, besteht Zielneutralität, während innerhalb des Bereichs Zielkonflikte auftreten.

Ein Verfahren, welches als Ziel die Ermittlung der Menge aller effizienten Lösungen $E(\Pi, z)$ hat, muß nicht nur globale Optima eines Zieles finden, sondern die bezüglich des Zielsystems effizienten globalen Optima aller Ziele. Daher ist es von Interesse, wie weit die effizienten globalen Optima der einzelnen Ziele voneinander entfernt liegen. In einem weiteren Experiment wurden daher die Mengen der globalen Optima der einzelnen Ziele vereinigt. Gleichzeitig wurden globale Optima, welche von anderen globalen Optima dominiert werden, entfernt. Es werden somit nur solche globalen Optima berücksichtigt, die auch Element der Menge aller effizienten Auftragsfolgen $E(\Pi, z)$ sind. Anschließend wurde erneut die echte Distanz von jedem dieser effizienten globalen Optima zu jedem anderen effizienten globalen Optimum bestimmt. Die Auswertungsgrößen beziehen sich nun also auf die effizienten globalen Optima aller drei Ziele. Das Ergebnis ist in Tabelle 5.25 angegeben.

Es zeigt sich, daß der Anteil am Lösungsraum, in dem die effizienten globalen Optima liegen, i.d.R. größer ist als die jeweiligen Anteile bei der isolierten Analyse. Bei $N = 4$ ist der Anteil jedoch kleiner als der, welcher bei der isolierten Analyse für die maximale Terminüberschreitung beobachtet wurde. Offensichtlich sind viele der multiplen globalen Optima, welche dort bei T_{max} beobachtet wurden, nicht effizient. Im Vergleich zum Fall, in dem alle globalen Optima betrachtet wurden, sind die Werte von $\%InMax$ nun kleiner.

Das Ergebnis bedeutet, daß sich die effizienten globalen Optima der einzelnen Ziele in unterschiedlichen Bereichen des Lösungsraums befinden, die sich jedoch weiterhin überschneiden können. Ein Verfahren, welches die globalen Optima aller Ziele finden möchte, muß somit einen deutlich

N	M	#O	MaxDist	%InMax
4	5	2,4000	2,3000	0,4542
4	10	2,1000	2,4000	0,4792
4	20	2,4000	2,8000	0,5625
4	∅	2,3000	2,5000	0,4986
6	5	2,8000	5,7000	0,3638
6	10	2,6000	6,1000	0,4335
6	20	4,4000	8,1000	0,6368
6	∅	3,2667	6,6333	0,4780
8	5	3,5000	10,0000	0,2779
8	10	2,5556	9,4444	0,2374
8	20	2,8000	10,7000	0,3142
8	∅	2,9655	10,0690	0,2778

Tabelle 5.25: Integrierte Analyse - Effiziente globale Optima

größeren Bereich bzw. verschiedene Bereiche des Lösungsraums absuchen.

5.5.3 Analyse der Distanz der Elemente der Menge aller effizienten Lösungen

Aus den Experimenten des letzten Abschnitts sind die Distanzen zwischen den effizienten globalen Optima der Ziele, und damit den Auftragsfolgen, bekannt, welche die Extrempunkte der effizienten Front bestimmen. Es fehlen jedoch Informationen über die restlichen Elemente der Menge aller effizienten Lösungen, welche kein globales Optimum bzgl. eines der drei Ziele darstellen und deren Zielfunktionsvektoren daher zwischen den Extrempunkten der effizienten Front liegen. Insbesondere ist dabei von Interesse, ob diese Elemente alle innerhalb des Bereichs des Lösungsraum liegen, in dem auch die effizienten globalen Optima der einzelnen Ziele liegen. Um dies zu untersuchen, wird für jedes Problem die durchschnittliche Distanz von jedem Element aus $E(\Pi, z)$ zu jedem anderen Element aus $E(\Pi, z)$ bestimmt.⁶² Die Auswertungsgrößen beziehen sich auf alle Elemente der Menge aller effizienten Lösungen $E(\Pi, z)$. Die Ergebnisse sind in Tabelle 5.26 angegeben.

Wie zu erkennen ist, steigen die Werte von *MaxDist* im Vergleich zu den Werten in Tabelle 5.25 an. Ebenso steigt *%InMax*. Somit liegen einige Elemente der Menge aller effizienten Lösungen außerhalb des Bereichs, in dem die effizienten globalen Optima der einzelnen Ziele liegen. Sinken die Werte von *%InMax* in Tabelle 5.25 mit steigender Auftragszahl, so ist diese Entwicklung nun nicht mehr zu beobachten.

⁶²Es wurde erneut die echte Distanz auf Basis des API-Operators bestimmt.

N	M	$ E(\Pi, z) $	MaxDist	%InMax
4	5	3,1000	2,4000	0,4792
4	10	3,7000	3,1000	0,5750
4	20	3,4000	3,1000	0,6208
4	\emptyset	3,4000	2,8667	0,5583
6	5	6,9000	7,9000	0,5771
6	10	4,5000	6,5000	0,4561
6	20	9,4000	9,0000	0,7064
6	\emptyset	6,9333	7,8000	0,5799
8	5	14,2000	12,0000	0,4394
8	10	12,3333	13,3333	0,5312
8	20	18,6000	15,8000	0,6498
8	\emptyset	15,1379	13,7241	0,5404

Tabelle 5.26: Distanzanalyse der effizienten Menge

5.6 Zusammenfassung

In diesem Kapitel wurde zunächst überprüft, ob oder wie häufig ein Nachbar im Lösungsraum auch ein Nachbar im Zielraum ist. Dazu war es zunächst notwendig, Konzepte für die Nachbarschaft im Zielraum zu entwickeln. Verschiedene experimentelle Analysen haben anschließend ergeben, daß ein Nachbar im Lösungsraum nur selten auch ein Nachbar im Zielraum ist. Für die API-Nachbarschaft ist die Wahrscheinlichkeit dafür noch am höchsten.

Anschließend wurde in diesem Abschnitt die Qualität von Maßen zur Approximation der echten Distanz zwischen zwei Auftragsfolgen untersucht. Dazu wurde ein rekursiver Algorithmus zur Berechnung der echten Distanz vorgeschlagen. Die Analyse der Differenzierungstiefe der Approximationsmaße und der Korrelation mit der echten Distanz hat gezeigt, daß keines der untersuchten Ersatzmaße in der Lage ist, eine gute Abschätzung der echten Distanz zu ermitteln.

Auf der Grundlage der echten Distanz wurden, für bzgl. der Anzahl der Aufträge kleine Probleme mit bis zu acht Aufträgen, verschiedene Experimente durchgeführt. Zuerst wurde die Korrelation zwischen der echten Distanz und den Zielfunktionsvektoren aller Elemente des Lösungsraum ermittelt. Die Ergebnisse laufen der Vorstellung der Existenz eines "Großen Tals" entgegen. Die im Anschluß vorgenommenen Distanzanalysen zwischen den globalen Optima der Ziele haben gezeigt, daß die globalen Optima eines Ziels in einem begrenzten Bereich des Lösungsraums liegen. Die Bereiche der einzelnen Ziele sind jedoch nicht deckungsgleich. Wird die Definition des "Großen Tals" dahingehend angepaßt, daß es vorliegt, falls

die globalen Optima eines Ziels in einem begrenzten Bereich des Lösungsraums zusammenliegen, so unterstützt dieses Ergebnis seine Existenz. Der zu untersuchende Bereich des Lösungsraums wächst, wenn alle Elemente der Menge aller effizienten Lösungen betrachtet werden.

Für einen Ansatz, welcher die Menge aller effizienten Lösungen bestimmen soll, bedeuten diese Resultate, daß unterschiedliche Bereiche des Lösungsraums ausgewertet werden müssen. Gleiches gilt für Heuristiken zur Bestimmung einer heuristisch effizienten Menge. Im Unterschied zu Ansätzen für nur ein Ziel oder a-priori Ansätzen müssen systematisch unterschiedliche Bereiche des Lösungsraums angesteuert und ausgewertet werden. Dies bedeutet insbesondere, daß die Heuristik Informationen über vielversprechende Bereiche des Lösungsraums wieder vergessen können muß. Diese Informationen werden bei ACO-Algorithmen in den Pheromoninformationen gespeichert. In den im folgenden Kapitel vorgestellten Ameisenalgorithmen ist diese Anforderung das wesentliche Argument für die wiederholte Durchführung von Neuinitialisierungen der Pheromoninformationen.

Kapitel 6

ACO Algorithmen für das multikriterielle PFSP

6.1 Grundlagen der ACO Metaheuristik

6.1.1 Die Natur als Vorbild

Neben Genetischen Algorithmen und dem Simulated Annealing gehört auch die Ant Colony Optimization (ACO) Metaheuristik zu einer Gruppe von Verfahren, welche einen natürlichen Prozeß als Vorbild hat.¹ Beim Simulated Annealing ist dieses Vorbild der physikalische Abkühlungsprozeß von Stoffen.² Genetische Algorithmen hingegen stellen eine Analogie des Vererbungsprozesses durch Crossover und Mutation sowie des Prinzips des 'Survival of the fittest' dar.³ Die ACO-Metaheuristik ist durch die Selbstorganisation von Ameisen bei der Futtersuche inspiriert.

Wie auch Termiten und einige Arten von Bienen und Wespen gehören Ameisen zur Gruppe der sozialen Insekten.⁴ Diese Gruppe von Insekten zeigt ein kollektives Problemlösungsverhalten, welches durch Selbstorganisation erreicht wird.⁵ Ein Beispiel für ein solches Problemlösungsverhalten läßt sich bei der Futtersuche der argentinischen Ameise (*Iridomyrmex humilis*) beobachten. Diese Ameisen sind in der Lage, den kürzesten Weg vom Nest zur Futterquelle zu identifizieren. Beobachtet und untersucht wurde dieses Verhalten unter anderem im sogenannten Doppelbrückenex-

¹In Colorni u. a. (1996, S.3) werden weiterhin Sampling & Clustering Ansätze, Tabu Search und Neuronale Netze zu dieser Gruppe von Verfahren gezählt.

²Vgl. Krikpatrick u. a. (1983, S.672f.) sowie Laarhoven, van u. Aarts (1987, S.7-15).

³Vgl. Colorni u. a. (1996, S.4).

⁴Vgl. Bonabeau u. a. (1999, S.1).

⁵Vgl. Bonabeau u. a. (1999, S.6).

periment von Goss u. a. (1989).⁶ Die Ergebnisse ihrer Arbeit sind Anstoß für die Entwicklung des ersten Ameisenalgorithmus durch Dorigo u. a. (1991b) sowie Dorigo u. a. (1991a) gewesen.⁷ Im Folgenden soll das Experiment von Goss u. a. (1989) und der Kommunikationsmechanismus, mit dessen Hilfe die Ameisen den kürzesten Weg finden, kurz dargestellt werden.

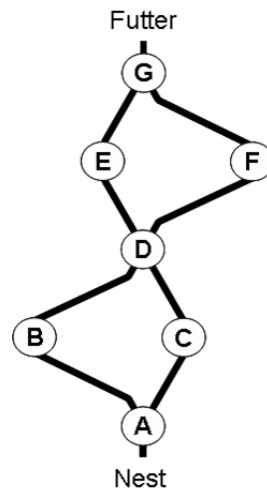


Abbildung 6.1: Doppelbrückenexperiment von Goss u. a. (1989)

Der Versuchsaufbau ist in Abbildung 6.1 dargestellt. Auf dem Weg von ihrem Nest zum Futterplatz haben die Ameisen zweimal die Wahl zwischen einem längeren und einem kürzeren Wegabschnitt. Zu Beginn des Versuchs haben die Ameisen alle Wegabschnitte gleichmäßig genutzt. Nach einiger Zeit haben jedoch die Ameisen alle den jeweils kürzeren Wegabschnitt gewählt, sind also der Strecke A,C,D,E,G gefolgt.⁸

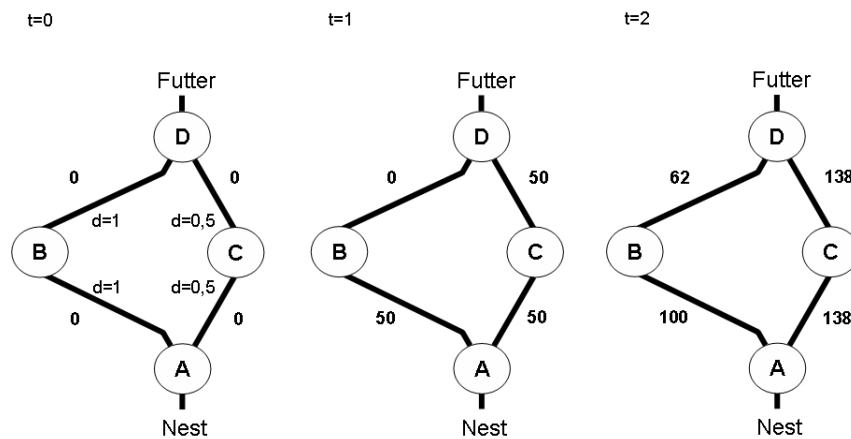
Die Beobachtung wird von Goss u. a. (1989) über einen indirekten Kommunikationsmechanismus mit Hilfe von Pheromonen erklärt, welche die Ameisen während der Wegsuche ablegen. In Abwesenheit von Pheromonen treffen die Ameisen ihre Wegentscheidungen zufällig. Sind jedoch Pheromoninformationen vorhanden, so beeinflussen diese die Entscheidung. Ein Weg, der eine höhere Pheromonkonzentration aufweist, hat eine höhere Wahrscheinlichkeit, von der Ameise ausgewählt zu werden als ein Weg mit einer niedrigeren Konzentration. Das Resultat ist ein autokatalytischer

⁶Vgl. Goss u. a. (1989, 579-581).

⁷Vgl. Dorigo u. a. (1991b, S.1-20), Dorigo u. a. (1991a, S.1-20). Dorigo hat sich auch im Rahmen seiner Dissertation mit Ameisenalgorithmen beschäftigt (vgl. Dorigo (1992)). Die Dissertation liegt nach Kenntnisstand des Verfassers nur in italienischer Sprache vor.

⁸Vgl. zu den bisherigen Ausführungen Goss u. a. (1989, S.579-581). Im Experiment wird das Verhältnis der Länge des längeren Wegabschnitts zur Länge des kürzeren Wegabschnitts variiert.

Prozeß, der die Pheromonkonzentration auf den kürzeren Wegabschnitten immer weiter ansteigen läßt, bis schließlich alle Ameisen diese Wegabschnitte wählen.⁹ Das folgende Zahlenbeispiel¹⁰ (Abbildung 6.2) soll den Prozeß der Selbstorganisation und seine Charakteristika verdeutlichen.



Annahmen:

- ▷ Zu jedem Zeitpunkt t starten 100 Ameisen vom Nest zur Futtersuche.
- ▷ Die Ameisen können in einer Zeiteinheit eine Distanz d von 1 zurücklegen.
- ▷ Jede Ameise legt eine Pheromonmenge von 1 auf dem zurückgelegten Weg ab.
- ▷ Die Verdunstung der Pheromone wird nicht berücksichtigt.

Abbildung 6.2: Indirekter Kommunikationsmechanismus

t=0: 100 Ameisen starten die Futtersuche im Nest und erreichen Punkt A. Aufgrund des Fehlens einer Pheromoninformation entscheiden sie sich zufällig für eine Richtung. Es gehen daher 50% über Punkt B und 50% über Punkt C.

t=1: Die 50 Ameisen, die über Punkt C gegangen sind, erreichen Punkt D und damit die Futterquelle. Die andere Gruppe von Ameisen erreicht erst Punkt B, da dieser Weg doppelt so lang ist. Auf dem Rückweg vom Futterplatz müssen sich die Ameisen an Punkt D erneut entscheiden. Nun existiert bereits eine Pheromonspur auf dem Weg in Richtung Punkt C, aber noch keine in Richtung Punkt B. Ein größerer Teil der 50 Ameisen wird sich daher für den Weg über Punkt C entscheiden. In diesem Beispiel werden 76% angenommen, also 38 Ameisen.

Gleichzeitig starten 100 neue Ameisen vom Nest zum Futterplatz. Beide Wege haben eine Pheromonmenge von 50, so daß sich erneut 50 Ameisen für den Weg über Punkt B und 50 Ameisen für den Weg über Punkt C entscheiden.

t=2: Die ersten 38 Ameisen mit Futter erreichen das Nest, gleichzeitig erreichen die 50 Ameisen, welche im Zeitpunkt $t=0$ den Weg über B gewählt haben,

⁹Vgl. zu diesen Ausführungen Goss u. a. (1989, 579f.).

¹⁰Das Beispiel wurde in Anlehnung an die Darstellung in Dorigo u. a. (1996, S.4) formuliert.

den Futterplatz. Auf dem Rückweg wird ein größerer Teil von ihnen, ca. 69%, über Punkt C zum Nest zurückkehren. Gleichzeitig wird ab diesem Zeitpunkt ein größerer Teil der Ameisen, ca. 57%, welche im Nest starten, ebenfalls den Weg über C wählen.

t=... Mit fortschreitender Zeit wird der Weg über C eine relativ immer höhere Pheromonkonzentration erhalten, so daß ihm immer mehr Ameisen folgen werden.

An dem Beispiel lassen sich die vier von Bonabeau u. a. (1999) angeführten Merkmale der Selbstorganisation, die sich auch in der ACO-Metaheuristik wiederfinden, verdeutlichen.¹¹ Zunächst zeichnet sich der Selbstorganisationsprozeß durch das Prinzip des positiven Feedback aus. Je häufiger ein Wegabschnitt von Ameisen benutzt wird, desto höher ist die Pheromonkonzentration, und desto häufiger wird er daher von nachfolgenden Ameisen verwendet. Dieser Prozeß der Selbstverstärkung wurde von Dorigo u. a. (1991b) auch als autokatalytischer Prozeß bezeichnet.¹² Pheromone verdunsten mit der Zeit.¹³ Dieser Prozeß wirkt dem Ablegen von Pheromonen durch die Ameisen entgegen. Ein solcher, dem positiven Feedback entgegenwirkender Prozeß wird auch negatives Feedback genannt und stellt das zweite Charakteristikum des Selbstorganisationsprozesses dar.¹⁴

Des weiteren hat der Prozeß der Selbstorganisation einen stochastischen Charakter.¹⁵ So entscheidet eine einzelne Ameise zufällig. Die Pheromonwerte beeinflussen lediglich die Auswahlwahrscheinlichkeiten, schalten aber die zufällige Entscheidung nicht vollständig ab. So ist es möglich, daß eine Ameise trotz extrem hoher Pheromonwerte für einen Weg einem anderen Weg folgt.

Als viertes Merkmal führt Bonabeau u. a. (1999) die multiple Interaktion auf. Interaktion oder Kommunikation zwischen den Mitgliedern der Kolonie kann direkt, z.B. durch Berührung, oder aber indirekt erfolgen. Indirekte Kommunikation findet über die Veränderung der Umwelt der Individuen statt. So verändert ein Individuum den Umweltzustand, ein anderes Individuum wertet den Umweltzustand aus und reagiert darauf.¹⁶

¹¹Vgl. Bonabeau u. a. (1999, S.9-12 und S.41).

¹²Vgl. hierzu und den folgenden Satz Dorigo u. a. (1991b, S.1), welcher auf Dorigo (1992) verweist.

¹³Dieser Prozeß wurde aus Übersichtlichkeitsgründen im Zahlenbeispiel nicht berücksichtigt.

¹⁴Vgl. Bonabeau u. a. (1999, S.10). Das negative Feedback ist notwendig, da nicht beschränkte autokatalytische Prozesse explodieren (vgl. Dorigo u. a. (1991b, S.1) oder Dorigo u. a. (1991a, S.1)).

¹⁵Vgl. Bonabeau u. a. (1999, S.10).

¹⁶Vgl. zur direkten und indirekten Kommunikation Bonabeau u. a. (1999, S.14).

Im Beispiel nutzen die Ameisen eine indirekte Kommunikation. Eine Ameise verändert ihre Umwelt durch das Ablegen von Pheromonen. Andere Ameisen erfassen die Pheromonkonzentration und berücksichtigen diese Information in ihrer Handlung. Des weiteren kann eine Ameise sowohl Pheromone ablegen als auch Pheromonkonzentrationen auswerten.

In den folgenden beiden Abschnitten wird dargestellt, wie die Funktionalität und Eigenschaften des Verhaltens der Ameisen zur Lösung kombinatorischer Optimierungsprobleme verwendet werden können. Dazu wird zunächst allgemein die ACO-Metaheuristik definiert und ihre Eigenschaften dargestellt. Anschließend werden für das Permutation Flow Shop Problem konkrete Ausgestaltungsmöglichkeiten aufgezeigt.

6.1.2 Definition und Eigenschaften der ACO-Metaheuristik

Damit ein Ameisenalgorithmus als Instanz der ACO-Metaheuristik klassifiziert werden kann, muß er die in Dorigo u. Di Caro (1999) und Dorigo u. a. (1999) beschriebenen Eigenschaften aufweisen.¹⁷ Diese sind sehr allgemein gehalten. Die wichtigsten und zentralen Eigenschaften lassen sich wie folgt zusammenfassen:¹⁸

- Der Algorithmus verwendet eine Kolonie künstlicher Ameisen.
- Die Ameisen nutzen eine indirekte Kommunikation in Form künstlicher Pheromone.
- Die einzelnen Ameisen konstruieren sukzessive eine Lösung.
- Die einzelnen Entscheidungen der Ameisen sind stochastisch und werden von lokal verfügbaren Informationen in Form künstlicher Pheromone beeinflusst. Des weiteren können optional heuristische Werte berücksichtigt werden.
- Jede Ameise hat ein Gedächtnis, in der sie die bisher in der Lösung enthaltenen Elemente speichert. Es wird dazu genutzt, die konstruierte Lösung zu bewerten, die Zulässigkeit der Lösung sicherzustellen und nach der Konstruktion der Lösung Pheromon abzulegen.
- Die Ameisen modifizieren die Pheromoninformation durch das Ablegen von Pheromonen und beeinflussen damit die Entscheidungen anderer Ameisen. Es sind zwei Strategien des Pheromonablegens möglich:

¹⁷Vgl. Dorigo u. Di Caro (1999, S.14-16) sowie Dorigo u. a. (1999, S.143-145).

¹⁸Vgl. Dorigo u. Di Caro (1999, S.16) sowie Merkle (2002, S.4).

- **Online Step-by-Step Pheromone Update:** Das Ablegen der Pheromone findet während der Lösungskonstruktion statt, d.h. unmittelbar nach dem Einfügen eines neuen Lösungselements.
- **Online Delayed Pheromone Update:** Das Ablegen der Pheromone findet erst nach der vollständigen Konstruktion einer Lösung statt.

- Neben dem Ablegen von Pheromon wird das Verdunsten von Pheromonen als eine von den Ameisen unabhängige Funktion modelliert.

Die künstlichen Ameisen der ACO-Metaheuristik weisen im Vergleich mit natürlichen Ameisen Gemeinsamkeiten, aber auch Unterschiede auf. Künstliche Ameisen haben ein Gedächtnis und können sich damit die bereits in die Lösung eingefügten Elemente merken, was die Strategie des Online Delayed Pheromone Updates ermöglicht. Die Menge des abgelegten Pheromons ist dabei in der Regel abhängig von der Lösungsqualität der konstruierten Lösung. Somit werden globale Informationen direkt in der Pheromoninformation gespeichert. Schließlich ist das Zeitumfeld der Ameisen in der ACO-Metaheuristik nicht stetig, sondern diskret.¹⁹

Weiterhin können in den Algorithmus Funktionen eingebaut werden, die sich in einer natürlichen Kolonie nicht finden. Dazu gehören die Möglichkeiten, lokale Suchverfahren aufzurufen oder auch Vorausschau oder Backtracking.²⁰ Zu diesen Funktionen gehört auch die Berücksichtigung von globalen Informationen durch die Ameisen. So ist das Ablegen einer von der Lösungsqualität abhängigen Pheromonmenge eine Berücksichtigung globaler Information. Ebenso können globale Informationen über heuristische Werte zusätzlich zu den Pheromonwerten bei der Auswahl des nächsten in die Lösung einzufügenden Elements berücksichtigt werden.

Im Algorithmus 2 ist die Basisstruktur eines ACO-Ansatzes wiedergegeben.

Zeile 1: Zunächst wird die Pheromonmatrix T initialisiert.²¹

Zeilen 2+14: Die Do-While-Schleife wird solange durchlaufen, bis die Abbruchbedingung erfüllt ist.

Zeilen 3+12: In dieser For-Schleife konstruiert jede Ameise *ant* der Kolonie eine Lösung. In der Kolonie gibt es insgesamt *Ants* Ameisen (Größe der Kolonie).

¹⁹Vgl. Dorigo u. a. (1999, S.141-143).

²⁰Vgl. Dorigo u. Di Caro (1999, S.17) und Dorigo u. a. (1999, S.142).

²¹Die Ausgestaltungsmöglichkeiten der Pheromonmatrix T werden in Abschnitt 6.1.3.1 betrachtet.

Algorithmus 2 Basisstruktur eines ACO-Algorithmus

```

1: Initialisiere Pheromonmatrix  $T$ 
2: do {
3:   for(  $ant=1$ ;  $ant \leq Ants$ ;  $ant=ant+1$  ) {
4:     Initialisiere Menge der nicht ausgewählten
       Elemente:  $U=\{1, 2, \dots, N\}$ ;
5:     Initialisiere aktuelle Teillösung:  $\tilde{\pi} = \emptyset$ ;
6:     while(  $U \neq \emptyset$  ) {
7:       Wähle ein Element  $n$  aus  $U$  aus;
8:       Füge  $n$  in  $\tilde{\pi}$  ein;
9:        $U=U \setminus \{n\}$ ;
10:    } // end while
11:    Führe lokale Suche aus // Optional
12:  } // end for
13:  Aktualisiere Pheromonmatrix  $T$ ;
14: } while( !Abbruchbedingung )

```

Zeile 4: Bevor eine Ameise ant eine Lösung konstruieren kann, wird die Menge U der (zulässigen) noch in die Lösung aufzunehmenden Elemente initialisiert. Im Fall des PFSP sind dies alle N Aufträge.

Zeile 5: Es wird die aktuelle Teillösung $\tilde{\pi}$ initialisiert. Da die Ameise ant noch nicht mit der Lösungskonstruktion begonnen hat, ist die aktuelle Teillösung zunächst leer. Im Fall des PFSP kann $\tilde{\pi}$ als aktuelle Teilauftragsfolge bezeichnet werden.

Zeilen 6+10: In diese While-Schleife konstruiert die Ameise ant eine Lösung. Dabei fügt sie solange Elemente in $\tilde{\pi}$ ein, bis alle Elemente aus U berücksichtigt sind, die Menge U also leer ist.

Zeile 7: Während der Lösungskonstruktion wählt die Ameise ant zunächst ein Element aus der Menge U aus.²²

Zeile 8: Das ausgewählte Element wird anschließend in die aktuelle Teillösung $\tilde{\pi}$ eingefügt.

Zeile 9: Schließlich ist das ausgewählte Element aus U zu entfernen, da es kein weiteres Mal ausgewählt und in die Lösung eingefügt werden darf.

Zeile 11: Nach Abschluß der Lösungskonstruktion durch Ameise ant ist es optional möglich, die konstruierte Lösung durch ein lokales Suchverfahren zu verbessern.

²²In Abschnitt 6.1.3.3 werden unterschiedliche Möglichkeiten aufgezeigt, wie beim PFSP Aufträge ausgewählt werden können.

Zeile 13: Nach der Konstruktion der Auftragsfolgen durch die Ameisen wird die Pheromoninformation aktualisiert.²³

Wie auch andere Metaheuristiken ist die ACO-Metaheuristik robust. Das bedeutet, sie kann auf verschiedene kombinatorische Optimierungsprobleme übertragen werden. Neben dem Traveling Salesman Problem, dem Quadratischen Zuordnungsproblem oder dem Tourenplanungsproblem gibt es für viele weitere Problemtypen ACO-Algorithmen. Einen guten Überblick geben Dorigo u. Di Caro (1999).²⁴ Auch für die von Dorigo u. Di Caro (1999), mit Ausnahme der SMTTP Probleme, nicht aufgeführten Scheduling Probleme gibt es mittlerweile eine Reihe von Ansätzen. Tabelle 6.1 gibt einen Überblick.

Problem	Jahr	Arbeit
JSP	1994	Colorni u. a. (1994)
JSP	1996	Dorigo u. a. (1996)
JSP	2001	Teich u. a. (2001)
PFSP	1998	Stützle (1998a)
PFSP	2002	T'kindt u. a. (2002)
PFSP	2002	Rajendran u. Ziegler (2002)
PFSP	2004	Rajendran u. Ziegler (2004)
PFSP	2004	Ying u. Liao (2004)
SMTTP	1999	Bauer u. a. (1999)
SMTWTP	2000	den Besten u. a. (2000)
SMTTP / SMTWTP	2000	Merkle u. Middendorf (2000)
SMTTP	2001	Gagne u. a. (2001)
SMTTP	2002	Gagne u. a. (2002)
SMTWTP	2003	Merkle u. Middendorf (2003)
SMTEMDP	2001	Merkle u. Middendorf (2002)
SMTWDP	2001	Merkle u. Middendorf (2001)
Single Machine Problem	2001	Iredi u. a. (2001)
Single Machine Problem	2002	Gravel u. a. (2002)

SMTEMDP: Single Machine Total Earliness with Multiple Due Dates Problem
 SMTWDP: Single Machine Total Weighted Deviation Problem

Tabelle 6.1: ACO Algorithmen für Scheduling Probleme

Um ein Problem mit Hilfe der ACO-Metaheuristik zu lösen, müssen folgende Designaufgaben gelöst werden.²⁵ Zunächst muß das Problem in

²³Vgl. zu unterschiedlichen Strategien der Pheromonaktualisierung auch Abschnitt 6.1.3.2 sowie Abschnitt 6.1.3.4.

²⁴Vgl. Dorigo u. Di Caro (1999, S.30).

²⁵Vgl. Dorigo u. a. (1991b, S.18).

Form eines Graphen repräsentiert werden, der dann von den Ameisen abgesucht werden kann.²⁶ Des Weiteren ist die Konstruktionsvorschrift für eine Lösung festzulegen sowie das Gedächtnis der Ameise zu modellieren. Abschließend gilt es, die Regeln für das Ablegen von Pheromon durch die Ameisen sowie die Verdunstung festzulegen und damit den autokatalytischen Prozeß zu modellieren. Im folgenden Abschnitt werden konkrete Ausgestaltungsmöglichkeiten der einzelnen Aspekte für das PFSP dargestellt.

6.1.3 Ausgestaltungsmöglichkeiten für das PFSP

6.1.3.1 Die Pheromonmatrix T

Die Pheromonmatrix ist das Langzeitgedächtnis der Heuristik und speichert die von den künstlichen Ameisen gelegte Pheromonspur.²⁷ Sie ist die Basis der Lösungskonstruktion durch eine einzelne Ameise. Daher muß zunächst für jedes kombinatorische Optimierungsproblem eine geeignete Form der Repräsentation in einer Pheromonmatrix gefunden werden. Für das Permutation Flow Shop Problem sollen zwei Möglichkeiten vorgestellt werden.

$i \setminus j$	1	...	j	...	N		$n \setminus h$	1	...	h	...	N	
1							1						
⋮							⋮						
i	τ_{ij}					n	τ_{nh}						
⋮							⋮						
N	Wie vorteilhaft ist es, Auftrag j direkt nach Auftrag i zu fertigen						N	Wie vorteilhaft ist es, Auftrag n auf Position h zu platzieren					

Abbildung 6.3: Kodierungsalternativen der Pheromonmatrix

Aufgrund der Ähnlichkeit mit dem Traveling Salesman Problem besteht eine Möglichkeit darin, die dort übliche Kodierung zu verwenden. Die Spalten und Zeilen enthalten die N Aufträge. Der Pheromonwert im Schnittpunkt gibt an, wie vorteilhaft es ist, Auftrag j unmittelbar nach Auftrag i zu fertigen (TSP-Kodierung). Für das Permutation Flow Shop Problem und Single Machine Problem hat sich jedoch eine alternative Kodierung durchgesetzt. In der Pheromonmatrix ist kodiert, wie vorteilhaft es ist, Auftrag n auf Position h der Auftragsfolge zu platzieren (Positionskodierung). Im

²⁶Vgl. zu den formalen Anforderungen an Probleme zur Repräsentation als Graph für ACO-Algorithmen Dorigo u. Di Caro (1999, S.14f.).

²⁷Vgl. Dorigo u. Di Caro (1999, S.15).

Folgenden beziehen sich alle Darstellungen auf diese zweite Form der Pheromonkodierung.

6.1.3.2 Anzahl der Ameisen *Ants* in einer Kolonie

Eine Kolonie kann aus einer oder mehreren Ameisen bestehen. Die Anzahl an Ameisen in einer Kolonie wird im Folgenden mit *Ants* bezeichnet. Gibt es in einer Kolonie mehr als eine Ameise ($Ants > 1$), so ist es denkbar, daß die einzelnen Ameisen in einer Iteration nacheinander oder parallel eine Lösung konstruieren. Im letzten Fall verwenden alle Ameisen die gleiche Pheromoninformation. Im ersten Fall ist es möglich, daß jede Ameise andere Pheromoninformationen verwendet. Dies hängt davon ab, wann die Pheromonmatrix T aktualisiert wird. Legt jede Ameise unmittelbar nach Abschluß der Lösungskonstruktion Pheromon ab, so verwenden die Ameisen einer Kolonie jeweils unterschiedliche Pheromoninformationen. Wird die Pheromonmatrix dagegen erst nach Abschluß der Lösungskonstruktion aller $ants = 1(1)Ants$ Ameisen aktualisiert, so verwenden die Ameisen in einer Iteration hingegen die gleichen Pheromoninformationen. Die Aktualisierung der Pheromoninformation sowie die Beurteilung von Elitestrategien wird in Abschnitt 6.1.3.4 beschreiben und diskutiert.

6.1.3.3 Konstruktion einer Auftragsfolge

Die einzelne Ameise konstruiert eine Auftragsfolge durch sukzessives Auswählen und Einfügen von Aufträgen in die aktuelle Teilauftragsfolge $\tilde{\pi}$.²⁸ Die Auswahl eines Auftrags ist dabei ein stochastisches Element im Ameisenalgorithmus. Aufgrund der in der Pheromonmatrix gespeicherten Informationen wird für jeden Auftrag eine Auswahlwahrscheinlichkeit berechnet. Viele Ansätze von ACO-Algorithmen unterscheiden sich unter anderem auch darin, wie diese Auswahlwahrscheinlichkeiten berechnet und anschließend die Auswahl von Aufträgen vorgenommen wird.

Die im als Ant System²⁹ bekannten ACO-Ansatz vorgeschlagene Berechnung der Auswahlwahrscheinlichkeiten für die Platzierung von Aufträgen auf Position h erfolgt gemäß der folgenden Gleichung:³⁰

²⁸Die aktuelle Teilauftragsfolge $\tilde{\pi}$ besteht aus den während der Lösungskonstruktion durch eine Ameise bereits ausgewählten Aufträgen. Vgl. auch Algorithmus 2 auf Seite 153 und die dazu angegebenen Erläuterungen.

²⁹Vgl. Dorigo u. a. (1991a, S.2-4), Dorigo u. a. (1991b, S.2-4), Dorigo u. a. (1996, S.5-8).

³⁰Vgl. Dorigo u. a. (1991a, S.4), Dorigo u. a. (1991b, S.4), Dorigo u. a. (1996, S.6). Die Autoren betrachten ein TSP und verwenden daher die TSP-Kodierung der Pheromonmatrix. Die Kodierung wurde hier auf die Positionskodierung angepaßt.

$$p_{nh} = \begin{cases} \frac{[\tau_{nh}]^\alpha \cdot [\eta_{nh}]^\beta}{\sum_{n \in U} [\tau_{nh}]^\alpha \cdot [\eta_{nh}]^\beta} & \text{falls } n \in U \\ 0 & \text{sonst} \end{cases} \quad (6.1)$$

τ_{nh} ist dabei der Pheromonwert, der angibt, wie vorteilhaft es ist, Auftrag n auf Position h zu plazieren. η_{nh} ist eine lokale Komponente. Mit den Parametern α und β kann die relative Bedeutung von Pheromoninformation und lokaler Komponente gesteuert werden.³¹ U ist die Menge der zulässigen, noch nicht in die Auftragsfolge eingefügten Elemente. Da in dem in dieser Arbeit betrachteten Modell keine speziellen Restriktionen (z.B. Reihenfolgebeziehungen zwischen den Aufträgen) berücksichtigt werden, sind alle noch nicht in der aktuellen Teilauftragsfolge $\tilde{\pi}$ enthaltenen Aufträge Element der Menge U ($U = \{1, 2, \dots, N\} \setminus \tilde{\pi}$).

Stützle (1998a) verwendet in seiner Auswahlvorschrift zur Konstruktion von Lösungen des Permutation Flow Shop Problems Elemente des Ant Systems sowie des als Ant Colony Systems³² bekannten ACO-Ansatzes.³³ Es gibt zwei mögliche Vorgehensweisen, mit denen ein Auftrag aus U ausgewählt werden kann. Welche dieser beiden Möglichkeiten verwendet wird, wird vor der Auswahl jedes Auftrags zufällig bestimmt. Dazu wird zunächst vor jeder Auswahlentscheidung eine gleichverteilte Zufallszahl u im Intervall $[0; 1]$ bestimmt und mit dem beim Start der Heuristik extern vorgegebenen Parameter p_0 ($0 \leq p_0 \leq 1$) verglichen. Es sind nun zwei Fälle möglich:

Fall 1: $u \leq p_0$: Mit einer Wahrscheinlichkeit von p_0 wird derjenige Auftrag n aus U auf Position h der Auftragsfolge plaziert, der den größten Pheromonwert für diese Position aufweist, also:

$$n \in \operatorname{argmax} \{ \tau_{nh} \mid n \in U \} \quad (6.2)$$

Fall 2: $u > p_0$: Mit der Gegenwahrscheinlichkeit von $1 - p_0$ wird ein Auftrag mit der Wahrscheinlichkeit p_{nh} ausgewählt. Die Wahrscheinlichkeit p_{nh} errechnet sich wie folgt:

$$p_{nh} = \begin{cases} \frac{\tau_{nh}}{\sum_{n \in U} \tau_{nh}} & \text{falls } n \in U \\ 0 & \text{sonst} \end{cases} \quad (6.3)$$

³¹Vgl. Dorigo u. a. (1991b, S.3f.).

³²Vgl. zum Ant Colony System Dorigo u. Gambardella (1997a, S.74-77), Dorigo u. Gambardella (1997b, S.55f.) sowie Gambardella u. Dorigo (1996, S.622-624).

³³Vgl. Stützle (1998a, S.1561f.).

Es wird im Fall $u > p_0$ also die lokale Pheromonauswertung³⁴ analog zum Ant System verwendet. Allerdings ist die Auswahlwahrscheinlichkeit eines Auftrags nun alleine vom Pheromonwert τ_{nh} abhängig, eine lokale Komponente η_{nh} , wie in Gleichung 6.1, wird nicht berücksichtigt.

Das Vorgehen, nicht jeden Auftrag mit Hilfe der lokalen Pheromonauswertung auszuwählen, sondern mit einer Wahrscheinlichkeit von p_0 denjenigen mit dem größten Pheromonwert auszuwählen, ist unter dem Begriff der Pseudo Random Proportionate Rule im Ant Colony System eingeführt worden.³⁵ Zweck ist es, die bekannten Informationen über eine gute Auftragsfolge stärker zu berücksichtigen (exploitation) und nur mit einer bestimmten Wahrscheinlichkeit neue Bereiche des Lösungsraums durch Anwendung der lokalen Pheromonauswertung zu untersuchen (exploration).³⁶

Im Gegensatz zu Stützle (1998a) verwenden Rajendran u. Ziegler (2002) anstelle der lokalen Pheromonauswertung die Summen-Pheromonauswertung.³⁷ Diese wurde neben der relativen Pheromonauswertung in Merkle (2002) als Möglichkeit vorgestellt, die von ihm beobachtete Verzerrung der Auswahlwahrscheinlichkeiten zu reduzieren.³⁸ Diese bezeichnet für Permutation Flow Shop Probleme die Beobachtung, daß die tatsächliche Verteilung der Auswahlwahrscheinlichkeiten der Aufträge auf die Positionen der Auftragsfolge nicht der theoretischen Verteilung, die sich aus den Pheromonwerten errechnet, entspricht.³⁹ Insbesondere mit der Summen-Pheromonauswertung sind gute Ergebnisse erzielt worden.⁴⁰ Während bei der lokalen Pheromonauswertung die Platzierungsentscheidung über einen Auftrag allein abhängig vom Wert τ_{nh} ist, haben bei der Summen-Pheromonauswertung alle Pheromonwerte von der ersten bis zur aktuellen Position Einfluß auf die Auswahlwahrscheinlichkeit. Dazu werden diese Pheromonwerte aufsummiert.⁴¹

³⁴Der Begriff für diese Form der Berechnung der Auswahlwahrscheinlichkeiten wurde von Merkle (2002, S.25) geprägt, um sie von der in seiner Arbeit untersuchten globalen Pheromonauswertungen (Summen-Pheromonauswertung und relative Pheromonauswertung) abzugrenzen.

³⁵Vgl. Dorigo u. Gambardella (1997a, S.75f.), Dorigo u. Gambardella (1997b, S.55f.).

³⁶Vgl. Dorigo u. Gambardella (1997b, S.56).

³⁷Vgl. Stützle (1998a, S.1561) und Rajendran u. Ziegler (2002, S.733f.).

³⁸Vgl. Merkle (2002, S.26-51). Siehe auch Merkle u. Middendorf (2000, S.290), Merkle u. Middendorf (2002, S.328f.). Diese beiden Arbeiten enthalten Teile der in Merkle (2002, S.26-51) vorgestellten Ergebnisse.

³⁹Vgl. Merkle (2002, S.20).

⁴⁰Vgl. Merkle (2002, S.25).

⁴¹Vgl. Merkle (2002, S.27).

$$T_{nh} = \sum_{q=1}^h \tau_{nq} \quad (6.4)$$

Die Auswahlwahrscheinlichkeit eines Auftrags n für Position h errechnet sich dann als:⁴²

$$p_{nh} = \frac{T_{nh}}{\sum_{n \in U} T_{nh}} \quad (6.5)$$

Neben der Summen-Pheromonauswertung hat Merkle (2002) auch eine Kombination aus Summen-Pheromonauswertung und lokaler Pheromonauswertung sowie die relative Pheromonauswertung vorgestellt.⁴³ Des Weiteren hat er Experimente über die Reihenfolge, in der die Positionen der Auftragsfolge belegt werden (vorwärts, rückwärts, zufällig), durchgeführt.⁴⁴ Auf eine Darstellung der Ergebnisse wird hier verzichtet.

Rajendran u. Ziegler (2002) wählen die Aufträge nicht aus der Menge U aus, sondern verwenden bei der Auswahl von Aufträgen eine Kandidatenliste π^{cand} .⁴⁵ Diese ergibt sich aus den ersten $cand$ noch nicht ausgewählten Aufträgen der bisher besten bekannten Auftragsfolge der Kolonie (π^{glb}).

Schließlich erweitern Rajendran u. Ziegler (2002) im von ihnen als ACO2 bezeichneten Ansatz die Pseudo Random Proportionate Rule um eine weitere dritte Möglichkeit, Aufträge auszuwählen. Diese sieht vor, den ersten noch nicht ausgewählten Auftrag aus π^{glb} auszuwählen.⁴⁶ Dies entspricht dem ersten Element der Kandidatenliste π^{cand} . Diese Strategie wird im Folgenden als erweiterte Pseudo Random Proportionate Rule bezeichnet. Das Vorgehen bei der Lösungskonstruktion ist im Algorithmus 3 wiedergegeben.⁴⁷ Die Parameter p_1 und p_2 werden beim Start des Verfahrens extern vorgegeben.⁴⁸

Zeile 1: Bevor die Ameise eine Lösung konstruieren kann, wird eine Menge U der (zulässigen) noch in die Lösung aufzunehmenden Elemente initialisiert. Im Fall des PFSP sind dies alle N Aufträge.

Zeile 2: Es wird die aktuelle Teilauftragsfolge $\tilde{\pi}$ initialisiert. Da die Ameise noch nicht mit der Lösungskonstruktion begonnen hat, ist die aktuelle Teilauftragsfolge zunächst leer.

⁴²Vgl. Merkle (2002, S.27).

⁴³Vgl. Merkle (2002, S.25-51).

⁴⁴Vgl. Merkle (2002, S.53-76).

⁴⁵Vgl. Rajendran u. Ziegler (2002, S.733f.).

⁴⁶Vgl. Rajendran u. Ziegler (2002, S.734).

⁴⁷Vgl. Rajendran u. Ziegler (2002, S.734).

⁴⁸In Rajendran u. Ziegler (2002) wird p_1 auf den Wert 0,4 und p_2 auf den Wert 0,8 gesetzt (vgl. Rajendran u. Ziegler (2002, S.734)).

Algorithmus 3 Konstruktion einer Auftragsfolge in ACO2 von Rajendran u. Ziegler (2002)

```

1: Initialisiere Menge der nicht ausgewählten
   Elemente:  $U = \{1, 2, \dots, N\}$ ;
2: Initialisiere:  $\tilde{\pi} = \emptyset$ ;
3: for(  $h=1$ ;  $h \leq N$ ;  $h=h+1$  ) {
4:    $\pi^{cand} =$  Erste cand noch nicht ausgewählte Aufträge
      aus  $\pi^{glb}$ ;
5:    $u \leftarrow U[0;1]$ ;
6:   if(  $u \leq p_1$  )
7:      $n =$  Erste Auftrag aus  $\pi^{cand}$ 
8:   if(  $p_1 < u \leq p_2$  )
9:      $n \in \operatorname{argmax} \left\{ T_{nh} = \sum_{q=1}^h \tau_{nq} \mid n \in \pi^{cand} \right\}$ 
10:  if (  $u > p_2$  )
11:    wähle einen Auftrag  $n$  aus  $\pi^{cand}$  mit
      Wahrscheinlichkeit  $p_{nh}$ :

$$p_{nh} = \frac{T_{nh}}{\sum_{n \in \pi^{cand}} T_{nh}} = \frac{\sum_{q=1}^h \tau_{nq}}{\sum_{n \in \pi^{cand}} \sum_{q=1}^h \tau_{nq}}$$

12:   Füge Auftrag  $n$  auf Position  $h$  in  $\tilde{\pi}$  ein;
13:    $U = U \setminus n$ ;
14: } // end for

```

Zeile 3+14: Es wird für jede der $h = 1(1)N$ Positionen der Auftragsfolge ein Auftrag ausgewählt. Es wird dabei bei der ersten Position begonnen.

Zeile 4: Die Kandidatenliste wird bestimmt. Sie besteht aus den ersten *cand* noch nicht ausgewählten Elementen der gegenwärtig besten bekannten Lösung.

Zeile 5: Es wird eine gleichverteilte Zufallszahl u aus dem Intervall $[0; 1]$ gezogen. Mit ihrer Hilfe wird entschieden, welche der drei Möglichkeiten zur Auswahl eines Auftrags für Position h verwendet wird.

Zeile 6+7: Gilt $0 \leq u \leq p_1$, so wird das erste Element aus π^{cand} ausgewählt.

Zeile 8+9: Gilt $p_1 < u \leq p_2$, wähle aus π^{cand} das Element mit dem größten Summen-Pheromonwert nach Gleichung 6.6 aus:

$$n \in \operatorname{argmax} \left\{ T_{nh} = \sum_{q=1}^h \tau_{nq} \mid n \in \pi^{cand} \right\} \quad (6.6)$$

Zeile 10-11: Gilt $p_2 < u \leq 1$, wähle einen Auftrag aus π^{cand} mit einer

Auswahlwahrscheinlichkeit gemäß Gleichung 6.7 aus:

$$p_{nh} = \frac{T_{nh}}{\sum_{n \in \pi^{cand}} T_{nh}} = \frac{\sum_{q=1}^h \tau_{nq}}{\sum_{n \in \pi^{cand}} \sum_{q=1}^h \tau_{nq}} \quad (6.7)$$

Zeile 12: Füge das ausgewählte Element auf Position h in $\tilde{\pi}$ ein.

Zeile 13: Entferne das ausgewählte Element aus U .

6.1.3.4 Modellierung des autokatalytischen Prozesses

Der autokatalytische Prozeß wird durch das Ablegen von künstlichem Pheromon durch die Ameisen und das Verdunsten von Pheromon modelliert. Die wesentlichen Gestaltungsparameter ergeben sich, indem variiert wird, wann welche Ameisen wieviel Pheromon ablegen dürfen. Dahingegen wird das Verdunsten des Pheromons weitgehend einheitlich über eine einfache Verdunstungsrate modelliert. Allgemein folgt eine Aktualisierung der Pheromoninformationen dem folgenden Prinzip:

$$\tau_{nh}(t) := \underbrace{\rho \cdot \tau_{nh}(t-1)}_{\text{Verdunstung}} + \underbrace{\Delta\tau_{nh}(t)}_{\text{Neues Pheromon}} \quad (6.8)$$

$\tau_{nh}(t)$ ist dabei der neue Pheromonwert im Zeitpunkt t und $\tau_{nh}(t-1)$ der alte Pheromonwert bzw. der Pheromonwert nach der letzten erfolgten Aktualisierung oder Initialisierung im Zeitpunkt $t-1$. ρ ist die sogenannte Trail-Persistence-Rate, so daß $1-\rho$ als Verdunstungsrate des Pheromons interpretiert werden kann. $\Delta\tau_{nh}(t)$ ist die neu hinzuzufügende Pheromonmenge.

Dorigo u. a. (1991b) haben für das Ant System drei prinzipielle Möglichkeiten für die Berechnung von $\Delta\tau_{nh}(t)$ untersucht.⁴⁹ Im Ant Density Ansatz ist $\Delta\tau_{nh}(t)$ eine Konstante.⁵⁰ Im Ant Quantity Ansatz ist die hinzugefügte Pheromonmenge abhängig von der Kantenbewertung, die beim dort betrachteten TSP der Entfernung zwischen den Städten entspricht.⁵¹ Als beste Vorgehensweise hat sich jedoch Ant Cycle herausgestellt.⁵² Hierbei ist die hinzugefügte Pheromonmenge abhängig von der Lösungsqualität der von der aktualisierenden Ameise erstellten Lösung.⁵³ Im Permutation

⁴⁹Vgl. Dorigo u. a. (1991b, S.4-8) oder Dorigo u. a. (1991a, S.4-7).

⁵⁰Vgl. Dorigo u. a. (1991b, S.4f.) oder Dorigo u. a. (1991a, S.4f.).

⁵¹Vgl. Dorigo u. a. (1991b, S.4f.) oder Dorigo u. a. (1991a, S.4f.).

⁵²Vgl. Dorigo u. a. (1991b, S.7) oder Dorigo u. a. (1991a, S.7).

⁵³Vgl. Dorigo u. a. (1991b, S.6f.) oder Dorigo u. a. (1991a, S.6f.).

Flow Shop Problem also:

$$\Delta\tau_{nh}(t) = \begin{cases} \frac{Q}{L(\pi^{ant})} & \text{falls Auftrag } n \text{ auf Position } h \text{ ist} \\ 0 & \text{sonst} \end{cases} \quad (6.9)$$

Q ist eine Konstante und wird gewöhnlich auf 1 gesetzt.⁵⁴ $L(\pi^{ant})$ steht allgemein für die Lösungsqualität der von Ameise ant konstruierten Auftragsfolge π^{ant} . In den in dieser Arbeit vorgestellten multikriteriellen Ansätzen wird $L(\pi^{ant})$ durch den Kompromißzielfunktionswert $\Psi(\pi^{ant})$ ersetzt.

Das Ablegen von Pheromon durch eine Ameise kann während der Lösungskonstruktion und zusätzlich oder alleine nach deren Abschluß erfolgen. Ist die Pheromonmenge abhängig von der Lösungsqualität, so kann das Ablegen offensichtlich nur nach Abschluß der Konstruktion einer Auftragsfolge erfolgen.

Grundsätzlich kann jede Ameise einer Kolonie Pheromon ablegen. Es hat sich aber bereits in den Experimenten zum Ant System von Dorigo u. a. (1991b, 1996) abgezeichnet, daß eine Elitestrategie eine Verbesserung der Optimierungseigenschaft des Algorithmus verspricht.⁵⁵ Eine Elitestrategie kann z.B. vorsehen, daß nur die beste Ameise einer Iteration die Pheromoninformation aktualisieren darf. Auch die Strategie, daß allein die global beste Lösung für die Aktualisierung verwendet wird, ist ebenso denkbar wie die Verwendung von Mischformen. Im AS^{rank} -Ansatz von Bullnheimer u. a. (1999) werden die Ameisen nach ihrer Lösungsgüte in eine Rangfolge gebracht und die besten $rank$ Ameisen dürfen eine Aktualisierung vornehmen.⁵⁶ Die Elitestrategie verstärkt damit, wie auch die Verwendung von Kandidatenmengen bei der Konstruktion von Lösungen, die Suche in einem vielversprechenden Bereich des Lösungsraums. Offensichtlich kann auch die Elitestrategie nur angewandt werden, wenn das Aktualisieren der Pheromonspur erfolgt, nachdem alle Ameisen eine Lösung konstruiert haben.

Unabhängig von der Aktualisierungsstrategie wird mit dem MAX-MIN Ant System der Grundgedanke eingeführt, die Pheromonwerte in der Pheromonmatrix nach oben und unten zu beschränken.⁵⁷ Das System soll unter anderem dazu beitragen, die Stagnation der Heuristik zu

⁵⁴Gambardella u. a. (1999, S.69f.) z.B. setzt $Q = \rho$.

⁵⁵Vgl. Dorigo u. a. (1991b, S.14f.) sowie Dorigo u. a. (1996, S.16).

⁵⁶Vgl. Bullnheimer u. a. (1999, S.30f.).

⁵⁷Vgl. Stützle (1998a, S.1561), Stützle (1999, S.64-69). Weitere Anwendungen und Arbeiten zum MAX-MIN Ant System finden sich z.B. in: Stützle u. Hoos (1997, S.309-312), Stützle u. Hoos (2000, S.898-904).

vermeiden.⁵⁸ Bei der Stagnation konstruieren alle Ameisen die gleiche Lösung.⁵⁹ Das Verhalten ist auf die Entwicklung der Pheromonwerte zurückzuführen, welche mit zunehmender Laufzeit der Heuristik bewirken, daß die Auswahlwahrscheinlichkeit für bestimmte Aufträge und Positionen auf nahe Null absinkt und für andere auf nahe Eins ansteigt. Durch die Beschränkung der Pheromonwerte wirkt das MAX-MIN Ant System dieser Entwicklung entgegen. Übersteigt ein Element der Pheromonmatrix einen oberen Grenzwert τ_{max} , so wird dieses Element gleich τ_{max} gesetzt. Analog werden Elemente der Pheromonmatrix, die den unteren Grenzwert τ_{min} unterschreiten, bis auf diesen heraufgesetzt.

Die bisher angeführten Ausgestaltungsmöglichkeiten geben die wesentlichen Entwicklungen von Strategien wieder. In vielen Ansätzen finden sich Mischformen und Abwandlungen der hier vorgestellten Ansätze.⁶⁰

6.1.4 Ausgestaltungsmöglichkeiten für a-posteriori Ansätze

6.1.4.1 Allgemeine Ausgestaltungsmöglichkeiten

A-posteriori Ansätze für multikriterielle Permutation Flow Shop Probleme müssen den Zielraum über die Variation der Parameter einer geeigneten Kompromißzielfunktion absuchen. Für jede Parametereinstellung existieren ein oder mehrere globale Optima für den Kompromißzielfunktionswert. Der Bereich des Lösungsraums, in dem diese globalen Optima liegen, ist zu identifizieren und abzusuchen. Die Ergebnisse der Ziel- und Lösungsraum-analyse aus Abschnitt 5.5.3 zeigen, daß die Elemente der Menge aller effizienten Lösungen in unterschiedlichen Bereichen des Lösungsraums liegen können. Daher müssen Ansätze zur Bestimmung einer heuristisch effizienten Menge in der Lage sein, gezielt unterschiedliche Bereiche des Lösungsraums zu identifizieren und abzusuchen. Diese Anforderung unterscheidet sie von Ansätzen, die nur ein Ziel berücksichtigen bzw. von a-priori Ansätzen. Diese können sich auf die Identifikation und Erkundung einer einzigen Region des Lösungsraums beschränken. In diesem Abschnitt sollen unterschiedliche denkbare Ausgestaltungsmöglichkeiten für a-posteriori Ansätze diskutiert werden. Einen ersten wesentlichen Gestaltungsaspekt stellt die Anzahl der verwendeten Kolonien und ihre Eigenschaften dar. Des weiteren können Varianten der Single Start Strategie und Varianten der Multi Start Strategie unterschieden werden.

⁵⁸Vgl. Stützle u. Hoos (2000, S.899).

⁵⁹Vgl. hierzu und im folgenden Dorigo u. a. (1996, S.11).

⁶⁰Vgl. dazu die in Tabelle 6.1 auf Seite 154 angegebenen Ansätze.

In Iredi u. a. (2001) werden zwei Ansätze vorgestellt, um eine heuristisch effiziente Front beim bikriteriellen Single Machine Problem zu ermitteln.⁶¹ Der erste Ansatz verwendet eine einzige heterogene Kolonie, wohingegen der zweite Ansatz mehrere heterogene Kolonien verwendet. Eine Kolonie ist eine Menge von Ameisen, wobei jede Ameise eine eigene Auftragsfolge konstruiert. Eine heterogene Kolonie kann dabei wie folgt beschrieben werden:

Heterogene Kolonie: Eine Kolonie wird als heterogen bezeichnet, falls die Ameisen in unterschiedlichen Bereichen der effizienten Front und damit in unterschiedlichen Bereichen des Zielraums suchen.⁶²

Analog zur heterogenen Kolonie kann eine homogene Kolonie definiert werden:

Homogene Kolonie: In einer homogenen Kolonie suchen alle Ameisen im gleichen Bereich der effizienten Front bzw. des Zielraums.

In den Ansätzen von Iredi u. a. (2001) teilen sich alle Ameisen einer heterogenen Kolonie die gleichen Pheromonmatrizen.⁶³ Problematisch erscheint dabei, daß sich in einer Matrix Informationen über unterschiedliche Bereiche des Lösungsraums mischen, da die Ameisen unter Umständen in unterschiedliche Bereiche des Lösungsraums streben. Es ist zu erwarten, daß sich dies bei der Konstruktion von guten Lösungen negativ auswirken kann. Bei homogenen Kolonien tritt dieses Problem nicht auf, da hier alle Ameisen die gleiche Region im Zielraum erforschen und somit auch in die gleiche Region des Lösungsraums streben sollten. Aufgrund dieser Überlegungen erscheint die Verwendung einer oder mehrerer homogener Kolonien der vielversprechendere Ansatz.

Single Start mit fixiertem Parametersatz (SSFP): Bei der Single Start Strategie wird zu Beginn der Laufzeit die Pheromoninformation einmalig initialisiert. Fixierter Parametersatz bedeutet, daß die den Ameisen zugeordneten Bereiche der effizienten Front bzw. des Zielraums über die gesamte Laufzeit hinweg unverändert bleiben. Diese Strategie setzt zur Ermittlung einer heuristisch effizienten Menge die Verwendung einer heterogenen oder mehrerer homogener Kolonien voraus. Eine einzige homogene Kolonie sucht nur in einem Bereich des Zielraums und ist die klassische Konfiguration für

⁶¹Vgl. Iredi u. a. (2001, S.361-366) sowie die Arbeit von Merkle (2002, S.115-128).

⁶²Vgl. Merkle (2002, S.116) und Iredi u. a. (2001, S.363).

⁶³Es wird für jedes Ziel eine eigene Matrix verwendet, die auch eine unterschiedliche Kodierung besitzen, vgl. auch Abschnitt 6.1.4.3.

Probleme mit nur einem zu berücksichtigendem Ziel oder für a-priori Ansätze. Die SSFP-Strategie wird in den Ansätzen von Iredi u. a. (2001) verwendet.⁶⁴

Single Start mit dynamischem Parametersatz (SSDP): Im Gegensatz zur SSFP-Strategie werden die Zuordnungen zu den Bereichen des Zielraums, in denen die Ameisen suchen, über die Laufzeit hinweg variiert. Die Pheromoninformation wird jedoch nur zu Beginn der Laufzeit einmalig initialisiert. Problematisch bei dieser Strategie erscheint, daß die Kolonien nach der Zuordnung eines neuen Bereichs im Zielraum lange benötigen, bis sie den neuen Bereich des Lösungsraums identifiziert haben. Die Pheromoninformationen lenkt die Suche zunächst noch in den alten Bereich des Lösungsraums. Sollen dynamische Parameter verwendet werden, so bietet sich daher eher die Verbundene Multi Start Strategie an.

Verbundene Multi Start Strategie (VMSS): Mit jedem Wechsel der Suchregion im Zielraum wird die Pheromoninformation neu initialisiert. Die Strategie wird als verbunden bezeichnet, da diese Initialisierung von der bisher erreichten Lösung abhängig sein kann. Die einzelnen Starts sind somit nicht völlig unabhängig voneinander.

Unverbundene Multi Start Strategie (UMSS): Bei dieser Strategie werden für jede Region im Zielraum ein oder mehrere vollständig, von vorangegangenen Läufen unabhängige Läufe der Heuristik durchgeführt. Aus den verschiedenen heuristisch effizienten Mengen der einzelnen Läufe wird anschließend die heuristisch effiziente Menge aller Läufe ermittelt.⁶⁵

Die in dieser Arbeit in den Abschnitten 6.3 und 6.4 vorgestellten Ameisenalgorithmen folgen der VMSS-Strategie. Der SCMO-Ansatz⁶⁶ verwendet eine einzelne homogene Kolonie. Der MCMO-Ansatz⁶⁷ setzt mehrere homogene Kolonien ein. Eine dritte denkbare Alternative stellt die Anwendung einer SSFP-Strategie in Verbindung mit mehreren homogenen Kolonien dar. Dieser Ansatz wird hier nicht verfolgt, da jeder Region des Zielraums eine eigene Kolonie zugeordnet werden muß. Insbesondere bei einer Suche mit vielen unterschiedlichen Suchbereichen im Zielraum würde

⁶⁴Vgl. Iredi u. a. (2001, S.361-366) sowie Merkle (2002, S.115-128).

⁶⁵Diese Möglichkeit wird auch in der Arbeit von Merkle (2002) diskutiert (vgl. Merkle (2002, S.118-128.), insbesondere die Seiten 118, 123f. u. 128).

⁶⁶SCMO steht für Single-Colony-Multi-Objektive. Der Ansatz wird in Abschnitt 6.3 beschrieben.

⁶⁷MCMO steht für Multi-Colony-Multi-Objektive. Der Ansatz wird in Abschnitt 6.4 beschrieben.

die Anzahl der Kolonien bei dieser Strategie stark ansteigen. Die UMSS-Strategie wird in dieser Arbeit ebenfalls nicht weiter verfolgt. Sie wurde bereits von Merkle (2002) untersucht. Dort hat sich gezeigt, daß Strategien, welche eine Kooperation von Kolonien verfolgen, bessere Ergebnisse liefern.⁶⁸

6.1.4.2 Multi Colony Ansätze und Kommunikation

Das Vorgehen, mehrere Kolonien einzusetzen, ist bereits in unterschiedlichen Arbeiten verwendet worden. Mögliche Einsatzgebiete sind multikriterielle Problemstellungen (z.B. Gambardella u. a. (1999), Iredi u. a. (2001)) sowie parallele Implementationen für Probleme mit einfacher Zielsetzung (z.B. Bullnheimer u. a. (1998), Stützle (1998b), Michel u. Middendorf (1999)).⁶⁹ Ein wesentliches Element der Multi-Kolonie-Ansätze ist in beiden Fällen die Kommunikation der Kolonien untereinander.

Stützle (1998b) untersucht die Vorteilhaftigkeit vieler kurzer, unabhängiger Läufe eines ACO-Algorithmus im Vergleich zu einem einzigen, langen Lauf.⁷⁰ Diese unabhängigen Läufe können parallel ausgeführt werden, was einen Geschwindigkeitsgewinn bewirkt. Stützle (1998b) hat beobachtet, daß sich durch die wiederholte oder parallele Ausführung kurzer Läufe anstelle von einem langen Lauf des MAX-MIN Ant Systems die Lösungsqualität steigern läßt.⁷¹ In seiner Untersuchung sind die Kolonien völlig unabhängig voneinander, eine Kommunikation findet nicht statt. Als Möglichkeit dazu werden von Stützle (1998b) zwei Strategien vorgeschlagen, aber nicht weiter untersucht. Zum einen können die vollständigen Pheromoninformationen, also die Pheromonmatrizen zwischen den Kolonien, ausgetauscht werden. Zum anderen können lediglich die besten Lösungen der einzelnen Kolonien untereinander ausgetauscht werden.⁷²

Merkle u. Middendorf (2002) weisen darauf hin, daß das Austauschen der besten Lösung die vorzuziehende Strategie ist.⁷³ Dieses Vorgehen findet sich auch in den Arbeiten von Bullnheimer u. a. (1998) und Michel u. Middendorf (1999)⁷⁴. Bei Michel u. Middendorf (1999) suchen mehrere Kolonien parallel zueinander nach Lösungen, wobei sie nach einer bestimmten

⁶⁸Diese Möglichkeit wird auch in der Arbeit von Merkle (2002) diskutiert (vgl. Merkle (2002, S.118-128.), insbesondere die Seiten 118, 123f. u. 128).

⁶⁹Vgl. Gambardella u. a. (1999, S.67-72), Iredi u. a. (2001, S.361-367), Bullnheimer u. a. (1998, S.91-99), Stützle (1998b, S.725-729), Michel u. Middendorf (1999, S.56f.).

⁷⁰Vgl. Stützle (1998b, S.726f.).

⁷¹Vgl. Stützle (1998b, S.728).

⁷²Vgl. Stützle (1998b, S.727).

⁷³Vgl. Middendorf u. a. (2000, S.646). Siehe auch Middendorf u. a. (2002, S.308f.). Beide Arbeiten verweisen auf ein unveröffentlichtes Manuskript von Krüger u. a. (1998).

⁷⁴Vgl. Michel u. Middendorf (1999, S.57) und Bullnheimer u. a. (1998, S.91-93).

Anzahl von Iterationen die beste gefundene Lösung austauschen. Ähnlich funktioniert auch der Ansatz von Bullnheimer u. a. (1998).⁷⁵

Die Verwendung mehrerer Kolonien war in den bisher erwähnten Ansätzen dazu bestimmt, eine Rechenzeitverkürzung oder Verbesserung der Lösungsqualität bei Problemen mit einfacher Zielsetzung zu erzielen. Die Ansätze von Gambardella u. a. (1999) und Iredi u. a. (2001) hingegen sind für multikriterielle Problemstellungen entwickelt worden und die Verwendung mehrerer Kolonien aus einem anderen Aspekt heraus motiviert.⁷⁶

Gambardella u. a. (1999) betrachten das Tourenplanungsproblem mit Zeitfensterrestriktionen, wobei sie eine lexikographische Ordnung der zwei Ziele Minimierung der Summe der Fahrzeiten und Minimierung der benötigten Fahrzeugzahl betrachten. Jedem der zwei verfolgten Ziele ist eine unabhängige Kolonie zugeordnet. Die Notwendigkeit der Kommunikation der Kolonien ergibt sich schon allein daraus, daß die von der ersten Kolonie ermittelte Fahrzeugzahl eine Restriktion für die zweite Kolonie ist, welche versucht, die Gesamtfahrzeit bei gegebener Fahrzeugzahl zu minimieren.⁷⁷

Iredi u. a. (2001) betrachten ein multikriterielles Single Machine Problem und lenken die einzelnen Kolonien in unterschiedliche Bereiche der heuristisch effizienten Front.⁷⁸ Der Ansatz wird im folgenden Abschnitt kurz beschrieben, wobei ein Schwerpunkt der Darstellung auf der Kommunikation zwischen den Kolonien liegt.

6.1.4.3 Der Multi Colony Ansatz von Iredi u. a. (2001)

Iredi u. a. (2001) betrachten ein Single Machine Problem mit den Zielen Minimierung der Summe der Verspätungen und Minimierung der Umrüstkosten.⁷⁹ Ziel ist die Bestimmung einer heuristisch effizienten Front. Die Suche wird von einer festzulegenden Zahl heterogener Kolonien durchgeführt. Jede Kolonie besitzt für jedes Ziel eine eigene Pheromonmatrix.⁸⁰ Die Zuordnung zu unterschiedlichen Suchbereichen im Zielraum erfolgt, indem die zwei Matrizen bzw. deren Pheromonwerte bei der Berechnung der Auswahlwahrscheinlichkeit für die Zuordnung eines Auftrags auf eine Position unterschiedlich gewichtet werden.⁸¹ Eine Zuordnung der Ameisen zu den

⁷⁵Vgl. Michel u. Middendorf (1999, S.57) und Bullnheimer u. a. (1998, S.91-93).

⁷⁶Vgl. Gambardella u. a. (1999, S.67-72) und Iredi u. a. (2001, S.361-367). Auch bei diesen Ansätzen ist eine Parallelisierung möglich, aber der Zeitgewinn scheint nicht die primäre Motivation zu sein.

⁷⁷Vgl. dazu Gambardella u. a. (1999, S.67-72).

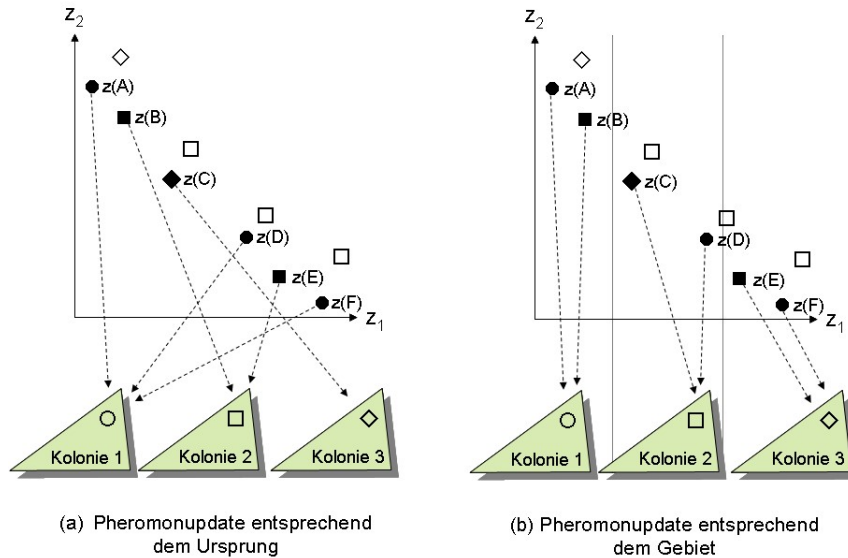
⁷⁸Vgl. Iredi u. a. (2001, S.361-367).

⁷⁹Vgl. für diesen Abschnitt Iredi u. a. (2001, S.360-372) und Merkle (2002, S.111-128).

⁸⁰Dieses Vorgehen wird damit begründet, daß die zwei Ziele unterschiedliche Kodierungen der Matrix erfordern.

⁸¹Auf die drei untersuchten unterschiedlichen Strategien zur Festlegung der Gewichtintervalle der Kolonien wird hier nicht genauer eingegangen.

Bereichen des Zielraums über eine Kompromißzielfunktion wird nicht vorgenommen. Die zu Beginn festgelegten Gewichte bleiben über die Laufzeit hinweg unverändert.



Siehe zu den Abbildungen: Iredi u. a. (2001, S.366) oder Merkle (2002, S.119f.). Die ausgefüllten Elemente kennzeichnen die (heuristisch) effiziente Front. Die Pfeile zeigen an, welche Pheromonmatrix welcher Kolonie durch die Auftragsfolge aktualisiert wird.

Abbildung 6.4: Aktualisierungsstrategien nach Iredi u. a. (2001)

Die Kolonien teilen sich eine globale, heuristisch effiziente Front (GNDF). Diese stellt die heuristisch effiziente Front bezogen auf die von allen Kolonien auf Effizienz überprüften Lösungen dar. Dagegen bezieht sich die lokale heuristisch effiziente Front (LNDF) nur auf die Lösungen einer einzelnen Kolonie. Die Kommunikation der Kolonien erfolgt über die global heuristisch effiziente Front bei der Pheromonaktualisierung, wobei zwei alternative Strategien verwendet werden.

Die erste Strategie wird als *Pheromonaktualisierung entsprechend dem Ursprung* bezeichnet. Dabei darf eine Ameise die Pheromonmatrizen ihrer eigenen Kolonie aktualisieren, falls ihre Lösung in der globalen heuristisch effizienten Front bereits enthalten ist oder dort eingefügt wurde. In Abbildung 6.4(a) findet Kolonie 1 die Auftragsfolgen A, D und F. Unabhängig von der Lage im Zielraum aktualisieren diese Lösungen stets die Pheromonmatrix der Kolonie 1.

Die zweite Strategie wird als *Pheromonaktualisierung entsprechend dem Gebiet* bezeichnet. Dabei darf eine Ameise eine Pheromonaktualisie-

rung vornehmen, wenn ihre Lösung in der globalen heuristisch effizienten Front bereits enthalten ist oder dort eingefügt wurde. Dabei aktualisiert sie jedoch die Matrizen derjenigen Kolonie, die dem Bereich der Front zugeordnet ist, in dem die Lösung liegt. Die Zuordnung einer Kolonie zu einem Bereich wird dabei in Abhängigkeit von der globalen heuristisch effizienten Front vorgenommen. Diese wird nach einem Zielkriterium sortiert und anschließend in so viele Teilintervalle zerlegt, wie es Kolonien gibt. Die Berechnung der Teilintervalle kann auch derart vorgenommen werden, daß sich die Intervalle überschneiden. Diese Teilintervalle werden den Kolonien zugeordnet. In Abbildung 6.4(b) aktualisiert die Auftragsfolge A die Pheromonmatrix von Kolonie 1, Auftragsfolge D die Pheromonmatrix von Kolonie 2 und Auftragsfolge F die Pheromonmatrix von Kolonie 3.

Überlappen sich die Intervalle der Kolonien bei der Pheromonaktualisierung entsprechend dem Ursprung, läßt sich beobachten, daß die beiden beschriebenen Kommunikationsstrategien bessere Ergebnisse generieren als eine Unverbundene Multi Start Strategie. Sind die beiden Strategien bei der Minimierung der Umrüstkosten etwa gleich gut, so ist in den restlichen Abschnitten der heuristisch effizienten Front die Pheromonaktualisierung entsprechend dem Gebiet vorteilhaft.

6.2 Lokale Suchstrategie in zwei Phasen

6.2.1 Motivation der zwei Phasen-Strategie

In diesem Abschnitt soll die von den später vorgestellten SCMO-Ansatz⁸² und MCMO-Ansatz⁸³ verwendete lokale Suchstrategie beschrieben werden. Dabei werden die Ergebnisse aus Kapitel 5 berücksichtigt.

Mit Hilfe der lokalen Suche soll ein konkreter Bereich des Lösungsraums gründlicher untersucht werden, indem die Nachbarschaft der Ausgangslösung im Lösungsraum unter Anwendung eines Nachbarschaftsoperators generiert und ausgewertet wird. In Kapitel 5 wurden unterschiedliche Nachbarschaftsoperatoren vorgestellt. Ebenso wurde untersucht, welcher dieser Nachbarschaftsoperatoren die größte Wahrscheinlichkeit aufweist, daß ein Nachbar im Lösungsraum auch ein Nachbar im Zielraum ist. Es wurde argumentiert, daß es zum Ende der lokalen Suche wünschenswert ist, verstärkt Nachbarn der Ausgangslösung im Zielraum zu generie-

⁸²SCMO steht für Single-Colony-Multi-Objektive. Der Ansatz wird in Abschnitt 6.3 beschrieben.

⁸³MCMO steht für Multi-Colony-Multi-Objektive. Der Ansatz wird in Abschnitt 6.4 beschrieben.

ren. Damit soll die Suche auf den von den Parametern der Kompromißziel-funktion eingestellten Bereich des Zielraums beschränkt bleiben. Es wurde bereits darauf hingewiesen, daß es nicht sinnvoll ist, für jede generierte Auftragsfolge die Nachbarschaft im Zielraum einer Auftragsfolge intensiv abzusuchen. Eine auf diese Weise untersuchte Auftragsfolge bzw. ihr Ziel-funktionsvektor sollte in dem durch die Parameter des Kompromißmodells eingestellten Bereich der effizienten Front liegen. Die Konstruktion einer Lösung durch eine Ameise stellt dies nicht grundsätzlich sicher. Die lokale Suche soll daher zunächst versuchen, ausgehend von der Lösung der Ameise eine Auftragsfolge zu finden, die im eingestellten Bereich der effizienten Front liegt. Die lokale Suche wird daher in zwei Phasen unterteilt:

Phase 1: In dieser Phase wird die Ausgangslösung lediglich bzgl. ihres Kompromißzielfunktionswertes verbessert. Ziel ist es, möglichst schnell eine Annäherung an den Zielbereich der effizienten Front zu erreichen. Große Sprünge im Zielraum sind daher explizit erwünscht. Dies kann über Nachbarschaftsoperatoren erreicht werden, die in den Experimenten aus Abschnitt 5.3 eine große Abweichung von der Nachbarschaft im Zielraum aufwiesen. Hier werden der Exchange-Operator und der Double-Shift-Operator eingesetzt. Der Einsatz zweier unterschiedlicher Operatoren soll es ermöglichen, die lokalen Optima eines Operators durch die Anwendung des anderen zu verlassen (vgl. Abschnitt 5.2.2).

Phase 2: In dieser Phase wird versucht, den Kompromißzielfunktionswert zu verbessern. Weiterhin wird bei Auftragsfolgen, die in die aktuelle heuristisch effiziente Menge eingefügt werden und gleichzeitig den aktuell besten bekannten Kompromißzielfunktionswert nicht verschlechtern, unter Anwendung des API-Operators die Suche in der Zielraumnachbarschaft verstärkt.

Die in Phase 1 verwendeten Schemata, das Job Swap Schema (JSS) und das Job Insertion Schema (JIS) werden in Abschnitt 6.2.2 beschrieben. Abschnitt 6.2.3 beschreibt anschließend das in Phase 2 verwendete Job Swap Schema mit rekursive API Suche (JSS-RAPI).

6.2.2 Lokale Suche in Phase 1: Job Swap Schema (JSS) und Job Insertion Schema (JIS)

Job Swap Schema (JSS)

Das Job Swap Schema benutzt den Exchange-Operator, vertauscht also jeweils zwei Aufträge in der Auftragsfolge. Jeder Auftrag wird, sofern möglich, mit den Aufträgen auf *range* Positionen vor und nach seiner aktuellen Position vertauscht.⁸⁴ Nachdem ein Auftrag auf alle zulässigen Positionen vertauscht wurde, wird der Zug mit der größten Vorteilhaftigkeit ausgewählt und durchgeführt. Erweist sich keiner der Züge als vorteilhaft, so findet kein Tausch statt. Anschließend wird der nächste Auftrag betrachtet. Das Verfahren ist abgeschlossen, wenn alle Aufträge auf Tauschmöglichkeiten hin überprüft wurden. Ob ein Zug als vorteilhaft beurteilt wird, ist zum einen davon abhängig, ob er den Kompromißzielfunktionswert verbessert und zum anderen davon, ob die Auftragsfolge in die heuristisch effiziente Menge eingefügt wird. Ein Zug gilt als vorteilhaft, wenn er den aktuellen Kompromißzielfunktionswert verbessert. Ein Zug gilt aber auch dann als vorteilhaft, wenn er den aktuellen Kompromißzielfunktionswert nicht verschlechtert, die Auftragsfolge aber in die aktuelle heuristisch effiziente Menge eingefügt wird. Das JSS ist im Algorithmus 4 beschrieben.

Zeile 1: Die der Funktion übergebene Auftragsfolge π^o wird in π^b gespeichert.

Zeilen 2+15: In der äußeren For-Schleife wird jede Position der Auftragsfolge angesteuert.

Zeile 3: Das InsertFlag merkt sich, ob bereits eine Auftragsfolge während der lokalen Suche in die NDS eingefügt wurde. Initialwert ist *false*.

Zeilen 4+12: In der inneren For-Schleife wird der Auftrag auf Position *i* mit allen Aufträgen, die sich auf *range* Positionen vor und nach ihm befinden, vertauscht.

Zeile 5: Hier wird der Auftrag auf Position *i* mit dem Auftrag auf Position *j* vertauscht. Die so erhaltene Auftragsfolge wird in π' gespeichert.

Zeile 6: Es wird versucht, π' in NDS einzufügen. Sollte dies erfolgreich sein, so werden gleichzeitig alle Elemente, die bisher in der NDS ent-

⁸⁴Ist die aktuelle Position *i* eines Auftrages kleiner als der Wert von *range*, so kann er offensichtlich nur mit den *i*-1 Aufträgen vor seiner aktuellen Position vertauscht werden. Analoge Überlegungen gelten, wenn die Anzahl der Positionen nach der aktuellen Position *i* kleiner sind als *range*. Diese Überlegungen werden in Zeile 4 im Algorithmus 4 berücksichtigt.

Algorithmus 4 Job Swap Schema (JSS)

```

// Parameterliste: ( $\pi^\circ$ ,  $\lambda$ , NDS)
//  $\pi^\circ$    übergebene Auftragsfolge
//  $\lambda$    aktuelle Parameter des Kompromiß-
//          modells
// NDS     Übergebene heuristisch effiziente Menge, in die
//          Lösungen eingefügt werden

1:  $\pi^b = \pi^\circ$ ;
2: for(  $i=0$ ;  $i<N$ ;  $i=i+1$ ) {
3:   InsertFlag = false;
4:   for(  $j=\max( 0; i-range)$ ;  $j<\min(N; i+range+1)$ ;  $j=j+1$ ) {
5:      $\pi' \leftarrow$ Vertausche in  $\pi^\circ$  die Aufträge auf Position
         $i$  und  $j$ ;
6:     Versuche  $\pi'$  in NDS einzufügen;
7:     if(  $\Psi_\lambda(\pi') == \Psi_\lambda(\pi^b)$  und  $\pi'$  in NDS eingefügt) {
8:        $\pi^b = \pi'$ ;
9:       InsertFlag = true;
10:    } // end if
11:    if(  $\Psi_\lambda(\pi') < \Psi_\lambda(\pi^b)$  )  $\pi^b = \pi'$ ;
12:  } // end for
13:  if(  $(\Psi_\lambda(\pi^b) < \Psi_\lambda(\pi^\circ))$  oder
         $(\Psi_\lambda(\pi^b) == \Psi_\lambda(\pi^\circ)$  und InsertFlag == true ))
14:     $\pi^\circ = \pi^b$ ;
15: } // end for

```

halten waren, nun aber von π' dominiert werden, aus der NDS entfernt.

Zeile 7-10: Ist der Kompromißzielfunktionswert von π' gleich dem von π^b und wurde π' in die NDS eingefügt, so setze π' als neues π^b . Gleichzeitig ist InsertFlag auf *true* zu setzen.

Zeile 11: Ist der Kompromißzielfunktionswert von π' besser als der von π^b , so setze $\pi^b = \pi'$. Es spielt in diesem Fall keine Rolle, ob π' in die NDS eingefügt wurde oder nicht.

Zeilen13-14: Der Auftrag auf Position i wurde nun mit allen Aufträgen auf den Positionen $i - range$ und $i + range$ vertauscht. Es wird nun der Zug zu der in π^b gespeicherten Auftragsfolge durchgeführt. Die if-Bedingung prüft lediglich ab, ob sich π^b überhaupt verändert hat. Dies dient allein dem Einsparen von Rechenzeit.

Job Insertion Schema (JIS)

Das Job Insertion Schema (JIS) benutzt den DSH-Operator. Um das JIS zu erhalten, muß im Algorithmus 4 lediglich Zeile 5 ersetzt werden gegen die

Anweisung:

```
5:  $\pi' \leftarrow$  Verschiebe in  $\pi^\circ$  den Auftrag von Position i  
auf Position j
```

6.2.3 Lokale Suche in Phase 2: Das Job Swap Schema mit rekursiver API Suche (JSS-RAPI)

Grundsätzlich folgt das Job Swap Schema mit rekursiver API Suche dem Job Swap Schema. Es wird lediglich der Aufruf einer rekursiven API Nachbarschaftssuche (RAPI) zwischen den Zeilen 9 und 10 im Algorithmus 4 eingefügt.

```
RAPI(  $\pi^b$ ,  $\lambda$ , NDS, LEVEL+1);
```

Wird also eine Auftragsfolge in die heuristisch effiziente Menge eingefügt, die den zu diesem Zeitpunkt besten bekannten Kompromißzielfunktionswert nicht verschlechtert, so wird innerhalb des Job Swap Schemas die rekursive API Suche ausgeführt.⁸⁵ Die rekursive API Suche (RAPI) ist im Algorithmus 5 angegeben.

⁸⁵Diese im Job Swap Schema mit π' bezeichnete Lösung wird als Parameter an RAPI übergeben, wo sie gemäß Parameterliste als π° empfangen wird. Diese eigenständige Notation von RAPI wurde gewählt, da die Prozedur prinzipiell auch alleine, ohne eine Einbindung in das JSS verwendet werden könnte.

Algorithmus 5 Rekursive API Suche (RAPI)

```

// Parameterliste: ( $\pi^\circ$ ,  $\lambda$ , NDS, LEVEL)
//  $\pi^\circ$ : übergebene Auftragsfolge
//  $\lambda$ : aktuelle Parameter des Kompromiß-
//       modells
// NDS: Übergebene heuristisch effiziente Menge, in die
//       Lösungen eingefügt werden
// LEVEL: Aktuelle Tiefe der Rekursion

// Sonstige Parameter:
// MAX_LEVEL: Maximal zulässige Tiefe der Rekursion

1:  $\pi^b = \pi^\circ$ ;
2: for(  $i=0$ ;  $i<N-1$ ;  $i=i+1$ ) {
3:    $\pi' \leftarrow$  Vertausche in  $\pi^\circ$  die Aufträge auf Position
       i und  $i+1$ ;
4:   Versuche  $\pi'$  in NDS einzufügen;
5:   if(  $\Psi_\lambda(\pi') == \Psi_\lambda(\pi^b)$  &&  $\pi'$  in NDS eingefügt )  $\pi^b = \pi'$ ;
6:   if(  $\Psi_\lambda(\pi') < \Psi_\lambda(\pi^b)$  )  $\pi^b = \pi'$ ;
7:   if(  $\pi'$  in NDS eingefügt &&  $\Psi_\lambda(\pi') == \Psi_\lambda(\pi^b)$  &&
       LEVEL < MAX_LEVEL)
8:     RAPI( $\pi^b$ ,  $\lambda$ , NDS, LEVEL+1);
9: } // end for
10: if(  $\Psi_\lambda(\pi^b) < \Psi_\lambda(\pi^\circ)$  )  $\pi^\circ = \pi^b$ ;

```

RAPI führt eine eigenständige Nachbarschaftssuche unter Anwendung des API Operators durch. Tritt während dieser Nachbarschaftssuche erneut die Situation auf, daß eine Auftragsfolge in die NDS eingefügt wird, ohne daß sich der zu diesem Zeitpunkt beste bekannte Kompromißzielfunktionswert verschlechtert, so ruft sich diese Funktion selbst auf (Zeile 7). Der Parameter *MAX_LEVEL* beschränkt dabei die Tiefe der Rekursion.

6.3 Der SCMO-Ansatz

6.3.1 Beschreibung des Algorithmus

6.3.1.1 Allgemeine Eigenschaften

Der Single-Colony-Multi-Objective Ansatz (SCMO-Ansatz) verwendet eine homogene Kolonie. Mit fortschreitender Laufzeit wechselt die Kolonie ihre Zielregion im Zielraum, wobei mit jedem Wechsel eine Neuinitialisierung der Pheromonmatrix erfolgt. Die Initialisierung wird auf Grundlage der bis dahin ermittelten heuristisch effizienten Menge durchgeführt (verbundene Multi Start Strategie). Die Kolonie verwendet die gewichtete Tchebycheff-Distanz als Kompromißmodell, wobei $z^{ut} = 0$ und $p = \infty$

gesetzt werden. Weiterhin werden die Werte z_k^{ut} und $z_k(\pi)$ jeweils noch durch z_k^{NORM} geteilt.⁸⁶ Der Kompromißzielfunktionswert errechnet sich dann gemäß der folgenden Gleichung:

$$\psi_\lambda(\pi) = \max_{k=1(1)K} \left\{ \lambda_k \frac{z_k(\pi)}{z_k^{NORM}} \right\} \quad (6.10)$$

Die mit der Größe z_k^{NORM} vorgenommene Normierung ist notwendig, um das unterschiedliche Niveau der Zielfunktionswerte der K Ziele zu berücksichtigen. z_k^{NORM} bestimmt sich wie folgt:

$$z_k^{NORM} = \begin{cases} \zeta_k^{NORM} & \text{falls } z_k^{EUB} = 0 \\ z_k^{EUB} & \text{sonst} \end{cases} \quad (6.11)$$

Die Größe z_k^{EUB} wird als endogene kleinste obere Schranke der Zielgröße k bezeichnet. Als endogen wird die Schranke bezeichnet, weil sie von der Heuristik selbst ermittelt wird und dieser nicht extern als Eingabeparameter vorgegeben werden muß. Dieses Vorgehen hat den Vorteil, daß die Heuristik auch für Probleme verwendet werden kann, für die noch keine kleinsten oberen Schranken mit anderen Verfahren ermittelt wurden. Nachteilig ist, daß von Beginn an eine ausreichend gute Qualität der endogenen oberen Schranken sichergestellt werden muß. Es ist jedoch nicht grundsätzlich ausgeschlossen, daß eine Zielgröße den Wert Null annehmen kann. Im in dieser Arbeit betrachteten Zielsystem ist dies bei der maximalen Terminüberschreitung möglich, d.h. es kann $z_{T_{max}}^{EUB} = 0$ gelten. In Gleichung 6.10 wäre daher bei Verwendung von z_k^{EUB} im Nenner eine Division durch Null nicht grundsätzlich auszuschließen. Daher wird $z_k^{NORM} = \zeta_k^{NORM}$ gesetzt, falls $z_k^{EUB} = 0$ gilt. Dabei ist ζ_k^{NORM} eine positive Zahl, die jedoch kleiner ist als der kleinste, theoretisch mögliche positive Wert für die Zielgröße k . Somit ist sichergestellt, daß der Nenner stets einen positiven Wert hat.⁸⁷

Eine weitere Eigenschaft des Algorithmus ist es, daß grundsätzlich alle generierten Auftragsfolgen auf ihre heuristische Effizienz hin überprüft werden. Dies ist ein wesentlicher Unterschied zu anderen Ansätzen, wie z.B. MOGLS von Ishibuchi u. Murata (1998) oder GPWGA von Chang u. a. (2002).⁸⁸

⁸⁶Vgl. zum Kompromißzielfunktionswert der gewichtete Tchebycheff-Distanz auch Gleichung 3.32 auf Seite 37.

⁸⁷Die drei in dieser Arbeit betrachteten Zielgrößen können keine negativen Werte annehmen. Für das in dieser Arbeit betrachtete Problem kann z.B. $\zeta_k^{NORM} = 0,9$ für alle K Zielgrößen gesetzt werden.

⁸⁸Vgl. Ishibuchi u. Murata (1998, S.392-399) und Chang u. a. (2002, S.3-8).

Die folgende Beschreibung stellt eine nach der Durchführung verschiedener Experimente als gut bewertete Ausgestaltung dar. Eine Beschreibung der Experimente erfolgt in Abschnitt 6.3.2.

6.3.1.2 Initialisierungsphase und Ermittlung der endogenen kleinsten oberen Schranken

Ziel der Initialisierungsphase ist es, zunächst erste, gute, endogene, kleinste obere Schranken für die einzelnen K Ziele zu bestimmen. Des Weiteren soll eine erste heuristisch effiziente Menge generiert werden. Dazu wird während der Initialisierungsphase für jedes der K Ziele eine einfache Konstruktionsheuristik verwendet und deren Ergebnisse mit Hilfe einer lokalen Suche verbessert.⁸⁹

Für das in dieser Arbeit betrachtete Problem wurden die Konstruktionsheuristiken von Nawaz u. a. (1983) (NEH-Heuristik) und Rajendran (1993) (RAJ93-Heuristik) verwendet.⁹⁰ Für die lokale Suche werden das Job Swap Schema (JSS) und das Job Insertion Schema (JIS) in der ersten Phase sowie das Job Swap Schema mit rekursiver API Suche (JSS-RAPI) in der zweiten Phase verwendet. Für jedes der drei betrachteten Ziele wird bei der dreistufigen lokalen Suche eine lokale heuristisch effiziente Menge (LNDS) verwendet. Lokal bezieht sich dabei auf die lokale Suche bezüglich eines einzelnen Ziels. Die Elemente dieser Menge werden nach Abschluß der lokalen Suche in die globale heuristisch effiziente Menge (GNDS) eingefügt. "Global" bezieht sich dabei auf die gesamte Heuristik. Die GNDS stellt auch die abschließende Lösung der Heuristik dar.

Die Ermittlung der Werte von z_k^{EUB} erfolge im SCMO- wie auch im später vorgestellten MCMO-Ansatz stets nach Gleichung 6.12:

$$z_k^{EUB} = \min \{z_k(\pi) \mid \pi \in NDS\} \quad (6.12)$$

Im Algorithmus 6 ist die Initialisierungsprozedur wiedergegeben.

Zeile 1+2: Zu Beginn der Heuristik ist noch kein heuristisch effizientes Element identifiziert worden. Daher wird die globale heuristisch effiziente Menge (GNDS) als leere Menge initialisiert.

Zeile 3: Die endogenen kleinsten oberen Schranken werden zunächst für alle drei betrachteten Ziele auf 1 gesetzt. Sie werden für die Berechnung der Kompromißzielfunktionswerte in den lokalen Suchverfahren

⁸⁹Wichtig ist dabei zu beachten, daß, obwohl allein nach einem Ziel minimiert wird, die anderen Zielgrößen dennoch ermittelt werden müssen. Nur so ist die Überprüfung der heuristischen Effizienz möglich.

⁹⁰Vgl. Nawaz u. a. (1983, S.92-94) und Rajendran (1993, S.66-69).

Algorithmus 6 SCMO-INIT

```

1: // Initialisiere die globale heuristisch
   effiziente Menge
2: GNDS =  $\emptyset$ ;
3:  $z^{EUB} = (1; 1; 1)$ ;
4: // Minimiere Zykluszeit
5:  $\lambda = (1; 0; 0)$ ;
6:  $\pi \leftarrow$ NEH-Heuristik;
7: LNDS =  $\{\pi\}$ ;
8: JSS( $\pi, \lambda, \text{LNDS}$ ); JIS( $\pi, \lambda, \text{LNDS}$ ); JSS-RAPI( $\pi, \lambda, \text{LNDS}$ );
9: Füge Elemente aus LNDS in GNDS ein;

10: // Minimiere mittlere Durchlaufzeit
11:  $\lambda = (0; 1; 0)$ ;
12:  $\pi \leftarrow$ RAJ93-Heuristik;
13: LNDS =  $\{\pi\}$ ;
14: JSS( $\pi, \lambda, \text{LNDS}$ ); JIS( $\pi, \lambda, \text{LNDS}$ ); JSS-RAPI( $\pi, \lambda, \text{LNDS}$ );
15: Füge Elemente aus LNDS in GNDS ein;

16: // Minimiere max. Terminüberschreitung
17:  $\lambda = (0; 0; 1)$ ;
18:  $\pi \leftarrow$ RAJ93-Heuristik;
19: LNDS =  $\{\pi\}$ ;
20: JSS( $\pi, \lambda, \text{LNDS}$ ); JIS( $\pi, \lambda, \text{LNDS}$ ); JSS-RAPI( $\pi, \lambda, \text{LNDS}$ );
21: Füge Elemente aus LNDS in GNDS ein;

```

ren benötigt. Sie haben während der Initialisierungsphase auf die Suche jedoch noch keinen Einfluß, da hier zu einem Zeitpunkt jeweils nur ein Ziel minimiert wird und die anderen Ziele daher in der Kompromißzielfunktion ein Gewicht von 0 haben.

Zeile 4+5: Zunächst wird nach einer Auftragsfolge gesucht, welche speziell die Zykluszeit minimiert. In dieser Darstellung ist λ_1 das Element des Gewichtevektors λ , welches der Zykluszeit zugeordnet ist. Es wird daher auf 1 gesetzt. Die anderen beiden Zielgrößen erhalten ein Gewicht von 0.

Zeile 6: Mit Hilfe der NEH-Heuristik⁹¹ wird unter alleiniger Berücksichtigung der Zykluszeit eine erste Auftragsfolge generiert.

Zeile 7: Die von der NEH-Heuristik ermittelte Auftragsfolge wird in die lokale heuristisch effiziente Menge (LNDS) eingefügt.

Zeile 8: Die von der NEH-Heuristik ermittelte Auftragsfolge wird durch lokale Suche verbessert. Dabei wird in Phase 1 JSS und JIS verwendet. Durch den anschließenden Aufruf von JSS-RAPI wird die Suche

⁹¹Die NEH-Heuristik bezeichnet hier die Heuristik von Nawaz u. a. (1983) (vgl. Nawaz u. a. (1983, S.92-94)).

im nun erreichten Bereich im Zielraum verstärkt (Phase 2 der lokalen Suche).

Zeile 9: Die lokale Suche hat die lokale heuristisch effiziente Menge (LNDS) durch neue als heuristisch effizient identifizierte Elemente aktualisiert. Die nun vorliegenden Elemente der LNDS werden in die noch leere GNDS eingefügt. Da diese noch leer ist, wird die Menge einfach kopiert.

Zeilen 10+11: Als nächstes wird eine Auftragsfolge gesucht, welche speziell das Ziel der mittleren Durchlaufzeit minimiert. In dieser Darstellung ist λ_2 das Element des Gewichtevektors λ , welches der mittleren Durchlaufzeit zugeordnet ist. Es wird daher auf 1 gesetzt. Die anderen beiden Zielgrößen erhalten ein Gewicht von 0.

Zeile 12: Mit Hilfe der RAJ93-Heuristik⁹² wird unter alleiniger Berücksichtigung der mittleren Durchlaufzeit eine Auftragsfolge generiert.

Zeile 13: Für die folgende lokale Suche wird die lokale heuristisch effiziente Menge neu initialisiert. Sie enthält zunächst alleine die von der RAJ93-Heuristik in Zeile 12 ermittelte Auftragsfolge.

Zeile 14: Die von der RAJ93-Heuristik ermittelte Auftragsfolge wird durch lokale Suche verbessert.

Zeile 15: Die lokale Suche hat die lokale heuristisch effiziente Menge (LNDS) durch neue als heuristisch effizient identifizierte Elemente aktualisiert. Es wird nun versucht, die Elemente aus der LNDS in die GNDS einzufügen. Wird ein Element aus der LNDS in die GNDS eingefügt, so werden gleichzeitig alle Elemente aus der GNDS, welche von diesem neuen Element dominiert werden, aus der GNDS entfernt.

Zeilen 16+17: Als nächstes wird eine Auftragsfolge gesucht, welche speziell das Ziel der maximalen Terminüberschreitung minimiert. In dieser Darstellung ist λ_3 das Element des Gewichtevektors λ , welches der maximalen Terminüberschreitung zugeordnet ist. Es wird daher auf 1 gesetzt. Die anderen beiden Zielgrößen erhalten ein Gewicht von 0.

⁹²Die RAJ93-Heuristik bezeichnet hier die Heuristik von Rajendran (1993) (vgl. Rajendran (1993, S.66-69)).

Zeile 18: Mit Hilfe der RAJ93-Heuristik⁹³ wird unter alleiniger Berücksichtigung der maximalen Terminüberschreitung eine Auftragsfolge generiert.

Zeile 19: Für die folgende lokale Suche wird die lokale heuristisch effiziente Menge neu initialisiert. Sie enthält zunächst alleine die von der RAJ93-Heuristik in Zeile 18 ermittelte Auftragsfolge.

Zeile 20: Die von der RAJ93-Heuristik ermittelte Auftragsfolge wird durch lokale Suche verbessert.

Zeile 21: Die lokale Suche hat die lokale heuristisch effiziente Menge (LNDS) durch neue, als heuristisch effizient identifizierte Elemente aktualisiert. Es wird nun versucht, die Elemente aus der LNDS in die GNDS einzufügen. Wird ein Element aus der LNDS in die GNDS eingefügt, so werden gleichzeitig alle Elemente aus der GNDS, welche von diesem neuen Element dominiert werden, aus der GNDS entfernt.

6.3.1.3 Suchschema im Zielraum

Für die Suche nach Elementen der heuristisch effizienten Menge wird der dreidimensionale⁹⁴ Zielraum mit Hilfe des Parameters B in \hat{B} Suchbereiche eingeteilt: \hat{B} errechnet sich wie folgt:

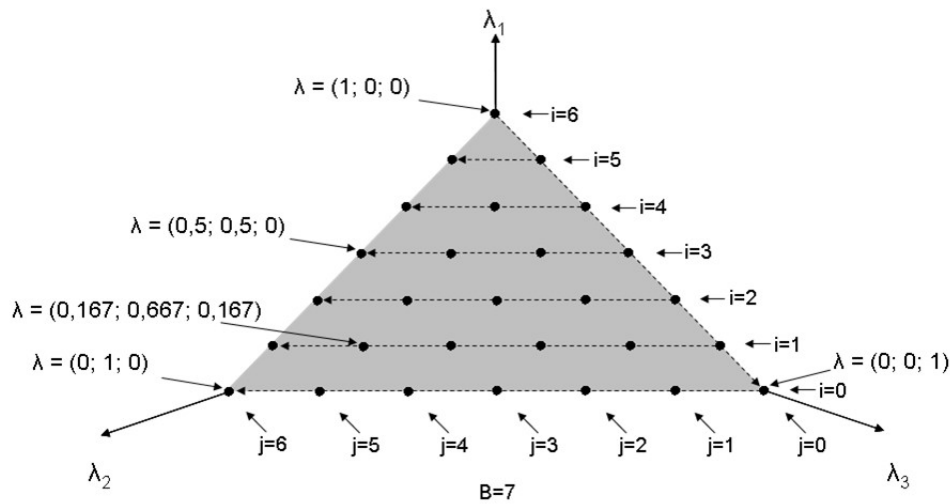
$$\hat{B} = \left(\frac{B}{2} + 0,5 \right) \cdot B \quad (6.13)$$

Abbildung 6.5 zeigt ein Beispiel für $B=7$.

Die einzelnen Bereiche werden, wie im Algorithmus 7 angegeben, generiert und abgesucht.

⁹³Die RAJ93-Heuristik bezeichnet hier die Heuristik von Rajendran (1993) (vgl. Rajendran (1993, S.66-69)).

⁹⁴Der Zielraum ist dreidimensional, da $K=3$ Ziele betrachtet werden.



$$\lambda = (\lambda_1; \lambda_2; \lambda_3)$$

Anzahl unterschiedlicher Suchregionen: $\hat{B} = (\frac{B}{2} + 0,5) \cdot B$; Hier $B = 7 \Rightarrow \hat{B} = 28$.

Abbildung 6.5: SCMO - Suchschema im Zielraum

Algorithmus 7 SCMO - Generierung der Suchbereiche

```

1: step =  $\frac{1}{B-1}$ ;
2: i = B-1;
3: while (i ≥ 0) {
4:   j = 0;
5:   while (j ≤ B-1-i) {
6:      $\lambda_1 = i \cdot \text{step}$ ;
7:      $\lambda_2 = j \cdot \text{step}$ ;
8:      $\lambda_3 = 1 - (\lambda_1 + \lambda_2)$ ;
9:     ... SCMO-SUCHE-REGION(  $\lambda$ , GNDS);
       (siehe Abschnitt 6.3.1.4)
10:    j=j+1;
11:  } // end while
12:  i=i-1;
13: } // end while

```

Die Suche beginnt demnach in der Region des Zielraums, in der die globalen Optima des mit λ_1 gewichteten Ziels liegen. Anschließend wird die Suche in Suchstreifen von oben nach unten fortgesetzt, wobei ein einzelner Streifen immer von rechts nach links abgesucht wird (siehe gestrichelte Pfeile in Abbildung 6.5). Für jeden Suchbereich wird die Prozedur SCMO-SUCHE-REGION aufgerufen, die im folgenden Abschnitt 6.3.1.4 beschrieben wird.

6.3.1.4 Die SCMO-SUCHE-REGION Prozedur

Diese Prozedur beschreibt die Suche in einem einzelnen Bereich des Zielraums und ist im Algorithmus 8 wiedergegeben. Die Prozedur beinhaltet die Konstruktion von Auftragsfolgen durch die Ameisen der Kolonie, die Verbesserung der Auftragsfolgen durch lokale Suche sowie die Verwaltung der Pheromoninformationen.

Zeile 1+2: Die endogenen kleinsten oberen Schranken werden ermittelt.

Zeilen 3-6 (Initialisierung der Pheromoninformation): Zunächst wird die Pheromonmatrix initialisiert. Dazu wird ein Element π^{init} aus der zu diesem Zeitpunkt vorliegenden heuristisch effizienten Menge bestimmt, das die Kompromißzielfunktion minimiert (Zeile 4). Anschließend wird die Pheromonmatrix mit der Pheromonmenge τ^{init} initialisiert (Zeile 5). Der Parameter Q wird beim Start der Heuristik extern vorgegeben und in dieser Arbeit immer gleich 1 gesetzt.

π^{init} ist gleichzeitig die Initialisierung für die global beste bekannte Lösung π^{glb} der Kolonie im betrachteten Zielgebiet des Zielraums. Mit dieser wird im Anschluß an die Initialisierung ein erstes Pheromonupdate nach Gleichung 6.8 auf Seite 161 durchgeführt (Zeile 6).⁹⁵

Zeile 7-8 (Initialisiere lokale NDS): Das Verfahren nutzt auch hier eine lokale heuristisch effiziente Menge (LNDS). Lokal bezieht sich dabei auf eine einzelne Zielregion im Zielraum. Die LNDS wird in jedem Zielbereich des Zielraums geleert und neu initialisiert. Sowohl die von den Ameisen konstruierten Auftragsfolgen als auch die während der lokalen Suche generierten Auftragsfolgen werden in diese lokale Menge eingefügt. Experimente (siehe Abschnitt 6.3.2) haben gezeigt, daß die Verwendung einer solchen Menge die Lösungsqualität verbessert.

Zeile 9+23: Für jede Region des Zielraums werden von der Kolonie für einen gegebenen Gewichtungsvektor *Iter*-mal Auftragsfolgen generiert und durch lokale Suche verbessert.

Zeile 10-16 (Konstruktion und Verbesserung von Auftragsfolgen): Jede Ameise der Kolonie erzeugt eine Auftragsfolge. Dabei wird die Konstruktionsvorschrift von Rajendran u. Ziegler (2002), wie in Algorithmus 3 auf Seite 160 dargestellt, unverändert übernommen (Zeile 11).⁹⁶ Es wird versucht, die konstruierte Auftragsfolge in die

⁹⁵ $\tau_{nh}(t-1)$ ist dabei der Pheromonwert nach Initialisierung der Pheromonmatrix.

⁹⁶Vgl. Rajendran u. Ziegler (2002, S.734).

Algorithmus 8 SCMO-SUCHE-REGION

```

// Parameterliste: ( $\lambda$ , GNDS)
//  $\lambda$     aktuelle Parameter des Kompromiß-
//          modells
// GNDS    übergebene (globale) heuristisch effiziente
//          Menge

// Weitere Parameter:
// Ants: Anzahl der Ameisen in der Kolonie
// Iter: Anzahl Iterationen
// LNDS: Lokale heuristisch effiziente Menge
//  $\pi^{glb}$ : Global beste Lösung der Kolonie
// Q: Parameter bei Pheromonaktualisierung, Wert bei Start der
//    Heuristik vorgegeben.
//  $\rho$ : Trail-Persistence-Rate, Wert bei Start der
//       Heuristik vorgegeben.
1: // Ermittle endogene kleinste obere Schranken
2:  $z_k^{EUB} = \min \{z_k(\pi) \mid \pi \in GNDS\}$ ;  $k=1(1)3$ ;

3: // Initialisierung der Pheromonmatrix
4:  $\pi^{glb} = \pi^{init} \in \operatorname{argmin} \{\Psi_\lambda(\pi) \mid \pi \in GNDS\}$ ;
5:  $\tau_{nh}^1 = \tau^{init} = \frac{Q}{\Psi_\lambda(\pi^{init})}$ ;  $n=1(1)N$ ;  $h=1(1)N$ ;
6: Führe Pheromonaktualisierung mit  $\pi^{glb}$  gemäß der
   Gleichung 6.8 auf Seite 161 aus,  $L(\pi^{ant}) = \Psi_\lambda(\pi^{init})$ ;

7: // Initialisiere lokale heuristisch effiziente
   Menge (LNDS)
8: LNDS =  $\emptyset$ ;

9: for(iter=1; iter<Iter, iter=iter+1) {
10:   for(ant=1; ant $\leq$ Ants; ant=ant+1) {
11:      $\pi^{ant} \leftarrow$  Konstruiere Auftragsfolge nach
        Algorithmus 3;
12:     Versuche  $\pi^{ant}$  in LNDS einzufügen;
13:     JIS( $\pi^{ant}$ ,  $\lambda$ , LNDS);
14:     JSS-RAPI( $\pi^{ant}$ ,  $\lambda$ , LNDS);
15:     if(  $\Psi_\lambda(\pi^{ant}) < \Psi_\lambda(\pi^{glb})$  )  $\pi^{glb} = \pi^{ant}$ ;
16:   } //end for
17:   Aktualisiere GNDS mit den Elementen aus LNDS;

18:   // Pheromonaktualisierung
19:   Führe Pheromonaktualisierung mit der besten Ameise
   der aktuellen Iteration gemäß Gleichung 6.8
   aus.  $L(\pi^{ant}) =$  Kompromißzielfunktionswert der
   besten Ameise der Iteration.
20:   Führe Pheromonaktualisierung mit  $\pi^{glb}$  gemäß
   Gleichungen 6.8 aus.  $L(\pi^{ant}) = \Psi_\lambda(\pi^{glb})$ 

21:   // Aktualisiere endogene kleinste obere Schranken
22:    $z_k^{EUB} = \min \{z_k(\pi) \mid \pi \in GNDS\}$ ;
23: } // end for

```

LNDS einzufügen. Wird die Auftragsfolge in die LNDS eingefügt, so werden gleichzeitig alle Elemente aus der LNDS, welche von diesem neuen Element dominiert werden, aus der LNDS entfernt (Zeile 12). Anschließend wird versucht, jede Auftragsfolge durch lokale Suche zu verbessern. Im Gegensatz zur Initialisierungsphase (vgl. Algorithmus 6) wird hier in Phase 1 der lokalen Suche nur noch JIS verwendet (Zeile 13). Die Phase 1 und die dort verwendeten Nachbarschaftsoperatoren wurden damit begründet, daß zunächst eine Annäherung an den durch die Parameter des Kompromißmodells eingestellten Bereich der effizienten Front stattfinden muß.⁹⁷ Mit zunehmender Laufzeit ist aber zu erwarten, daß sich die Suche immer mehr den Elementen der effizienten Front annähert, die im angesteuerten Bereich des Zielraums liegen. Die Phase 1 der lokalen Suche kann somit verkürzt werden. Daher wird auf JSS verzichtet. Dies bewirkt letztendlich auch eine Reduzierung der Rechenzeit. In Phase 2 der lokalen Suche wird wieder JSS-RAPI verwendet (Zeile 14). In Zeile 15 wird schließlich noch überprüft, ob π^{glb} verbessert wurde.

Zeile 17 (Aktualisierung GNDS): Nachdem alle Ameisen eine Auftragsfolge generiert haben und diese verbessert wurde, wird versucht, die Elemente der lokalen heuristisch effizienten Menge in die globale heuristisch effiziente Menge einzufügen. Wird ein Element aus der LNDS in die GNDS eingefügt, so werden gleichzeitig alle Elemente der GNDS entfernt, die von diesem neuen Element dominiert werden.

Die in der LNDS enthaltenen Elemente werden jedoch nicht aus der lokalen heuristisch effizienten Menge entfernt.

Zeile 18-20 (Pheromonaktualisierung): Zunächst wird eine Pheromonaktualisierung durch die beste Ameise der Iteration durchgeführt. Anschließend wird eine Aktualisierung durch die bisher beste gefundene Lösung, bezogen auf den Kompromißzielfunktionswert, durchgeführt. In beiden Fällen wird $L(\pi^{ant})$ gleich dem Kompromißzielfunktionswert der für die Aktualisierung verwendeten Auftragsfolge gesetzt. Die Parameter Q und ρ werden zu Beginn des Verfahrens vorgegeben. Die für sie verwendeten Werte werden in Abschnitt 6.3.2.2 diskutiert.

Ziel des doppelten Pheromonupdate ist es, sowohl Informationen über neue, vielversprechende Bereiche des Lösungsraum zu berücksichti-

⁹⁷Vgl. die Ausführungen in Abschnitt 6.2.1.

gen (1. Update), aber gleichzeitig über das 2. Update Informationen über die bisher beste bekannte Auftragsfolge π^{glb} nicht zu vernachlässigen.

Zeile 21-22: Die endogenen kleinsten oberen Schranken werden aktualisiert.

6.3.2 Experimentelle Analyse

6.3.2.1 Aufbau der Experimente und Bewertungsmethoden

In den folgenden Abschnitten werden Experimente beschrieben, deren Ergebnisse zur beschriebenen Ausgestaltung des SCMO-Ansatzes geführt haben. Des Weiteren wird der Einfluß von unterschiedlichen Parametereinstellungen auf die Lösungsqualität des SCMO-Ansatzes untersucht. Dazu werden die drei Qualitätsmaße SQ_l , DUB_l^k sowie AEX_l^k herangezogen.

Als Testprobleme für die durchgeführten Experimente werden die ersten 90 Probleme der in Taillard (1993) definierten Flow Shop Probleme verwendet.⁹⁸ Es werden Probleme mit 20, 50 und 100 Aufträgen sowie 5, 10 und 20 Maschinen betrachtet. Jedes dieser Probleme wird von einem Lösungsverfahren bzw. einer konkreten Konfiguration der Parameter einer Heuristik v $Runs$ -mal gelöst. Für jedes getestete Lösungsverfahren oder für jede getestete Konfiguration wird die durchschnittliche Lösungsgüte über die $Runs$ Läufe ermittelt. Bei der Bestimmung der $TMPS$ werden alle $Runs$ Lösungen jedes der V in den Vergleich aufgenommenen Lösungsverfahren berücksichtigt.

Ist $SQ_{v,run}$ der S-Maß-Quotient der Lösung von Lösungsverfahren v in Rechenlauf run ($run=1(1)Runs$), so ergibt sich für ein Lösungsverfahren der durchschnittliche S-Maß-Quotient der $Runs$ Lösungen als:

$$SQ_v^\mu = \frac{1}{Runs} \cdot \sum_{run=1}^{Runs} SQ_{v,run} \quad (6.14)$$

Analog ergeben sich:⁹⁹

$$DUB_v^{\mu k} = \frac{1}{Runs} \cdot \sum_{run=1}^{Runs} DUB_{v,run}^k \quad (6.15)$$

⁹⁸Vgl. Taillard (1993, S.280).

⁹⁹Für keine der in dieser Arbeit betrachteten Probleminstanzen wurde eine Lösung ermittelt, welche eine maximale Terminüberschreitung von Null enthält (vgl. dazu auch die Ergebnisse in Abschnitt 6.5.2.2). Daher wurde bei der Berechnung von $DUB_{v,run}^k$ der Wert Null für ζ^k , $k = 1(1)K$ gewählt.

$$AEX_v^{\mu k} = \frac{1}{Runs} \cdot \sum_{run=1}^{Runs} AEX_{v,run}^k \quad (6.16)$$

Neben dem durchschnittlichen Wert einer Auswertungsgröße für die *Runs* Läufe eines Lösungsverfahrens *v* wird auch die Standardabweichung der Auswertungsgrößen ermittelt:

$$SQ_v^{\sigma} = \sqrt{\frac{1}{Runs} \cdot \sum_{run=1}^{Runs} (SQ_{v,run}^{\mu} - SQ_v^{\mu})^2} \quad (6.17)$$

$$DUB_v^{\sigma k} = \sqrt{\frac{1}{Runs} \cdot \sum_{run=1}^{Runs} (DUB_{v,run}^k - DUB_v^{\mu k})^2} \quad (6.18)$$

$$AEX_v^{\sigma k} = \sqrt{\frac{1}{Runs} \cdot \sum_{run=1}^{Runs} (AEX_{v,run} - AEX_v^{\mu})^2} \quad (6.19)$$

Zur Berechnung des DUB^k -Maßes ist die Vorgabe von exogenen kleinsten oberen Schranken nötig (vgl. Abschnitt 4.2.4). Die in dieser Arbeit verwendeten kleinsten oberen Schranken für die einzelnen Ziele werden aus den Ergebnissen unterschiedlicher Heuristiken ermittelt. Das Vorgehen bei der Ermittlung dieser kleinsten oberen Schranken und die Werte sind in Anhang C für die betrachteten 90 Benchmarkprobleme von Taillard (1993) angegeben.¹⁰⁰

Wie bereits in Kapitel 4 angesprochen, wird es bei der Verwendung mehrerer Qualitätsmaße in der Regel nicht möglich sein, die beurteilten Lösungen in eine eindeutige Rangfolge bezüglich ihrer Lösungsqualität zu bringen. Die Qualitätsmaße der einzelnen Lösungen bilden selbst wieder Vektoren, welche zu vergleichen sind. Als Beurteilungsmaßstab bleibt somit alleine das Effizienzkriterium. Das Ziel der folgenden Experimente ist es daher nicht, eine allgemein und objektiv beste Strategie zu identifizieren, sondern die Auswirkungen unterschiedlicher Ausgestaltungsmöglichkeiten und Parametereinstellungen auf die Lösungsqualität der betrachteten Aspekte aufzuzeigen. Um die Anzahl der überprüften Varianten zu begrenzen, ist es notwendig, bestimmte Konfigurationen von weiterführenden Experimenten auszuschließen bzw. einzelne Varianten für weiterführende Experimente auszuwählen. Sofern diese in dieser Arbeit getroffenen Entscheidungen nicht allein aus der Anwendung des Effizienzkriteriums resultieren, können sie aus dem oben angesprochenen Grund keinen Anspruch auf Objektivität und Allgemeingültigkeit erheben, sondern unter-

¹⁰⁰Vgl. Taillard (1993, S.280.).

liegen vielmehr subjektiven Einschätzungen des Verfassers.

Neben der Auswertung der Zielgrößen wird bei einigen Experimenten auch die benötigte Rechenzeit für die Beurteilung eines Verfahrens verwendet. Die Rechenzeit wird hier in Form von CPU-Ticks und nicht in Sekunden erfaßt.¹⁰¹

Alle in den Experimenten verwendeten Algorithmen wurden in C++ programmiert. Ausgeführt wurden die Experimente auf zwei Rechnern mit unterschiedlicher Hardwareausstattung. Der erste Rechner war mit einem Intel Pentium IV Prozessor mit 2,4 GHz und 512 MB Arbeitsspeicher ausgerüstet (Rechnerkonfiguration 1). Der zweite Rechner hatte einen Mobile Intel Pentium IV mit 2 GHz und ebenfalls 512 MB Arbeitsspeicher. (Rechnerkonfiguration 2). Die unterschiedliche Konfiguration hat einen Einfluß auf die Rechenzeit der Heuristiken. Sofern die Rechenzeit in der Beurteilung der Ergebnisse diskutiert wird, wurden die einzelnen Experimente jeweils nur auf einer Rechnerkonfiguration durchgeführt. Hier wurden jeweils nur solche Ergebnisse miteinander verglichen, welche auf der gleichen Rechnerkonfiguration ermittelt wurden.

6.3.2.2 Ausgangsparametereinstellungen

Die einzelnen im folgenden durchgeführten Experimente variieren jeweils maximal zwei Faktoren bzw. Parameter des SCMO-Ansatzes. In diesem Abschnitt werden daher die Ausgangsparametereinstellungen angegeben. In den einzelnen Experimenten werden anschließend nur noch die Abweichungen von diesen Einstellungen angeführt.

range: In der lokalen Suche kann die Anzahl der berücksichtigten Positionen vor und nach dem betrachteten Auftrag über den Parameter *range* eingestellt werden. Dies soll insbesondere die Rechenzeit der lokalen Suche beschränken und in praktikablen Grenzen halten. Der Parameter wird hier in Abhängigkeit von der Problemgröße, also der Auftragszahl N , festgelegt.¹⁰² Weiterhin wird eine Staffelung vorgenommen. Die lokale Suche besteht in der Initialisierungsphase aus drei Stufen, ansonsten aus zwei. Dabei nimmt der Wert von *range* mit jeder Stufe ab. Die im Folgenden dargestellten Gleichungen zur Berechnung von *range* haben sich in Vorexperimenten als guter Kompromiß zwischen benötigter Rechenzeit und

¹⁰¹Dabei kann eine Umrechnung von CPU-Ticks in Sekunden grob vorgenommen werden, indem die CPU-Ticks durch den Faktor 1000 geteilt werden.

¹⁰²Die Problemgröße wird hier allein über die Anzahl der Aufträge N definiert. Die Größe des Lösungsraums Π ist allein von N abhängig und steigt exponentiell mit N . Auch die Anzahl der Maschinen M hat Einfluß auf die Rechenzeit. Diese steigt jedoch lediglich linear mit steigendem M .

Lösungsgüte herausgestellt.

In der ersten Stufe wird $range$ nach folgender Gleichung bestimmt:

$$range = \left\lceil \lg(N) \cdot \frac{275}{N} \right\rceil \quad (6.20)$$

In der zweiten Stufe nach:

$$range = \left\lceil \lg(N) \cdot \frac{250}{N} \right\rceil \quad (6.21)$$

In der dritten Stufe der Initialisierungsphase wird $range$ schließlich wie folgt bestimmt:

$$range = \left\lceil \lg(N) \cdot \frac{225}{N} \right\rceil \quad (6.22)$$

Zuordnung der Ziele zu den Elementen des Gewichtungsvektors

λ : Eine Variation der Zuordnung beeinflusst, in welcher Reihenfolge der Zielraum abgesucht wird. Es kann bestimmt werden, nach welchem Ziel zu Beginn individuell optimiert wird und für welches Ziel erst am Ende der Laufzeit die individuelle Optimierung durchgeführt wird. Die Analyse der unterschiedlichen Zuordnungsmöglichkeiten auf die Lösungsqualität wird in Experiment 1 vorgenommen. Aufgrund der dort erhaltenen Ergebnisse wird die Zuordnung $C_{max} \rightarrow \lambda_1$, $F_{mid} \rightarrow \lambda_2$ sowie $T_{max} \rightarrow \lambda_3$ gewählt und für alle folgenden Experimente beibehalten.

Anzahl der Suchbereiche im Zielraum, Parameter B : Soweit nichts anderes angegeben ist, wird der Parameter B auf den Wert 7 gesetzt. In Experiment 3 wird der Einfluß unterschiedlicher Werte des Parameters auf die Lösungsqualität und die Rechenzeit untersucht.

Anzahl der Iterationen in jedem Suchbereich, $Iter$: Der Parameter steuert die Anzahl der Iterationen einer Kolonie innerhalb einer Suchregion. Als Standardeinstellung ist er auf 10 gesetzt.

Die Anzahl der Ameisen $Ants$: Die Anzahl der Ameisen in einer Kolonie. Hier wird nur eine einzelne Ameise pro Kolonie verwendet. In Experiment 4 wird analysiert, ob es vorteilhaft ist, mehrere Iterationen in einem Suchbereich auszuführen oder aber mehrere Ameisen zu verwenden.

Die Größe der Kandidatenliste *cand*: Die Größe der Kandidatenliste wird festgelegt als:¹⁰³

$$cand = \left\lfloor 4 + \frac{N}{20} \right\rfloor \quad (6.23)$$

Die Parameter der erweiterten Pseudo Random Proportionate Rule p_1 und p_2 : Den Parametern der erweiterten Pseudo Random Proportionate Rule werden die in Rajendran u. Ziegler (2002) verwendeten Werte zugeordnet: $p_1 = 0,4$; $p_2 = 0,8$.¹⁰⁴

Die Trail-Persistence-Rate ρ und Parameter Q : Die Trail-Persistence-Rate ρ wird auf den Wert 0,75 gesetzt. Die Verdunstungsrate entspricht damit 0,25. In Experiment 5 wird der Einfluß unterschiedlicher Werte der Trail-Persistence-Rate auf die Lösungsqualität untersucht. Der Parameter Q wird auf den Wert 1 gesetzt.

Werte für ζ_k^{NORM} : Die beiden Zielgrößen Zykluszeit und mittlere Durchlaufzeit können nur positive Werte annehmen, daher werden ζ_{Cmax}^{NORM} und ζ_{Fmid}^{NORM} gleich Null gesetzt. Die Zielgröße der maximalen Terminüberschreitung kann den Wert Null annehmen. Der kleinste mögliche positive Wert ist, da alle Bearbeitungszeiten und Fertigstellungstermine ganzzahlig sind, 1. Für ζ_{Tmax}^{NORM} wird daher der Wert 0,9 gewählt.¹⁰⁵

Runs: Die Anzahl der *Runs* wird in allen Experimenten auf 10 gesetzt.

¹⁰³Diese Einstellung hat sich in Vorexperimenten als gute Einstellung dargestellt.

¹⁰⁴Vgl. Rajendran u. Ziegler (2002, S.734).

¹⁰⁵Für keine der in dieser Arbeit betrachteten Probleminstanzen wurde eine Lösung ermittelt, welche eine maximale Terminüberschreitung von Null enthält (vgl. dazu auch die Ergebnisse in Abschnitt 6.5.2.2). Daher hat die Wahl des Wertes für ζ_{Tmax}^{NORM} keinen Einfluß auf die von den hier untersuchten SCMO-Ansatz und MCMO-Ansätzen ermittelten Lösungen.

6.3.2.3 Experiment 1: Systematische Suche im Zielraum und Verwendung der LNDS

In diesem Experiment wird untersucht, welche Zuordnung der drei Ziele zu den Elementen des Gewichtungsvektors λ vorteilhaft ist. Die bei drei Zielen existierenden Möglichkeiten sind in Tabelle 6.2 angegeben.

Variante	λ_1	λ_2	λ_3
1	C_{max}	F_{mid}	T_{max}
2	C_{max}	T_{max}	F_{mid}
3	F_{mid}	C_{max}	T_{max}
4	F_{mid}	T_{max}	C_{max}
5	T_{max}	C_{max}	F_{mid}
6	T_{max}	F_{mid}	C_{max}

Tabelle 6.2: SCMO - Alternative Suchrichtungen im Zielraum

Gleichzeitig soll betrachtet werden, wie sich die Lösungsgüte bei den einzelnen Varianten unterscheidet, falls zum einen eine lokale heuristisch effiziente Menge verwendet und zum anderen auf diese verzichtet wird und alleine mit der globalen heuristisch effizienten Menge gearbeitet wird. Im Algorithmus 8 wird also anstelle der LNDS in den Zeilen 13 und 14 die GNDS an die lokalen Suchverfahren übergeben.¹⁰⁶ Auch in der Initialisierungsphase in Algorithmus 6 werden keine lokalen heuristisch effizienten Mengen mehr verwendet. Vielmehr werden die von der NEH-Heuristik und RAJ93-Heuristik generierten Auftragsfolgen stets in die globale heuristisch effiziente Menge eingefügt. Diese wird anschließend auch an die lokalen Suchverfahren übergeben. Die Ergebnisse sind in den Tabellen 6.3 bis 6.7 wiedergegeben.¹⁰⁷

SQ-Maß: Die Ergebnisse des SQ-Maßes in Tabelle 6.3 zeigen, daß die Verwendung einer lokalen heuristisch effizienten Menge die Lösungsqualität in den meisten Fällen steigen läßt. Mit Ausnahme der Probleme mit 100 Aufträgen in den Varianten 3 und 4 hat die Konfiguration mit LNDS größere Werte für SQ_v^μ als die Variante ohne LNDS. Weiterhin hat (außer für diese Fälle und die Variante 5) die Konfiguration mit LNDS auch die niedrigere Standardabweichung und dominiert somit die Konfiguration ohne LNDS.

Bei der Verwendung der LNDS fällt auf, daß die Werte für SQ_v^μ innerhalb einer Suchvariante für steigende Problemgrößen systematisch sinken. Eine Beobachtung, die sich bei der Konfiguration ohne LNDS nicht machen

¹⁰⁶Die Zeilen 7, 8 und 17 sind damit überflüssig.

¹⁰⁷Für das Experiment wurde die Rechnerkonfiguration 1 verwendet.

Variante	N	Mit LNDS		Ohne LNDS	
		SQ^μ	SQ^σ	SQ^μ	SQ^σ
1	20	0,8461	0,0368	0,7669	0,0526
	50	0,7415	0,0496	0,5946	0,0565
	100	0,6387	0,0534	0,6042	0,0531
	Avg	0,7421	0,0466	0,6552	0,0541
2	20	0,8469	0,0396	0,7482	0,0517
	50	0,7331	0,0466	0,5512	0,0612
	100	0,6281	0,0475	0,5518	0,0529
	Avg	0,7360	0,0446	0,6171	0,0553
3	20	0,8515	0,0378	0,7649	0,0483
	50	0,7486	0,0523	0,6120	0,0573
	100	0,6179	0,0550	0,6397	0,0598
	Avg	0,7393	0,0484	0,6722	0,0551
4	20	0,8459	0,0376	0,7659	0,0523
	50	0,7488	0,0537	0,6104	0,0571
	100	0,6150	0,0534	0,6342	0,0603
	Avg	0,7366	0,0482	0,6701	0,0565
5	20	0,7066	0,0420	0,5969	0,0414
	50	0,5409	0,0466	0,3769	0,0344
	100	0,4693	0,0437	0,3743	0,0285
	Avg	0,5723	0,0441	0,4494	0,0348
6	20	0,8441	0,0385	0,7528	0,0522
	50	0,7327	0,0529	0,5864	0,0629
	100	0,6402	0,0496	0,5938	0,0547
	Avg	0,7390	0,0470	0,6443	0,0566

Tabelle 6.3: SCMO - Exp. 1: SQ -Maß

läßt.

Eine Aussage, welche Suchvariante bei Verwendung einer lokalen heuristisch effizienten Menge die beste ist, ist nicht eindeutig möglich. Wird alleine auf SQ_v^μ abgestellt, so ist für Probleme mit 20 Aufträgen die Variante 3 die beste Möglichkeit. Für Probleme mit 50 Aufträgen hingegen Variante 4 und für 100 Aufträge Variante 6. Im Durchschnitt ist Variante 1 die beste.

DUB^k -Maß: Die Werte für das DUB^k -Maß werden für die Konfiguration ohne LNDS in Tabelle 6.4 und für die Konfiguration mit LNDS in Tabelle 6.5 wiedergegeben.

Zunächst sollen die Ergebnisse der Konfiguration ohne LNDS betrachtet werden. Dort läßt sich beobachten, daß für ein Ziel immer eine Variante den kleinsten Wert für $DUB_v^{\mu k}$ aufweist, welche die Suche mit diesem Ziel beginnt. Dies bedeutet zunächst, daß die endogenen kleinsten oberen Schranken nach der Initialisierungsphase verbessert werden können. Diese Verbesserung fällt stärker aus, wenn unmittelbar nach der Initialisie-

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0096	0,0035	1,0096	0,0034	1,0285	0,0115
	50	1,0177	0,0028	1,0269	0,0021	1,0244	0,0067
	100	1,0128	0,0016	1,0295	0,0010	1,0115	0,0036
	Avg	1,0134	0,0026	1,0220	0,0022	1,0215	0,0073
2	20	1,0097	0,0035	1,0107	0,0033	1,0268	0,0115
	50	1,0173	0,0027	1,0291	0,0011	1,0240	0,0064
	100	1,0127	0,0016	1,0309	0,0004	1,0123	0,0039
	Avg	1,0132	0,0026	1,0236	0,0016	1,0210	0,0073
3	20	1,0135	0,0034	1,0070	0,0038	1,0305	0,0117
	50	1,0232	0,0020	1,0228	0,0033	1,0281	0,0066
	100	1,0152	0,0014	1,0271	0,0014	1,0121	0,0036
	Avg	1,0173	0,0022	1,0190	0,0028	1,0236	0,0073
4	20	1,0139	0,0037	1,0067	0,0037	1,0259	0,0107
	50	1,0233	0,0019	1,0232	0,0033	1,0259	0,0064
	100	1,0154	0,0012	1,0272	0,0013	1,0110	0,0037
	Avg	1,0175	0,0022	1,0190	0,0028	1,0209	0,0070
5	20	1,0115	0,0034	1,0166	0,0004	1,0575	0,0080
	50	1,0189	0,0025	1,0315	0,0002	1,0482	0,0057
	100	1,0140	0,0013	1,0313	0,0000	1,0231	0,0029
	Avg	1,0148	0,0024	1,0265	0,0002	1,0430	0,0055
6	20	1,0133	0,0033	1,0092	0,0037	1,0234	0,0109
	50	1,0205	0,0030	1,0276	0,0019	1,0197	0,0067
	100	1,0145	0,0013	1,0299	0,0007	1,0099	0,0037
	Avg	1,0161	0,0025	1,0222	0,0021	1,0177	0,0071

Tabelle 6.4: SCMO - Exp. 1: Variante ohne LNDS: DUB^k -Maß

rungsphase in der Region des Zielraums gesucht wird, in der das Minimum eines Ziels liegt. Da eine einzelne homogene Kolonie eingesetzt wird, kann nur das individuelle Optimum eines der drei Ziele unmittelbar nach der Initialisierungsphase gesucht werden. Bei den verwendeten Suchvarianten wird außerdem das individuelle Optimum eines Ziels erst im letzten Schritt gezielt gesucht. Neben der Initialisierung der Kolonien hat insbesondere die an die lokalen Suchverfahren übergebene heuristisch effiziente Menge einen Einfluß auf die Suche im Lösungsraum. Je besser diese Menge ist, desto weniger Züge werden aufgrund der fehlenden heuristischen Effizienz ausgeführt und die rekursive API Suche nicht gestartet. Die lokale Suche ist damit aufgrund der bereits in der heuristisch effizienten Menge enthaltenen Lösungen blockiert. Die Verwendung einer LNDS soll diese Blockade reduzieren, indem zu Beginn der Suche in einem Bereich des Zielraums eine leere heuristisch effiziente Menge verwendet wird und damit in früheren Schritten ermittelte heuristisch effiziente Lösungen die Suche nicht

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0067	0,0029	1,0031	0,0023	1,0126	0,0075
	50	1,0152	0,0024	1,0080	0,0035	1,0132	0,0061
	100	1,0139	0,0012	1,0118	0,0027	1,0108	0,0040
	Avg	1,0119	0,0022	1,0076	0,0028	1,0122	0,0059
2	20	1,0070	0,0029	1,0031	0,0023	1,0128	0,0077
	50	1,0152	0,0025	1,0079	0,0029	1,0143	0,0060
	100	1,0139	0,0012	1,0122	0,0028	1,0114	0,0037
	Avg	1,0120	0,0022	1,0077	0,0027	1,0128	0,0058
3	20	1,0065	0,0027	1,0033	0,0021	1,0127	0,0073
	50	1,0154	0,0028	1,0077	0,0030	1,0155	0,0061
	100	1,0139	0,0012	1,0121	0,0028	1,0129	0,0041
	Avg	1,0120	0,0022	1,0077	0,0026	1,0137	0,0058
4	20	1,0066	0,0031	1,0033	0,0021	1,0145	0,0073
	50	1,0155	0,0025	1,0077	0,0031	1,0170	0,0066
	100	1,0139	0,0012	1,0121	0,0028	1,0141	0,0036
	Avg	1,0120	0,0023	1,0077	0,0026	1,0152	0,0058
5	20	1,0052	0,0027	1,0156	0,0013	1,0440	0,0078
	50	1,0131	0,0026	1,0301	0,0006	1,0355	0,0057
	100	1,0124	0,0012	1,0310	0,0001	1,0207	0,0030
	Avg	1,0102	0,0021	1,0256	0,0007	1,0334	0,0055
6	20	1,0066	0,0031	1,0032	0,0024	1,0151	0,0085
	50	1,0156	0,0027	1,0082	0,0035	1,0178	0,0056
	100	1,0137	0,0014	1,0118	0,0028	1,0120	0,0035
	Avg	1,0120	0,0024	1,0077	0,0029	1,0150	0,0058

Tabelle 6.5: SCMO - Exp. 1: Variante mit LNDS: DUB^k -Maß

mehr blockieren können.

Die Analyse der Ergebnisse der Konfiguration mit einer LNDS zeigt, daß nun für die einzelnen Ziele in der Regel eine Konfiguration am besten ist, die keine individuelle Optimierung des Ziels zu Beginn der Suche durchführt.¹⁰⁸ Dies wird insbesondere bei den Zielen C_{max} und T_{max} deutlich. Für die Zykluszeit ist Variante 5 gut, während für die maximale Terminüberschreitung und die mittlere Durchlaufzeit Variante 1 gut ist. Der Unterschied zwischen den einzelnen Varianten für eine Problemgröße ist, im Vergleich zur Alternative ohne LNDS, geringer.

AEX^k -Maß: Abschließend sollen noch die Ergebnisse für das AEX^k -Maß ausgewertet werden. Dabei ist zu beachten, daß dies nicht unabhängig vom DUB^k -Maß ist. So ist es nicht überraschend, daß auch hier die Konfiguration mit der LNDS überwiegend bessere Werte für $AEX_v^{\mu k}$

¹⁰⁸Nur bei der mittleren Durchlaufzeit und Problemen mit 50 Aufträgen haben die Varianten 3 und 4 weiterhin den besten Wert für $DUB_v^{\mu Fmid}$.

Variante	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0217	0,0583	0,0067	0,0200	0,0033	0,0100
	50	0,0033	0,0100	0,0000	0,0000	0,0000	0,0000
	100	0,0189	0,0544	0,0000	0,0000	0,0100	0,0300
	Avg	0,0146	0,0409	0,0022	0,0067	0,0044	0,0133
2	20	0,0100	0,0300	0,0033	0,0100	0,0033	0,0100
	50	0,0000	0,0000	0,0000	0,0000	0,0067	0,0200
	100	0,0164	0,0493	0,0000	0,0000	0,0083	0,0250
	Avg	0,0088	0,0264	0,0011	0,0033	0,0061	0,0183
3	20	0,0033	0,0067	0,0433	0,1067	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0033	0,0100
	100	0,0033	0,0100	0,0000	0,0000	0,0033	0,0100
	Avg	0,0022	0,0056	0,0144	0,0356	0,0022	0,0067
4	20	0,0033	0,0100	0,0400	0,0839	0,0078	0,0167
	50	0,0000	0,0000	0,0000	0,0000	0,0033	0,0100
	100	0,0000	0,0000	0,0000	0,0000	0,0100	0,0300
	Avg	0,0011	0,0033	0,0133	0,0280	0,0070	0,0189
5	20	0,0033	0,0100	0,0000	0,0000	0,0000	0,0000
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0100	0,0276	0,0000	0,0000	0,0000	0,0000
	Avg	0,0044	0,0125	0,0000	0,0000	0,0000	0,0000
6	20	0,0000	0,0000	0,0267	0,0733	0,0078	0,0233
	50	0,0000	0,0000	0,0000	0,0000	0,0033	0,0100
	100	0,0033	0,0100	0,0000	0,0000	0,0200	0,0600
	Avg	0,0011	0,0033	0,0089	0,0244	0,0104	0,0311

Tabelle 6.6: SCMO - Exp. 1: Variante ohne LNDS: AEX^k -Maß

aufweist. Ausnahmen gibt es aber bei Problemgrößen mit 100 Aufträgen. Bei der Zykluszeit sind die Varianten 1, 2 und 3 und bei der maximalen Terminüberschreitung die Varianten 2, 4 und 6 bei Verzicht auf die LNDS besser als die entsprechenden Konfigurationen mit LNDS. Werden für die Varianten mit LNDS die durchschnittlichen Werte von $AEX_v^{\mu^k}$ betrachtet, so erweist sich auch hier für die Zykluszeit Variante 5 als die beste. Für die mittlere Durchlaufzeit ist es Variante 6 und für die maximale Terminüberschreitung Variante 1. Häufig sind die Werte Null in den Tabellen zu finden. Diese treten auf, da nur sehr wenige der *Runs* heuristisch effizienten Mengen einer Konfiguration v einen positiven Wert für $AEX_v^{\mu^k}$ aufweisen. Die Anzahl der Lösungen mit einem positiven Wert reichen nicht aus, damit der Durchschnittswert von $AEX_v^{\mu^k}$ bei vier Nachkommastellen einen positiven Wert aufweist.

Insgesamt hat die Einführung einer lokalen heuristisch effizienten

Variante	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0417	0,0840	0,2233	0,2469	0,0812	0,1759
	50	0,0044	0,0133	0,0367	0,1100	0,0233	0,0700
	100	0,0014	0,0043	0,0333	0,1000	0,0108	0,0325
	Avg	0,0159	0,0339	0,0978	0,1523	0,0385	0,0928
2	20	0,0450	0,0813	0,2083	0,2863	0,0800	0,1616
	50	0,0053	0,0158	0,0367	0,1100	0,0153	0,0458
	100	0,0014	0,0043	0,0267	0,0800	0,0058	0,0175
	Avg	0,0172	0,0338	0,0906	0,1588	0,0337	0,0750
3	20	0,0383	0,0784	0,2117	0,2545	0,0567	0,1254
	50	0,0083	0,0250	0,0200	0,0600	0,0156	0,0467
	100	0,0008	0,0025	0,0300	0,0900	0,0117	0,0350
	Avg	0,0158	0,0353	0,0872	0,1348	0,0280	0,0690
4	20	0,0591	0,1242	0,2150	0,2675	0,0704	0,1251
	50	0,0125	0,0375	0,0200	0,0600	0,0192	0,0575
	100	0,0100	0,0300	0,0267	0,0800	0,0033	0,0100
	Avg	0,0272	0,0639	0,0872	0,1358	0,0310	0,0642
5	20	0,1147	0,1915	0,0000	0,0000	0,0100	0,0233
	50	0,0706	0,1993	0,0000	0,0000	0,0100	0,0300
	100	0,0330	0,0882	0,0000	0,0000	0,0033	0,0100
	Avg	0,0727	0,1597	0,0000	0,0000	0,0078	0,0211
6	20	0,0563	0,1133	0,2400	0,2632	0,0633	0,1555
	50	0,0042	0,0125	0,0333	0,1000	0,0033	0,0100
	100	0,0100	0,0300	0,0433	0,1300	0,0133	0,0400
	Avg	0,0235	0,0519	0,1056	0,1644	0,0266	0,0685

Tabelle 6.7: SCMO - Exp. 1: Variante mit LNDS: AEX^k -Maß

Menge einen positiven Einfluß auf die Lösungsqualität. Dennoch kann sie die Lösungsqualität der einzelnen Varianten bzgl. $DUB_v^{\mu k}$ nicht ausreichend angleichen. Es ist zu bedenken, daß auch die Entwicklung der endogenen kleinsten oberen Schranken über die Laufzeit einen Einfluß hat. Diese müßten für Ziele, die erst zu einem späteren Zeitpunkt individuell optimiert werden, tendenziell schlechter sein. Um dem entgegenzuwirken, müßte sichergestellt werden, daß unmittelbar nach der Initialisierungsphase der Heuristik alle Regionen im Zielraum, welche die individuellen Optima der Ziele enthalten, betrachtet werden. Dies fordert die vollständige Neudefinition der Suchstrategie. Diese Neudefinition soll hier für den SCMO-Ansatz nicht vorgenommen werden. Die Überlegungen fließen jedoch in die Ausgestaltung des zweiten beschriebenen Ansatzes ein, dem MCMO-Ansatz.

In den folgenden Experimenten wird die Verwendung einer LNDS

beibehalten. Von den sechs überprüften Varianten, die eine LNDS verwenden, wird Variante 1 für die weitere Verwendung ausgewählt. Für SQ_v^μ , $DUB_v^{\mu Fmid}$, $DUB_v^{\mu Tmax}$ und $AEX_v^{\mu Tmax}$ werden mit dieser Variante im Durchschnitt über alle Problemgrößen die besten Ergebnisse erzielt. Für $DUB_v^{\mu Cmax}$ und $AEX_v^{\mu Fmid}$ ist sie im Durchschnitt die zweitbeste Variante. Nur bei $AEX_v^{\mu Cmax}$ ist sie deutlich schlechter als die beste Variante 5.¹⁰⁹

6.3.2.4 Experiment 2: Einfluß der Neuinitialisierung der Pheromonmatrix

In diesem Experiment wird der Einfluß der Neuinitialisierung der Pheromonmatrix beim Wechsel des Suchbereichs der Kolonie untersucht. Die Pheromonmatrix wird nur einmal zu Beginn der Suche im ersten Suchbereich initialisiert und anschließend kontinuierlich fortgeschrieben. Anstelle einer verbundenen Multi Start Strategie wird also eine SSDP-Strategie verwendet. Die Ergebnisse sind in den Tabellen 6.8 bis 6.10 angegeben.¹¹⁰

N	Mit Neuinit.		Ohne Neuinit.	
	SQ^μ	SQ^σ	SQ^μ	SQ^σ
20	0,8626	0,0429	0,7789	0,0606
50	0,7806	0,0619	0,6283	0,0696
100	0,6688	0,0661	0,6681	0,0653
Avg	0,7707	0,0570	0,6918	0,0652

Tabelle 6.8: SCMO - Exp. 2: SQ -Maß

SQ -Maß: Die Variante mit der Neuinitialisierung der Pheromonmatrix dominiert die Variante ohne Neuinitialisierung bei Problemen mit 20 und 50 Aufträgen. Bei Problemen mit 100 Aufträgen hat sie den besseren Wert für SQ_v^μ .

DUB^k -Maß: Beim DUB^k -Maß in Tabelle 6.9 dominiert die Strategie mit Neuinitialisierung bei Problemen mit 20 Aufträgen die Strategie ohne Neuinitialisierung für alle drei Zielsetzungen. Bei 50 Aufträgen dominiert sie bezüglich der Zykluszeit und der maximalen Terminüberschreitung. Bei der mittleren Durchlaufzeit hat sie den besseren $DUB_v^{\mu k}$ -Wert. Für große Probleme mit 100 Aufträgen liefert die Strategie ohne Neuinitialisierung

¹⁰⁹Es sei an dieser Stelle nochmals darauf hingewiesen, daß sich diese Entscheidung nicht vollständig objektiv begründen läßt.

¹¹⁰Für das Experiment wurde die Rechnerkonfiguration 1 verwendet.

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
mit	20	1,0067	0,0029	1,0030	0,0023	1,0105	0,0075
Neunit.	50	1,0152	0,0024	1,0071	0,0035	1,0100	0,0061
	100	1,0138	0,0012	1,0116	0,0027	1,0087	0,0040
	Avg	1,0119	0,0022	1,0073	0,0028	1,0097	0,0059
ohne	20	1,0093	0,0035	1,0094	0,0032	1,0242	0,0123
Neunit.	50	1,0165	0,0029	1,0263	0,0020	1,0196	0,0063
	100	1,0121	0,0014	1,0292	0,0009	1,0085	0,0035
	Avg	1,0126	0,0026	1,0216	0,0020	1,0174	0,0074

Tabelle 6.9: SCMO - Exp. 2: DUB^k -Maß

Variante	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEQ^σ	AEX^μ	AEQ^σ
mit	20	0,1100	0,2483	0,3133	0,3666	0,1556	0,3085
Neunit.	50	0,0656	0,1828	0,1000	0,3000	0,0867	0,2568
	100	0,0107	0,0298	0,1000	0,3000	0,0533	0,1600
	Avg	0,0621	0,1536	0,1711	0,3222	0,0985	0,2418
ohne	20	0,0472	0,1201	0,0067	0,0200	0,0250	0,0750
Neunit.	50	0,0344	0,1000	0,0000	0,0000	0,0133	0,0400
	100	0,0893	0,2526	0,0000	0,0000	0,0467	0,1400
	Avg	0,0570	0,1576	0,0022	0,0067	0,0283	0,0850

Tabelle 6.10: SCMO - Exp. 2: AEX^k -Maß

tendenziell bessere Ergebnisse. Bei der maximalen Terminüberschreitung dominiert sie die Variante mit Neuinitialisierung. Bei der Zykluszeit hat sie den besseren $DUB_v^{\mu k}$ -Wert.

AEX^k -Maß: Beim AEX^k -Maß zeigt sich, daß die Strategie der Neuinitialisierung bzgl. $AEX_v^{\mu k}$ bessere Ergebnisse hervorbringt. Lediglich bei Problemen mit 100 Aufträgen ist die Strategie ohne Neuinitialisierung für die Zielgröße der Zykluszeit besser, vgl. Tabelle 6.10.

Insgesamt erscheint die Strategie mit Neuinitialisierung der Strategie ohne Neuinitialisierung überlegen. Wie bereits bei der Diskussion unterschiedlicher Strategien zur Ausgestaltung von multikriteriellen Ansätzen angesprochen, kann ein Grund für dieses Ergebnis darin liegen, daß die Ameisen zu Beginn der Suche in einer Zielregion zu lange von den alten Pheromoninformationen der Vorgängerregion beeinflusst werden. Es dauert eine gewisse Zeit, bis diese Informationen vergessen sind. Diese Vermutung

wird unterstützt von Ergebnissen, die von Goss u. a. (1989) im Rahmen ihres Doppelbrückenexperiments erzielt wurden.¹¹¹ Dort wurde den Ameisen zunächst nur der längere Weg angeboten. Erst nach 30 Minuten, in denen sich auf dem längeren Weg eine Pheromonspur ausbilden konnte, wird auch der kürzere Weg den Ameisen zugänglich gemacht. Es wurde beobachtet, daß die Ameisen weiterhin überwiegend den längeren Weg nutzen und nicht auf den kürzeren Weg wechseln.¹¹² Als Begründung für dieses Verhalten wird angeführt, daß die Ameisen die Information über den längeren Weg vergessen müßten, was aber durch eine zu langsame Verdunstung der Pheromone verhindert wird.¹¹³

Auf ACO-Algorithmen übertragen bedeutet dieses Ergebnis, daß am Anfang der Suche in einer Region eine höhere Verdunstungsrate zu verwenden ist oder aber die Pheromonmatrix gleich neu zu initialisieren ist. Letztere Strategie hat den Vorteil, daß weniger Iterationen in einem Suchbereich benötigt werden, da zu Beginn der Suche grundsätzlich keine zusätzlichen Iterationen zum Vergessen alter Information benötigt werden. Auch wird das Risiko ausgeschlossen, daß die Suche aufgrund der alten Pheromoninformationen in den falschen Bereich des Lösungsraums geleitet wird, weil die alte Information nicht vergessen wird.

¹¹¹Vgl. Goss u. a. (1989, S.580f.).

¹¹²Vgl. Goss u. a. (1989, S.580f.).

¹¹³Vgl. Dorigo u. Stützle (2004, S.5) sowie Goss u. a. (1989, S.580).

6.3.2.5 Experiment 3: Variation von B

In diesem Experiment soll der Einfluß auf die Lösungsqualität und Rechenzeit einer Variation des Parameters B und damit einer unterschiedlich feinen Einteilung des Zielraums in Suchregionen untersucht werden. Unter Beibehaltung aller anderen Einstellungen wird der Parameter B im Bereich von 4 bis 10 in Einerschritten variiert.¹¹⁴

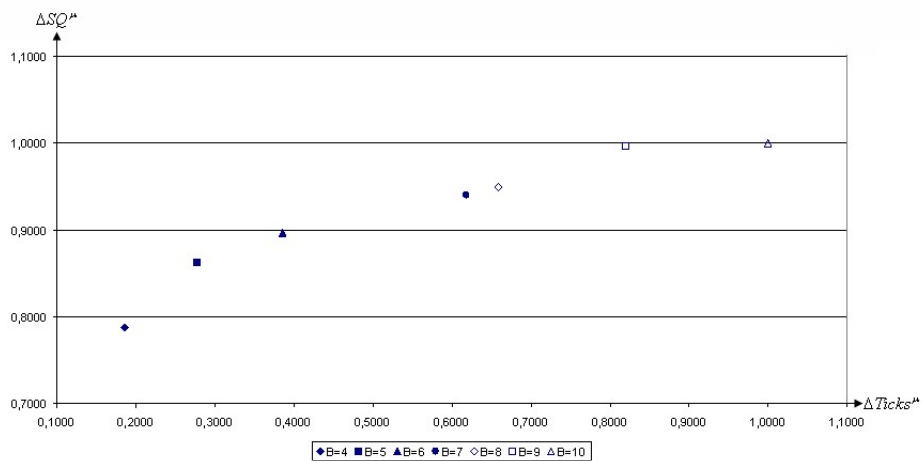
In den Tabellen 6.11 bis 6.13 wird die Relation der Werte der Maße zur Beurteilung der Lösungsqualität für die $v = 1(1)7$ Konfigurationen bezogen auf die Werte der Konfiguration mit $B=10$ wiedergegeben. Diese wurde als Basis gewählt, da sie in der Regel die beste Lösungsgüte aller Konfigurationen liefert, aber auch die größte Rechenzeit aufweist. Die Berechnung ist in Gleichung 6.24 beispielhaft für ΔSQ_v^μ dargestellt, die anderen Werte berechnen sich analog:

$$\Delta SQ_v^\mu = \frac{SQ_v^\mu}{SQ_{B=10}^\mu}, v = 1(1)7 \quad (6.24)$$

SQ-Maß: Es ist zu erkennen, daß sich die Werte von SQ_v^μ mit steigendem B verbessern. In Tabelle 6.11 ist weiterhin der Anteil der bei unterschiedlich B insgesamt benötigten Rechenzeit an der bei $B=10$ benötigten Rechenzeit angegeben. So ist zu erkennen, daß die Konfiguration $B=4$ im Durchschnitt 78,75 % der Lösungsqualität der Konfiguration $B=10$ erreicht und dabei nur 18,55 % der Rechenzeit benötigt. Die Reduktion von B bewirkt bei großen Problemen ein stärkeres Absinken der Lösungsqualität als bei kleinen Problemen. So beträgt z.B. die Lösungsqualität für Probleme mit 20 Aufträgen bei $B=4$ noch 86,81% der Lösungsqualität bei $B=10$. Für Probleme mit 100 Aufträgen beträgt die Lösungsqualität nur noch 70,88%. Die durchschnittlichen Werte von ΔSQ_v^μ und $\Delta Tickets_v^\mu$ einer Variante v über alle Probleminstanzen sind für alle $v = 1(1)V$ betrachteten Varianten in Abbildung 6.6 grafisch dargestellt.

¹¹⁴Dieses Experiment wurde auf der Rechnerkonfiguration 1 durchgeführt.

B	N	ΔSQ^μ	ΔSQ^σ	$\Delta Ticks^\mu$	$\Delta Ticks^\sigma$
4	20	0,8681	1,4683	0,1887	0,6281
	50	0,7720	1,1151	0,1867	0,1823
	100	0,7088	0,8760	0,1852	0,3272
	Avg	0,7875	1,0983	0,1855	0,2973
5	20	0,9291	1,2537	0,2802	0,7986
	50	0,8495	1,1532	0,2784	0,3134
	100	0,7964	0,8493	0,2768	0,4979
	Avg	0,8621	1,0497	0,2771	0,4585
6	20	0,9479	1,2772	0,3888	1,4761
	50	0,8899	1,1258	0,3855	0,3347
	100	0,8431	0,9259	0,3851	0,5763
	Avg	0,8965	1,0778	0,3852	0,5331
7	20	0,9676	1,1400	0,6240	2,4925
	50	0,9368	1,0480	0,6189	0,6538
	100	0,9109	0,9685	0,6178	1,0005
	Avg	0,9400	1,0364	0,6181	0,9420
8	20	0,9731	1,1949	0,6625	2,3067
	50	0,9467	1,0697	0,6591	0,7171
	100	0,9248	0,9238	0,6585	0,9617
	Avg	0,9495	1,0381	0,6587	0,9254
9	20	0,9963	1,0243	0,8212	1,5161
	50	0,9993	1,0227	0,8198	0,5995
	100	0,9964	1,0095	0,8201	0,9608
	Avg	0,9973	1,0176	0,8201	0,8830
10	20	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000

Tabelle 6.11: SCMO - Exp. 3: ΔSQ_v und $\Delta Ticks_v$ Abbildung 6.6: SCMO - Exp. 3: Trade Off: ΔSQ_v^μ - $\Delta Ticks_v^\mu$

B	N	C_{max}		F_{mid}		T_{max}	
		ΔDUB^μ	ΔDUB^σ	ΔDUB^μ	ΔDUB^σ	ΔDUB^μ	ΔDUB^σ
4	20	1,0041	1,1651	1,0028	1,6420	1,0107	1,6732
	50	1,0046	1,0513	1,0052	1,1518	1,0122	1,1283
	100	1,0022	0,8578	1,0080	0,9739	1,0088	0,9080
	Avg	1,0037	1,0642	1,0053	1,1943	1,0106	1,3023
5	20	1,0029	1,0801	1,0023	1,6163	1,0071	1,4027
	50	1,0037	1,0827	1,0042	1,0765	1,0077	1,1300
	100	1,0019	0,8322	1,0069	0,9377	1,0060	1,0514
	Avg	1,0028	1,0353	1,0045	1,1457	1,0070	1,2241
6	20	1,0022	1,1011	1,0014	1,3700	1,0041	1,3489
	50	1,0025	0,9464	1,0026	1,0909	1,0062	1,1468
	100	1,0013	0,9609	1,0043	0,9767	1,0040	1,1467
	Avg	1,0020	1,0152	1,0028	1,1100	1,0048	1,2294
7	20	1,0012	1,0534	1,0010	1,3529	1,0027	1,1608
	50	1,0018	0,9404	1,0017	1,2724	1,0037	1,0462
	100	1,0008	1,0334	1,0030	0,9607	1,0028	1,1502
	Avg	1,0013	1,0059	1,0019	1,1684	1,0031	1,1160
8	20	1,0008	1,0197	1,0005	1,2374	1,0023	1,2415
	50	1,0012	0,9085	1,0008	1,1234	1,0019	1,0470
	100	1,0005	0,9874	1,0017	1,0080	1,0017	1,0973
	Avg	1,0009	0,9706	1,0010	1,1042	1,0020	1,1376
9	20	1,0002	1,0179	1,0004	1,0634	1,0002	1,0011
	50	1,0003	1,0059	1,0003	1,1000	1,0007	0,9618
	100	1,0001	1,0422	1,0012	0,9297	1,0002	1,1502
	Avg	1,0002	1,0177	1,0006	1,0247	1,0004	1,0195
10	20	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Tabelle 6.12: SCMO - Exp. 3: ΔDUB_v^k

Zwischen B=4 und B=7 ist zu erkennen, daß der Nutzen einer weiteren Einheit Rechenzeit abnimmt. Zwischen B=7 und B=8 ist dieser abnehmende Grenznutzen nicht festzustellen. Insbesondere zwischen B=9 und B=10 ist der Unterschied in der Lösungsgüte sehr gering, wohingegen die Rechenzeit erheblich ansteigt.

DUB^k -Maß: Auch $DUB_v^{\mu k}$ verbessert sich bei allen drei Zielen bei steigendem B. Die Unterschiede in der Lösungsgüte für die verschiedenen Ausprägungen von B sind jedoch nicht so groß wie bei SQ_v^μ . Wie Tabelle 6.12 zeigt, ist die maximale Terminüberschreitung im Durchschnitt am sensibelsten gegenüber Variationen von B, wohingegen die Zykluszeit im Durchschnitt

B	N	C_{max}		F_{mid}		T_{max}	
		ΔAEX^μ	ΔAEX^σ	ΔAEX^μ	ΔAEX^σ	ΔAEX^μ	ΔAEX^σ
4	20	0,1790	0,2320	0,3260	0,4585	0,2218	0,3481
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Avg	0,1121	0,1206	0,2599	0,2524	0,1479	0,1724
5	20	0,2430	0,3230	0,3923	0,6698	0,4013	0,5646
	50	0,0057	0,0060	0,2000	0,2000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Avg	0,1536	0,1699	0,3216	0,3882	0,2677	0,2796
6	20	0,3069	0,3242	0,4862	0,8429	0,5715	0,7560
	50	0,0000	0,0000	0,6000	0,6000	0,0000	0,0000
	100	0,0671	0,0481	0,0000	0,0000	0,0833	0,0833
	Avg	0,2002	0,1757	0,4141	0,5226	0,3940	0,3939
7	20	0,4433	0,4796	0,6519	0,8295	0,7174	0,9007
	50	0,1385	0,1458	0,8000	0,8000	0,2857	0,2857
	100	0,5246	0,4943	0,0556	0,0556	0,1667	0,1667
	Avg	0,3754	0,3720	0,5639	0,5543	0,5554	0,5626
8	20	0,6496	0,7092	0,7624	0,8924	0,8416	1,0076
	50	0,2990	0,3148	0,8000	0,8000	0,4286	0,4286
	100	0,8253	0,8870	0,2222	0,2222	0,4167	0,4167
	Avg	0,5814	0,6063	0,6784	0,6476	0,7022	0,7126
9	20	0,7945	0,8378	0,8674	1,0793	0,9402	0,9493
	50	0,4664	0,4507	2,6000	2,6000	0,4286	0,4286
	100	1,9165	2,0486	0,3889	0,3889	0,8333	0,8333
	Avg	0,8449	0,8934	0,8678	0,9850	0,8320	0,7807
10	20	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Tabelle 6.13: SCMO - Exp. 3: ΔAEX_v^k

die kleinste prozentuale Veränderung aufweist.¹¹⁵

AEX^k -Maß: Schließlich zeigt auch $AEX_v^{\mu k}$ in Tabelle 6.13 eine deutliche Tendenz zur Verbesserung bei steigendem B. Jedoch gibt es hier auch einige Ausnahmen. Z.B. verschlechtert sich $AEX_v^{\mu C_{max}}$ für N=50 und B=6 im Vergleich zu B=5. Das Gleiche gilt für Probleme mit 100 Aufträgen beim Vergleich von B=9 und B=10. Schließlich ist auch $AEX_v^{\mu F_{mid}}$ für N=50 und B=10 schlechter als bei B=9.

Insgesamt kann mit einer feineren Einteilung des Zielraums in Suchregionen die Lösungsqualität gesteigert werden, wobei jedoch die Rechenzeit überproportional ansteigt.

¹¹⁵Ob diese Beobachtung mit der in Experiment 1 ausgewählten Suchvariante zusammenhängt, wird hier nicht weiter untersucht.

6.3.2.6 Experiment 4: Variation der Parameter *Iter* und *Ants*

Wie das Erhöhen von B , so hat auch die Variation der Parameter *Iter* und *Ants* einen Einfluß auf die Intensität der Suche. Im folgenden Experiment soll untersucht werden, welchen Einfluß deren Variation auf die Lösungsqualität und die benötigte Rechenzeit hat.

Die untersuchten Einstellungen sind derart ausgelegt, daß die Gesamtzahl der von Ameisen konstruierten und verbesserten Auftragsfolgen in allen getesteten Konfigurationen, mit Ausnahme der Variante 1, identisch ist. Tabelle 6.14 gibt einen Überblick über die getesteten Varianten.

Variante	B	Iter	Ants	\hat{B}	$\hat{B} \cdot \text{Iter} \cdot \text{Ants}$
1	5	10	1	15	150
2	9	10	1	45	450
3	5	6	5	15	450
4	5	10	3	15	450
5	5	30	1	15	450

Tabelle 6.14: SCMO - Konfigurationen Experiment 4

Die Ergebnisse der Varianten 1 und 2 werden dabei aus Experiment 3 übernommen. Mit Hilfe von Variante 2 soll überprüft werden, ob eine Variation von B alleine besser ist als eine Variation der Parameter *Iter* und *Ants*. Mit Hilfe von Variante 1 soll überprüft werden, ob überhaupt eine Verbesserung der Lösungsqualität im Vergleich zur Einstellung mit $B=5$ aus Experiment 3 auftritt und wie stark diese ist. Die Ergebnisse sind in den Tabellen 6.15 bis 6.17 angegeben.¹¹⁶

¹¹⁶Für dieses Experiment wurde Rechnerkonfiguration 1 verwendet.

SQ-Maß: Tabelle 6.15 zeigt zunächst das wenig überraschende Ergebnis, daß die Varianten 2, 3, 4 und 5 für alle Problemgrößen bzgl. SQ_v^μ besser sind als Variante 1. Mit Ausnahme der Probleme mit 100 Aufträgen dominieren diese Varianten die Variante 1.

Variante	N	SQ^μ	SQ^σ
1	20	0,8047	0,0459
	50	0,6541	0,0626
	100	0,5452	0,0516
	Avg	0,6680	0,0534
2	20	0,8649	0,0374
	50	0,7751	0,0558
	100	0,6870	0,0610
	Avg	0,7757	0,0515
3	20	0,8672	0,0371
	50	0,7166	0,0518
	100	0,5861	0,0568
	Avg	0,7233	0,0486
4	20	0,8522	0,0399
	50	0,7452	0,0580
	100	0,6349	0,0576
	Avg	0,7441	0,0518
5	20	0,8128	0,0447
	50	0,7388	0,0591
	100	0,7606	0,0659
	Avg	0,7707	0,0566

Tabelle 6.15: SCMO - Exp. 4: SQ -Maß

Ein Vergleich der Varianten 3, 4 und 5 alleine zeigt, daß sie jeweils für unterschiedliche Problemgrößen vorteilhaft sind. Für Probleme mit 20 Aufträgen dominiert Variante 3. Für Probleme mit 50 Aufträgen dominiert Variante 4 die Variante 5 und hat im Vergleich zu Variante 3 den besseren Durchschnittswert. Für Probleme mit 100 Aufträgen hat Variante 5 den besten SQ_v^μ -Wert, dominiert aber aufgrund der größten Standardabweichung nicht die anderen beiden Varianten.

Wird Variante 2 hinzugezogen, so ist diese die vorzuziehende Alternative für Probleme mit 50 Aufträgen.

DUB^k -Maß: Beim DUB^k -Maß (vgl. Tabelle 6.16) sind die Varianten 3, 4 und 5 für die einzelnen Problemgrößen, bezogen auf $DUB_v^{\mu k}$, besser als Variante 1. Dabei sind unterschiedliche Varianten für unterschiedliche Ziele und Problemgrößen vorteilhaft.

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0083	0,0030	1,0046	0,0027	1,0160	0,0091
	50	1,0171	0,0027	1,0107	0,0029	1,0178	0,0066
	100	1,0149	0,0010	1,0161	0,0027	1,0151	0,0037
	Avg	1,0135	0,0022	1,0105	0,0028	1,0163	0,0065
2	20	1,0056	0,0028	1,0026	0,0018	1,0091	0,0065
	50	1,0137	0,0025	1,0068	0,0030	1,0107	0,0056
	100	1,0132	0,0012	1,0104	0,0026	1,0092	0,0040
	Avg	1,0108	0,0022	1,0066	0,0025	1,0097	0,0054
3	20	1,0060	0,0024	1,0024	0,0019	1,0103	0,0065
	50	1,0156	0,0025	1,0082	0,0030	1,0142	0,0061
	100	1,0145	0,0012	1,0139	0,0026	1,0143	0,0040
	Avg	1,0120	0,0020	1,0081	0,0025	1,0129	0,0055
4	20	1,0064	0,0028	1,0031	0,0021	1,0105	0,0065
	50	1,0147	0,0026	1,0078	0,0034	1,0125	0,0063
	100	1,0138	0,0012	1,0121	0,0030	1,0119	0,0040
	Avg	1,0116	0,0022	1,0077	0,0028	1,0116	0,0056
5	20	1,0081	0,0032	1,0039	0,0025	1,0140	0,0071
	50	1,0147	0,0022	1,0092	0,0035	1,0137	0,0057
	100	1,0116	0,0014	1,0129	0,0030	1,0076	0,0038
	Avg	1,0115	0,0023	1,0087	0,0030	1,0118	0,0056

Tabelle 6.16: SCMO - Exp. 4: DUB^k -Maß

Wird Variante 2 hinzugezogen, so ist diese eindeutig die beste Variante. Mit Ausnahme von $DUB_v^{\mu F_{mid}}$ bei N=20 und $DUB_v^{\mu T_{max}}$ sowie $DUB_v^{\mu C_{max}}$ bei N=100 hat sie immer die besten $DUB_v^{\mu k}$ -Werte aller vier Alternativen.

AEX^k -Maß: Beim AEX^k -Maß (vgl. Tabelle 6.17) gilt, daß die Varianten 3 und 4 für die einzelnen Problemgrößen bessere $AEX_v^{\mu k}$ Werte haben als Variante 1. Variante 5 ist bei Problemen mit 20 Aufträgen bzgl. $AEX_v^{\mu T_{max}}$ schlechter. Beim Vergleich der Varianten 3, 4 und 5 zeigen sich erneut unterschiedliche Varianten in Abhängigkeit von Problemgröße und Ziel als vorteilhaft, wenn $AEX_v^{\mu k}$ betrachtet wird.

B	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0117	0,0350	0,1150	0,1786	0,0774	0,1488
	50	0,0004	0,0011	0,0033	0,0100	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Avg	0,0040	0,0120	0,0394	0,0629	0,0258	0,0496
2	20	0,0750	0,1549	0,2550	0,2905	0,1732	0,2439
	50	0,0564	0,1552	0,0400	0,1200	0,0333	0,1000
	100	0,0153	0,0425	0,0600	0,1800	0,0200	0,0600
	Avg	0,0489	0,1175	0,1183	0,1968	0,0755	0,1346
3	20	0,0583	0,1270	0,2517	0,3464	0,1322	0,2206
	50	0,0073	0,0220	0,0200	0,0600	0,0167	0,0500
	100	0,0078	0,0233	0,0033	0,0100	0,0033	0,0100
	Avg	0,0245	0,0575	0,0917	0,1388	0,0507	0,0935
4	20	0,0761	0,1665	0,1783	0,2526	0,1261	0,2069
	50	0,0159	0,0456	0,0267	0,0800	0,0333	0,1000
	100	0,0008	0,0025	0,0200	0,0600	0,0183	0,0550
	Avg	0,0310	0,0715	0,0750	0,1309	0,0593	0,1206
5	20	0,0322	0,0583	0,1300	0,2017	0,0614	0,1205
	50	0,0200	0,0578	0,0100	0,0300	0,0167	0,0500
	100	0,0761	0,2175	0,0167	0,0500	0,0583	0,1750
	Avg	0,0428	0,1112	0,0522	0,0939	0,0455	0,1152

Tabelle 6.17: SCMO - Exp. 4: AEX^k -Maß

Wird Variante 2 hinzugezogen, so ist diese überwiegend die beste Variante. Bei Problemen mit 20 Aufträgen ist bei $AEX_v^{\mu C_{max}}$ Variante 4 besser. Bei Problemen mit 100 Aufträgen ist für $AEX_v^{\mu C_{max}}$ und $AEX_v^{\mu T_{max}}$ Variante 5 besser.

Zusammenfassend zeigt sich, daß eine Erhöhung der insgesamt von Ameisen generierten und verbesserten Auftragsfolgen zu einer Verbesserung der Lösungsqualität führt. Bei Variation der Anzahl der Ameisen und des Parameters $Iter$ sowie konstantem Parameter B zeichnet sich aber keine der drei untersuchten Varianten (3, 4, 5) als eindeutig beste Möglichkeit ab. Wird die Variation von B zusätzlich berücksichtigt, so zeigt sich, daß es tendenziell vorteilhaft ist, nur Parameter B zu variieren. Nur für große Probleme kann mitunter auch eine Variation des Parameters $Iter$ vorteilhaft sein. Die Rechenzeiten für die in diesem Experiment überprüften Varianten unterliegen nur kleinen Schwankungen und sind stochastischer Natur, da die Gesamtzahl der von Ameisen generierten Auftragsfolgen gleich ist. Auf eine Darstellung der Rechenzeiten wird daher verzichtet.

6.3.2.7 Experiment 5: Variation des Parameters ρ

In diesem Experiment soll die Veränderung der Lösungsgüte aufgrund einer Veränderung der Trail-Persistence-Rate analysiert werden. Als mögliche Werte für ρ werden dabei untersucht: 0,55, 0,65, 0,75, 0,85 sowie 0,95. Die Ergebnisse sind in den Tabellen 6.18 bis 6.20 angegeben.

ρ	N	SQ^μ	SQ^σ
0,55	20	0,8046	0,0484
	50	0,7435	0,0569
	100	0,7639	0,0556
	Avg	0,7706	0,0536
0,65	20	0,8309	0,0443
	50	0,7639	0,0585
	100	0,7161	0,0573
	Avg	0,7703	0,0534
0,75	20	0,8432	0,0408
	50	0,7486	0,0578
	100	0,6364	0,0586
	Avg	0,7427	0,0524
0,85	20	0,8520	0,0421
	50	0,7192	0,0561
	100	0,5914	0,0512
	Avg	0,7209	0,0498
0,95	20	0,8511	0,0362
	50	0,7107	0,0559
	100	0,5768	0,0512
	Avg	0,7129	0,0478

Tabelle 6.18: SCMO - Exp. 5: SQ -Maß

SQ -Maß: Tabelle 6.18 zeigt für das SQ -Maß, daß es mit steigender Problemgröße vorteilhaft ist, einen kleineren Wert für den Parameter ρ und damit eine größere Verdunstungsrate zu wählen. Für $N=20$ ist ein Wert von 0,85, für $N=50$ ein Wert von 0,65 und für Probleme mit 100 Aufträgen ein Wert von 0,55 vorteilhaft.

ρ	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
0,55	20	1,0086	0,0034	1,0043	0,0028	1,0162	0,0092
	50	1,0154	0,0024	1,0095	0,0035	1,0144	0,0061
	100	1,0116	0,0012	1,0130	0,0028	1,0074	0,0038
	Avg	1,0119	0,0023	1,0089	0,0031	1,0126	0,0064
0,65	20	1,0071	0,0032	1,0034	0,0023	1,0137	0,0084
	50	1,0147	0,0023	1,0083	0,0033	1,0121	0,0063
	100	1,0125	0,0013	1,0120	0,0028	1,0084	0,0042
	Avg	1,0114	0,0022	1,0079	0,0028	1,0114	0,0063
0,75	20	1,0067	0,0029	1,0032	0,0023	1,0118	0,0075
	50	1,0152	0,0024	1,0079	0,0035	1,0129	0,0061
	100	1,0139	0,0012	1,0120	0,0027	1,0118	0,0040
	Avg	1,0119	0,0022	1,0077	0,0028	1,0122	0,0059
0,85	20	1,0064	0,0027	1,0029	0,0022	1,0111	0,0072
	50	1,0161	0,0023	1,0080	0,0031	1,0140	0,0056
	100	1,0147	0,0011	1,0136	0,0026	1,0144	0,0037
	Avg	1,0124	0,0020	1,0082	0,0026	1,0131	0,0055
0,95	20	1,0063	0,0028	1,0026	0,0019	1,0110	0,0064
	50	1,0166	0,0027	1,0081	0,0031	1,0152	0,0060
	100	1,0150	0,0011	1,0139	0,0026	1,0149	0,0039
	Avg	1,0126	0,0022	1,0082	0,0025	1,0137	0,0054

Tabelle 6.19: SCMO - Exp. 5: DUB^k -Maß

DUB^k -Maß: Für das DUB^k -Maß (vgl. Tabelle 6.19) läßt sich ebenfalls die Tendenz beobachten, daß es vorteilhaft ist, mit steigender Problemgröße kleinere Werte für ρ zu wählen. Für alle drei Zielgrößen ist bei 20 Aufträgen ein Wert von 0,95 gut. Für die mittlere Durchlaufzeit ist für die verbleibenden untersuchten Problemgrößen ein kleinerer Wert von 0,75 vorteilhaft. Für die Zykluszeit und die maximale Terminüberschreitung sollte ρ bei N=50 auf 0,65 und bei N=100 auf 0,55 gesetzt werden.

ρ	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
0,55	20	0,0328	0,0833	0,1500	0,1879	0,0522	0,1367
	50	0,0235	0,0694	0,0100	0,0300	0,0167	0,0500
	100	0,0828	0,2278	0,0233	0,0700	0,0467	0,1400
	Avg	0,0463	0,1269	0,0611	0,0960	0,0385	0,1089
0,65	20	0,0797	0,1637	0,1900	0,2643	0,0894	0,1687
	50	0,0379	0,1107	0,0267	0,0800	0,0333	0,1000
	100	0,0128	0,0342	0,0400	0,1200	0,0467	0,1400
	Avg	0,0435	0,1029	0,0856	0,1548	0,0565	0,1362
0,75	20	0,0511	0,1056	0,2067	0,2517	0,0912	0,1993
	50	0,0228	0,0683	0,0233	0,0700	0,0300	0,0900
	100	0,0000	0,0000	0,0267	0,0800	0,0033	0,0100
	Avg	0,0246	0,0580	0,0856	0,1339	0,0415	0,0998
0,85	20	0,0803	0,1631	0,2267	0,3115	0,0839	0,1655
	50	0,0092	0,0277	0,0233	0,0700	0,0100	0,0300
	100	0,0044	0,0133	0,0033	0,0100	0,0033	0,0100
	Avg	0,0313	0,0680	0,0844	0,1305	0,0324	0,0685
0,95	20	0,0769	0,1831	0,2200	0,2271	0,0902	0,1953
	50	0,0067	0,0200	0,0167	0,0500	0,0100	0,0300
	100	0,0000	0,0000	0,0067	0,0200	0,0000	0,0000
	Avg	0,0279	0,0677	0,0811	0,0990	0,0334	0,0751

Tabelle 6.20: SCMO - Exp. 5: AEX^k -Maß

AEX^k -Maß: Die Auswertung der Ergebnisse des AEX^k -Maßes (vgl. Tabelle 6.20) zeigt, daß für größere Probleme kleinere Werte für ρ gewählt werden sollten. Wie schon beim SQ_v -Maß ist im Fall der Zykluszeit für $N=20$ ein Wert von 0,85, für $N=50$ ein Wert von 0,65 und für $N=100$ ein Wert von 0,55 vorteilhaft. Bei der mittleren Durchlaufzeit sollte bei Problemen mit 20 Aufträgen ebenfalls ρ auf den Wert 0,85 gesetzt werden. Für die restlichen untersuchten Problemgrößen ist ein Wert von 0,65 eine gute Wahl. Auch bei der maximalen Terminüberschreitung ist ein Wert von 0,65 bei $N=50$ und $N=100$ vorteilhaft. Bei Problemen mit 20 Aufträgen sollte ρ auf 0,75 gesetzt werden.

Insgesamt läßt sich bei allen drei betrachteten Qualitätsmaßen die Tendenz beobachten, daß für steigende Problemgrößen eine höhere Verdunstungsrate verwendet werden sollte. Soll ρ unabhängig von der Problemgröße eingestellt werden, so erscheint unter Berücksichtigung aller drei Qualitätsmaße ein Wert von 0,65 als eine gute Einstellung.

6.4 Der MCMO-Ansatz

6.4.1 Beschreibung des Algorithmus - Variante 1 (MCMO1)

6.4.1.1 Allgemeine Eigenschaften

Der Multi-Colony-Multi-Objective-Ansatz (MCMO-Ansatz) verwendet drei homogene Kolonien, da das Zielsystem aus drei Zielen besteht. In vielen Punkten bestehen Gemeinsamkeiten mit dem SCMO-Ansatz. Die Kolonien wechseln mit fortschreitender Laufzeit ihre Zielregion im Zielraum, wobei stets eine Neuinitialisierung der Pheromonmatrix analog zum SCMO-Ansatz erfolgt (verbundene Multi Start Strategie). Als Kompromißmodell wird die gewichtete Tchebycheff-Distanz verwendet. Der Kompromißzielfunktionswert ermittelt sich analog zu Gleichung 6.10 auf Seite 175. Die Ansätze haben auch die Verwendung von lokalen heuristisch effizienten Mengen und die Initialisierungsphase gemeinsam. Auch beim MCMO-Ansatz werden grundsätzlich alle generierten Auftragsfolgen auf heuristische Effizienz hin überprüft.

Neben der Anzahl der Kolonien besteht der wesentliche Unterschied in der Strategie, mit der der Zielraum abgesucht wird und bei Variante 1 zusätzlich in der Konstruktion der Auftragsfolgen durch die Ameisen. Während der Laufzeit suchen zwei oder drei Kolonien gleichzeitig aktiv in unterschiedlichen Bereichen des Zielraums. Treffen sich zwei oder drei Kolonien in einem einzelnen Bereich, so wird die Suche nur von einer Kolonie durchgeführt. Während der Suche kommunizieren die aktiv suchenden Kolonien miteinander, sowohl während als auch nach der Lösungskonstruktion. Dies macht im Vergleich zum SCMO-Ansatz eine abgewandelte Konstruktionsprozedur für Auftragsfolgen notwendig. Die später vorgestellte Variante 2 des MCMO-Ansatzes verzichtet auf die Kommunikation der Kolonien während der Lösungskonstruktion. Abbildung 6.7 zeigt die in Variante 1 des MCMO-Ansatzes verwendeten Prozeduren und ihre Beziehungen miteinander. Die einzelnen Prozeduren werden in den folgenden Abschnitten genauer beschrieben.

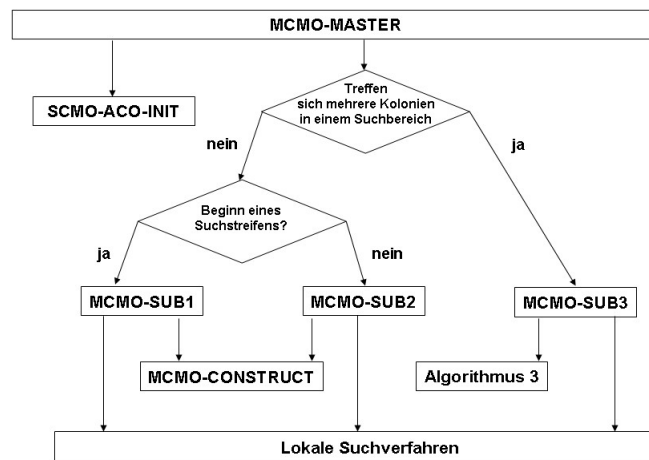


Abbildung 6.7: MCMO1 - Übersicht der Prozeduren

6.4.1.2 MCMO-MASTER - Suche im Zielraum

MCMO-MASTER ist die zentrale Prozedur der Heuristik. Nachdem sie die Initialisierungsprozedur SCMO-INIT aufgerufen hat, ist es ihre Aufgabe, die systematische Suche im Zielraum zu organisieren. Der Zielraum wird in von außen nach innen geschachtelte Suchstreifen eingeteilt (vgl. Abbildung 6.8). Über die Laufzeit des Algorithmus werden die Suchstreifen von außen nach innen abgesucht. Die rautierten Felder kennzeichnen jeweils die Suchregion, in denen die Suche auf einem Suchstreifen beginnt (Startregion).

Jede der drei Kolonien sucht dabei nur in einer bestimmten Region des Zielraums. Die drei Kolonien können daher als Spezialisten für jeweils ein Ziel betrachtet werden. Zu Beginn der Suche verfolgen die Kolonien alleine ihr spezialisiertes Ziel. Dies wird erreicht, indem Kolonie 1 mit dem Gewichtungsvektor $\lambda^1(1; 0; 0)$, Kolonie 2 mit $\lambda^2 = (0; 1; 0)$ und Kolonie 3 mit $\lambda^3 = (0; 0; 1)$ sucht. Anders als im SCMO-Ansatz werden hier somit alle Regionen des Zielraums, in dem die individuellen Optima der einzelnen Ziele liegen, unmittelbar nach der Initialisierungsphase abgesucht.

Nachdem die Suche in den Bereichen, in denen die globalen Optima liegen, abgeschlossen ist, nähern sich jeweils zwei Kolonien bzgl. ihrer Suchbereiche auf einem Streifen an, wobei nur die sich annähernden Kolonien aktiv suchen und die jeweils dritte Kolonie keine Suchaktivitäten ausführt. Dabei wird jeweils ein Annäherungsschritt auf jedem Suchstreifen durchgeführt. Treffen sich zwei oder drei Kolonien in einem Suchbereich (in Abbildung 6.8 die grauen Felder) so wird der Suchbereich nur von einer der

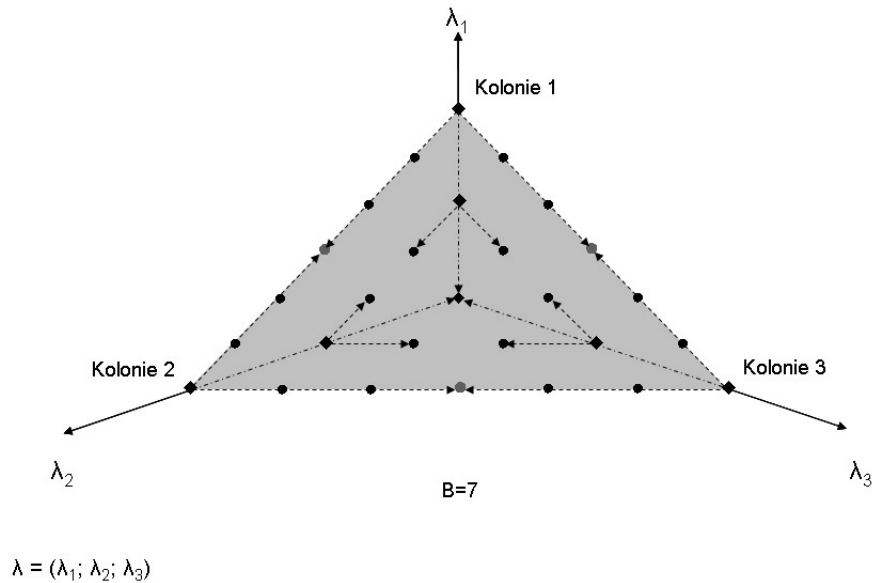


Abbildung 6.8: MCMO - Suchschema im Zielraum

Kolonien abgesucht. Sind alle Suchbereiche der drei äußeren Suchstreifen abgearbeitet worden, so wird der nächste Suchstreifen eröffnet.

In Abhängigkeit davon, wieviele Kolonien zu einem Zeitpunkt aktiv suchen, werden unterschiedliche Prozeduren aufgerufen. Dabei findet zwischen den aktiv suchenden Kolonien sowohl während der Konstruktion von Auftragsfolgen als auch bei der Aktualisierung der Pheromoninformationen ein Informationsaustausch statt. Suchen im ersten Suchbereich eines Suchstreifens alle drei Kolonien aktiv (Rautenfelder), so wird die Prozedur MCMO-SUB1 verwendet. Nähern sich zwei Kolonien auf einem Suchstreifen an, so suchen nur diese beiden Kolonien aktiv (weder Rautenfelder noch graue Felder). Es wird die Prozedur MCMO-SUB2 aufgerufen. Treffen sich zwei oder drei Kolonien zu einem Zeitpunkt in einem Suchbereich (graue Felder), so wird die Suche dort von nur einer Kolonie ausgeführt und daher MCMO-SUB3 aufgerufen. Die Suche wird nur von einer Kolonie ausgeführt, da ansonsten Bereiche des Zielraums, in denen sich mehrere Kolonien treffen, im Vergleich zu anderen Bereichen überproportional intensiv untersucht würden.

Im Algorithmus 9 ist die Variation der Gewichte und der Aufruf der jeweiligen Prozeduren dargestellt.

Zeile 1: Die Initialisierung entspricht dem Vorgehen im SCMO-Ansatz, weshalb die Prozedur SCMO-INIT verwendet wird.

Algorithmus 9 MCMO-MASTER

```

// Parameter:
// B:           Parameter zum Festlegen der Anzahl der
                Suchbereiche
//  $T^1, T^2, T^3$ : Pheromonmatrix der Kolonie 1, 2, 3
//  $\lambda^1, \lambda^2, \lambda^3$ : Gewichtungsvektor der Kolonie 1, 2, 3
// GNDS:       Global heuristisch effiziente Menge,
                wird in SCMO-INIT initialisiert

1: SCMO-INIT;

2: step =  $\frac{1}{B-1}$ ;
3: for( i=0; i  $\leq$   $\lfloor \frac{B}{3} \rfloor$ ; i=i+1 ) {
4:     j = (B-1) - (2*i);
5:     // Suche im ersten Bereich der Suchstreifen:
6:      $\lambda_1^1 = \lambda_2^2 = \lambda_3^3 = j * \text{step}$ ;
7:      $\lambda_3^1 = \lambda_2^2 = \lambda_1^3 = i * \text{step}$ ;
8:      $\lambda_2^1 = \lambda_1^2 = \lambda_3^3 = 1 - (j * \text{step}) - (i * \text{step})$ ;
9:     if(  $\lambda^1 == \lambda^2 == \lambda^3$  ) MCMO-SUB3( $\lambda^1$ , GNDS);
10:    else MCMO-SUB1( $\lambda^1, \lambda^2, \lambda^3$ , GNDS);

11:    // Kopiere die Pheromonmatrizen der Kolonien
12:     $T^{K1} = T^1$ ;  $T^{K2} = T^2$ ;  $T^{K3} = T^3$ ;

13:    // Suche in restlichen Bereichen des Suchstreifens
14:    for( s=j-1; s  $\geq$   $\lfloor \frac{B-i}{2} \rfloor$ ; s=s-1 ) {
15:        // Streifen der Kolonien 1 und 2
16:         $\lambda_1^1 = \lambda_2^2 = s * \text{step}$ ;
17:         $\lambda_3^1 = \lambda_2^3 = i * \text{step}$ ;
18:         $\lambda_2^1 = \lambda_1^2 = 1 - (i * \text{step}) - (s * \text{step})$ ;
19:        if(  $\lambda^1 == \lambda^2$  ) MCMO-SUB3( $\lambda^1$ , GNDS);
20:        else MCMO-SUB2( $T^1, T^2, T^{K3}, \lambda^1, \lambda^2$ , GNDS);

21:        // Streifen der Kolonien 1 und 3
22:         $\lambda_1^1 = \lambda_3^3 = s * \text{step}$ ;
23:         $\lambda_2^1 = \lambda_2^3 = i * \text{step}$ ;
24:         $\lambda_3^1 = \lambda_1^3 = 1 - (i * \text{step}) - (s * \text{step})$ ;
25:        if(  $\lambda^1 == \lambda^3$  ) MCMO-SUB3( $\lambda^3$ , GNDS);
26:        else MCMO-SUB2( $T^1, T^3, T^{K2}, \lambda^1, \lambda^3$ , GNDS);

27:        // Streifen der Kolonien 2 und 3
28:         $\lambda_2^2 = \lambda_3^3 = s * \text{step}$ ;
29:         $\lambda_1^2 = \lambda_1^3 = i * \text{step}$ ;
30:         $\lambda_3^2 = \lambda_2^3 = s - (i * \text{step}) - (s * \text{step})$ ;
31:        if(  $\lambda^2 == \lambda^3$  ) MCMO-SUB3( $\lambda^2$ , GNDS);
32:        else MCMO-SUB2( $T^2, T^3, T^{K1}, \lambda^2, \lambda^3$ , GNDS);
33:    } // end for
34: } // end for

```

Zeile 2: Die Schrittweite, mit der die Elemente der Gewichtvektoren variiert werden (*step*), wird berechnet.

Zeile 3+34: In der äußeren der beiden For-Schleifen werden die Suchstreifen von außen nach innen abgearbeitet.

Zeilen 4-8: Bei der Suche in den ersten Suchbereichen der Suchstreifen sind alle drei Kolonien aktiv. Für jede Kolonie wird ein Gewichtungsvektor für die Kompromißzielfunktion berechnet.

Zeilen 9-10: Die hier vorgenommene Fallunterscheidung ist notwendig, da überprüft werden muß, ob sich die Kolonien in einem Bereich des Zielraums treffen oder nicht. Sind die Gewichtungsvektoren gleich, so treffen sie sich in einem Suchbereich. Dieser Fall kann bei ungeraden Werten von B auftreten, dann treffen sich die drei Kolonien zum Abschluß der Suche in der "Mitte des Zielraums". Sind die Gewichtungsvektoren unterschiedlich (ELSE-Zweig), so wird die Prozedur MCMO-SUB1 aufgerufen, in der jede Kolonie in dem ihr zugeteilten Bereich des Zielraums sucht. Daß sich zu Beginn eines Suchstreifens nur zwei Kolonien in einem Bereich treffen, ist nicht möglich.

Zeilen 11-12: Nähern sich zwei Kolonien bzgl. ihrer Suchbereiche auf einem Suchstreifen an, so suchen nur diese beiden Kolonien aktiv. Dennoch berücksichtigen sie bei ihrer Suche auch Informationen, welche nach Abschluß von MCMO-SUB1 in der Pheromonmatrix der dritten Kolonie gespeichert sind. Dabei greifen sie auf die Information zu, wie sie nach Abschluß der Suche auf dem ersten Suchbereich eines Suchstreifens vorliegt. Die folgenden Suchaktivitäten der Kolonien auf den Suchstreifen würden diese Information zerstören. Um sie zu erhalten, werden die Pheromonmatrizen der Kolonien an dieser Stelle kopiert.

Zeilen 14+33: In der inneren For-Schleife werden die übrigen Suchbereiche eines Suchstreifens bearbeitet.

Zeilen 15-20: Zunächst wird auf dem Streifen zwischen den Suchbereichen gesucht, welchem die Kolonien 1 und 2 zugeordnet sind. Dazu werden zunächst die Zielgewichte bestimmt. Anschließend muß erneut überprüft werden, ob sich die zwei aktiven Kolonien in einem Suchbereich treffen. Dies ist dann der Fall, wenn die Gewichtungsvektoren gleich sind, was den Aufruf der Prozedur MCMO-SUB3 auslöst. Ansonsten wird MCMO-SUB2 aufgerufen, wo jede der zwei Kolonien in dem ihr zugeteilten Bereich des Zielraums sucht.

Zeilen 21-32: Hier wird analog zu den Zeilen 15-20 zunächst der Suchstreifen zwischen den Suchbereichen betrachtet, denen Kolonie 1 und 3 zugeordnet sind. Anschließend wird der dritte, noch fehlende Streifen betrachtet.

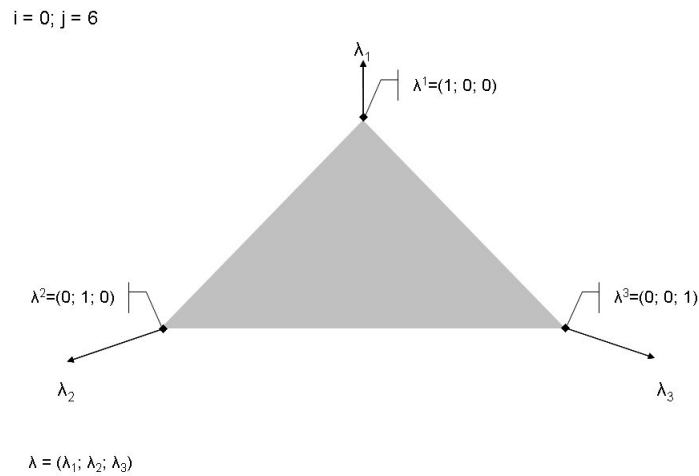
Das folgende ausführliche grafische Beispiel soll die systematische Suche im Zielraum nochmal verdeutlichen.

Ein grafisches Beispiel

Nach der Initialisierungsphase (Aufruf von SCMO-INIT in Zeile 1) beginnt das systematische Absuchen des Zielraums. Dazu wird zuerst für ein gegebenes B die Größe der Intervalle für die Variation der Gewichte berechnet. In diesem Beispiel sei $B = 7$. Es ergibt sich in Zeile 2 daher: $step = \frac{1}{7-1} = 0,166\bar{6}$. Insgesamt wird der Zielraum in $\hat{B} = (\frac{7}{2} + 0,5) \cdot 7 = 28$ Bereiche eingeteilt.

i=0; j=6

Es gilt $i = 0 \leq \lfloor \frac{7}{3} \rfloor = 2$. Die äußere For-Schleife wird das erste Mal durchlaufen. In Zeile 4 errechnet sich j daher zu: $j = (7 - 1) - (2 \cdot 0) = 6$. In den Zeilen 5-8 werden anschließend die Gewichte der Kompromißzielfunktion für die drei Kolonien berechnet. Da sich die Gewichtevektoren der drei Kolonien unterscheiden, wird in der If-Else-Verzweigung in den Zeilen 9 und 10 die Prozedur MCMO-SUB1 aufgerufen und damit die Suche in den ersten Suchbereichen des äußersten Suchstreifens im Zielraum gestartet. Alle drei Kolonien suchen aktiv in den ihnen zugeordneten Bereichen des Zielraums.

Abbildung 6.9: MCMO: $i=0; j=6$

Nach Abschluß der Suche werden Kopien der Pheromonmatrizen der drei Kolonien erstellt (Zeilen 11 u. 12).

$i=0; j=6; s=5$ Es gilt $s = 5 \geq \lfloor \frac{7-0}{2} \rfloor = 3$. Die innere For-Schleife wird durchlaufen. Damit beginnt die Suche in den restlichen Suchbereichen des äußeren Suchstreifens. Dabei nähern sich jeweils zwei Kolonien bzgl. ihrer Suchbereiche im Zielraum einander an. So nähern sich in den Zeilen 15 bis 20 zunächst die Kolonien 1 und 2 einander an, während Kolonie 3 steht. Es werden zunächst die Gewichtungsvektoren der Kolonie 1 (λ^1) und der Kolonie 2 (λ^2) berechnet. Da sie sich unterscheiden, suchen die Kolonien in unterschiedlichen Bereichen des Zielraums, und es wird MCMO-SUB2 aufgerufen. Bei der Konstruktion der Auftragsfolgen wird neben den Pheromonmatrizen der Kolonien 1 und 2 auch die Pheromoninformation der nicht aktiv suchenden Kolonie 3 benötigt. Es wird daher die Kopie der Pheromonmatrix von Kolonie 3 (T^{K3}) an MCMO-SUB2 übergeben.

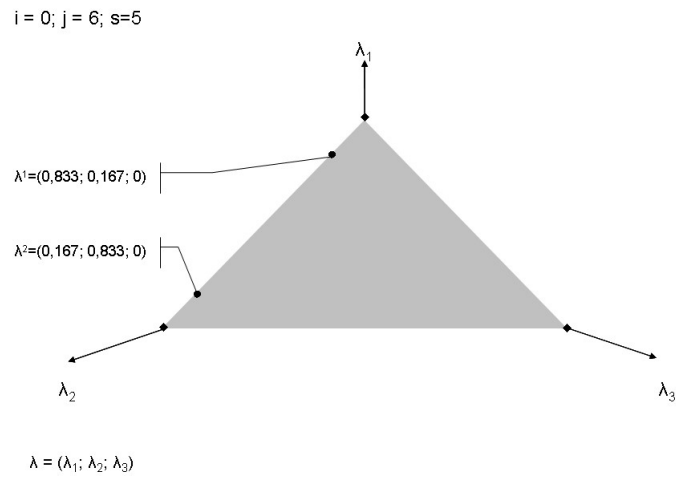


Abbildung 6.10: MCMO: $i=0; j=6; s=5$; Kolonie 1 & 2

Im Anschluß beginnt der erste Schritt der Annäherung der Kolonien 1 und 3, während Kolonie 2 steht (Zeilen 21-26).

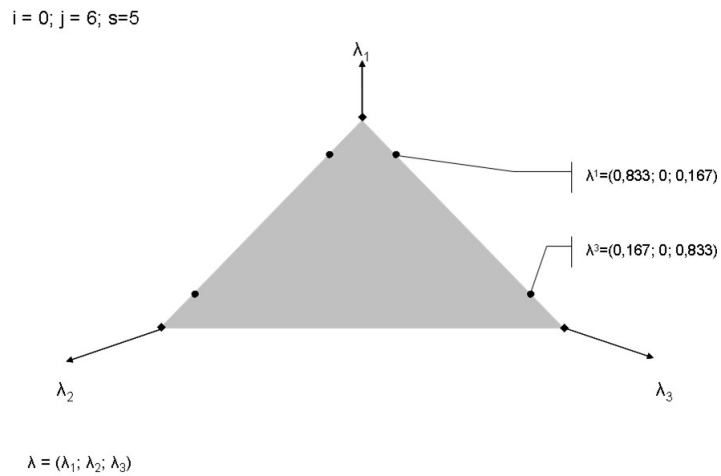


Abbildung 6.11: MCMO: $i=0; j=6; s=5$; Kolonie 1 & 3

Zum Abschluß suchen Kolonie 2 und 3 aktiv auf ihrem Suchstreifen (Zeilen 27-32).

$i = 0; j = 6; s = 5$

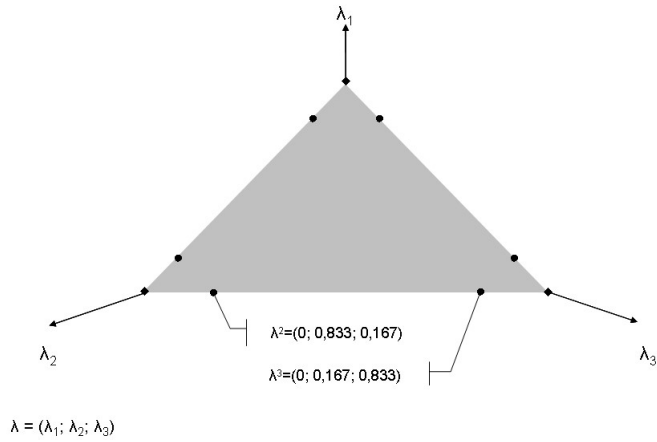


Abbildung 6.12: MCMO: $i=0; j=6; s=5$; Kolonie 2 & 3

$i=0; j=6; s=4$ Es gilt $s = 4 \geq \lfloor \frac{7-0}{2} \rfloor = 3$. Die innere For-Schleife wird ein weiteres Mal durchlaufen und die Annäherung auf den drei Suchstreifen damit weitergeführt.

$i = 0; j = 6; s = 4$

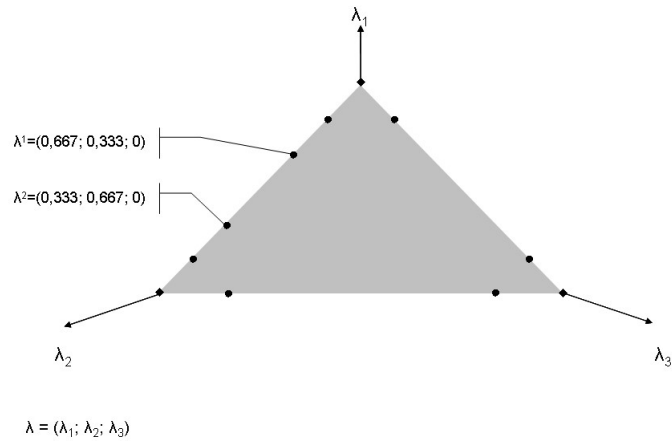
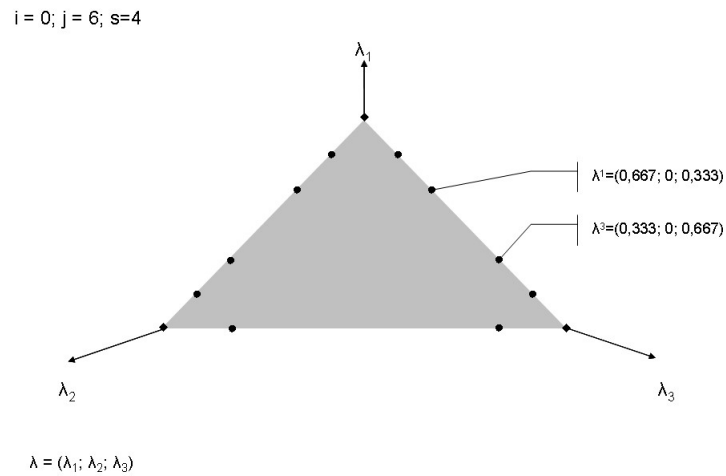
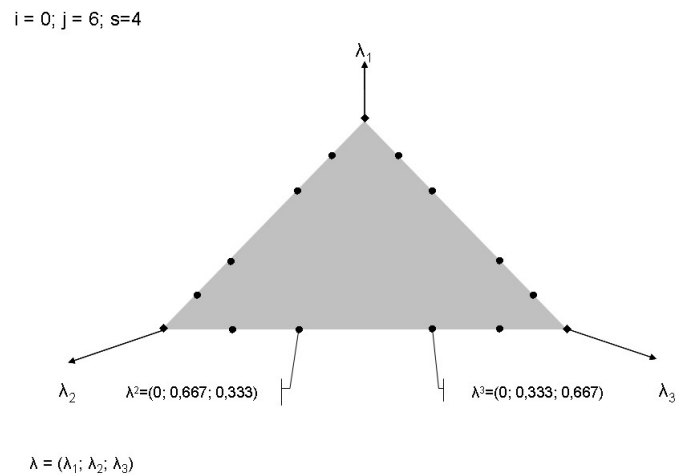
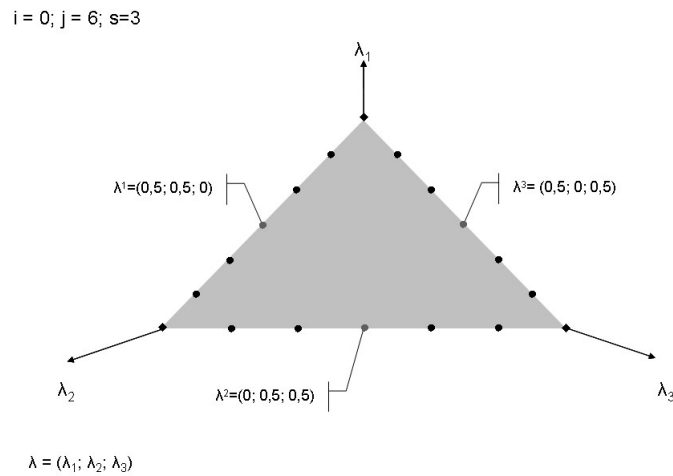


Abbildung 6.13: MCMO: $i=0; j=6; s=4$; Kolonie 1 & 2

Abbildung 6.14: MCMO: $i=0; j=6; s=4$; Kolonie 1 & 3Abbildung 6.15: MCMO: $i=0; j=6; s=4$; Kolonie 2 & 3

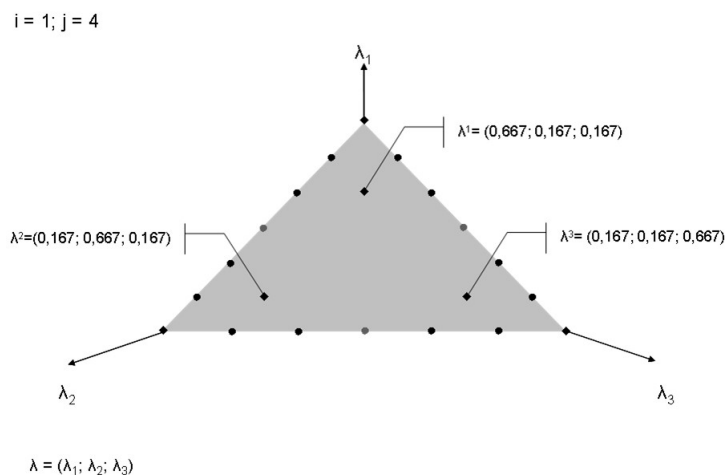
$i=0; j=6; s=3$ Es gilt $s = 3 \geq \lfloor \frac{7-0}{2} \rfloor = 3$. Die innere For-Schleife wird ein weiteres Mal durchlaufen und die Annäherung auf den drei Suchstreifen damit weitergeführt. Da sich jeweils zwei Kolonien in einem Suchbereich treffen, wird die Suche nur von jeweils einer der beiden Kolonien durchgeführt und daher die Prozedur MCMO-SUB3 verwendet.

Abbildung 6.16: MCMO: $i=0; j=6; s=3$

$i=0; j=6; s=2$ Es gilt $s = 2 \not\geq \lfloor \frac{7-0}{2} \rfloor = 3$. Die Abbruchbedingung der inneren For-Schleife ist erfüllt und die Suche auf den äußeren Suchstreifen damit beendet. Die Suche auf dem nächsten Suchstreifen kann beginnen.

$i=1; j=4$

$i = 1 \leq \lfloor \frac{7}{3} \rfloor = 2$. Die äußere For-Schleife wird ein weiteres Mal durchlaufen. In Zeile 4 errechnet sich j zu: $j = (7 - 1) - (2 \cdot 1) = 4$. Erneut werden zuerst die Endbereiche des Suchstreifens abgesucht und anschließend Kopien der Pheromonmatrizen der drei Kolonien erstellt.

Abbildung 6.17: MCMO: $i=1; j=4$

i=1; j=4; s=3 Es gilt $s = 3 \geq \lfloor \frac{7-1}{2} \rfloor = 3$. Damit beginnt die Suche in den restlichen Suchbereichen des Suchstreifens.

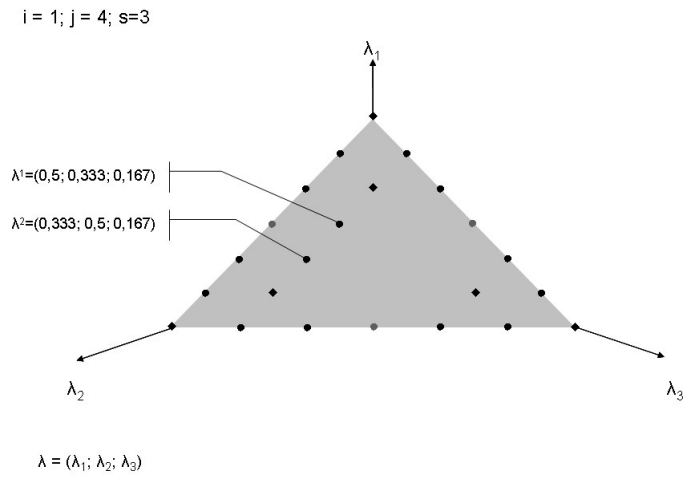


Abbildung 6.18: MCMO: $i=1; j=4; s=3$; Kolonie 1 & 2

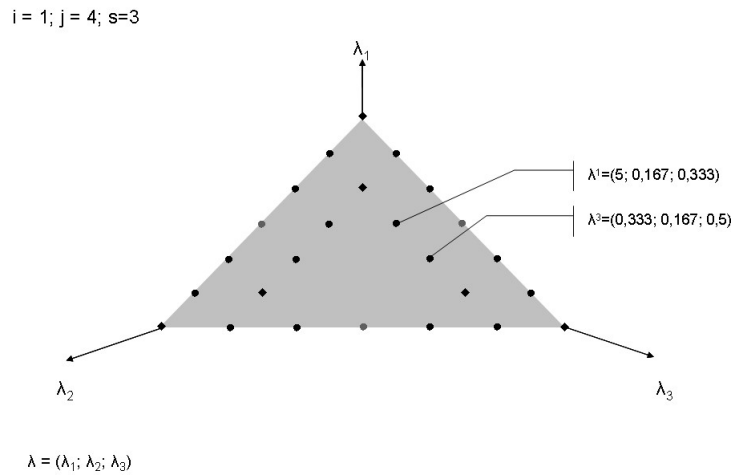


Abbildung 6.19: MCMO: $i=1; j=4; s=3$; Kolonie 1 & 3

$i = 1; j = 4; s = 3$

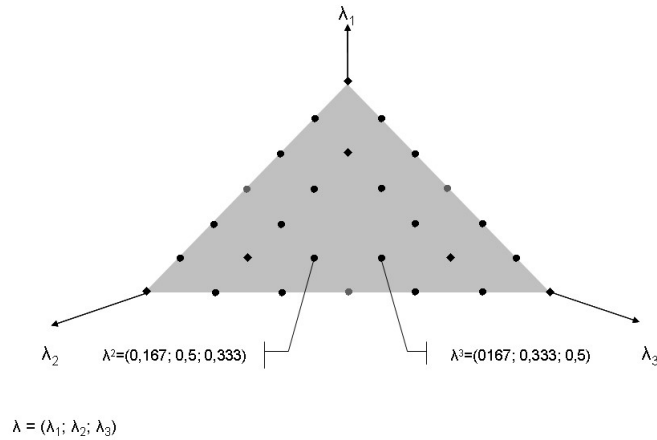


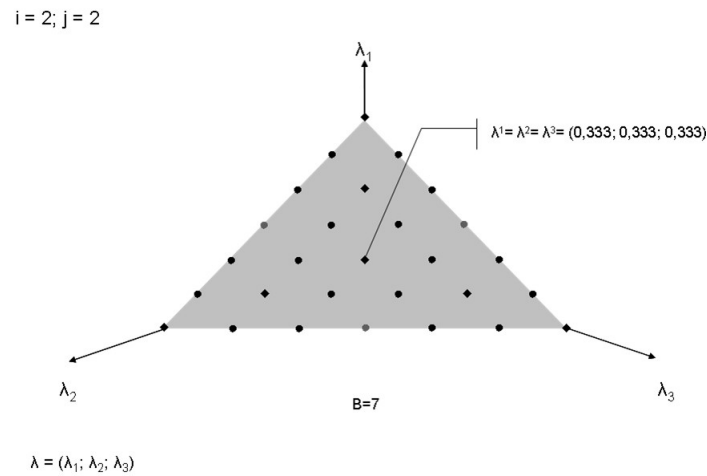
Abbildung 6.20: MCMO: $i=1; j=4; s=3$; Kolonie 2 & 3

$i=1; j=4; s=2$ Es gilt $s = 2 \not\geq \lfloor \frac{7-1}{2} \rfloor = 3$. Die Abbruchbedingung der inneren For-Schleife ist erfüllt und die Suche auf den Suchstreifen damit beendet. Die Suche auf dem nächsten Suchstreifen kann beginnen.

$i=2; j=2$

Es gilt $i = 2 \leq \lfloor \frac{7}{3} \rfloor = 2$. Die äußere For-Schleife wird ein weiteres Mal durchlaufen. In Zeile 4 errechnet sich j zu: $j = (7 - 1) - (2 \cdot 2) = 2$.

In den Zeilen 5-8 werden anschließend die Gewichte der Kompromißzielfunktion für die drei Kolonien berechnet. Da die Gewichtvektoren der drei Kolonien nun alle gleich sind, wird in der If-Else-Verzweigung in den Zeilen 9 und 10 anstelle von MCMO-SUB1 die Prozedur MCMO-SUB3 ausgeführt.

Abbildung 6.21: MCMO: $i=2; j=2$

$i=2; j=2; s=1$ Es gilt $s = 1 \not\geq \lfloor \frac{7-2}{2} \rfloor = 1$. Die innere For-Schleife wird nicht durchlaufen.

$i=3$

Es gilt $i = 3 \not\geq \lfloor \frac{7}{3} \rfloor = 2$. Die Abbruchbedingung der äußeren For-Schleife ist erfüllt. Die Heuristik wird beendet.

6.4.1.3 MCMO-CONSTRUCT - Konstruktion von Auftragsfolgen

Die Prozedur MCMO-CONSTRUCT wird von den Prozeduren MCMO-SUB1 und MCMO-SUB2 zur Konstruktion von Auftragsfolgen durch die Ameisen aufgerufen. D.h. dann, wenn zwei oder drei Kolonien gleichzeitig in unterschiedlichen Bereichen des Zielraums suchen. Die Prozedur ermöglicht eine Kommunikation der Kolonie während der Konstruktion einer Auftragsfolge, weshalb die Pheromoninformation aller drei Kolonien zur Konstruktion einer Auftragsfolge durch eine Ameise benötigt wird. Die Konstruktionsprozedur einer Kolonie ist im Algorithmus 10 angegeben.

Die Auswahl eines Elements aus der Menge der noch nicht zugeordneten Elemente erfolgt nach der erweiterten Pseudo Random Proportionate Rule von Rajendran u. Ziegler (2002), ebenso wird eine Summen-Pheromonauswertung verwendet.¹¹⁷ Der wesentliche Unterschied besteht in der Berechnung der Summen-Pheromonwerte. Die Auswahlentscheidung über die Platzierung eines Auftrags n auf Position h

¹¹⁷Vgl. Rajendran u. Ziegler (2002, S.734).

Algorithmus 10 MCMO-CONSTRUCT

```

// Parameterliste: ( $T^1, T^2, T^3, \omega$ )
//  $T^1, T^2, T^3$ : Übergebene Pheromonmatrizen der drei
//                Kolonien
//  $\omega$ :                Pheromongewichte

// Sonstige Parameter:
// cand: Größe der Kandidatenliste
//  $\pi^{glb}$ : Global beste Lösung der konstruierenden
//           (aktuellen) Kolonie

1: // Berechne aggregierte Summen-Pheromonmatrix  $T^\omega$ 
2: for(  $h=1; h \leq N; h=h+1$  ) {
3:   for(  $n=1; n \leq N; n=n+1$  ) {
4:      $T_{nh}^\omega = 0;$ 
5:     for(  $q=1; q \leq h; q=q+1$  ) {
6:        $T_{nh}^\omega = T_{nh}^\omega + \max \{ \omega_1 T_{nq}^1; \omega_2 T_{nq}^2; \omega_3 T_{nq}^3 \};$ 
7:     } // end for
8:   } // end for
9: } // end for

10: Initialisiere Menge der nicht zugeordneten
//     Elemente:  $U = \{1, 2, \dots, N\};$ 
11: Initialisiere  $\tilde{\pi}$ :  $\tilde{\pi} = \emptyset;$ 

12: for(  $h=1; h \leq N; h=h+1$  ) {
13:    $\pi^{cand} =$  Erste cand noch nicht ausgewählte Aufträge
//           aus  $\pi^{glb};$ 
14:    $u \leftarrow U[0;1];$ 
15:   if(  $u \leq p_1$  )
16:      $n =$  Erster Auftrag aus  $\pi^{cand};$ 
17:   if(  $p_1 \leq u \leq p_2$  )
18:      $n \in \operatorname{argmax} \{ T_{nh}^\omega \mid n \in \pi^{cand} \};$ 
19:   if(  $u \geq p_2$  )
//     wähle ein Auftrag  $n$  aus  $\pi^{cand}$  mit
//     Wahrscheinlichkeit  $p_{nh}$ :
//     
$$p_{nh} = \frac{T_{nh}^\omega}{\sum_{n \in \pi^{cand}} T_{nh}^\omega};$$

20:   Füge Auftrag  $n$  auf Position  $h$  in  $\tilde{\pi}$  ein;
21:    $U = U \setminus n;$ 
22: } // end for

```

wird auf Basis der aggregierten Summen-Pheromonmatrix (T^w) getroffen. Die aggregierte Summen-Pheromonmatrix wird in den Zeilen 1 bis 9 berechnet.

Die Kolonien suchen aufgrund der Parameter des Kompromißmodells nach Auftragsfolgen, deren Zielfunktionsvektoren in unterschiedlichen Bereichen des Zielraums liegen. Wie in Abschnitt 6.4.1.2 beschrieben, werden die Parameter des Kompromißmodells derart variiert, daß sich die Bereiche im Zielraum, in dem die Kolonien suchen, schrittweise annähern. Um den Annäherungsprozeß der Kolonien zu unterstützen, erhalten sie Zugriff auf die Pheromonmatrizen der jeweils anderen Kolonien. Sie erhalten damit Informationen über die Bereiche des Lösungsraums, in denen die anderen Kolonien suchen und damit Informationen darüber, über welche Auftragsfolgen die Bereiche des Zielraums erreicht werden können, in denen die anderen Kolonie aktuell suchen. Je weiter die Bereiche im Zielraum, in denen die Kolonien suchen, voneinander entfernt liegen, desto geringer ist der Einfluß "fremder" Pheromoninformationen auf die Auswahlentscheidung einer Kolonie. Der Grad, mit dem die jeweiligen Pheromoninformationen von einer Kolonie berücksichtigt werden, wird über den Gewichtevektor w eingestellt. Die Kolonien berücksichtigen stets den größten der gewichteten Pheromonwerte der drei Pheromonmatrizen. Ist nun der gewichtete Pheromonwert einer "fremden" Kolonie größer als der gewichtete Pheromonwert der eigenen Pheromonmatrix, so wird die "fremde" Pheromoninformation verwendet. In dieser Situation richtet die Kolonie ihre Suche dann verstärkt in Richtung des Bereiches im Zielraums, in dem die fremde Kolonie nach Elementen der heuristisch effizienten Menge sucht.

Sucht zu einem Zeitpunkt nur eine einzelne Kolonie, so wird anstelle der MCMO-CONSTRUCT Prozedur die Konstruktionsvorschrift von Rajendran u. Ziegler (2002) gemäß Algorithmus 3 auf Seite 160 verwendet.

6.4.1.4 MCMO-SUB1, MCMO-SUB2 und MCMO-SUB3

MCMO-SUB1

Die Prozedur MCMO-SUB1 wird verwendet, wenn alle drei Kolonien gleichzeitig in unterschiedlichen Bereichen des Zielraums suchen.

Dabei konstruieren die einzelnen Kolonien jeweils nacheinander eine Lösung unter Verwendung der MCMO-CONSTRUCT Prozedur und verbessern diese durch den Aufruf von lokalen Suchverfahren. An die Erstellung der Auftragsfolgen schließt sich die Aktualisierung der Pheromoninformation an. Diese Schritte werden *Iter*-mal wiederholt. Die Prozedur ist im Al-

gorithmus 11 angegeben.

Zeilen 1-2: Zunächst werden die Werte der aktuellen endogenen kleinsten oberen Schranken ermittelt.

Zeilen 3-7: Anschließend werden die Pheromonmatrizen der drei Kolonien initialisiert. Die Menge des bei der Initialisierung verwendeten Pheromons τ^{init} ist abhängig von der Initialisierungsauftragsfolge π^{init} , die aus der aktuellen heuristisch effizienten Menge ermittelt wird. π^{init} ist damit gleichzeitig die Initialisierung für die global beste bekannte Lösung der Kolonie im betrachteten Zielgebiet des Zielraums. Im Anschluß an die Initialisierung wird ein erstes Pheromonupdate mit π^{init} nach Gleichung 6.8 auf Seite 161 durchgeführt.

Zeilen 8-9: Da jede Kolonie in einem eigenen Bereich des Zielraums sucht, erhält jede ihre eigene lokale heuristisch effiziente Menge, welche zunächst als leere Menge initialisiert wird.

Zeilen 10+26: Für jede Region des Zielraums werden von den Kolonien *Iter*-mal Auftragsfolgen generiert und verbessert.

Zeilen 11-16: Jede Ameise der Kolonie 1 erzeugt unter Anwendung der MCMO-CONSTRUCT Prozedur eine Auftragsfolge. Anschließend wird versucht, die konstruierte Auftragsfolge in die LNDS1 einzufügen. Wird die Auftragsfolge in die LNDS1 eingefügt, so werden gleichzeitig alle Elemente aus LNDS1, welche von diesem neuen Element dominiert werden, aus der LNDS1 entfernt. Schließlich wird versucht, die konstruierte Auftragsfolge durch die sequentielle Anwendung der JIS und JSS-RAPI Verfahren zu verbessern.

Zeile 17: Die Zeilen 11-16 werden analog für Kolonie 2 (mit λ^2 und LNDS2) sowie Kolonie 3 (mit λ^3 und LNDS3) ausgeführt.

Zeilen 18: Im Anschluß an die Generierung und Verbesserung der Auftragsfolgen werden die Elemente der lokalen heuristisch effizienten Mengen in die globale heuristisch effiziente Menge eingefügt. Sollten Elemente aus den lokalen heuristisch effizienten Mengen in die globale heuristisch effiziente Menge aufgenommen werden, so werden gleichzeitig alle von diesen neuen Elementen dominierten Elemente aus der global heuristisch effizienten Menge entfernt. Die in den lokalen heuristisch effizienten Mengen enthaltenen Elemente werden jedoch nicht aus diesen entfernt. Die lokalen heuristisch effizienten Mengen bleiben bei dieser Prozedur also unverändert.

Algorithmus 11 MCMO-SUB1

```

// Parameterliste: ( $\lambda^1, \lambda^2, \lambda^3$  GNDS)
//  $\lambda^1, \lambda^2, \lambda^3$ : Gewichtungsvektoren des Kompromißmodells
// der drei Kolonien
// GNDS: übergebene globale heuristisch effiziente
// Menge

// Weitere Parameter:
// Ants: Anzahl der Ameisen in den Kolonien
//  $T^1, T^2, T^3$ : Pheromonmatrizen der Kolonien 1, 2 und 3.
// Q: Parameter bei Pheromonaktualisierung, Wert bei Start der
// Heuristik vorgegeben.
//  $\rho$ : Trail-Persistence-Rate, Wert bei Start der
// Heuristik vorgegeben.

1: // Ermittle endogene kleinste obere Schranken
2:  $z_k^{EUB} = \min\{z_k(\pi) \mid \pi \in GNDS\}$ ;  $k=1(1)3$ ;

3: Initialisierung der Pheromonmatrizen der drei Kolonien:
4:  $\pi^{init} \in \operatorname{argmin}\{\Psi_{\lambda^1}(\pi) \mid \pi \in GNDS\}$ ;
5:  $\tau_{nh}^1 = \tau^{init} = \frac{Q}{\Psi_{\lambda^1}(\pi^{init})}$ ;  $n=1(1)N$ ;  $h=1(1)N$ ;
   Q wie bei Start der Heuristik vorgegeben.
6: Führe Pheromonaktualisierung mit  $\pi^{init}$  gemäß der
   Gleichung 6.8 auf Seite 161 aus,  $L(\pi^{ant}) = \Psi_{\lambda^1}(\pi^{init})$ ;
7: ... Analog für die Kolonien 2 ( $\lambda^2$ ) und Kolonie 3 ( $\lambda^3$ )
8: // Initialisiere lokale heuristisch effiziente
   Mengen der Kolonien (LNDS)
9: LNDS1 =  $\emptyset$ ; LNDS2 =  $\emptyset$ ; LNDS3 =  $\emptyset$ ;

10: for( iter=1; iter $\leq$ Iter; iter=iter+1 ) {
11:   for( ant=1; ant $\leq$ Ants; ant=ant+1 ) {
12:      $\pi^{ant} \leftarrow \text{MCMO-CONSTRUCT}( T^1, T^2, T^3, \lambda^1 )$ ;
13:     Versuche  $\pi^{ant}$  in LNDS1 einzufügen
14:     JIS(  $\pi^{ant}, \lambda^1, \text{LNDS1}$  );
15:     JSS-RAPI(  $\pi^{ant}, \lambda^1, \text{LNDS1}$  );
16:   } // end for
17:   ... Analog für Kolonie 2 ( $\lambda^2, \text{LNDS2}$ )
     und Kolonie 3 ( $\lambda^3, \text{LNDS3}$ )

18: Versuche die Elemente aus LNDS1, LNDS2 und LNDS3
   in GNDS einzufügen;

19: // Aktualisiere Pheromonmatrix der Kolonie 1:
20: Führe Pheromonaktualisierung mit der besten Ameise
   der aktuellen Iteration gemäß Gleichung 6.8
   aus.  $L(\pi^{ant}) = \text{Kompromißzielfunktionswert der}$ 
   besten Ameise der Iteration.
21: Führe Pheromonaktualisierung mit  $\pi^{glb}$  gemäß
   Gleichungen 6.8 aus.  $L(\pi^{ant}) = \Psi_{\lambda^1}(\pi^{glb})$ .
22: Führe Pheromonupdate mit einem beliebigen Element
   aus GNDS durch, für das gilt:
    $\pi^* \in \operatorname{argmin}\{\Psi_{\lambda^1}(\pi) \mid \pi \in GNDS\}$ ;  $L(\pi^{ant}) = \Psi_{\lambda^1}(\pi^*)$ .

23: ... Analog für die Pheromonmatrizen der Kolonie 2
   und 3.

24: // Aktualisiere endogene kleinste obere Schranken
25:  $z_k^{EUB} = \min\{z_k(\pi) \mid \pi \in GNDS\}$ ;  $k=1(1)3$ ;
26: } // end for

```

Zeilen 19-23: An dieser Stelle erfolgt die Aktualisierung der Pheromoninformationen der Kolonien. Für jede Kolonie werden drei Pheromonupdates durchgeführt.

Zunächst wird eine Pheromonaktualisierung durch die beste Ameise der Iteration durchgeführt. Anschließend wird eine Aktualisierung durch die bisher beste gefundene Lösung, bezogen auf den Kompromißzielfunktionswert, durchgeführt. Ziel dieser Strategie ist es, sowohl Informationen über neue, vielversprechende Bereiche des Lösungsraum zu berücksichtigen (1. Update) als auch gleichzeitig, durch das 2. Update, Informationen über die bisher beste bekannte Auftragsfolge π^{glb} nicht zu vernachlässigen. Soweit entspricht das Vorgehen dem Pheromonupdate des SCMO-Ansatzes.

Zusätzlich zum SCMO-Ansatz erfolgt hier ein drittes Pheromonupdate. Dieses wird durch ein Element der aktuellen globalen heuristisch effizienten Menge durchgeführt, welches den Kompromißzielfunktionswert der Kolonie minimiert. Diese Auftragsfolge wird analog zu π^{init} ermittelt. Dieses letzte Update ermöglicht die Kommunikation von Lösungen zwischen den Kolonien, da die in diesem Schritt ausgewählte Auftragsfolge nicht von derselben Kolonie ermittelt worden sein muss, deren Pheromonmatrix gerade aktualisiert wird.

In allen drei Pheromonaktualisierungen wird $L(\pi^{ant})$ gleich dem Kompromißzielfunktionswert der für die Aktualisierung verwendeten Auftragsfolge gesetzt. Die Parameter Q und ρ werden zu Beginn des Verfahrens vorgegeben. Für die für sie verwendeten Werte vergleiche Abschnitt 6.3.2.2.

Zeile 24-25: Zum Abschluß werden die endogenen kleinsten oberen Schranken aktualisiert, die sich durch die Schritte in den Zeilen 11-18 verbessert haben können.

MCMO-SUB2

Die Prozedur MCMO-SUB2 wird verwendet, wenn zwei Kolonien parallel in unterschiedlichen Bereichen des Zielraums suchen. Vom grundsätzlichen Aufbau her entspricht diese Prozedur MCMO-SUB1. Damit die Ameisen mit Hilfe von MCMO-CONSTRUCT eine Auftragsfolge konstruieren können, benötigen sie jedoch auch Zugriff auf die Pheromoninformationen der dritten, nicht aktiv suchenden Kolonie. Diese erhalten sie, indem MCMO-SUB2 die in MCMO-MASTER erstellte Kopie der Pheromonmatrix der dritten, nicht aktiv suchenden Kolonie übergeben wird. Die Prozedur ist im Algorithmus 12 wiedergegeben.

Algorithmus 12 MCMO-SUB2

```

// Parameterliste: ( $T^A$ ,  $T^B$ ,  $T^{fix}$ ,  $\lambda^A$ ,  $\lambda^B$ , GNDS)
//  $T^A$ ,  $T^B$ : An die Prozedur übergebene Pheromonmatrix der
//           Kolonie A bzw. B
//  $\lambda^A$ ,  $\lambda^B$ : Gewichtungsvektoren des Kompromißmodells der
//           aktiven Kolonien A und B
//  $T^{fix}$ : Pheromonmatrix der inaktiven Kolonie
// GNDS: übergebene globale heuristisch effiziente Menge

// Weitere Parameter:
// Ants: Anzahl der Ameisen in den Kolonien
// Q: Parameter bei Pheromonaktualisierung, Wert bei Start der
//   Heuristik vorgegeben.
//  $\rho$ : Trail-Persistence-Rate, Wert bei Start
//      der Heuristik vorgegeben

1: // Ermittle endogene kleinste obere Schranken
2:  $z_k^{EUB} = \min\{z_k(\pi) \mid \pi \in GNDS\}$ ;  $k=1(1)3$ ;

3: Initialisierung der Pheromonmatrizen der zwei Kolonien:
4:  $\pi^{init} \in \operatorname{argmin}\{\Psi_{\lambda^A}(\pi) \mid \pi \in GNDS\}$ ;
5:  $\tau_{nh}^1 = \tau^{init} = \frac{Q}{\Psi_{\lambda}(\pi^{init})}$ ;  $n=1(1)N$ ;  $h=1(1)N$ ;
6: Führe Pheromonaktualisierung mit  $\pi^{init}$  gemäß der
   Gleichung 6.8 auf Seite 161 aus,  $L(\pi^{ant}) = \Psi_{\lambda}(\pi^{init})$ ;

7: ... Analog für die Kolonien B ( $\lambda^B$ )

8: // Initialisiere lokale heuristisch effiziente
   Mengen der Kolonien (LNDS)
9: LNDSA =  $\emptyset$ ; LNDSB =  $\emptyset$ ;

10: for( iter=1; iter $\leq$ Iter; iter=iter+1 ) {
11:   for( ant=1; ant $\leq$ Ants; ant=ant+1 ) {
12:      $\pi^{ant} \leftarrow \text{MCMO-CONSTRUCT}( T^A, T^B, T^{fix}, \lambda^A )$ ;
13:     Versuche  $\pi^{ant}$  in LNDSA einzufügen
14:     JIS(  $\pi^{ant}$ ,  $\lambda^A$ , LNDSA );
15:     JSS-RAPI(  $\pi^{ant}$ ,  $\lambda^A$ , LNDSA );
16:   } // end for
17:   ... Analog für Kolonie B ( $\lambda^B$ , LNDSB)

18: Versuche die Elemente aus LNDSA und LNDSB
   in GNDS einzufügen;

19: // Aktualisiere Pheromonmatrix der Kolonie A:
20: Führe Pheromonaktualisierung mit der besten Ameise
   der aktuellen Iteration von Kolonie A gemäß
   Gleichung 6.8 aus.
    $L(\pi^{ant}) =$  Kompromißzielfunktionswert der
   besten Ameise der Iteration.
21: Führe Pheromonaktualisierung mit  $\pi^{glb}$  gemäß
   Gleichungen 6.8 aus.  $L(\pi^{ant}) = \Psi_{\lambda}(\pi^{glb})$ .
22: Führe Pheromonupdate mit einem beliebigen Element
   aus GNDS durch, für das gilt:
    $\pi^* \in \operatorname{argmin}\{\Psi_{\lambda^1}(\pi) \mid \pi \in GNDS\}$ ;  $L(\pi^{ant}) = \Psi_{\lambda}(\pi^*)$ .
23: ... Analog für die Pheromonmatrix der Kolonie B

24: // Aktualisiere endogene kleinste obere Schranken
25:  $z_k^{EUB} = \min\{z_k(\pi) \mid \pi \in GNDS\}$ ;  $k=1(1)3$ ;
26: } // end for

```

MCMO-SUB3

Diese Prozedur wird aufgerufen, wenn sich zwei oder drei Kolonien in der gleichen Suchregion treffen. In diesem Fall sucht nur eine einzige Kolonie. Auf eine Kommunikation zwischen den Kolonien kann verzichtet werden. Die Prozedur entspricht somit dem Ausführen der SCMO-SUCHE-REGION Prozedur.

6.4.2 Modifikation des MCMO-Ansatz - Variante 2 (MCMO2)

Der wesentliche Unterschied zur Variante 1 ist, daß in Variante 2 auf die Kommunikation zwischen den Kolonien bei der Lösungskonstruktion verzichtet wird. Die Systematik, mit der der Zielraum abgesucht wird, bleibt unverändert. Die Kommunikation während der Lösungskonstruktion wird in Variante 1 durch die Verwendung der Prozedur MCMO-CONSTRUCT und der dort berechneten Pheromonmatrix T^w realisiert. Soll nun diese Kommunikation unterbunden werden, so dürfen die Ameisen einer Kolonie bei der Konstruktion von Auftragsfolgen nur auf die Pheromoninformationen ihrer Kolonie zugreifen. Im Variante 2 des MCMO-Ansatzes wird die Prozedur MCMO-CONSTRUCT daher nicht verwendet.

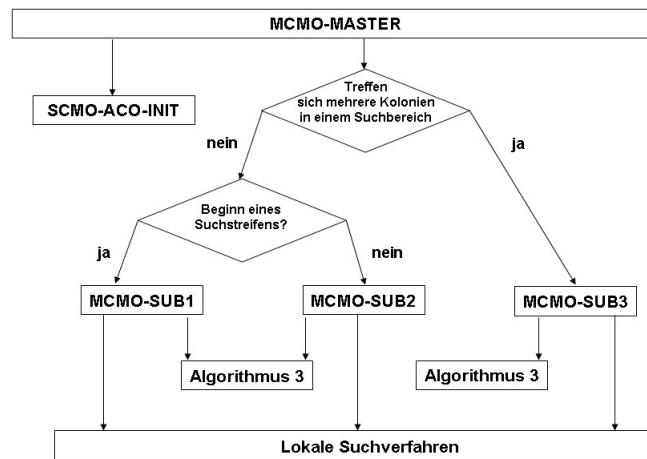


Abbildung 6.22: MCMO2 - Übersicht der Prozeduren

Die Ameisen konstruieren die Auftragsfolgen nach dem von Rajendran u. Ziegler (2002) vorgestellten Verfahren.¹¹⁸ Diese Konstruktionsprozedur wurde bereits im SCMO-Ansatz und in MCMO-SUB3 verwendet und ist im

¹¹⁸Vgl. Rajendran u. Ziegler (2002, S.734).

Algorithmus 3 auf Seite 160 beschrieben. Abbildung 6.22 stellt den veränderten Zusammenhang der Prozeduren dar.

Trotz dieser Modifikation entspricht MCMO2 nicht dem SCMO-Ansatz, insbesondere ist es kein Single-Colony-Ansatz. Es besteht zwar im Vergleich zum SCMO-Ansatz kein Unterschied bei der Lösungskonstruktion mehr, jedoch wird die Suche weiterhin gleichzeitig von mehreren Kolonien in unterschiedlichen Bereichen des Zielraums durchgeführt. So wird z.B. weiterhin unmittelbar nach der Initialisierungsphase in allen Regionen des Zielraums, in denen die individuellen globalen Optima der einzelnen Ziele liegen, gleichzeitig gesucht. Im SCMO-Ansatz wird nach der Initialisierungsphase nur die Region des Zielraums untersucht, in der die globalen Optima des mit λ_1 gewichteten Ziels liegen. Die Region des Zielraums, in dem die globalen Optima des mit λ_3 gewichteten Ziels liegen, wird hingegen als letzter der Bereiche untersucht. Weiterhin bleibt die Kommunikation der Kolonien bei der Aktualisierung der Pheromoninformationen erhalten.

6.4.3 Experimentelle Analyse

6.4.3.1 Experiment 1: Kommunikationsstrategien

Die in diesem und den folgenden Experimenten verwendeten Testprobleme und Auswertungsgrößen entsprechen denen, die bereits in der experimentellen Analyse des SCMO-Ansatzes verwendet wurden (vgl. Abschnitt 6.3.2.1). Auch die Ausgangsparametereinstellungen wurden vollständig übernommen (vgl. Abschnitt 6.3.2.2).

In diesem ersten Experiment wird die Vorteilhaftigkeit der Kommunikationsstrategien zwischen den drei Kolonien überprüft. Die Kommunikation nach der Lösungskonstruktion wird erreicht, indem ein Pheromonupdate durch ein Element der aktuellen heuristisch effizienten Menge ausgeführt wird, welches den Kompromißzielfunktionswert der Kolonie minimiert.¹¹⁹ Für diese Kommunikationsform wurden drei Varianten überprüft:

Variante 1: Die Kommunikation nach der Lösungskonstruktion wird nicht verwendet, d.h. das Update grundsätzlich nicht ausgeführt.

Variante 2: Die Kommunikation nach der Lösungskonstruktion wird grundsätzlich verwendet, d.h. das Update grundsätzlich ausgeführt.

Variante 3: Die Kommunikation nach der Lösungskonstruktion wird nur dann ausgeführt, wenn der Kompromißzielfunktionswert des

¹¹⁹Vgl. MCMO-SUB1 (Algorithmus 11), Zeile 22 und MCMO-SUB2 (Algorithmus 12), Zeile 22.

Elements aus der heuristisch effizienten Menge kleiner ist als der Kompromißzielfunktionswert der global besten Lösung der Kolonie.

Die Vorteilhaftigkeit dieser drei Varianten wurde für MCMO1 und MCMO2 überprüft. Die Ergebnisse sind in den Tabellen 6.21 bis 6.25 angegeben.

Variante	N	MCMO1		MCMO2	
		SQ^μ	SQ^σ	SQ^μ	SQ^σ
1	20	0,8568	0,0422	0,7999	0,0469
	50	0,7358	0,0522	0,6763	0,0569
	100	0,6506	0,0632	0,6045	0,0582
	Avg	0,7477	0,0525	0,6936	0,0540
2	20	0,8527	0,0428	0,7887	0,0485
	50	0,7614	0,0589	0,6941	0,0625
	100	0,6824	0,0629	0,6900	0,0663
	Avg	0,7655	0,0549	0,7243	0,0591
3	20	0,8578	0,0401	0,8021	0,0455
	50	0,7433	0,0547	0,6762	0,0594
	100	0,6572	0,0603	0,6032	0,0571
	Avg	0,7528	0,0517	0,6938	0,0540

Tabelle 6.21: MCMO - Exp. 1: SQ -Maß

SQ -Maß: Beim SQ -Maß ist MCMO1, außer bei Updatevariante 2 und Problemen mit 100 Aufträgen, bzgl. SQ_v^μ immer besser als MCMO2. Bei Problemen mit 20 und 50 Aufträgen dominiert MCMO1 MCMO2. Die Kommunikation während der Lösungskonstruktion erweist sich bei Betrachtung des SQ -Maßes als vorteilhaft. Bei $N=20$ ist es sowohl bei MCMO1 als auch bei MCMO2 vorteilhaft, Updatevariante 3 zu verwenden. Für $N=50$ und $N=100$ ist Updatevariante 2 zu bevorzugen.

DUB^k -Maß: Die Ergebnisse des DUB^k -Maßes zeigen, daß bei der Zykluszeit, mit Ausnahme von Variante 2, bei Problemen mit 20 Aufträgen MCMO2 bessere Werte bzgl. $DUB_v^{\mu Cmax}$ hat als MCMO1. Bei den Varianten 1 und 3 dominiert MCMO2 MCMO1 sogar. Bei der mittleren Durchlaufzeit ist bei allen drei Updatevarianten, außer bei Problemen mit 20 Aufträgen, $DUB_v^{\mu Fmid}$ bei MCMO2 besser als bei MCMO1. Auch bei der maximalen Terminüberschreitung ist MCMO2 bei Problemgrößen mit 50 und 100 Aufträgen bei allen Updatevarianten bezüglich $DUB_v^{\mu Tmax}$ besser als MCMO1. Im Fall $N=20$ hat auch hier MCMO1 die besseren Werte für $DUB_v^{\mu Tmax}$. Im Gegensatz zum SQ -Maß erweist sich beim DUB^k -Maß der Verzicht auf eine Kommunikation während der Lösungskonstruktion als vorteilhaft.

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0062	0,0030	1,0028	0,0020	1,0119	0,0074
	50	1,0166	0,0025	1,0082	0,0028	1,0164	0,0061
	100	1,0144	0,0012	1,0135	0,0028	1,0113	0,0033
	Avg	1,0124	0,0022	1,0082	0,0025	1,0132	0,0056
2	20	1,0064	0,0030	1,0031	0,0021	1,0117	0,0084
	50	1,0155	0,0025	1,0079	0,0030	1,0147	0,0063
	100	1,0137	0,0011	1,0130	0,0027	1,0095	0,0036
	Avg	1,0119	0,0022	1,0080	0,0026	1,0120	0,0061
3	20	1,0060	0,0030	1,0028	0,0021	1,0120	0,0073
	50	1,0168	0,0026	1,0082	0,0028	1,0160	0,0057
	100	1,0144	0,0012	1,0133	0,0027	1,0110	0,0034
	Avg	1,0124	0,0022	1,0081	0,0025	1,0130	0,0055

Tabelle 6.22: MCMO1 - Exp. 1: DUB^k -Maß

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0060	0,0029	1,0032	0,0021	1,0132	0,0076
	50	1,0149	0,0024	1,0077	0,0030	1,0152	0,0057
	100	1,0135	0,0012	1,0123	0,0029	1,0104	0,0033
	Avg	1,0115	0,0022	1,0077	0,0027	1,0129	0,0056
2	20	1,0065	0,0031	1,0036	0,0024	1,0147	0,0077
	50	1,0140	0,0025	1,0074	0,0033	1,0141	0,0064
	100	1,0119	0,0012	1,0113	0,0026	1,0069	0,0035
	Avg	1,0108	0,0023	1,0074	0,0027	1,0119	0,0059
3	20	1,0060	0,0029	1,0032	0,0022	1,0130	0,0078
	50	1,0148	0,0026	1,0077	0,0030	1,0151	0,0059
	100	1,0135	0,0012	1,0124	0,0028	1,0102	0,0032
	Avg	1,0114	0,0022	1,0078	0,0027	1,0128	0,0056

Tabelle 6.23: MCMO2 - Exp. 1: DUB^k -Maß

AEX^k -Maß: Bei kleinen Problemen mit 20 Aufträgen ist, außer bei $AEX_3^{\mu Cmax}$, MCMO1 besser als MCMO2. Bei größeren Problemen ist von zwei Ausnahmen abgesehen MCMO2 besser. Bei Problemen mit 100 Aufträgen ist $AEX_1^{\mu Fmid}$ und $AEX_3^{\mu Tmax}$ bei MCMO1 besser als bei MCMO2.

Variante	N	$Cmax$		$Fmid$		$Tmax$	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0892	0,1735	0,3217	0,3346	0,0746	0,1855
	50	0,0067	0,0200	0,0167	0,0500	0,0100	0,0300
	100	0,0057	0,0170	0,0167	0,0500	0,0000	0,0000
	Avg	0,0339	0,0702	0,1183	0,1449	0,0282	0,0718
2	20	0,0740	0,1454	0,2567	0,3076	0,0683	0,1680
	50	0,0113	0,0340	0,0167	0,0500	0,0167	0,0500
	100	0,0067	0,0200	0,0100	0,0300	0,0200	0,0600
	Avg	0,0307	0,0665	0,0944	0,1292	0,0350	0,0927
3	20	0,0737	0,1391	0,3267	0,3558	0,0797	0,1874
	50	0,0067	0,0200	0,0133	0,0400	0,0067	0,0200
	100	0,0007	0,0020	0,0000	0,0000	0,0100	0,0300
	Avg	0,0270	0,0537	0,1133	0,1319	0,0321	0,0791

Tabelle 6.24: MCMO1 - Exp. 1: AEX^k -Maß

Variante	N	$Cmax$		$Fmid$		$Tmax$	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0723	0,1115	0,2633	0,2699	0,0704	0,1583
	50	0,0074	0,0223	0,0200	0,0600	0,0133	0,0400
	100	0,0119	0,0358	0,0133	0,0400	0,0067	0,0200
	Avg	0,0306	0,0566	0,0989	0,1233	0,0301	0,0728
2	20	0,0678	0,1391	0,2283	0,2718	0,0309	0,0740
	50	0,0562	0,1652	0,0267	0,0800	0,0400	0,1200
	100	0,0724	0,2092	0,0400	0,1200	0,0633	0,1867
	Avg	0,0655	0,1712	0,0983	0,1573	0,0447	0,1269
3	20	0,0784	0,1276	0,2717	0,2681	0,0710	0,1447
	50	0,0241	0,0723	0,0133	0,0400	0,0167	0,0500
	100	0,0108	0,0325	0,0200	0,0600	0,0000	0,0000
	Avg	0,0378	0,0775	0,1017	0,1227	0,0292	0,0649

Tabelle 6.25: MCMO2 - Exp. 1: AEX^k -Maß

Bei alleiniger Berücksichtigung von MCMO2 ist Updatevariante 2 unter Beachtung aller drei Auswertungsgrößen eine gute Strategie. Im Durchschnitt über alle Problemgrößen weist sie für SQ_v^μ wie auch für $DUB_v^{\mu k}$ bei allen drei Zielgrößen und bei $AEX^{\mu k}$ für die Zielgrößen Zykluszeit und maximale Terminüberschreitung die besten Werte auf.

Bei alleiniger Berücksichtigung von MCMO1 ist ebenfalls Updatevariante 2 unter Beachtung aller drei Auswertungsgrößen eine gute Strategie. Sie hat im Durchschnitt über alle Problemgrößen die besten Werte für SQ_v^μ und $DUB_v^{\mu k}$, und zwar bei allen drei Zielgrößen. Bei $AEX_v^{\mu k}$ ist dies nur bei der maximalen Terminüberschreitung der Fall. Für die anderen beiden Zielgrößen hat Updatevariante 1 die besseren Werte.

6.4.3.2 Experiment 2: Variation von B

In diesem Experiment soll der Einfluß einer unterschiedlich feinen Einteilung des Zielraums in Suchregionen auf die Lösungsqualität und Rechenzeit untersucht werden. Als Werte für B werden die Werte 4, 5, 6, 7, 8, 9 und 10 gewählt.¹²⁰ Die Tabellen 6.26 bis 6.31 geben die Relation der Werte der Maße zur Beurteilung der Lösungsqualität für die $v = 1(1)7$ Konfigurationen, bezogen auf die Werte der Konfiguration, mit $B=10$ wieder.

B	N	ΔSQ^μ	ΔSQ^σ	$\Delta Ticks^\mu$	$\Delta Ticks^\sigma$
4	20	0,8610	1,2464	0,1906	0,4582
	50	0,7612	1,0091	0,1887	0,4319
	100	0,7244	0,9173	0,1855	0,2707
	Avg	0,7862	1,0353	0,1861	0,3013
5	20	0,9165	1,3128	0,2819	0,5565
	50	0,8456	0,9764	0,2772	0,4756
	100	0,8078	1,0048	0,2742	0,3449
	Avg	0,8598	1,0741	0,2748	0,3725
6	20	0,9567	1,1178	0,4457	0,6773
	50	0,9029	0,9745	0,4431	0,6266
	100	0,8719	0,9853	0,4420	0,4880
	Avg	0,9129	1,0156	0,4423	0,5156
7	20	0,9652	1,0822	0,5162	0,6286
	50	0,9402	1,0394	0,5119	0,7168
	100	0,9069	1,0492	0,5090	0,4918
	Avg	0,9391	1,0542	0,5096	0,5288
8	20	0,9726	1,0792	0,6550	0,7055
	50	0,9537	1,0661	0,6557	0,6984
	100	0,9293	1,0406	0,6545	0,6000
	Avg	0,9531	1,0597	0,6547	0,6183
9	20	0,9889	1,0581	0,8067	0,7104
	50	0,9810	0,9614	0,8155	0,7433
	100	0,9817	1,1452	0,8156	0,6800
	Avg	0,9841	1,0566	0,8154	0,6901
10	20	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000

Tabelle 6.26: MCMO1 - Exp. 2: ΔSQ_v und $\Delta Ticks_v$

Diese wurde als Basis gewählt, da sie in der Regel die beste Lösungsgüte aller untersuchten Konfigurationen aufweist. Lediglich in drei Fällen besitzt eine andere Konfiguration eine bessere Lösungsgüte. Bei MCMO1 ist dies $DUB_{B=9}^{\mu Cmax}$ bei Problemen mit 20 Aufträgen sowie $AEX_{B=9}^{\mu Cmax}$ bei

¹²⁰Für dieses und alle folgenden Experimente wurde Rechnerkonfiguration 2 verwendet.

Problemen mit 100 Aufträgen. Bei MCMO2 ist bei 50 Aufträgen $AEX_{B=9}^{\mu T_{max}}$ besser.

SQ-Maß: Beim SQ-Maß zeigt sich für MCMO1 (vgl. Tabelle 6.26), daß sich die Werte von SQ_v^μ mit steigendem B verbessern. Die Reduktion von B bewirkt bei großen Problemen ein stärkeres Absinken der Lösungsgüte als bei kleineren Problemen. So beträgt die Lösungsqualität bei B=4 und Problemen mit 20 Aufträgen noch 86,10% der Lösungsqualität bei B=10. Für Probleme mit 100 Aufträgen beträgt sie nur noch 72,44%.

Die durchschnittlichen Werte von ΔSQ_v^μ und $\Delta Ticks_v^\mu$ einer Konfiguration v über alle Probleminstanzen aus Tabelle 6.26 sind für alle $v = 1(1)7$ betrachteten Varianten in Abbildung 6.23 wiedergegeben.

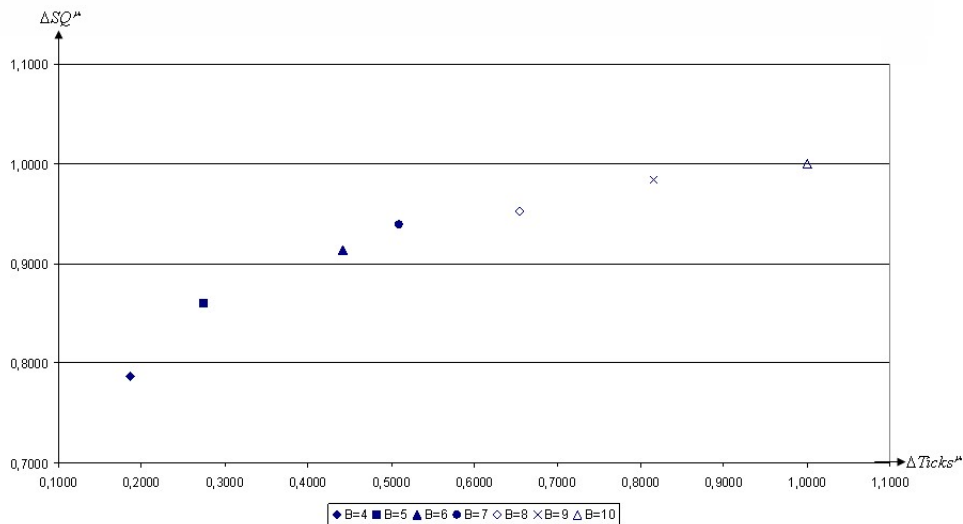


Abbildung 6.23: MCMO1 - Trade Off: $\Delta SQ^\mu - \Delta Ticks_v^\mu$

Das Erhöhen des Parameters B läßt die Rechenzeit ansteigen. Es ist zu erkennen, daß der Anstieg der Lösungsqualität unterproportional zum Anstieg der Rechenzeit ist bzw. der Grenznutzen einer weiteren Erhöhung von B abnehmend ist.

Im Gegensatz zu MCMO1 verschlechtert sich bei MCMO2 (vgl. Tabelle 6.27) für alle Problemgrößen die Lösungsqualität bzgl. $SQ_v^{\mu^k}$ bei B=7 im Vergleich zu B=6 wieder. Ansonsten steigen auch hier die Werte von SQ_v^{μ} mit steigendem B. Auch bei MCMO2 bewirkt das Verringern von B=10 auf B=4 bei kleinen Problemen einen geringeren Rückgang der Lösungsqualität als bei großen.

B	N	ΔSQ^{μ}	ΔSQ^{σ}	$\Delta Ticks^{\mu}$	$\Delta Ticks^{\sigma}$
4	20	0,8664	1,3046	0,1932	0,4676
	50	0,7623	0,9501	0,1898	0,4063
	100	0,6565	0,8316	0,1862	0,3467
	Avg	0,7673	0,9899	0,1868	0,3624
5	20	0,8986	1,1723	0,2838	0,5126
	50	0,8141	1,0240	0,2791	0,5128
	100	0,7141	0,7960	0,2749	0,4320
	Avg	0,8138	0,9693	0,2757	0,4488
6	20	0,9996	1,0603	0,4453	0,7568
	50	0,9996	1,0616	0,4442	0,5601
	100	0,9362	0,9495	0,4445	0,5522
	Avg	0,9797	1,0165	0,4445	0,5645
7	20	0,9483	1,1662	0,5370	1,4219
	50	0,9164	1,0491	0,5166	0,9089
	100	0,8638	0,9666	0,5123	0,6977
	Avg	0,9116	1,0449	0,5135	0,7694
8	20	0,9709	1,1838	0,6726	0,9835
	50	0,9478	1,0911	0,6613	0,8325
	100	0,9125	0,9951	0,6597	0,7340
	Avg	0,9452	1,0755	0,6602	0,7626
9	20	0,9838	1,0364	0,8271	0,9769
	50	0,9703	1,0415	0,8217	0,8289
	100	0,9567	0,9680	0,8204	0,7864
	Avg	0,9710	1,0109	0,8207	0,8032
10	20	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000

Tabelle 6.27: MCMO2 - Exp. 2: SQ -Maß

Das Erhöhen des Parameters B läßt die Rechenzeit ansteigen. Auch hier ist der Anstieg der Lösungsqualität tendenziell unterproportional zum Anstieg der Rechenzeit (vgl. Abbildung 6.24).

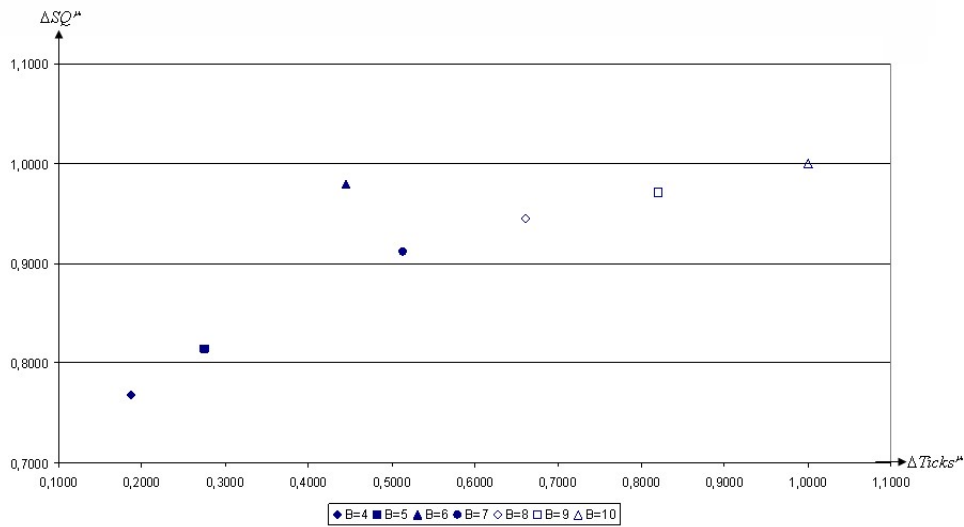


Abbildung 6.24: MCMO2 - Trade Off: $\Delta SQ^\mu - \Delta Ticks_v^\mu$:

Deutlich zu erkennen ist der oben angesprochene Ausreißerwert bei $B=6$. Im Durchschnitt über alle Problemgrößen ist erst die Konfiguration $B=10$ wieder besser als $B=6$. Um zu überprüfen, ob dieser Ausreißerwert zufällig ist, wurde die Lösung für $B=6$ ein zweites Mal mit anderen Zufallszahlenströmen berechnet. Auch diese Berechnung erzeugt einen Ausreißer. Für diesen Ausreißerwert gibt es zur Zeit keine logische Erklärung, zumal er nicht bei allen Auswertungsgrößen gleichermaßen auftritt.

B	N	C_{max}		F_{mid}		T_{max}	
		ΔDUB^μ	ΔDUB^σ	ΔDUB^μ	ΔDUB^σ	ΔDUB^μ	ΔDUB^σ
4	20	1,0036	1,1770	1,0027	1,5077	1,0107	1,5901
	50	1,0042	1,2197	1,0057	0,9949	1,0118	1,0723
	100	1,0018	0,9625	1,0068	0,9102	1,0066	1,0503
	Avg	1,0032	1,1531	1,0051	1,0790	1,0097	1,2676
5	20	1,0027	1,2277	1,0019	1,4729	1,0077	1,6199
	50	1,0039	1,1330	1,0048	0,9634	1,0093	0,9890
	100	1,0016	1,0004	1,0057	0,8525	1,0055	1,0536
	Avg	1,0027	1,1497	1,0041	1,0366	1,0075	1,2466
6	20	1,0018	1,0825	1,0014	1,3565	1,0031	1,1862
	50	1,0021	1,2179	1,0029	0,9904	1,0055	0,9902
	100	1,0011	1,1027	1,0037	0,9735	1,0035	1,0925
	Avg	1,0016	1,1372	1,0027	1,0673	1,0040	1,0879
7	20	1,0009	1,1686	1,0008	1,1949	1,0028	1,4304
	50	1,0013	1,1532	1,0022	1,0114	1,0035	1,0363
	100	1,0008	1,0076	1,0032	0,8939	1,0022	1,1013
	Avg	1,0010	1,1328	1,0021	1,0078	1,0028	1,2025
8	20	1,0003	1,1087	1,0004	1,1103	1,0019	1,2155
	50	1,0009	1,0453	1,0011	0,9653	1,0027	0,9546
	100	1,0005	1,0355	1,0015	0,9723	1,0015	1,0146
	Avg	1,0006	1,0712	1,0010	1,0011	1,0020	1,0683
9	20	0,9998	1,0904	1,0001	1,0336	1,0014	1,1477
	50	1,0005	0,9709	1,0005	0,9975	1,0014	0,9430
	100	1,0002	1,0400	1,0012	0,9367	1,0006	1,0068
	Avg	1,0002	1,0361	1,0006	0,9822	1,0011	1,0358
10	20	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Tabelle 6.28: MCMO1 - Exp. 2: ΔDUB_v^k -

DUB^k -Maß: Beim DUB^k -Maß kann bei MCMO1 (vgl. Tabelle 6.28) beobachtet werden, daß sich $DUB_v^{\mu k}$ für die mittlere Durchlaufzeit und die maximale Terminüberschreitung mit steigendem B verbessert. Bei der Zykluszeit steigt die Lösungsqualität mit der Ausnahme, daß bei N=20 der Wert von $DUB_v^{\mu Cmax}$ bei B=9 besser ist als bei B=10. Die Unterschiede in der Lösungsgüte für die verschiedenen Ausprägungen von B sind nicht so groß wie beim SQ -Maß.

B	N	C_{max}		F_{mid}		T_{max}	
		ΔDUB^μ	ΔDUB^σ	ΔDUB^μ	ΔDUB^σ	ΔDUB^μ	ΔDUB^σ
4	20	1,0036	1,2900	1,0030	1,4425	1,0120	1,4953
	50	1,0037	1,0282	1,0050	1,0462	1,0118	1,1095
	100	1,0026	0,9047	1,0058	0,9115	1,0083	0,9777
	Avg	1,0033	1,1047	1,0046	1,1000	1,0107	1,2325
5	20	1,0020	1,2060	1,0027	1,4168	1,0085	1,3793
	50	1,0024	0,9522	1,0040	1,0921	1,0100	1,1115
	100	1,0019	0,9023	1,0042	0,9374	1,0071	0,9407
	Avg	1,0021	1,0404	1,0036	1,1199	1,0086	1,1788
6	20	1,0024	1,2272	1,0015	1,2381	1,0030	1,1414
	50	1,0023	0,9825	1,0026	1,1127	1,0056	1,0162
	100	1,0014	0,8683	1,0030	0,9099	1,0041	0,9352
	Avg	1,0020	1,0543	1,0024	1,0711	1,0042	1,0474
7	20	1,0010	1,2432	1,0011	1,1885	1,0044	1,2192
	50	1,0009	0,9216	1,0017	1,1438	1,0048	1,0803
	100	1,0008	0,9541	1,0019	0,9324	1,0031	0,9646
	Avg	1,0009	1,0526	1,0016	1,0781	1,0041	1,1090
8	20	1,0007	1,1200	1,0005	1,1237	1,0025	1,2358
	50	1,0009	0,8200	1,0010	1,0889	1,0029	0,9569
	100	1,0008	0,9730	1,0010	0,9745	1,0019	1,0552
	Avg	1,0008	0,9667	1,0009	1,0561	1,0024	1,0901
9	20	1,0002	1,1343	1,0003	1,0879	1,0016	1,1540
	50	1,0006	0,8786	1,0007	1,0476	1,0018	1,0588
	100	1,0003	0,9802	1,0005	1,0099	1,0009	0,9632
	Avg	1,0004	0,9979	1,0005	1,0443	1,0014	1,0747
10	20	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Tabelle 6.29: MCMO2 - Exp. 2: ΔDUB_v^k

Auch bei MCMO2 (vgl. Tabelle 6.29) führt eine Erhöhung von B in der Regel zu einer Verbesserung der Lösungsqualität bzgl. $DUB_v^{\mu k}$. Ausnahmen existieren bei Problemgrößen mit 20 Aufträgen: Bei der Zykluszeit sinkt die Lösungsqualität beim Wechsel von B=5 auf B=6, bei der maximalen Terminüberschreitung läßt sich ein Absinken der Lösungsqualität beim Sprung von B=6 auf B=7 beobachten.

B	N	C_{max}		F_{mid}		T_{max}	
		ΔAEX^μ	ΔAEX^σ	ΔAEX^μ	ΔAEX^σ	ΔAEX^μ	ΔAEX^σ
4	20	0,1564	0,2271	0,4217	0,5980	0,1960	0,2792
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0581	0,0593	0,0000	0,0000	0,0000	0,0000
	Avg	0,1033	0,1219	0,3097	0,2956	0,1234	0,1330
5	20	0,2227	0,3210	0,5361	0,7268	0,2065	0,2502
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,3295	0,3362	0,0000	0,0000	0,0769	0,0769
	Avg	0,1837	0,2209	0,3938	0,3593	0,1438	0,1386
6	20	0,3697	0,3763	0,6265	0,8004	0,6123	0,6913
	50	0,0769	0,0783	0,0714	0,0714	0,0714	0,0714
	100	0,3566	0,3244	0,0000	0,0000	0,1538	0,1538
	Avg	0,2956	0,2707	0,4690	0,4125	0,4267	0,3876
7	20	0,5735	0,6569	0,7229	0,9693	0,6279	0,7514
	50	0,2867	0,2919	0,2143	0,2143	0,4286	0,4286
	100	0,4767	0,4865	0,0625	0,0625	0,3077	0,3077
	Avg	0,4885	0,5069	0,5664	0,5466	0,5326	0,5519
8	20	0,5545	0,6397	0,7952	1,0206	0,7361	0,7976
	50	0,3741	0,3809	0,4286	0,4286	0,2143	0,2143
	100	0,6919	0,7061	0,1875	0,1875	0,3462	0,3462
	Avg	0,5304	0,5694	0,6637	0,6562	0,5664	0,5255
9	20	0,8436	0,9315	0,8976	0,9967	0,7383	0,5818
	50	0,3601	0,3666	0,4286	0,4286	0,4286	0,4286
	100	1,1570	1,0414	0,6250	0,6250	0,4231	0,4231
	Avg	0,7710	0,7713	0,8009	0,7624	0,6226	0,5002
10	20	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Tabelle 6.30: MCMO1 - Exp. 2: ΔAEX_v^k

AEX^k -**Maß:** Bei MCMO1 (vgl. Tabelle 6.30) verbessert sich bei der mittleren Durchlaufzeit die Lösungsqualität bzgl. $AEX_v^{\mu F_{mid}}$ mit steigendem B. Bei der Zykluszeit und der maximalen Terminüberschreitung ist das Steigen der Lösungsqualität nicht grundsätzlich der Fall. Bei der Zykluszeit gibt es Verschlechterungen für Probleme mit 20 Aufträgen beim Sprung von B=7 auf B=8. Für Probleme mit 50 Aufträgen verschlechtert sich die Lösungsqualität beim Wechsel von B=8 auf B=9 und für Probleme mit 100 Aufträgen zwischen B=9 und B=10. Auch bei der maximalen Terminüberschreitung gibt es eine Ausnahme. $AEX_v^{\mu T_{max}}$ verschlechtert sich zwischen B=7 und B=8 für Probleme mit 50 Aufträgen.

B	N	C_{max}		F_{mid}		T_{max}	
		ΔAEX^μ	ΔAEX^σ	ΔAEX^μ	ΔAEX^σ	ΔAEX^μ	ΔAEX^σ
4	20	0,2437	0,3047	0,4433	0,5106	0,1505	0,1831
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Avg	0,1359	0,1343	0,3359	0,2778	0,1027	0,1013
5	20	0,4197	0,4818	0,4639	0,5225	0,3190	0,3830
	50	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Avg	0,2340	0,2124	0,3516	0,2842	0,2175	0,2118
6	20	0,4341	0,4605	0,5979	0,6539	0,5842	0,6412
	50	0,0907	0,0916	0,0667	0,0667	0,0000	0,0000
	100	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	Avg	0,2622	0,2291	0,4609	0,3704	0,3984	0,3546
7	20	0,6724	0,7509	0,7010	0,7926	0,4358	0,4481
	50	0,4828	0,4873	0,2667	0,2667	0,5000	0,5121
	100	0,2709	0,2731	0,1250	0,1250	0,1711	0,1711
	Avg	0,5418	0,5447	0,5781	0,5194	0,3999	0,3937
8	20	0,8538	0,9125	0,8247	0,8865	0,6588	0,7532
	50	0,1134	0,1145	0,3333	0,3333	0,4000	0,4097
	100	0,1505	0,1570	0,2500	0,2500	0,7017	0,7017
	Avg	0,5344	0,4779	0,6953	0,6146	0,6282	0,6708
9	20	0,9179	0,9441	0,8866	0,9772	0,7563	0,7799
	50	0,4451	0,4174	0,5333	0,5333	1,1000	1,1266
	100	0,6313	0,6506	0,5000	0,5000	0,6944	0,6944
	Avg	0,7498	0,7135	0,7969	0,7669	0,7961	0,8296
10	20	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	50	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	100	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000
	Avg	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000

Tabelle 6.31: MCMO2 - Exp. 2: ΔAEX_v^k

Eine Analyse der Ergebnisse des AEX^k -Maßes von MCMO2 (vgl. Tabelle 6.31) zeigt, daß sich auch hier für die mittlere Durchlaufzeit die Lösungsqualität bzgl. $AEX_v^{\mu Fmid}$ mit steigendem B verbessert. Bei der Zykluszeit verschlechtert sich die Lösungsqualität zwischen B=7 und B=8 für Probleme mit 50 und 100 Aufträgen. Bei der maximalen Terminüberschreitung gibt es ebenfalls Ausnahmen. Für N=20 tritt eine Verschlechterung der Lösungsqualität zwischen B=6 und B=7 ein. Für N=50 verschlechtert sich die Lösungsqualität zwischen B=7 und B=8 sowie beim Wechsel von B=9 auf B=10. Für N=100 tritt eine Verschlechterung der Lösungsqualität beim Sprung von B=8 auf B=9 ein.

Insgesamt zeigt sich sowohl für MCMO1 als auch für MCMO2, daß mit einer feineren Einteilung des Zielraums in Suchregionen die Lösungsqualität trotz einiger Unregelmäßigkeiten tendenziell gesteigert werden kann,

wobei jedoch die Rechenzeit ansteigt.

6.4.3.3 Experiment 3: Variation der Parameter *Iter* und *Ants*

Im folgenden Experiment soll für beide Varianten des MCMO-Ansatzes der Einfluß einer Variation der Parameter *Iter* und *Ants* auf die Lösungsgüte und Rechenzeit untersucht werden. Die getesteten Varianten sind wie beim entsprechenden Experiment beim SCMO-Ansatz aufgebaut (vgl. Abschnitt 6.3.2.6, insbesondere Tabelle 6.14 auf Seite 202). Die Ergebnisse sind in den Tabellen 6.32 bis 6.36 wiedergegeben.

Variante	N	MCMO1		MCMO2	
		SQ^μ	SQ^σ	SQ^μ	SQ^σ
1	20	0,7985	0,0511	0,7596	0,0490
	50	0,6678	0,0541	0,6323	0,0652
	100	0,5793	0,0574	0,5733	0,0575
	Avg	0,6819	0,0542	0,6551	0,0572
2	20	0,8642	0,0411	0,8337	0,0436
	50	0,7772	0,0530	0,7572	0,0659
	100	0,7061	0,0651	0,7633	0,0705
	Avg	0,7825	0,0531	0,7847	0,0600
3	20	0,8667	0,0380	0,8339	0,0431
	50	0,7377	0,0501	0,6971	0,0631
	100	0,6086	0,0574	0,5684	0,0538
	Avg	0,7377	0,0485	0,6998	0,0534
4	20	0,8604	0,0383	0,8179	0,0433
	50	0,7600	0,0552	0,7123	0,0615
	100	0,6624	0,0578	0,6582	0,0613
	Avg	0,7609	0,0505	0,7295	0,0554
5	20	0,8415	0,0396	0,8232	0,0450
	50	0,7635	0,0547	0,7027	0,0596
	100	0,7459	0,0681	0,6114	0,0660
	Avg	0,7837	0,0541	0,7125	0,0569

Tabelle 6.32: MCMO - Exp. 3: SQ_v

SQ -Maß: Tabelle 6.32 zeigt zunächst, daß sowohl bei MCMO1 als auch bei MCMO2 die Varianten 3, 4 und 5 bis auf eine Ausnahme bessere Werte für SQ_v^μ haben als Variante 1. Bei MCMO2 ist für $N=100$ SQ_1^μ besser als SQ_3^μ .

Werden bei MCMO1 die Varianten 3, 4 und 5 alleine verglichen, so zeigt sich, daß für unterschiedliche Problemgrößen unterschiedliche Varianten vorteilhaft sind. Wird Variante 2 hinzugezogen, so ist diese bei Problemen mit 50 Aufträgen die beste der vier Varianten.

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0082	0,0031	1,0043	0,0026	1,0176	0,0095
	50	1,0181	0,0025	1,0107	0,0029	1,0195	0,0060
	100	1,0145	0,0011	1,0158	0,0026	1,0128	0,0034
	Avg	1,0136	0,0022	1,0102	0,0027	1,0166	0,0063
2	20	1,0053	0,0028	1,0024	0,0018	1,0113	0,0067
	50	1,0147	0,0021	1,0064	0,0030	1,0116	0,0057
	100	1,0132	0,0011	1,0112	0,0028	1,0078	0,0033
	Avg	1,0110	0,0020	1,0067	0,0025	1,0102	0,0052
3	20	1,0057	0,0028	1,0023	0,0018	1,0113	0,0067
	50	1,0157	0,0023	1,0074	0,0027	1,0152	0,0055
	100	1,0148	0,0009	1,0133	0,0024	1,0116	0,0033
	Avg	1,0120	0,0020	1,0077	0,0023	1,0127	0,0052
4	20	1,0058	0,0030	1,0026	0,0019	1,0109	0,0070
	50	1,0154	0,0022	1,0073	0,0028	1,0139	0,0057
	100	1,0136	0,0012	1,0129	0,0025	1,0096	0,0035
	Avg	1,0116	0,0021	1,0076	0,0024	1,0115	0,0054
5	20	1,0061	0,0027	1,0030	0,0022	1,0128	0,0071
	50	1,0156	0,0026	1,0077	0,0029	1,0128	0,0056
	100	1,0129	0,0013	1,0127	0,0026	1,0081	0,0032
	Avg	1,0115	0,0022	1,0078	0,0025	1,0112	0,0053

Tabelle 6.33: MCMO1 - Exp. 3: DUB^k -Maß

Bei MCMO2 stellt man bei einem isolierten Vergleich der Alternativen 3, 4 und 5 fest, daß Variante 5 für keine Problemgröße am besten ist. Bei $N=20$ ist Variante 3, ansonsten Variante 4 die beste. Wird Variante 2 mit in den Vergleich aufgenommen, so ist diese bei Problemen mit 50 und 100 Aufträgen die vorteilhafteste der vier Varianten bzgl. SQ_v^μ .

DUB^k -Maß: Beim DUB^k -Maß (Tabellen 6.33 und 6.34) sind bei MCMO1, mit Ausnahme von $DUB_3^{\mu C_{max}}$ bei $N=100$, die Varianten 3, 4 und 5 bzgl. $DUB_v^{\mu k}$ besser als Variante 1. Ein isolierter Vergleich der Varianten 3, 4 und 5 zeigt, daß keine der Varianten eindeutig vorteilhaft ist. Je nach betrachteter Ziel- und Problemgröße ist eine andere Variante besser. Wird Variante 2 hinzugezogen, so ist diese, bis auf drei Ausnahmen, die beste Variante. Für die Zykluszeit und $N=100$ ist $DUB_5^{\mu C_{max}}$ besser. Des Weiteren ist für die mittlere Durchlaufzeit und $N=20$ Variante 3 sowie bei der maximalen Terminüberschreitung und $N=20$ Variante 4 besser.

Variante	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
1	20	1,0075	0,0030	1,0053	0,0028	1,0182	0,0087
	50	1,0156	0,0025	1,0101	0,0031	1,0204	0,0066
	100	1,0130	0,0012	1,0139	0,0026	1,0127	0,0034
	Avg	1,0121	0,0022	1,0098	0,0028	1,0171	0,0063
2	20	1,0057	0,0028	1,0029	0,0022	1,0111	0,0073
	50	1,0138	0,0024	1,0068	0,0030	1,0121	0,0063
	100	1,0115	0,0013	1,0102	0,0028	1,0064	0,0035
	Avg	1,0103	0,0022	1,0066	0,0027	1,0099	0,0057
3	20	1,0053	0,0025	1,0028	0,0020	1,0110	0,0068
	50	1,0142	0,0026	1,0070	0,0027	1,0163	0,0059
	100	1,0139	0,0011	1,0127	0,0027	1,0136	0,0032
	Avg	1,0111	0,0021	1,0075	0,0025	1,0136	0,0053
4	20	1,0058	0,0027	1,0031	0,0020	1,0119	0,0071
	50	1,0139	0,0024	1,0074	0,0028	1,0157	0,0062
	100	1,0121	0,0013	1,0111	0,0028	1,0095	0,0031
	Avg	1,0106	0,0021	1,0072	0,0025	1,0124	0,0055
5	20	1,0053	0,0025	1,0029	0,0023	1,0116	0,0078
	50	1,0142	0,0025	1,0073	0,0029	1,0155	0,0054
	100	1,0131	0,0013	1,0116	0,0027	1,0114	0,0032
	Avg	1,0109	0,0021	1,0073	0,0027	1,0128	0,0055

Tabelle 6.34: MCMO2 - Exp. 3: DUB^k -Maß

Werden die Ergebnisse für MCMO2 ausgewertet, so zeigt sich, daß nur bei der mittleren Durchlaufzeit die Varianten 3, 4 und 5 grundsätzlich besser sind als Variante 1. Bei der maximalen Terminüberschreitung ist Variante 3 für $N=100$ schlechter als Variante 1. Bei der Zykluszeit sind bei Problemgrößen mit 100 Aufträgen die Varianten 3 und 5 schlechter als Variante 1. Wie bereits bei MCMO1 beobachtet, ist bei einem isolierten Vergleich der drei Varianten 3, 4 und 5 keine Variante eindeutig vorteilhaft. Wird erneut Variante 2 in den Vergleich mit aufgenommen, so ist diese außer bei $N=20$ für alle drei Zielgrößen die beste Variante. Bei 20 Aufträgen ist bei allen drei Zielgrößen Variante 3 besser. Bei der Zykluszeit zusätzlich Variante 5, welche die gleiche Lösungsqualität besitzt wie Variante 3.

Variante	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0244	0,0553	0,1567	0,2215	0,0149	0,0448
	50	0,0003	0,0010	0,0000	0,0000	0,0033	0,0100
	100	0,0050	0,0150	0,0000	0,0000	0,0033	0,0100
	Avg	0,0099	0,0238	0,0522	0,0738	0,0072	0,0216
2	20	0,0894	0,1856	0,2600	0,2999	0,0930	0,1563
	50	0,0487	0,1300	0,0400	0,1200	0,0572	0,1650
	100	0,0241	0,0601	0,0667	0,2000	0,0467	0,1400
	Avg	0,0541	0,1252	0,1222	0,2066	0,0656	0,1538
3	20	0,0628	0,1398	0,2767	0,2824	0,0947	0,1980
	50	0,0150	0,0450	0,0100	0,0300	0,0067	0,0200
	100	0,0016	0,0048	0,0067	0,0200	0,0067	0,0200
	Avg	0,0265	0,0632	0,0978	0,1108	0,0360	0,0793
4	20	0,0489	0,0943	0,2700	0,2925	0,0671	0,1458
	50	0,0219	0,0657	0,0300	0,0900	0,0200	0,0600
	100	0,0310	0,0907	0,0133	0,0400	0,0200	0,0600
	Avg	0,0339	0,0836	0,1044	0,1408	0,0357	0,0886
5	20	0,0489	0,1005	0,2267	0,2625	0,0878	0,1707
	50	0,0174	0,0507	0,0200	0,0600	0,0161	0,0483
	100	0,0384	0,1151	0,0133	0,0400	0,0233	0,0700
	Avg	0,0349	0,0888	0,0867	0,1208	0,0424	0,0963

Tabelle 6.35: MCMO1 - Exp. 3: AEX^k -Maß

AEX^k -Maß: Beim AEX^k -Maß (Tabellen 6.35 und 6.36) sind bei MCMO1 die Varianten 3, 4 und 5, außer $AEX_3^{\mu C_{max}}$ bei N=100, bzgl. $AEX_v^{\mu k}$ besser als Variante 1. Ein isolierter Vergleich der Varianten 3, 4 und 5 zeigt, daß keine der Varianten eindeutig vorteilhaft ist. Wird Variante 2 in den Vergleich mit aufgenommen, so ist diese die überwiegend beste Variante. Bei N=20 und den Zielgrößen der mittleren Durchlaufzeit und der maximalen Terminüberschreitung ist Variante 3 besser. Ebenso ist bei N=100 $AEX_5^{\mu C_{max}}$ besser als $AEX_2^{\mu C_{max}}$.

Werden die Ergebnisse von MCMO2 ausgewertet, so sind die Varianten 3, 4 und 5 nicht schlechter als Variante 1. Außer bei N=100 und der maximalen Terminüberschreitung, wo Variante 3 die gleiche Lösungsqualität besitzt, sind sie sogar besser als Variante 1. Erneut erweist sich bei isolierter Betrachtung der Varianten 3, 4 und 5 keine als eindeutig vorteilhaft. Bei Hinzunahme von Variante 2 ist diese, bis auf zwei Ausnahmen, die Variante mit den besten Werten für $AEX_v^{\mu k}$. Bei N=20 und der Zielgröße der Zykluszeit ist Variante 3 und bei der Zielgröße der mittleren Durchlaufzeit Variante 5 besser.

Variante	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
1	20	0,0344	0,0616	0,1133	0,1565	0,0228	0,0617
	50	0,0011	0,0033	0,0000	0,0000	0,0033	0,0100
	100	0,0010	0,0029	0,0000	0,0000	0,0000	0,0000
	Avg	0,0122	0,0226	0,0378	0,0522	0,0087	0,0239
2	20	0,0628	0,1095	0,2300	0,3017	0,0828	0,1681
	50	0,0403	0,1063	0,0333	0,1000	0,0592	0,1775
	100	0,0752	0,2063	0,0567	0,1700	0,0792	0,2375
	Avg	0,0594	0,1407	0,1067	0,1906	0,0737	0,1944
3	20	0,0933	0,2041	0,2433	0,2757	0,0739	0,1517
	50	0,0163	0,0488	0,0133	0,0400	0,0067	0,0200
	100	0,0033	0,0100	0,0033	0,0100	0,0000	0,0000
	Avg	0,0376	0,0876	0,0867	0,1086	0,0269	0,0572
4	20	0,0928	0,1833	0,2200	0,2374	0,0522	0,1049
	50	0,0139	0,0417	0,0300	0,0900	0,0208	0,0625
	100	0,0159	0,0455	0,0300	0,0900	0,0200	0,0600
	Avg	0,0409	0,0901	0,0933	0,1391	0,0310	0,0758
5	20	0,0750	0,1229	0,2867	0,3115	0,0710	0,1814
	50	0,0285	0,0854	0,0233	0,0700	0,0100	0,0300
	100	0,0046	0,0138	0,0100	0,0300	0,0008	0,0025
	Avg	0,0360	0,0740	0,1067	0,1372	0,0273	0,0713

Tabelle 6.36: MCMO2 - Exp. 3: AEX^k -Maß

Insgesamt zeigt sich in diesem Experiment, daß Variante 2 eine gute Variante ist. Sowohl bei MCMO1 als auch bei MCMO2 ist sie von den Varianten 2, 3, 4 und 5 im Durchschnitt über alle Probleminstanzen in der Regel die beste Variante. Bei MCMO1 ist sie bzgl. SQ^μ die zweitbeste Variante nach Variante 5. Bei MCMO2 ist sie bzgl. $AEX^{\mu F_{mid}}$ ebenfalls die zweitbeste Variante nach Variante 5. Soll die Anzahl der untersuchten und ausgewerteten Auftragsfolgen erhöht werden, so ist die alleinige Erhöhung des Parameters B und damit eine feinere Unterteilung des Zielraums in Suchregionen die vorzuziehende Strategie.

6.4.3.4 Experiment 4: Variation des Parameters ρ

In diesem Experiment soll die Veränderung der Lösungsgüte aufgrund einer Veränderung der Trail-Persistence-Rate analysiert werden. Als mögliche Werte für ρ werden dabei untersucht: 0,55, 0,65, 0,75, 0,85 sowie 0,95. Die Ergebnisse sind in den Tabellen 6.37 bis 6.42 angegeben.

ρ	N	MCMO1		MCMO2	
		SQ^μ	SQ^σ	SQ^μ	SQ^σ
0,55	20	0,8330	0,0427	0,7892	0,0487
	50	0,7607	0,0632	0,6836	0,0725
	100	0,7413	0,0691	0,7192	0,0676
	Avg	0,7783	0,0583	0,7306	0,0630
0,65	20	0,8319	0,0436	0,7999	0,0477
	50	0,7633	0,0641	0,6999	0,0698
	100	0,7048	0,0647	0,7160	0,0674
	Avg	0,7667	0,0575	0,7386	0,0616
0,75	20	0,8420	0,0438	0,8146	0,0489
	50	0,7531	0,0589	0,7128	0,0670
	100	0,6615	0,0610	0,6855	0,0685
	Avg	0,7522	0,0546	0,7376	0,0615
0,85	20	0,8471	0,0411	0,8328	0,0455
	50	0,7312	0,0538	0,7033	0,0664
	100	0,6247	0,0621	0,5960	0,0590
	Avg	0,7344	0,0523	0,7107	0,0570
0,95	20	0,8482	0,0411	0,8343	0,0446
	50	0,7124	0,0516	0,6791	0,0609
	100	0,6064	0,0571	0,5545	0,0560
	Avg	0,7223	0,0500	0,6893	0,0538

Tabelle 6.37: MCMO - Exp. 4: SQ -Maß

SQ -Maß: Die Auswertung des SQ -Maßes zeigt, daß es sowohl bei MCMO1 als auch für MCMO2 vorteilhaft ist, für steigende Problemgrößen einen kleineren Wert für ρ und damit eine höhere Verdunstungsrate zu verwenden (vgl. Tabelle 6.37). Bei MCMO1 ist für N=20 ein Wert von 0,95, für N=50 ein Wert von 0,65 und für Probleme mit 100 Aufträgen ein Wert von 0,55 vorteilhaft. Bei MCMO2 ist für N=50 ein Wert von 0,75 vorteilhaft. Für N=20 und N=100 sollten die Werte wie bei MCMO1 gewählt werden.

ρ	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
0,55	20	1,0064	0,0031	1,0034	0,0023	1,0131	0,0083
	50	1,0158	0,0024	1,0077	0,0030	1,0133	0,0061
	100	1,0130	0,0012	1,0126	0,0028	1,0078	0,0034
	Avg	1,0117	0,0023	1,0079	0,0027	1,0114	0,0059
0,65	20	1,0067	0,0033	1,0033	0,0023	1,0120	0,0079
	50	1,0155	0,0022	1,0075	0,0030	1,0130	0,0057
	100	1,0133	0,0010	1,0128	0,0028	1,0084	0,0033
	Avg	1,0118	0,0022	1,0079	0,0027	1,0111	0,0056
0,75	20	1,0064	0,0030	1,0032	0,0021	1,0125	0,0084
	50	1,0155	0,0025	1,0081	0,0030	1,0127	0,0062
	100	1,0138	0,0011	1,0130	0,0027	1,0088	0,0036
	Avg	1,0119	0,0022	1,0081	0,0026	1,0113	0,0061
0,85	20	1,0062	0,0032	1,0029	0,0021	1,0116	0,0076
	50	1,0166	0,0026	1,0076	0,0030	1,0133	0,0059
	100	1,0145	0,0011	1,0127	0,0027	1,0105	0,0036
	Avg	1,0124	0,0023	1,0077	0,0026	1,0118	0,0057
0,95	20	1,0062	0,0028	1,0027	0,0019	1,0125	0,0077
	50	1,0165	0,0028	1,0085	0,0029	1,0144	0,0053
	100	1,0149	0,0010	1,0138	0,0027	1,0115	0,0034
	Avg	1,0125	0,0022	1,0083	0,0025	1,0128	0,0055

Tabelle 6.38: MCMO1 - Exp. 4: DUB^k -Maß

ρ	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
0,55	20	1,0067	0,0030	1,0043	0,0026	1,0169	0,0098
	50	1,0150	0,0024	1,0098	0,0032	1,0156	0,0066
	100	1,0113	0,0013	1,0131	0,0031	1,0081	0,0037
	Avg	1,0110	0,0022	1,0091	0,0030	1,0135	0,0067
0,65	20	1,0069	0,0032	1,0037	0,0023	1,0153	0,0090
	50	1,0146	0,0026	1,0083	0,0031	1,0153	0,0064
	100	1,0115	0,0013	1,0123	0,0031	1,0079	0,0036
	Avg	1,0110	0,0024	1,0081	0,0028	1,0129	0,0063
0,75	20	1,0065	0,0031	1,0035	0,0024	1,0143	0,0077
	50	1,0140	0,0025	1,0077	0,0033	1,0140	0,0064
	100	1,0120	0,0012	1,0114	0,0026	1,0091	0,0035
	Avg	1,0108	0,0023	1,0076	0,0027	1,0125	0,0059
0,85	20	1,0058	0,0028	1,0030	0,0021	1,0126	0,0085
	50	1,0147	0,0026	1,0074	0,0031	1,0148	0,0062
	100	1,0137	0,0013	1,0122	0,0026	1,0123	0,0034
	Avg	1,0114	0,0022	1,0075	0,0026	1,0132	0,0060
0,95	20	1,0058	0,0024	1,0028	0,0020	1,0117	0,0084
	50	1,0154	0,0025	1,0078	0,0030	1,0161	0,0063
	100	1,0144	0,0011	1,0139	0,0028	1,0145	0,0036
	Avg	1,0119	0,0020	1,0081	0,0026	1,0141	0,0061

Tabelle 6.39: MCMO2 - Exp. 4: DUB^k -Maß

DUB^k -Maß: Beim DUB^k -Maß ist es sowohl bei MCMO1 als auch bei MCMO2 bei allen drei Zielgrößen vorteilhaft, für steigende Problemgrößen einen kleineren Wert für ρ und damit eine größere Verdunstungsrate zu verwenden (vgl. Tabelle 6.38 und 6.39). Die folgende Tabelle zeigt, welche Werte von ρ für die unterschiedlichen Problemgrößen vorteilhaft sind.

N	MCMO1			MCMO2		
	C_{max}	F_{mid}	T_{max}	C_{max}	F_{mid}	T_{max}
20	0,95	0,95	0,85	0,95	0,95	0,95
50	0,65	0,65	0,75	0,75	0,85	0,75
100	0,55	0,55	0,55	0,55	0,75	0,65

Tabelle 6.40: Wahl von ρ in Abhängigkeit von der Problemgröße

AEX^k -Maß: Die Ergebnisse des AEX^k -Maßes von MCMO1 (vgl. Tabelle 6.41) zeigen, daß es auch hier vorteilhaft ist, für steigende Problemgrößen kleinere Werte für ρ zu verwenden. Im Fall der Zykluszeit ist bei N=20 und N=50 ein Wert von 0,75 und für N=100 ein Wert von 0,55 vorteilhaft. Bei der mittleren Durchlaufzeit sollte bei Problemen mit 20 Aufträgen ρ auf den Wert 0,95 gesetzt werden. Für die Problemgrößen mit 50 Aufträgen ist ein Wert von 0,85 und für Probleme mit 100 Aufträgen ein Wert von 0,75 eine gute Wahl. Bei der maximalen Terminüberschreitung ist ein Wert von 0,85 bei N=20 vorteilhaft. Bei Problemen mit 50 Aufträgen sollte ρ auf 0,75 gesetzt werden. Bei 100 Aufträgen ist einer der Werte 0,55 oder 0,65 zu wählen.

Bei MCMO2 (vgl. Tabelle 6.42) ist die Beobachtung, daß es vorteilhaft ist, für steigende Problemgrößen kleinere Werte für ρ zu verwenden, nur bei der Zykluszeit zu machen. Sowohl bei der mittleren Durchlaufzeit als auch bei der maximalen Terminüberschreitung haben die Probleme mit 20 Aufträgen von allen untersuchten Problemgrößen weiterhin den größten Wert für ρ . Die Probleme mit 50 Aufträgen haben aber den kleinsten Wert für ρ .

ρ	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
0,55	20	0,0667	0,1260	0,1833	0,2595	0,0687	0,1574
	50	0,0262	0,0750	0,0233	0,0700	0,0172	0,0517
	100	0,0573	0,1646	0,0233	0,0700	0,0267	0,0800
	Avg	0,0501	0,1218	0,0767	0,1332	0,0375	0,0963
0,65	20	0,0450	0,1081	0,2033	0,2585	0,0908	0,1836
	50	0,0102	0,0286	0,0200	0,0600	0,0133	0,0400
	100	0,0250	0,0731	0,0167	0,0500	0,0267	0,0800
	Avg	0,0267	0,0699	0,0800	0,1228	0,0436	0,1012
0,75	20	0,0683	0,1284	0,1967	0,2743	0,0717	0,1780
	50	0,0348	0,1015	0,0067	0,0200	0,0400	0,1171
	100	0,0078	0,0233	0,0267	0,0800	0,0200	0,0600
	Avg	0,0370	0,0844	0,0767	0,1248	0,0439	0,1184
0,85	20	0,0750	0,1456	0,2117	0,2607	0,1011	0,2132
	50	0,0178	0,0533	0,0367	0,1100	0,0211	0,0633
	100	0,0078	0,0233	0,0233	0,0700	0,0167	0,0500
	Avg	0,0335	0,0740	0,0906	0,1469	0,0463	0,1088
0,95	20	0,0650	0,1183	0,2117	0,2554	0,0919	0,1938
	50	0,0113	0,0340	0,0133	0,0400	0,0083	0,0250
	100	0,0054	0,0162	0,0100	0,0300	0,0100	0,0300
	Avg	0,0272	0,0562	0,0783	0,1085	0,0367	0,0829

Tabelle 6.41: MCMO1 - Exp. 4: AEX^k -Maß

ρ	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
0,55	20	0,0728	0,1540	0,1750	0,2093	0,0744	0,1242
	50	0,0101	0,0303	0,0067	0,0200	0,0300	0,0867
	100	0,0403	0,1210	0,0200	0,0600	0,0333	0,1000
	Avg	0,0411	0,1018	0,0672	0,0964	0,0459	0,1036
0,65	20	0,0533	0,1129	0,1883	0,2026	0,0942	0,1587
	50	0,0226	0,0639	0,0300	0,0900	0,0167	0,0500
	100	0,0432	0,1261	0,0167	0,0500	0,0433	0,1300
	Avg	0,0397	0,1010	0,0783	0,1142	0,0514	0,1129
0,75	20	0,0483	0,0955	0,2317	0,2818	0,0556	0,0883
	50	0,0368	0,1105	0,0233	0,0700	0,0289	0,0867
	100	0,0158	0,0473	0,0333	0,1000	0,0200	0,0600
	Avg	0,0336	0,0844	0,0961	0,1506	0,0348	0,0783
0,85	20	0,0792	0,1533	0,2850	0,2847	0,1191	0,1974
	50	0,0204	0,0584	0,0267	0,0800	0,0100	0,0300
	100	0,0007	0,0020	0,0200	0,0600	0,0033	0,0100
	Avg	0,0334	0,0713	0,1106	0,1416	0,0441	0,0791
0,95	20	0,0703	0,1309	0,2550	0,3018	0,1289	0,2158
	50	0,0100	0,0300	0,0167	0,0500	0,0144	0,0433
	100	0,0000	0,0000	0,0100	0,0300	0,0000	0,0000
	Avg	0,0268	0,0536	0,0939	0,1273	0,0478	0,0864

Tabelle 6.42: MCMO2 - Exp. 4: AEX^k -Maß

Insgesamt kann unter Berücksichtigung aller Auswertungsgrößen sowohl bei MCMO1 als auch MCMO2 die Verwendung kleinerer Werte für ρ für größere Probleme als vorteilhaft beurteilt werden. Ein Vergleich der guten Werte für ρ bei unterschiedlichen Problemgrößen zwischen MCMO1 und MCMO2 zeigt, daß die Werte bei MCMO1 tendenziell kleiner zu wählen sind als bei MCMO2. Die Verwendung einer Kommunikation während der Lösungskonstruktion verlangt damit tendenziell eine höhere Verdunstungsrate. Für MCMO1 ist $\rho=0,55$ eine gute Einstellung. Für große Probleme ist sie in der Regel die beste Einstellung. Für die restlichen Problemgrößen sind die Unterschiede zu den besten Werten gering. Für MCMO2 kann $\rho=0,65$ als gute Einstellung angesehen werden.

6.5 Vergleich mit einem Referenzverfahren

6.5.1 Grundsätzliche Überlegungen

In den letzten beiden Abschnitten sind mehrere Ansätze von Ameisenalgorithmen zur Lösung des multikriteriellen PFSP isoliert voneinander vorgestellt worden. Die experimentellen Analysen haben sich dabei immer nur auf einen einzelnen Ansatz bzw. eine seiner Varianten bezogen. In diesem Abschnitt sollen der SCMO-Ansatz und die zwei Varianten des MCMO-Ansatzes miteinander verglichen werden. Die Ansätze werden, wie in Tabelle 6.43 angegebenen, konfiguriert.

Parameter	SCMO	MCMO1	MCMO2a	MCMO2b
<i>B</i>	7	7	7	6
<i>Ants</i>	1	1	1	1
<i>Iter</i>	10	10	10	10
ρ	0,65	0,55	0,65	0,65

Tabelle 6.43: Konfigurationen von SCMO, MCMO1 und MCMO2

Der Vergleich soll unter der Bedingung (annähernd) gleicher Rechenzeit durchgeführt werden. Dies wird durch das einheitliche Festlegen der Parameter *B*, *Ants* und *Iter* für SCMO, MCMO1 und MCMO2a erreicht. Leichte Schwankungen in der Rechenzeit sind allein durch unterschiedliche Laufzeiten der lokalen Suche begründet. Die Anzahl der von den Ameisen konstruierten Auftragsfolgen ist in den drei Ansätzen gleich. MCMO2b hat eine geringere Rechenzeit, da ein kleinerer Wert von *B* die Anzahl der von den Ameisen konstruierten Auftragsfolgen verringert und damit auch die Rechenzeit. Die Variante wird dennoch aufgrund des Ausreißerwertes beim *SQ*-Maß in Experiment 3 im Vergleich berücksichtigt.

Außerdem soll ein Vergleich der entwickelten Ameisenalgorithmen mit dem MOGLS-Ansatz von Ishibuchi u. Murata (1998) vorgenommen werden.¹²¹ Damit ein Vergleich bei (annähernd) gleicher Rechenzeit ermöglicht wird, muß MOGLS bezüglich seiner Abbruchbedingung modifiziert werden. In der Arbeit von Ishibuchi u. Murata (1998) wird eine Anzahl von Iterationen als Abbruchbedingung verwendet.¹²² Diese Bedingung ist dahingehend zu modifizieren, daß die Heuristik nach Ablauf einer vorgegebenen Rechenzeit abbricht. Auf den Vergleich mit dem GPWGA-Ansatz von Chang u. a. (2002) wird verzichtet.¹²³ Als Abbruchbedingung wird dort eine Anzahl ge-

¹²¹Vgl. Ishibuchi u. Murata (1998, S.392-398).

¹²²Vgl. Ishibuchi u. Murata (1998, S.397).

¹²³Vgl. Chang u. a. (2002, S.3-8).

nerierter Populationen verwendet.¹²⁴ Aus dieser Vorgabe werden weitere Parameter, insbesondere die Variation der Gewichte, berechnet. Bei Vorgabe einer erlaubten Rechenzeit ist die von Chang u. a. (2002) vorgesehene Berechnung der Gewichte nicht mehr möglich.¹²⁵

6.5.2 Experimentelle Analyse

6.5.2.1 Vergleich der Lösungsqualität von SCMO, MCMO und MOGLS

In den Tabellen 6.44 bis 6.46 sind die Ergebnisse des Vergleichs zwischen dem SCMO-Ansatz, den Varianten MCMO1, MCMO2a und MCMO2b des MCMO-Ansatzes und MOGLS angegeben.¹²⁶

Heuristik	N	SQ^μ	SQ^σ
SCMO	20	0,8402	0,0444
	50	0,7348	0,0566
	100	0,6377	0,0576
	Avg	0,7376	0,0529
MCMO1	20	0,8509	0,0412
	50	0,7390	0,0632
	100	0,6326	0,0660
	Avg	0,7408	0,0568
MCMO2a	20	0,7808	0,0464
	50	0,6511	0,0650
	100	0,5872	0,0603
	Avg	0,6730	0,0572
MCMO2b	20	0,8292	0,0441
	50	0,7339	0,0615
	100	0,6490	0,0577
	Avg	0,7374	0,0545
MOGLS	20	0,4360	0,0627
	50	0,3872	0,0836
	100	0,4524	0,0927
	Avg	0,4252	0,0796

Tabelle 6.44: Vergleich SCMO, MCMO und MOGLS: SQ -Maß

SQ -Maß: Aus Experiment 1 des MCMO-Ansatzes (vgl. Abschnitt 6.4.3.1 ab Seite 230) ist bekannt, daß MCMO2a bezüglich SQ_v^μ für alle Problemgrößen schlechter ist als MCMO1. Obwohl MCMO2b bezüglich des SQ -Maßes

¹²⁴Vgl. Vgl. Chang u. a. (2002, S.8).

¹²⁵Vgl. Chang u. a. (2002, S.6-8).

¹²⁶Zur Konfiguration von SCMO, MCMO1 und MCMO2 siehe Tabelle 6.43 auf der vorherigen Seite. MOGLS wurde, wie in Tabelle C.1 auf Seite 291 in Anhang C beschrieben, konfiguriert.

in Experiment 2 (vgl. Abschnitt 6.4.3.2 ab Seite 235) einen Ausreißer darstellte und fast die Lösungsqualität der Variante mit $B=10$ erreichte, ist er nur bei Problemen mit 100 Aufträgen besser als MCMO1.

Sollen SCMO, MCMO1, MCMO2a und MCMO2b bezüglich der in Tabelle 6.44 angegebenen Werte von SQ_v^μ in eine Rangfolge gebracht werden, so ergibt sich für Probleme mit 20 und 50 Aufträgen: $\text{MCMO1} \succ \text{SCMO} \succ \text{MCMO2b} \succ \text{MCMO2a}$. Bei Problemen mit 100 Aufträgen ist MCMO2b am besten und SCMO besser als MCMO1, so daß gilt: $\text{MCMO2b} \succ \text{SCMO} \succ \text{MCMO1} \succ \text{MCMO2a}$.

Die MOGLS-Heuristik ist deutlich schlechter als alle in den Vergleich aufgenommenen Ameisenalgorithmen, wie aus Tabelle 6.44 deutlich zu erkennen ist.

Heuristik	N	C_{max}		F_{mid}		T_{max}	
		DUB^μ	DUB^σ	DUB^μ	DUB^σ	DUB^μ	DUB^σ
SCMO	20	1,0071	0,0032	1,0034	0,0023	1,0118	0,0084
	50	1,0147	0,0023	1,0082	0,0033	1,0113	0,0063
	100	1,0125	0,0013	1,0119	0,0028	1,0099	0,0042
	Avg	1,0114	0,0022	1,0078	0,0028	1,0110	0,0063
MCMO1	20	1,0064	0,0031	1,0033	0,0023	1,0119	0,0083
	50	1,0158	0,0024	1,0077	0,0030	1,0165	0,0061
	100	1,0130	0,0012	1,0128	0,0028	1,0136	0,0034
	Avg	1,0117	0,0023	1,0079	0,0027	1,0140	0,0059
MCMO2a	20	1,0069	0,0032	1,0038	0,0023	1,0153	0,0090
	50	1,0146	0,0026	1,0082	0,0031	1,0166	0,0064
	100	1,0115	0,0013	1,0122	0,0031	1,0108	0,0036
	Avg	1,0110	0,0024	1,0081	0,0028	1,0142	0,0063
MCMO2b	20	1,0083	0,0033	1,0044	0,0025	1,0136	0,0083
	50	1,0156	0,0026	1,0090	0,0031	1,0171	0,0061
	100	1,0121	0,0013	1,0131	0,0030	1,0116	0,0033
	Avg	1,0120	0,0024	1,0088	0,0029	1,0141	0,0059
MOGLS	20	1,0525	0,0050	1,0263	0,0074	1,0700	0,0153
	50	1,0434	0,0046	1,0404	0,0077	1,0372	0,0092
	100	1,0296	0,0028	1,0423	0,0067	1,0130	0,0050
	Avg	1,0419	0,0041	1,0363	0,0073	1,0401	0,0098

Tabelle 6.45: Vergleich SCMO, MCMO und MOGLS: DUB^k -Maß

DUB^k -Maß: Die Rangfolgen der Heuristiken bzgl. $DUB_v^{\mu k}$ (vgl. Tabelle 6.45) verändert sich in Abhängigkeit von der Ziel- und Problemgröße. Werden alleine die Durchschnittswerte über alle Probleminstanzen betrachtet, so ergibt sich, daß SCMO bei den Zielgrößen der mittleren Durchlaufzeit und der maximalen Terminüberschreitung besser ist als die MCMO-Ansätze. Bei der Zykluszeit ist SCMO die zweitbeste Heuristik hinter MC-

MO2a. Auffällig ist, daß bei der maximalen Terminüberschreitung MC-MO2b Rang drei hinter SCMO und MCMO1 belegt.

Der Vergleich der MOGLS-Heuristik mit den Ameisenalgorithmen zeigt, daß sie für alle drei Zielgrößen deutlich schlechter als die Ameisenalgorithmen ist.

Heuristik	N	C_{max}		F_{mid}		T_{max}	
		AEX^μ	AEX^σ	AEX^μ	AEX^σ	AEX^μ	AEX^σ
SCMO	20	0,0833	0,1746	0,2117	0,2659	0,1383	0,2540
	50	0,0176	0,0484	0,0400	0,1200	0,0592	0,1746
	100	0,0142	0,0375	0,0367	0,1100	0,0300	0,0900
	Avg	0,0384	0,0868	0,0961	0,1653	0,0758	0,1729
MCMO1	20	0,0978	0,1801	0,2133	0,2695	0,1037	0,1860
	50	0,0150	0,0450	0,0333	0,1000	0,0167	0,0500
	100	0,0067	0,0161	0,0167	0,0500	0,0000	0,0000
	Avg	0,0398	0,0804	0,0878	0,1398	0,0401	0,0787
MCMO2a	20	0,0756	0,1710	0,1617	0,1893	0,0842	0,1435
	50	0,0521	0,1492	0,0200	0,0600	0,0167	0,0500
	100	0,0700	0,1970	0,0367	0,1100	0,0233	0,0700
	Avg	0,0659	0,1724	0,0728	0,1198	0,0414	0,0878
MCMO2b	20	0,0347	0,0857	0,1333	0,1698	0,0794	0,1317
	50	0,0119	0,0357	0,0067	0,0200	0,0075	0,0225
	100	0,0058	0,0175	0,0100	0,0300	0,0133	0,0400
	Avg	0,0175	0,0463	0,0500	0,0733	0,0334	0,0647
MOGLS	20	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	50	0,0033	0,0100	0,0000	0,0000	0,0000	0,0000
	100	0,0033	0,0071	0,0000	0,0000	0,0333	0,1000
	Avg	0,0022	0,0057	0,0000	0,0000	0,0111	0,0333

Tabelle 6.46: Vergleich SCMO, MCMO und MOGLS: AEX^k -Maß

AEX^k -Maß: Wie beim DUB^k -Maß verändert sich auch bzgl. $AEX_v^{\mu k}$ die Rangfolge der Heuristiken in Abhängigkeit von der Ziel- und Problemgröße (vgl. Tabelle 6.46). Es werden erneut alleine Durchschnittswerte über alle Probleminstanzen betrachtet. Bei der mittleren Durchlaufzeit und der maximalen Terminüberschreitung ist die SCMO-Heuristik erneut besser als die MCMO-Ansätze. Bei der Zykluszeit belegt sie Rang drei hinter MC-MO2a und MCMO1.

Wie schon bei den beiden anderen Auswertungsgrößen ist auch hier MOGLS deutlich schlechter als die Ameisenalgorithmen.

Zum Abschluß des Vergleichs der Heuristiken soll noch ein Vergleich der erzeugten und auf heuristische Effizienz überprüften Auftragsfolgen, sowie ein Vergleich des Anteils der lokalen Suche an der Rechenzeit durchgeführt

werden. In Tabelle 6.47 sind die benötigte Gesamtrechenzeit ($TicksT_v^\mu$), die Rechenzeit der lokalen Suche ($TicksLS_v^\mu$) und der Anteil der lokalen Suche an der Gesamtrechenzeit ($\%Ticks_v^\mu = TicksLS_v^\mu / TicksT_v^\mu$) angegeben.

Heuristik	N	$TicksT^\mu$	$TicksT^\sigma$	$TicksLS^\mu$	$TicksLS^\sigma$	$\%Ticks^\mu$
SCMO	20	1.637,0000	36,3419	1.185,1367	60,0772	0,7240
	50	11.836,8100	141,2584	5.828,5933	150,7887	0,4924
	100	66.328,4333	544,7932	19.837,4400	542,5728	0,2991
	AVG	26.600,7478	240,7978	8.950,3900	251,1462	0,3365
MCMO1	20	1.679,2967	12,5132	1.192,2300	45,2662	0,7100
	50	12.129,7433	79,6556	5.765,6167	89,3394	0,4753
	100	68.541,7567	463,3267	19.414,0833	454,6549	0,2832
	AVG	27.450,2656	185,1652	8.790,6433	196,4201	0,3202
MCMO2a	20	1.807,5767	47,8836	1.230,9000	79,7689	0,6810
	50	12.173,2433	111,2411	5.774,2233	105,4162	0,4743
	100	68.589,1900	455,3894	19.673,8100	455,1176	0,2868
	AVG	27.523,3367	204,8380	8.892,9778	213,4342	0,3231
MCMO2b	20	1.525,4600	34,8078	1.058,5033	57,3194	0,6939
	50	10.483,9300	83,5326	4.970,7000	93,3875	0,4741
	100	59.377,6133	431,3464	17.087,1500	429,2444	0,2878
	AVG	23.795,6678	183,2290	7.705,4511	193,3171	0,3238
MOGLS	20	1.811,7433	49,8023	423,0400	234,9188	0,2335
	50	12.218,5500	103,0143	1.892,2767	878,4146	0,1549
	100	68.891,1100	406,3374	7.227,8600	2.949,0520	0,1049
	AVG	27.640,4678	186,3847	3.181,0589	1.354,1285	0,1151

Tabelle 6.47: Vergleich SCMO, MCMO und MOGLS: Struktur der Rechenzeiten

In Tabelle 6.48 sind die Anzahl der insgesamt generierten Auftragsfolgen ($SolT_v^\mu$), die Anzahl der in der lokalen Suche generierten Auftragsfolgen ($SolLS_v^\mu$) sowie deren Anteil an der Anzahl der insgesamt generierten Auftragsfolgen ($\%Sol_v^\mu = SolLS_v^\mu / SolT_v^\mu$) wiedergegeben.

Zunächst ist zu beobachten, daß bei allen Heuristiken der Anteil der lokalen Suche an der Gesamtrechenzeit mit steigender Problemgröße sinkt. Gleichzeitig bleibt der Anteil der in der lokalen Suche generierten Auftragsfolgen an der Gesamtzahl der erzeugten Auftragsfolgen annähernd gleich. Die für die lokale Suche benötigte Rechenzeit steigt also relativ geringer an als die außerhalb der lokalen Suche benötigte Rechenzeit, obwohl der Anteil der in der lokalen Suche erzeugten Lösungen nahezu unverändert bleibt. Die Rechenzeit außerhalb der lokalen Suche wird insbesondere für die Verwaltung der Pheromonmatrizen und die Konstruktion von Auftragsfolgen durch die Ameisen aufgebracht. Falls lokale heuristisch effiziente Mengen zum Einsatz kommen, müssen deren Elemente auf globale heuristische Effizienz hin überprüft werden. Gleichzeitig wird durch die bei

Heuristik	N	$SolT^\mu$	$SolT^\sigma$	$SolLS^\mu$	$SolLS^\sigma$	$\%Sol^\mu$
SCMO	20	218.412,9000	1.003,9080	218.132,9000	1.003,9080	0,9987
	50	532.956,7000	8.118,6807	532.676,7000	8.118,6807	0,9995
	100	965.955,4000	21.935,2243	965.675,4000	21.935,2243	0,9997
	AVG	572.441,6667	10.352,6043	572.161,6667	10.352,6043	0,9995
MCMO1	20	218.391,1333	904,7564	218.111,1333	904,7564	0,9987
	50	528.652,5000	6.221,2590	528.372,5000	6.221,2590	0,9995
	100	950.738,1333	22.017,0833	950.458,1333	22.017,0833	0,9997
	AVG	565.927,2556	9.714,3662	565.647,2556	9.714,3662	0,9995
MCMO2a	20	217.868,7667	874,9379	217.588,7667	874,9379	0,9987
	50	526.575,2000	7.028,5207	526.295,2000	7.028,5207	0,9995
	100	958.835,7000	20.514,8980	958.555,7000	20.514,8980	0,9997
	AVG	567.759,8889	9.472,7855	567.479,8889	9.472,7855	0,9995
MCMO2b	20	187.676,8000	840,6911	187.436,8000	840,6911	0,9987
	50	455.356,8667	6.133,9363	455.116,8667	6.133,9363	0,9995
	100	833.966,4333	19.547,5707	833.726,4333	19.547,5707	0,9997
	AVG	492.333,3667	8.840,7327	492.093,3667	8.840,7327	0,9995
MOGLS	20	254.632,3000	47.793,3800	164.381,7333	30.861,9967	0,6456
	50	1.188.825,3333	106.099,8433	767.692,6000	68.482,2233	0,6458
	100	4.342.676,0000	232.848,0633	2.804.374,3333	150.335,6900	0,6458
	AVG	1.928.711,2111	128.913,7622	1.245.482,8889	83.226,6367	0,6458

Tabelle 6.48: Vergleich SCMO, MCMO und MOGLS: Anzahl generierter Auftragsfolgen

steigendem N sinkenden Werte von *range* die relative Größe der Nachbarschaft kleiner. Diese kleinere Nachbarschaft muß aber häufiger generiert werden, weil N steigt.

Weiterhin ist zu erkennen, daß MOGLS prozentual weniger Zeit in der lokalen Suche verbringt als die Ameisenalgorithmen. Auch der Anteil der in der lokalen Suche generierten Auftragsfolgen an der Anzahl der insgesamt generierten Auftragsfolgen ist geringer.

Abschließend kann zunächst festgehalten werden, daß MOGLS eindeutig schlechter ist als die Ameisenalgorithmen, selbst wenn diese eine kürzere Rechenzeit zur Verfügung haben, wie dies bei MCMO2b der Fall ist. Von den drei untersuchten Ameisenalgorithmen (SCMO, MCMO1 und MCMO2) ist keiner der eindeutig beste Ansatz. Bezüglich des *SQ*-Maßes ist SCMO den MCMO-Ansätzen unterlegen. Dafür liefert die Single-Colony-Strategie beim *DUB^k*-Maß und beim *AEX^k*-Maß insbesondere bei der mittleren Durchlaufzeit und der maximalen Terminüberschreitung tendenziell die besseren Ergebnisse.

6.5.2.2 Verbesserung der kleinsten oberen Schranken durch SC-MO und MCMO

In dieser, die experimentellen Analysen abschließenden Auswertung soll dargestellt werden, wie die in der Arbeit entwickelten Ameisenalgorithmen die kleinsten oberen Schranken aus Anhang C für die drei Zielgrößen verbessern. Dabei werden die in Tabelle 6.43 auf Seite 253 angegebenen Konfigurationen von SCMO, MCMO1 und MCMO2a verwendet.

Bei der Zykluszeit wird die kleinste obere Schranke nur bei Problem 62 von allen drei Ameisenalgorithmen von 5274 um 0,11% auf 5268 verbessert.

Die erzielten Verbesserungen der mittleren Durchlaufzeit sind in den Tabellen 6.49 bis 6.51 angegeben. Bei Problemen mit 20 und 50 Aufträgen werden 19 bzw. 20 der kleinsten oberen Schranken verbessert. Bei großen Problemen mit 100 Aufträgen werden nur noch 8 kleinste obere Schranken verbessert. Die durchschnittliche Verbesserung liegt dabei mit 0,16% bei Problemen mit 20 Aufträgen, 0,55% bei Problemen mit 50 Aufträgen und 0,17% bei Problemen mit 100 Aufträgen deutlich unter 1 Prozent.

Die Verbesserungen der maximalen Terminüberschreitung sind in den Tabellen 6.52 bis 6.54 angegeben. Außer bei Problem 40 werden alle kleinsten oberen Schranken verbessert. Im Gegensatz zu den kleinsten oberen Schranken der Zykluszeit und der mittleren Durchlaufzeit sind die alten kleinsten oberen Schranken der maximalen Terminüberschreitung jedoch nicht von auf die Zielgröße spezialisierten Lösungsverfahren ermittelt worden. Insofern bringen die Verbesserungen bei der maximalen Terminüberschreitung lediglich zum Ausdruck, daß die Ameisenalgorithmen bei der individuellen Minimierung der maximalen Terminüberschreitung in der Regel besser sind als die von der GPWGA- und die MOGLS-Heuristik in Anhang C ermittelten Werte. Bei Problemen mit 20 Aufträgen beträgt die Verbesserung im Durchschnitt 3,74%. Bei 50 Aufträgen werden die kleinsten oberen Schranken durchschnittlich um 3,06% und bei 100 Aufträgen um 1,78% verbessert.

Nr.	N	M	UB alt	SCMO	MCMO1	MCMO2	UB verb.	$\frac{UB\ alt}{UB\ verb} - 1$
1	20	5	702,05	702,05	701,65	701,70	701,65	0,0006
2	20	5	757,55	758,45	757,55	757,95	-	-
3	20	5	670,65	665,05	665,05	665,05	665,05	0,0084
4	20	5	774,30	773,90	772,35	772,85	772,35	0,0025
5	20	5	676,45	676,45	676,45	676,45	-	-
6	20	5	656,95	656,15	656,15	656,15	656,15	0,0012
7	20	5	677,95	677,40	677,40	677,40	677,40	0,0008
8	20	5	698,40	697,40	697,40	697,40	697,40	0,0014
9	20	5	715,85	714,75	714,75	714,75	714,75	0,0015
10	20	5	648,40	648,80	648,80	648,80	-	-
11	20	10	1.047,90	1045,55	1045,55	1045,55	1045,55	0,0022
12	20	10	1.122,00	1122	1122,00	1122,00	-	-
13	20	10	991,65	991,65	991,65	991,65	-	-
14	20	10	936,20	935,85	938,25	938,25	935,85	0,0004
15	20	10	932,20	932,05	932,60	932,60	932,05	0,0002
16	20	10	962,25	966,75	966,90	966,20	-	-
17	20	10	918,80	918,15	918,15	918,80	918,15	0,0007
18	20	10	1.012,05	1012,05	1012,05	1012,05	-	-
19	20	10	1.016,50	1016,50	1016,50	1016,50	-	-
20	20	10	1.066,00	1066,00	1066,00	1066,75	-	-
21	20	20	1.681,15	1681,15	1681,15	1681,15	-	-
22	20	20	1.580,20	1579,35	1579,35	1579,85	1579,35	0,0005
23	20	20	1.696,00	1696,00	1696,00	1698,50	-	-
24	20	20	1.584,90	1585,35	1583,05	1586,90	1583,05	0,0012
25	20	20	1.729,65	1730,25	1729,50	1729,30	1729,30	0,0002
26	20	20	1.631,85	1628,20	1628,20	1628,20	1628,20	0,0022
27	20	20	1.651,90	1646,10	1653,45	1655,70	1646,10	0,0035
28	20	20	1.622,20	1620,60	1620,60	1623,05	1620,6	0,0010
29	20	20	1.681,25	1680,00	1681,15	1680,60	1680,00	0,0007
30	20	20	1.615,75	1613,45	1613,10	1615,85	1613,10	0,0016
Avg								0,0016

Tabelle 6.49: Verbesserte kleinste obere Schranken - F_{mid} : TAIL001-TAIL030

Nr.	N	M	UB alt	SCMO	MCMO1	MCMO2	UB verb.	$\frac{UB\ alt}{UB\ verb.} - 1$
31	50	5	1.308,56	1301,20	1302,78	1304,58	1301,20	0,0057
32	50	5	1.373,28	1370,68	1368,68	1372,12	1368,68	0,0034
33	50	5	1.283,32	1277,56	1280,42	1280,28	1277,56	0,0045
34	50	5	1.382,26	1380,16	1374,96	1378,56	1374,96	0,0053
35	50	5	1.406,62	1399,86	1398,56	1399,82	1398,56	0,0058
36	50	5	1.351,26	1351,46	1352,48	1355,34	-	-
37	50	5	1.340,28	1332,62	1327,24	1332,70	1327,24	0,0098
38	50	5	1.297,26	1297,56	1295,86	1298,00	1295,86	0,0011
39	50	5	1.274,70	1266,4	1264,70	1266,48	1264,70	0,0079
40	50	5	1.405,12	1389,66	1390,16	1390,44	1389,66	0,0111
41	50	10	1.775,40	1782,94	1768,88	1776,16	1768,88	0,0037
42	50	10	1.672,24	1680,42	1689,12	1696,60	-	-
43	50	10	1.629,28	1615,62	1617,42	1615,90	1615,62	0,0085
44	50	10	1.751,58	1753,36	1751,30	1750,42	1750,42	0,0007
45	50	10	1.766,46	1746,98	1745,00	1756,66	1745,00	0,0123
46	50	10	1.760,22	1749,38	1751,70	1759,10	1749,38	0,0062
47	50	10	1.799,12	1799,38	1792,24	1793,60	1792,24	0,0038
48	50	10	1.766,60	1762,78	1767,04	1759,58	1759,58	0,0040
49	50	10	1.739,50	1740,58	1742,86	1745,24	-	-
50	50	10	1.786,64	1779,44	1783,86	1792,02	1779,44	0,0040
51	50	20	2.537,48	2557,66	2552,32	2550,94	-	-
52	50	20	2.424,16	2411,96	2421,90	2408,96	2408,96	0,0063
53	50	20	2.343,36	2369,34	2366,28	2361,54	-	-
54	50	20	2.458,12	2446,46	2448,96	2447,50	2446,46	0,0048
55	50	20	2.398,40	2419,62	2413,98	2417,62	-	-
56	50	20	2.447,38	2449,56	2453,50	2447,38	-	-
57	50	20	2.506,34	2515,30	2512,98	2507,44	-	-
58	50	20	2.490,86	2500,24	2497,56	2495,28	-	-
59	50	20	2.476,84	2480,46	2488,04	2492,30	-	-
60	50	20	2.524,06	2519,56	2528,94	2524,78	2519,56	0,0018
Avg								0,0055

Tabelle 6.50: Verbesserte kleinste obere Schranken - F_{mid} : TAIL031-TAIL060

Nr.	N	M	UB alt	SCMO	MCMO1	MCMO2	UB verb.	$\frac{UB\ alt}{UB\ verb} - 1$
61	100	5	2.567,89	2566,11	2567,64	2569,08	2566,11	0,0007
62	100	5	2.456,09	2454,75	2453,75	2456,85	2453,75	0,0010
63	100	5	2.405,37	2408,18	2410,97	2409,73	-	-
64	100	5	2.304,80	2298,01	2302,97	2302,55	2298,01	0,0030
65	100	5	2.430,13	2431,24	2437,38	2437,52	-	-
66	100	5	2.357,93	2358,51	2363,78	2358,38	-	-
67	100	5	2.437,41	2432,09	2438,65	2434,42	2432,09	0,0022
68	100	5	2.344,53	2340,21	2337,31	2345,73	2337,31	0,0031
69	100	5	2.512,91	2522,74	2521,99	2520,08	-	-
70	100	5	2.462,61	2465,02	2465,83	2468,62	-	-
71	100	10	3.039,10	3050,55	3044,54	3037,40	3037,40	0,0006
72	100	10	2.785,96	2781,01	2795,66	2784,47	2781,01	0,0018
73	100	10	2.953,89	2976,81	2976,89	2977,34	-	-
74	100	10	3.068,50	3076,56	3077,51	3080,94	-	-
75	100	10	2.890,60	2945,21	2930,62	2932,06	-	-
76	100	10	2.753,84	2762,91	2770,48	2758,63	-	-
77	100	10	2.858,58	2862,70	2868,72	2867,35	-	-
78	100	10	2.961,30	2965,96	2975,47	2963,36	-	-
79	100	10	3.096,64	3106,30	3092,38	3102,61	3092,38	0,0014
80	100	10	2.964,62	3012,47	3015,83	2987,66	-	-
81	100	20	3.721,56	3782,92	3785,18	3781,28	-	-
82	100	20	3.806,44	3877,87	3869,87	3886,02	-	-
83	100	20	3.775,79	3824,51	3845,47	3840,58	-	-
84	100	20	3.802,01	3879,89	3905,6	3908,01	-	-
85	100	20	3.772,68	3810,39	3827,12	3805,19	-	-
86	100	20	3.788,57	3853,93	3863,24	3845,63	-	-
87	100	20	3.815,31	3863,46	3867,04	3858,15	-	-
88	100	20	3.926,97	3986,02	3991,22	3973,87	-	-
89	100	20	3.825,52	3847,64	3860,19	3857,95	-	-
90	100	20	3.872,82	3952,56	3952,42	3942,21	-	-
Avg								0,0017

Tabelle 6.51: Verbesserte kleinste obere Schranken - F_{mid} : TAIL061-TAIL090

Nr.	N	M	UB alt	SCMO	MCMO1	MCMO2	UB verb.	$\frac{UB\ alt.}{UB\ verb.} - 1$
1	20	5	549	523	523	523	523	0,0497
2	20	5	498	478	478	478	478	0,0418
3	20	5	677	646	646	647	646	0,0480
4	20	5	543	529	534	529	529	0,0265
5	20	5	524	488	488	488	488	0,0738
6	20	5	656	651	651	652	651	0,0077
7	20	5	587	574	576	576	574	0,0226
8	20	5	652	644	643	643	643	0,0140
9	20	5	697	689	689	689	689	0,0116
10	20	5	573	539	538	538	538	0,0651
11	20	10	861	855	855	859	855	0,0070
12	20	10	892	864	864	860	860	0,0372
13	20	10	529	488	488	490	488	0,0840
14	20	10	659	637	637	655	637	0,0345
15	20	10	929	905	910	908	905	0,0265
16	20	10	1.018	975	975	975	975	0,0441
17	20	10	1.059	1051	1051	1051	1051	0,0076
18	20	10	657	640	640	638	638	0,0298
19	20	10	814	779	779	779	779	0,0449
20	20	10	746	744	743	744	743	0,0040
21	20	20	1.257	1212	1205	1211	1205	0,0432
22	20	20	826	784	784	815	784	0,0536
23	20	20	1.153	1106	1102	1106	1102	0,0463
24	20	20	926	901	901	901	901	0,0277
25	20	20	394	362	363	363	362	0,0884
26	20	20	1.046	1031	1031	1031	1031	0,0145
27	20	20	810	767	775	778	767	0,0561
28	20	20	843	806	806	816	806	0,0459
29	20	20	686	668	671	671	668	0,0269
30	20	20	763	735	735	735	735	0,0381
Avg								0,0374

Tabelle 6.52: Verbesserte kleinste obere Schranken - T_{max} : TAIL001-TAIL030

Nr.	N	M	UB alt	SCMO	MCMO1	MCMO2	UB verb.	$\frac{UB\ alt. - 1}{UB\ verb.}$
31	50	5	1.797	1765	1772	1772	1765	0,0181
32	50	5	2.715	2710	2721	2710	2710	0,0018
33	50	5	1.773	1749	1763	1765	1749	0,0137
34	50	5	1.867	1827	1836	1836	1827	0,0219
35	50	5	2.390	2317	2317	2317	2317	0,0315
36	50	5	2.316	2244	2244	2244	2244	0,0321
37	50	5	1.394	1352	1356	1356	1352	0,0311
38	50	5	1.876	1823	1822	1828	1822	0,0296
39	50	5	2.298	2284	2284	2284	2284	0,0061
40	50	5	2.563	2564	2564	2564	-	-
41	50	10	2.202	2141	2138	2165	2138	0,0299
42	50	10	2.167	2122	2154	2147	2122	0,0212
43	50	10	1.856	1817	1815	1816	1815	0,0226
44	50	10	1.929	1847	1880	1881	1847	0,0444
45	50	10	2.209	2119	2143	2152	2119	0,0425
46	50	10	1.905	1851	1872	1888	1851	0,0292
47	50	10	1.947	1906	1904	1921	1904	0,0226
48	50	10	2.149	2119	2131	2111	2111	0,0180
49	50	10	2.086	2027	2043	2038	2027	0,0291
50	50	10	2.344	2274	2292	2263	2263	0,0358
51	50	20	2.620	2520	2507	2514	2507	0,0451
52	50	20	2.700	2569	2599	2592	2569	0,0510
53	50	20	2.246	2188	2195	2192	2188	0,0265
54	50	20	2.853	2735	2739	2721	2721	0,0485
55	50	20	2.667	2586	2608	2614	2586	0,0313
56	50	20	2.203	2104	2098	2085	2085	0,0566
57	50	20	2.675	2576	2585	2575	2575	0,0388
58	50	20	2.613	2565	2596	2574	2565	0,0187
59	50	20	1.839	1734	1778	1785	1734	0,0606
60	50	20	2.169	2120	2113	2107	2107	0,0294
Avg								0,0306

Tabelle 6.53: Verbesserte kleinste obere Schranken - T_{max} : TAIL031-TAIL060

Nr.	N	M	UB alt	SCMO	MCMO1	MCMO2	UB verb.	$\frac{UB\ alt}{UB\ verb} - 1$
61	100	5	4.417	4400	4431	4417	4400	0,0039
62	100	5	4.459	4435	4424	4424	4424	0,0079
63	100	5	4.782	4715	4772	4748	4715	0,0142
64	100	5	3.743	3726	3734	3750	3726	0,0046
65	100	5	4.045	4013	4031	4052	4013	0,0080
66	100	5	4.485	4412	4439	4462	4412	0,0165
67	100	5	4.489	4441	4441	4439	4439	0,0113
68	100	5	4.049	3958	3967	3964	3958	0,0230
69	100	5	4.770	4703	4715	4719	4703	0,0142
70	100	5	4.026	3988	4014	4021	3988	0,0095
71	100	10	4.803	4690	4703	4672	4672	0,0280
72	100	10	4.888	4910	4938	4913	-	-
73	100	10	4.826	4747	4774	4750	4747	0,0166
74	100	10	5.091	5024	5040	5040	5024	0,0133
75	100	10	4.537	4443	4443	4439	4439	0,0221
76	100	10	4.714	4622	4619	4620	4619	0,0206
77	100	10	5.039	4977	4972	4972	4972	0,0135
78	100	10	4.518	4475	4447	4423	4423	0,0215
79	100	10	4.174	4136	4158	4150	4136	0,0092
80	100	10	4.523	4482	4495	4470	4470	0,0119
81	100	20	4.654	4627	4697	4666	4627	0,0058
82	100	20	5.003	4861	4861	4851	4851	0,0313
83	100	20	4.891	4827	4837	4778	4778	0,0237
84	100	20	5.158	5011	5067	5038	5011	0,0293
85	100	20	4.374	4260	4311	4274	4260	0,0268
86	100	20	4.641	4546	4588	4558	4546	0,0209
87	100	20	5.133	4978	5017	4982	4978	0,0311
88	100	20	5.364	5233	5289	5266	5233	0,0250
89	100	20	5.037	4899	4938	4903	4899	0,0282
90	100	20	4.952	4831	4859	4839	4831	0,0250
Avg								0,0178

Tabelle 6.54: Verbesserte kleinste obere Schranken - T_{max} : TAIL061-TAIL090

6.6 Zusammenfassung

In diesem Kapitel wurde im ersten Abschnitt zunächst die indirekte Kommunikation von Ameisen bei der Futtersuche als Motivation für die Entwicklung der Ameisenalgorithmen vorgestellt. Darauf aufbauend wurde die ACO-Metaheuristik definiert und für das Permutation Flow Shop Problem konkrete, in der Literatur verwendete, Ausgestaltungsmöglichkeiten aufgezeigt. Anschließend sind unterschiedliche Möglichkeiten zum Aufbau von a-posteriori Ameisenalgorithmen diskutiert wurden. Dabei sind Erkenntnisse aus der Parallelisierung von ACO-Heuristiken und Multi-Colony-Ansätzen in die Entwicklung und Beurteilung eingeflossen.

Im zweiten Abschnitt wurde, motiviert durch die Ergebnisse der Ziel- und Lösungsraumanalyse aus Kapitel 5, eine zweiphasige lokale Suche entwickelt. Neben dem Job Swap Schema und dem Job Insertion Schema wurde insbesondere das Job Swap Schema mit rekursiver API Suche eingeführt. Motivation der rekursiven API Suche ist es dabei, die Suche verstärkt in oder zumindest in der Nähe der Nachbarschaft im Zielraum einer Ausgangslösung durchzuführen.

Im dritten Abschnitt wurde mit dem Single-Colony-Multi-Objective-Ansatz eine erste Möglichkeit eines a-posteriori Ameisenalgorithmus entwickelt und analysiert. Der Ansatz verwendet die gewichtete Tchebycheff-Distanz als Kompromißmodell. Durch die systematische Variation der Gewichte wird die Suche einer einzelnen, homogenen Kolonie in unterschiedliche Bereich des Zielraums gelenkt und die Elemente einer heuristisch effizienten Menge gesucht.

Ebenfalls aufbauend auf die systematische Variation der Gewichte suchen die zwei Varianten des in Abschnitt vier vorgestellten Multi-Colony-Multi-Objective-Ansatzes nach Elementen einer heuristisch effizienten Menge. Im Gegensatz zum SCMO-Ansatz werden dabei drei unterschiedliche, jedoch nicht unabhängige, homogene Kolonien eingesetzt. Zwischen den Kolonien findet indirekt eine Kommunikation über die globale heuristisch effiziente Menge statt.

Wurden sowohl für den SCMO-Ansatz als auch für die beiden MCMO-Ansätze ausführliche experimentelle Analysen zunächst isoliert durchgeführt, so wurden sie im fünften Abschnitt dieses Kapitels unter der Bedingung (annähernd) gleicher Rechenzeit miteinander verglichen. Dabei hat sich keiner der Ansätze als eindeutig vorteilhaft erwiesen. Ein ebenfalls durchgeführter Vergleich mit der MOGLS-Heuristik hat ergeben, daß dieser deutlich schlechter ist als der SCMO-Ansatz und die beiden MCMO-

Ansätze.

Schließlich wurde gezeigt, daß die in diesem Kapitel entwickelten Ameisenalgorithmen die kleinsten oberen Schranken für die maximale Terminüberschreitung und insbesondere auch viele der kleinsten oberen Schranken für die mittlere Durchlaufzeit verbessern. Insbesondere bei der mittleren Durchlaufzeit fällt die Verbesserung aber gering aus.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurde das Permutation Flow Shop Problem mit mehrfacher Zielsetzung betrachtet. Ziel war die Ermittlung einer heuristisch effizienten Menge von Auftragsfolgen. Die Lösung des Problems sollte mit Hilfe von Ameisenalgorithmen erfolgen. Diese haben sich in den letzten Jahren als effektive und effiziente Methode zur Lösung unterschiedlicher kombinatorischer Optimierungsprobleme, hauptsächlich bei einfacher Zielsetzung, erwiesen.

Nach der Definition des behandelten Permutation Flow Shop Problems wurden kurz die entscheidungsrelevanten Kosten der Reihenfolgeplanung diskutiert. Darauf aufbauend ist das in dieser Arbeit betrachtete Zielsystem, bestehend aus der Zykluszeit, der mittleren Durchlaufzeit und der maximalen Terminüberschreitung, abgeleitet worden. Die drei Ziele sind alle zu minimieren. Bei der Diskussion der entscheidungsrelevanten Kosten wurde insbesondere festgestellt, daß die Leerzeitkosten, welche häufig mit der Zykluszeit in Verbindung gebracht werden, keine Komponente der entscheidungsrelevanten Kosten sind. Die Zykluszeit, welche die Leerzeiten der Maschinen minimiert, wurde dennoch als Steuer- und Kontrollparameter in das Zielsystem aufgenommen. Ihre Minimierung stellt sicher, daß möglichst früh ein neuer Block von Aufträgen in das Fertigungssystem eingeplant werden kann, was die Flexibilität in der Zukunft erhöht. Weiterhin kann ein zu großzügiger Umgang mit den Kapazitäten dazu führen, daß Aufträge in der Zukunft aus Kapazitätsmangel abgelehnt werden müssen. Diese eventuell entgangenen Gewinne können durchaus als Opportunitätskosten betrachtet werden. Abschließend wurde eine Korrelations-

analyse der Zielgrößen durchgeführt. Zusammen mit den Ergebnissen aus Abschnitt 5.5.2 kann vermutet werden, daß bereichsweise Zielbeziehungen vorliegen. Außerhalb des Bereichs des Lösungsraums, in dem die globalen Optima liegen, kann eine Zielneutralität vermutete werden. Innerhalb des Bereichs des Lösungsraums, in dem die globalen Optima liegen, liegt dagegen in der Regel ein Zielkonflikt vor.

Bei dem vorliegenden Problem handelt es sich um ein diskretes Vektorminimierungsproblem. Im dritten Kapitel der Arbeit wurden daher unterschiedliche Kompromißmodelle zur Ermittlung effizienter Lösungen diskutiert. Der Schwerpunkt der Betrachtung lag dabei auf der Möglichkeit, durch ein geeignetes, systematisches Variieren der Parameter des Kompromißmodells alle effizienten Lösungen zu bestimmen. Insbesondere mußte darauf geachtet werden, daß auch die sogenannten nicht-unterstützten effizienten Lösungen ermittelt werden können. Für die Verwendung in den zu entwickelnden Ameisenalgorithmen ist schließlich das Kompromißmodell der gewichteten Tchebycheff-Distanz ausgewählt worden.

Im vierten Kapitel wurde die Beurteilung der Qualität von heuristisch effizienten Mengen ausführlich diskutiert. In der Literatur ist dabei noch kein einheitliches Vorgehen erkennbar. Unter der Betrachtung von drei Qualitätsaspekten wurden unterschiedliche Maße vorgestellt und gezeigt, daß ein Großteil der bisher in der Literatur verwendeten Maße allein aus theoretischen Überlegungen nicht dazu geeignet ist, die Lösungsqualität korrekt wiederzugeben. So ist mit dem S-Maß Quotienten alleine ein Qualitätsmaß zur Messung der "Distanz zur effizienten Menge" in das Beurteilungssystem aufgenommen worden. Die anderen beiden, häufig in der Literatur betrachteten Qualitätsaspekte "Verteilung der Elemente der heuristisch effizienten Menge im Zielraum" sowie die "Größe der aufgezeigten Variationsbreiten der Zielfunktionswerte" sind mangels geeigneter Maße im Beurteilungssystem nicht berücksichtigt worden. Zusätzlich wurden Maße zur Erfassung des bisher in der Literatur nicht betrachteten Aspekts der "Optimierungseigenschaft bzgl. der einzelnen Ziele" entwickelt und in das Beurteilungssystem aufgenommen.

In Kapitel fünf sind allgemeine, für die Steuerung der Suche nach Elementen der effizienten Menge relevante Überlegungen angestellt worden. Zunächst wurde überprüft, ob und wie häufig ein Nachbar im Lösungsraum auch ein Nachbar im Zielraum ist. Dabei wurde auf unterschiedliche, bereits existierende Nachbarschaften im Lösungsraum zurückgegriffen. Für die Nachbarschaft im Zielraum waren zunächst eigene Konzepte zu entwickeln. Verschiedene experimentelle Analysen haben anschließend gezeigt, daß ein Nachbar im Lösungsraum nur selten auch ein Nachbar im

Zielraum ist. Für die API-Nachbarschaft ist die Wahrscheinlichkeit dafür am größten. Die Ergebnisse sind in Kapitel sechs in die Entwicklung einer speziellen zweiphasigen lokalen Suche eingeflossen.

Des Weiteren wurde in Kapitel fünf die Existenz des sogenannten "Großen Tals" überprüft. Dazu wurde zunächst die Approximationsqualität der bisher in der Literatur verwendeten Maße zur Messung der Distanz zwischen zwei Permutationen untersucht. Diese werden verwendet, weil für die Berechnung der echten Distanz zwischen zwei Permutationen kein polynomialer Algorithmus bekannt ist. In den Untersuchungen konnte festgestellt werden, daß die Approximationsqualität nicht ausreichend ist. Die in dieser Arbeit durchgeführten Analysen beschränken sich daher auf kleine Problemgrößen mit bis zu acht Aufträgen. Für diese Problemgrößen kann mit Hilfe eines vorgestellten, rekursiven Algorithmus die echte Distanz in angemessener Rechenzeit ermittelt werden. Wird das "Große Tal" derart definiert, daß es vorliegt, falls die globalen Optima eines Ziels in einem begrenzten Bereich des Lösungsraums zusammenliegen, so unterstützen die Ergebnisse der durchgeführten Untersuchungen seine Existenz.¹ Weiter konnte gezeigt werden, daß sich die Elemente der effizienten Menge in unterschiedlichen Regionen des Lösungsraums befinden.

Im letzten Kapitel sind mit dem Single-Colony-Multi-Objective-Ansatz und zwei Varianten des Multi-Colony-Multi-Objective-Ansatzes unterschiedliche Ameisenalgorithmen zur Problemlösung entwickelt worden, welche beide die gewichtete Tchebycheff-Distanz als Kompromißmodell verwenden. Im SCMO-Ansatz wird durch die systematische Variation der Gewichte die Suche einer einzelnen homogenen Kolonie in unterschiedliche Bereiche des Zielraums gelenkt und die Elemente einer heuristisch effizienten Menge gesucht. Im Gegensatz zum SCMO-Ansatz werden im MCMO-Ansatz drei unterschiedliche, jedoch nicht unabhängige, homogene Kolonien eingesetzt, die miteinander auf indirektem Weg kommunizieren. In ausführlichen experimentellen Analysen wurde die Vorteilhaftigkeit unterschiedlicher Parametereinstellungen überprüft. Weder der SCMO-Ansatz noch eine der beiden Varianten des MCMO-Ansatzes sind unter Anwendung des verwendeten Systems zur Beurteilung der Lösungsqualität zu bevorzugen. Es konnte festgestellt werden, daß die als externes Vergleichsverfahren verwendete MOGLS-Heuristik bei gleicher Rechenzeit deutlich schlechter ist als die entwickelten Ameisenalgorithmen. Abschließend wurde gezeigt, daß, obwohl es sich

¹Damit unterscheidet sich die Definition von der in den Arbeiten von Boese u. a. (1993), Boese u. a. (1994). Das "Große Tal" beschreibt in den dort durchgeführten Versuchen die Möglichkeit, mit nur wenigen Zügen eines Nachbarschaftsoperators, von einem lokalen Optimum in ein anderes zu gelangen (vgl. Boese u. a. (1993, S.4), Boese u. a. (1994, S.103f.)).

bei den entwickelten Ameisenalgorithmen um a-posteriori Ansätze für den Fall mehrfacher Zielsetzung handelt, sie insbesondere für die mittlere Durchlaufzeit die bisher besten bekannten kleinsten oberen Schranken für viele der ersten 90 Taillardbenchmarkprobleme leicht verbessern. Dabei ist zu berücksichtigen, daß die alten kleinsten oberen Schranken der mittleren Durchlaufzeit i.d.R. von auf dieses Ziel spezialisierten Verfahren ermittelt wurden.

Aus den in dieser Arbeit erzielten Ergebnissen ergeben sich Ansatzpunkte für weiterführende Untersuchungen:

So besteht weiterhin ein Bedarf an Maßen zur Erfassung unterschiedlicher Aspekte der Lösungsqualität heuristisch effizienter Mengen. Ebenso sollte versucht werden, bessere Maße zur Abschätzung der echten Distanz im Lösungsraum zwischen zwei Auftragsfolgen (bzw. allgemein Permutationen) zu entwickeln.

Auch bei den Ameisenalgorithmen besteht weiterer Forschungsbedarf. So ist z.B. bei den MCMO-Ansätzen bisher nicht berücksichtigt worden, daß die einzelnen Kolonien mit unterschiedlichen Parametereinstellungen, z.B. für die Verdunstungsrate, suchen können. Ebenso wird sowohl im SCMO- als auch im MCMO-Ansatz die Steuerung der Suche im Zielraum noch statisch vorgenommen. Hier könnte versucht werden, Methoden zu entwickeln, die es der Heuristik ermöglichen, selber zu entscheiden, in welchen Bereichen des Zielraums eine verstärkte Suche notwendig sein könnte.

Literaturverzeichnis

[Bagchi 1999]

BAGCHI, T. P.: *Multiobjective Scheduling by Genetic Algorithms*. Kluwer Academic Publishers, Boston, Dordrecht, London, 1999

[Bauer u. a. 1999]

BAUER, A.; BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C.: An Ant Colony Optimization Approach for the Single Machine Total Tardiness Problem. In: *Proceedings of the 1999 Congress on Evolutionary Computation (CEC'99), Washington, D.C., USA*, IEEE Service Center, Piscataway, NJ, 1999, S. 1445–1450

[Blohm u. a. 1988]

BLOHM, H.; BEER, T.; SEIDENBERG, U.; SILBER, H.: *Produktionswirtschaft*. 2. unveränd. Auflage. Verlag Neue Wirtschafts-Briefe, Herne, Berlin, 1988

[Boese u. a. 1993]

BOESE, K. D.; KAHNG, A. B.; MUDDU, S.: On the Big Valley and Adaptive Multit-Start for Discrete Global Optimizations (CSD-930015) / Computer Science Department, University of California, Los Angeles, CA 90024-1596 USA. 1993. – Forschungsbericht

[Boese u. a. 1994]

BOESE, K. D.; KAHNG, A. B.; MUDDU, S.: A new adaptive multi-start technique for combinatorial global optimization. In: *Operations Research Letters* 16 (1994), S. 101–113

[Bonabeau u. a. 1999]

BONABEAU, E.; DORIGO, M.; THERAULAZ, G.: *Swarm Intelligence From Nature to Artificial Systems*. Oxford University Press, New York, Oxford, 1999

[Borges u. Hansen 1998]

BORGES, P. C.; HANSEN, M. P.: A basis for future successes in multiob-

jective combinatorial optimization / Institute of Mathematical Modelling, Technical University of Denmark, 2800 Lyngby. 1998 (IMM-REP-1998-8). – Forschungsbericht

[Bowman Jr. 1976]

BOWMAN JR., J. V.: On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In: *Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems 130*, Springer-Verlag, Berlin, Heidelberg, New-York, 1976, S. 76–85

[Bullnheimer u. a. 1999]

BULLNHEIMER, B.; HARTL, R. F.; STRAUSS, C.: A New Rank Based Version of the Ant System - A Computational Study. In: *Central European Journal of Operations Research (CEJOR)* 7 (1999), S. 25–38

[Bullnheimer u. a. 1998]

BULLNHEIMER, B.; KOTSIS, G.; STRAUSS, C.: Parallelization Strategies for the Ant System. In: *High Performance Algorithms and Software in Nonlinear Optimization*. De Leone, R. and Murli, A. and Pardalos, P. M. and Toraldo, G. Kluwer Academic Publishers, Dordrecht, Bosten, London, 1998, S. 87–100

[Chang u. a. 2002]

CHANG, P.-C.; HSIEH, J.-C.; LIN, S.-G.: The development of gradual priority weighting approach for the multi-objective flowshop scheduling problem. In: *International Journal of Production Economics* 79 (2002), S. 171–183

[Colorni u. a. 1996]

COLORNI, A.; DORIGO, M.; MAFFIOLI, F.; MANIEZZO, V.; RIGHINI, G.; TRUBIAN, M.: Heuristics from Nature for Hard Combinatorial Optimization Problems. In: *International Transactions in Operational Research* 3 (1996), Nr. 1, S. 1–21

[Colorni u. a. 1994]

COLORNI, A.; DORIGO, M.; MANIEZZO, V.; TRUBIAN, M.: Ant system for Job-shop scheduling. In: *Belgian journal of Operations Research, Statistics and Computer Science* 34 (1994), Nr. 1, S. 39–53

[Conway u. a. 1967]

CONWAY, R. W.; MAXWELL, W. L.; MILLER, L. W.: *Theory of scheduling*. Addison-Wesley, Reading, Massachusetts, Palo Alto, London, Don Mills, Ontario, 1967

[Czyzak u. Jaszkievicz 1998]

CZYZAK, P.; JASZKIEWICZ, A.: Pareto Simulated Annealing - A Metaheuristic Technique for Multiple-Objective Combinatorial Optimization. In: *Journal of Multi-Criteria Decision Analysis* 7 (1998), S. 34–47

[Daniels 1992]

DANIELS, R. L.: Analytical Evaluation of Multi-Criteria Heuristics. In: *Management Science* 38 (1992), Nr. 4, S. 501–513

[Daniels u. Chambers 1990]

DANIELS, R. L.; CHAMBERS, R. J.: Multiobjective Flow-Shop Scheduling. In: *Naval Research Logistics* 37 (1990), S. 981–995

[Deb u. a. 2000]

DEB, K.; AGRAWAL, S.; PRATAP, A.; MEYARIVAN, T.: A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In: *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN-VI), Paris, Frankreich, LNCS 1917*, Springer-Verlag, Berlin, 2000, S. 849–858

[den Besten u. a. 2000]

DEN BESTEN, M.; STÜTZLE, T.; DORIGO, M.: Ant Colony Optimization for the Total Weighted Tardiness Problem. In: *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature (PPSN-IV), Paris, Frankreich, LNCS 1917*, Springer-Verlag, Berlin, 2000, S. 611–620

[Dietmar 1991]

DIETMAR, H.: *Statistik in C, Methoden der bivarianten Statistik effizient programmiert*. Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig, 1991

[Dinkelbach 1962]

DINKELBACH, W.: Unternehmerische Entscheidung bei mehrfacher Zielsetzung. In: *ZfB* 32 (1962), S. 739–747

[Dinkelbach 1977]

DINKELBACH, W.: Ablaufplanung in entscheidungstheoretischer Sicht. In: *Zeitschrift für Betriebswirtschaft* 47 (1977), Nr. 9, S. 543–566

[Dinkelbach 1982]

DINKELBACH, W.: *Entscheidungsmodelle*. Walter de Gruyter, Berlin, New York, 1982

[Domschke u. Drexl 1998]

DOMSCHKE, W.; DREXL, A.: *Einführung in Operations Research*. 4., verbesserte Auflage. Springer-Verlag, Berlin, Heidelberg, New York, 1998

[Dorigo 1992]

DORIGO, M.: *Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale (Optimization, learning and natural algorithms)*, Dipartimento di Elettronica, Politecnico di Milano, Italien, Diss., 1992. – (in italienisch)

[Dorigo u. Di Caro 1999]

DORIGO, M.; DI CARO, G.: The Ant Colony Optimization Meta-Heuristic. In: *New Ideas in Optimization*. Corne, D. and Dorigo, M. and Glover, F. McGrawHill, London, 1999, S. 11–32

[Dorigo u. a. 1999]

DORIGO, M.; DI CARO, G.; GAMBARDELLA, L. M.: Ant Algorithms for Discrete Optimization. In: *Artificial Life* 5 (1999), Nr. 2, S. 137–172

[Dorigo u. Gambardella 1997a]

DORIGO, M.; GAMBARDELLA, L. M.: Ant colonies for the traveling salesman problem. In: *BioSystems* 43 (1997), S. 73–81

[Dorigo u. Gambardella 1997b]

DORIGO, M.; GAMBARDELLA, L. M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. In: *IEEE Transactions on Evolutionary Computation* 1 (1997), Nr. 1, S. 53–66

[Dorigo u. a. 1991a]

DORIGO, M.; MANIEZZO, V.; COLORNI, A.: Ant System: An Autocatalytic Optimizing Process (91-016 (überarbeitete Fassung)) / Dipartimento di Elettronica e Informazione Politecnico di Milano, Italien. 1991. – Forschungsbericht

[Dorigo u. a. 1991b]

DORIGO, M.; MANIEZZO, V.; COLORNI, A.: Positive feedback as a search strategy (91-016) / Dipartimento di Elettronica e Informatica, Politecnico di Milano, Italien. 1991. – Forschungsbericht

[Dorigo u. a. 1996]

DORIGO, M.; MANIEZZO, V.; COLORNI, A.: The Ant System: Optimization by a colony of cooperating agents. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part B* 26 (1996), Nr. 1, S. 1–13

[Dorigo u. Stützle 2004]

DORIGO, M.; STÜTZLE, T.: *Ant Colony Optimization*. Cambridge, Massachusetts, London: The MIT Press, 2004

[Ehrgott u. Gandibleux 2000]

EHRGOTT, M.; GANDIBLEUX, X.: A survey and annotated bibliography of multiobjective combinatorial optimization. In: *OR Spektrum* 22 (2000), S. 425–460

[Ehrgott u. Tenfelde-Podehl 2003]

EHRGOTT, M.; TENFELDE-PODEHL, D.: Computation of ideal and Nadir values and implications for their use in MCDM methods. In: *European Journal of Operational Research* 151 (2003), S. 119–139

[Farhang-Mehr u. Azarm 2002]

FARHANG-MEHR, A.; AZARM, S.: Diversity Assessment of Pareto Optimal Solution Sets: An Entropy Approach. In: *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02, Honolulu, Hawaii*, IEEE Service Center, Piscataway, NJ, 2002, S. 723–728

[Farhang-Mehr u. Azarm 2003]

FARHANG-MEHR, A.; AZARM, S.: Minimal Sets of Quality Metrics. In: *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003, Faro, Portugal, LNCS 2632*, Springer-Verlag, Berlin, Heidelberg, 2003, S. 405–417

[Farhang-Mehr u. a. 2001]

FARHANG-MEHR, A.; WU, J.; AZARM, S.: Some preliminary results on the development and comparison of a new multi-objective genetic algorithm. In: *Proceedings of DETC'01, ASME 2001 Design Engineering Technical Conferences, Pittsburgh, Pennsylvania*, 2001

[Fleischer 2002]

FLEISCHER, M.: The Measure of Pareto Optima: Applications to Multiobjective Metaheuristics (CSHCN TR 2002-17 / ISR TR 2002-32) / Institute for Systems Research, University of Maryland, College Park, MD. 2002. – Forschungsbericht

[Fleischer 2003]

FLEISCHER, M.: *The Measure of Pareto Optima Applications to Multiobjective Metaheuristics*. Fonseca, C. M. and Fleming, P. J. and Zitzler, E. and Deb, K. and Thiele, L. Springer-Verlag, Berlin, Heidelberg, New York, 2003. – 519–533 S

[Framinan u. a. 2002]

FRAMINAN, J. M.; LEISTEN, R.; RUIZ-USANO, R.: Efficient heuristics for flowshop sequencing with the objectives of makespan and flowtime minimisation. In: *European Journal of Operational Research* 141 (2002), S. 559–569

[Gagne u. a. 2001]

GAGNE, C.; PRICE, W. L.; GRAVEL, M.: Scheduling a single machine with sequence dependent setup time using ant colony optimization (2001-003) / Universität Laval, Quebec, Canada. 2001. – Forschungsbericht

[Gagne u. a. 2002]

GAGNE, C.; PRICE, W. L.; GRAVEL, M.: Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times. In: *Journal of the Operational Research Society* 53 (2002), Nr. 8, S. 895–906

[Gambardella u. Dorigo 1996]

GAMBARDELLA, L. M.; DORIGO, M.: Solving Symmetric and Asymmetric TSPs by Ant Colonies. In: *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC'96), Nagoya University, Japan, 1996*, S. S.622–627

[Gambardella u. a. 1999]

GAMBARDELLA, L. M.; TAILLARD, E.; AGAZZI, G.: MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: *New Ideas in Optimization*. Corne, D. and Dorigo, M. and Glover, F. McGraw-Hill. London, 1999, S. 63–76

[Gangadharan u. Rajendran 1994]

GANGADHARAN, R.; RAJENDRAN, C.: A Simulated Annealing Heuristic for Scheduling in a Flowshop with Bicriteria. In: *Computers & Industrial Engineering* 27 (1994), S. 473–476

[Geoffrion 1968]

GEOFFRION, A. M.: Proper Efficiency and the Theory of Vector Maximization. In: *Journal of Mathematical Analysis and Applications* 22 (1968), S. 618–630

[Goss u. a. 1989]

GOSS, S.; ARON, S.; DENEUBOURG, J. L.; PASTEELS, J. M.: Self-organized Shortcuts in the Argentine Ant. In: *Naturwissenschaften* 76 (1989), S. 579–581

[Gravel u. a. 2002]

GRAVEL, M.; PRICE, W. L.; GAGNE, C.: Scheduling continuous casting of aluminium using a multiple objective ant colony optimization meta-heuristic. In: *European Journal of Operational Research* 143 (2002), S. 218–229

[Günther 1971]

GÜNTHER, H.: *Das Dilemma der Arbeitsablaufplanung*. Erich Schmidt Verlag, Berlin, 1971

[Gutenberg 1951]

GUTENBERG, E.: *Grundlagen der Betriebswirtschaft, 1. Band, Die Produktion*. 2. Aufl. Springer Verlag, Berlin, Göttingen, Heidelberg, 1951

[Hansen u. Jazzkiewicz 1998]

HANSEN, M. P.; JASZKIEWICZ, A.: Evaluating the quality of approximations to the non-dominated set (IMM-REP-1998-7) / Institute of Mathematical Modelling, Technical University of Denmark, 2800 Lyngby, Denmark. 1998. – Forschungsbericht

[Holthaus 1996]

HOLTHAUS, O.: *Ablaufplanung bei Werkstattfertigung*. Dt. Univ.-Verl., Gabler, Wiesbaden, 1996. – Zugl.: Universität Paderborn, Diss., 1996, u.d.T.: Koordinationsregeln für dezentrale Werkstattsteuerungssysteme

[Huckert u. a. 1980]

HUCKERT, K.; RHODE, R.; ROGLIN, O.; WEBER, R.: On the Interactive Solution to a Multicriteria Scheduling Problem. In: *Zeitschrift für Operations Research* 24 (1980), S. 47–60

[Iredi u. a. 2001]

IREDI, S.; MERKLE, D.; MIDDENDORF, M.: *Bi-Criterion Optimization with Multi Colony Ant Algorithms*. Zitzler, E. and Thiele, L. and Deb, K. and Coello Coello, C. A. and Corne, D. Springer-Verlag, Berlin, Heidelberg, New York, 2001. – 359–372 S

[Ishibuchi u. Murata 1998]

ISHIBUCHI, H.; MURATA, T.: A Multi-Objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. In: *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews* 28 (1998), Nr. 3, S. 392–403

[Johnson 1954]

JOHNSON, S. M.: Optimal Two- and Three-Stage Production Schedu-

les with Setup Times included. In: *Naval research logistics quarterly* 1 (1954), Nr. 1, S. 61–68

[Kim u. a. 2001]

KIM, B.; GEL, E. S.; CARLYLE, W. M.; FOWLER, J. W.: A New Technique to Compare Algorithms for Bi-criteria Combinatorial Optimization Problems. In: *Multiple Criteria Decision Making in the New Millenium, Lecture Notes in Economics and Mathematical Systems 507*. Köksalan, M. and Zions, S. Springer-Verlag, Berlin, Heidelberg, 2001, S. 113–123

[Knowles 2002]

KNOWLES, J. D.: *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*, University of Reading, Department of Computer Science, Reading, UK, Diss., 2002

[Knowles u. Corne 2002]

KNOWLES, J. D.; CORNE, D. W.: On Metrics for Comparing Nondominated Sets. In: *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02, Honolulu, Hawaii*, IEEE Service Center, Piscataway, NJ, 2002, S. 711–716

[Krikpatrick u. a. 1983]

KRIKPATRICK, S.; GELATT JR., C. D.; VECCHI, M. P.: Optimization by Simulated Annealing. In: *Science* 220 (1983), Nr. 4598, S. 671–680

[Krüger u. a. 1998]

KRÜGER, F.; MIDDENDORF, M.; MERKLE, D.: *Studies on a Parallel Ant System for the BSP Modell*. 1998. – unveröffentlichtes Manuskript

[Laarhoven, van u. Aarts 1987]

LAARHOVEN, VAN, P. J. M.; AARTS, E. H. L.: *Simulated Annealing: Theory and Applications*. D. Reidel Publ. Co., 1987

[Liu u. Reeves 2001]

LIU, J.; REEVES, C. R.: Constructive and composite heuristic solutions to the $P//\sum C_i$ scheduling problem. In: *European Journal of Operational Research* 132 (2001), S. 439–542

[Loukil u. a. 2005]

LOUKIL, T.; TEGHEM, J.; TUYTTENS, D.: Solving multi-objective production scheduling problems using metaheuristics. In: *European Journal of Operational Research* 161 (2005), S. 42–61. – to appear

[Mensch 1972]

MENSCH, G.: Das Trilemma der Ablaufplanung. In: *Zeitschrift für Betriebswirtschaft* 42 (1972), Nr. 2, S. 77–88

[Merkle 2002]

MERKLE, D.: *Ameisenalgorithmen - Optimierung und Modellierung*, Fakultät für Wirtschaftswissenschaften der Universität Fridericiana zu Karlsruhe, Diss., 2002

[Merkle u. Middendorf 2000]

MERKLE, D.; MIDDENDORF, M.: An Ant Algorithm with a New Pheromone Evaluation Rule for Total Tardiness Problems. In: *Real-World Applications of Evolutionary Computing, Proceedings of the EvoWorkshops 2000, Edinburgh, Schottland, LNCS 1803*, Springer-Verlag, Berlin, Heidelberg, New York, 2000, S. 287–296

[Merkle u. Middendorf 2001]

MERKLE, D.; MIDDENDORF, M.: A New Approach to Solve Permutation Scheduling Problems with Ant Colony Optimization. In: *Applications of Evolutionary Computing, Proceedings of the EvoWorkshops 2001, Como, Italien, LNCS 2037*, Springer-Verlag, Berlin, Heidelberg, New York, 2001, S. 484–494

[Merkle u. Middendorf 2002]

MERKLE, D.; MIDDENDORF, M.: Ant Colony Optimization with the Relative Pheromone Evaluation Method. In: *Applications of Evolutionary Computing, Proceedings of the EvoWorkshops 2002, Kinsale, Ireland, LNCS 2279*, Springer-Verlag, Berlin, Heidelberg, New York, 2002, S. 325–333

[Merkle u. Middendorf 2003]

MERKLE, D.; MIDDENDORF, M.: Ant Colony Optimization with Global Pheromone Evaluation for Scheduling a Single Machine. In: *Applied Intelligence* 18 (2003), Nr. 1, S. 105–111

[Michel u. Middendorf 1999]

MICHEL, R.; MIDDENDORF, M.: An ACO Algorithm for the Shortest Common Supersequence Problem. In: *New Ideas in Optimization*. Corne, D. and Dorigo, M. and Glover, F. McGraw-Hill, London, 1999, S. 51–61

[Middendorf u. a. 2000]

MIDDENDORF, M.; REISCHLE, F.; SCHMECK, H.: Information Exchange

in Multi Colony Ant Algorithms. In: *Parallel and Distributed Processing, Proceedings of the 15 IPDPS 2000 Workshop, Cancun, Mexiko, LNCS 1800*, Springer-Verlag, Berlin, Heidelberg, 2000, S. 645–652

[Middendorf u. a. 2002]

MIDDENDORF, M.; REISCHLE, F.; SCHMECK, H.: Multi Colony Ant Algorithms. In: *Journal of Heuristics* 8 (2002), S. 305–320

[Nagar u. a. 1995]

NAGAR, A.; HADDOCK, J.; HERAGU, S.: Multiple and bicriteria scheduling: A literature survey. In: *European Journal of Operational Research* 81 (1995), S. 88–104

[Nawaz u. a. 1983]

NAWAZ, M.; ENSCORE JR., E.; HAM, I.: Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem. In: *OMEGA, The Int. Jl. of Mgmt. Sci.* 11 (1983), Nr. 1, S. 91–95

[Nelson u. a. 1986]

NELSON, R. T.; SARIN, R. K.; DANIELS, R. L.: Scheduling with multiple performance measures: the one-machine case. In: *Management Science* 32 (1986), Nr. 4, S. 464–479

[Olkin u. Pratt 1958]

OLKIN, I.; PRATT, J. W.: Unbiased estimation of certain correlation coefficients. In: *Annals of Mathematical Statistics* 29 (1958), S. 201–211

[Paulik 1984]

PAULIK, R.: *Kostenorientierte Reihenfolgeplanung in der Werkstattfertigung*. Verlag Paul Haupt, Bern, Stuttgart, 1984

[Pinedo 1995]

PINEDO, M.: *Scheduling: Theory, Algorithms, and Systems*. Englewood Cliffs, New Jersey, 1995

[Rajendran 1993]

RAJENDRAN, C.: Heuristic algorithm for scheduling in a flowshop to minimize total flowtime. In: *International Journal of Production Economics* 29 (1993), S. 65–73

[Rajendran 1994]

RAJENDRAN, C.: A heuristic for scheduling in flowshop and flowline-based manufacturing cell with multi-criteria. In: *International Journal of Production Research* 32 (1994), Nr. 11, S. 2541–2558

[Rajendran 1995]

RAJENDRAN, C.: Heuristics for scheduling in flowshop with multiple objectives. In: *European Journal of Operational Research* 82 (1995), S. 540–555

[Rajendran u. Ziegler 1997]

RAJENDRAN, C.; ZIEGLER, H.: Heuristics for scheduling in a flowshop with setup, processing and removal times separated. In: *Production Planning & Control* 8 (1997), Nr. 6, S. 568–576

[Rajendran u. Ziegler 2002]

RAJENDRAN, C.; ZIEGLER, H.: Two Ant-Colony Algorithms for Minimizing Total Flowtime in Flowshops. In: *Proceedings of the 30th International Conference on Computers and Industrial Engineering, Tinos Island, Griechenland* Bd. 2, 2002, S. 731–736

[Rajendran u. Ziegler 2004]

RAJENDRAN, C.; ZIEGLER, H.: Ant-colony algorithms for permutation flowshop scheduling to minimize makespan / total flowtime of jobs. In: *European Journal of Operational Research* 155 (2004), S. 426–438

[Reeves 1999]

REEVES, C. R.: Landscapes, operators and heuristic search. In: *Annals of Operations Research* 86 (1999), S. 473–490

[Rosenthal 1985]

ROSENTHAL, R. E.: Concepts, Theory, and Techniques Principles of Multiobjective Optimization. In: *Decision Sciences* 16 (1985), S. 133–152

[Schaffer 1985]

SCHAFFER, J. D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: *Proceedings of the first International Conference on Genetic Algorithms and their Applications*, 1985, S. 93–100

[Schaich u. a. 1993]

SCHAICH, E.; KÖHLE, D.; SCHWEITZER, W.; WEGNER, F.: *Statistik I für Volkswirte, Betriebswirte und Soziologen*. Verlag Franz Vahlen München, 1993

[Schweitzer 1967]

SCHWEITZER, M.: Methodologische und entscheidungstheoretische Grundfragen der betriebswirtschaftlichen Prozeßstrukturierung. In: *Zeitschrift für betriebswirtschaftliche Forschung* 19 (1967), S. 279–296

[Seelbach 1975]

SEELBACH, H.: *Ablaufplanung*. Physica-Verlag, Würzburg, Wien, 1975

[Seelbach u. Zimmermann 1973]

SEELBACH, H.; ZIMMERMANN, H.-G.: Quantitative Kapitalbedarfsanalyse. In: *Zeitschrift für Betriebswirtschaft* 43 (1973), Nr. 5, S. 329–350

[Selen u. Hott 1986]

SELEN, W. J.; HOTT, D. D.: A Mixed-Integer Goal-Programming Formulation of the Standard Flow-Shop Scheduling Problem. In: *Journal of the Operational Research Society* 37 (1986), Nr. 12, S. 1121–1128

[Soland 1979]

SOLAND, R. M.: Multicriteria optimization: a general characterization of efficient solutions. In: *Decision Sciences* 10 (1979), S. 26–38

[Srinivas u. Deb 1994]

SRINIVAS, N.; DEB, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. In: *Evolutionary Computation* 2 (1994), Nr. 3, S. 221–248

[Stefenelli 2000]

STEFENELLI, M.: *Kostenorientierte Reihenfolgeplanung durch Verknüpfung von Prioritätsregeln mit lokalen Suchverfahren*. Shaker Verlag, Aachen, 2000. – Zugl.: Universität Passau, Diss., 2000

[Steuer 1986]

STEUER, R. E.: *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapur, 1986

[Stützle 1998a]

STÜTZLE, T.: An Ant Approach to the Flow Shop Problem. In: *Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EU-FIT'98), Aachen, Deutschland* Bd. 3, Verlag Mainz, Aachen, 1998, S. 1560–1564

[Stützle 1998b]

STÜTZLE, T.: Parallelization Strategies for Ant Colony Optimization. In: *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN-V), Berlin, LNCS 1498*, Springer Verlag, Berlin, Heidelberg, New York, 1998, S. 722–731

[Stützle 1999]

STÜTZLE, T.: *Local Search Algorithms for Combinatorial Problems - Analysis, Algorithms, and New Applications*. Infix, Sankt Augustin, 1999. – Zugl.: Techn. Univ. Darmstadt, Diss., 1998

[Stützle u. Hoos 1997]

STÜTZLE, T.; HOOS, H. H.: MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation, ICEC'97, Indianapolis, USA*, IEEE Service Center, Piscataway, 1997, S. 309–314

[Stützle u. Hoos 2000]

STÜTZLE, T.; HOOS, H. H.: MAX-MIN Ant System. In: *Future Generation Computer Systems* 16 (2000), S. 889–914

[Taillard 1993]

TAILLARD, E.: Benchmarks for basic scheduling problems. In: *European Journal of Operational Research* 64 (1993), S. 278–285

[Teich u. a. 2001]

TEICH, T.; FISCHER, M.; VOGEL, A.; FISCHER, J.: A New Ant Colony Algorithm for the Job Shop Scheduling Problem. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, Morgan Kaufmann, 2001, S. 803

[T'Kindt u. Billaut 2001]

T'KINDT, V.; BILLAUT, J.-C.: Multicriteria Scheduling Problems: A Survey. In: *RAIRO Operations Research* 35 (2001), S. 143–163

[T'Kindt u. Billaut 2002]

T'KINDT, V.; BILLAUT, J.-C.: *Multicriteria Scheduling*. Springer-Verlag, Berlin, Heidelberg, New York, 2002

[T'kindt u. a. 2002]

T'KINDT, V.; MONMARCHE, N.; TERCINET, F.; LAUGT, D.: An Ant Colony Optimization algorithm to solve a 2-machine bicriteria flowshop scheduling problem. In: *European Journal of Operational Research* 142 (2002), S. 250–257

[Tuyttens u. a. 2000]

TUYTTENS, D.; TEGHEM, J.; FORTEMPS, P.; VAN NIEUWENHUYZE, K.: Performance of the MOSA Method for the Bicriteria Assignment Problem. In: *Journal of Heuristics* 6 (2000), S. 295–310

[Ulungu u. Teghem 1994]

ULUNGU, E. L.; TEGHEM, J.: Multi-objective Combinatorial Optimization Problems: A Survey. In: *Journal of Multi-Criteria Decision Analysis* 3 (1994), S. 83–104

[Van Veldhuizen 1999]

VAN VELDHUIZEN, D. A.: *Multiobjective Evolutionary Algorithms: Classification, Analyses, and New Innovations*, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, Diss., May 1999

[Van Veldhuizen u. Lamont 2000]

VAN VELDHUIZEN, D. A.; LAMONT, G. B.: Multiobjective Evolutionary Algorithms: Analysing the State-of-the-Art. In: *Evolutionary Computation* 8 (2000), Nr. 2, S. 125–147

[Wu u. Azarm 2001]

WU, J.; AZARM, S.: Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set. In: *Transactions of the ASME* 123 (2001), March, S. 18–25

[Ying u. Liao 2004]

YING, K.-C.; LIAO, C.-L.: An ant colony system for permutation flow-shop sequencing. In: *Computers & Operations Research* 31 (2004), S. 791–801

[Zäpfel 1982]

ZÄPFEL, G.: *Produktionswirtschaft*. Walter de Gruyter, Berlin, New York, 1982

[Zitzler 1999]

ZITZLER, E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Shaker Verlag, Aachen, 1999. – Zugl.: Swiss Federal Institute of Technology Zurich, Diss., ETH Nr. 13398, 1999

[Zitzler u. a. 2000]

ZITZLER, E.; DEB, K.; THIELE, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. In: *Evolutionary Computation* 8 (2000), Nr. 2, S. 173–195

[Zitzler u. Thiele 1998]

ZITZLER, E.; THIELE, L.: Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*

(PPSN-V), Berlin, LNCS 1498, Springer-Verlag, Berlin, Heidelberg, New York, 1998, S. 292–301

Anhang A

Generierung von Testproblemen

In diesem Anhang wird beschrieben, wie kleine Testprobleme mit 4, 6, 8 und 10 Aufträgen für das Flow Shop Problem generiert werden können.

Grundlage bilden die von Taillard (1993) veröffentlichten Benchmarkprobleme.¹ Seine Flow Shop Probleme werden verkleinert, indem den Problemen eine neue Auftrags- und Maschinenzahl zugeordnet wird. Die neue Zuordnung ist in Tabelle A.1 angegeben.

Problemnummern Problem	Taillard		Kleine Testprobleme	
	N	M	N	M
1-10	20	5	4	5
11-20	20	10	4	10
21-30	20	20	4	20
31-40	50	5	6	5
41-50	50	10	6	10
51-60	50	20	6	20
61-70	100	5	8	5
71-80	100	10	8	10
81-90	100	20	8	20
91-100	200	10	10	5
101-110	200	20	10	10
111-120	500	20	10	20

Tabelle A.1: Kleine Testprobleme

Die Bearbeitungszeiten werden anschließend analog zu den Benchmarkproblemen von Taillard (1993) generiert.

¹Vgl. Taillard (1993, S.279-281).

Anhang B

Generierung von Fertigstellungsterminen

In der Arbeit von Taillard (1993) ist die Bestimmung von Bearbeitungszeiten für Instanzen von Flow Shop Problemem beschrieben.¹ Da in dieser Arbeit auch das Ziel der Minimierung der maximalen Terminüberschreitung verfolgt wird, werden zusätzlich Fertigstellungstermine für die Aufträge benötigt. Im Folgenden soll beschrieben werden, wie, sowohl für die Benchmarkprobleme von Taillard (1993) als auch für die kleinen Testprobleme aus Anhang A, diese Fertigstellungstermine generiert werden können.

Grundsätzlich erscheint es sinnvoll, den Fertigungstellungstermin eines Auftrags n derart zu setzen, daß der Auftrag theoretisch pünktlich fertiggestellt werden kann. Es solle also gefordert werden, daß gilt:

$$d_n \geq r_n + \sum_{m=1}^M p_{nm} \quad (\text{B.1})$$

Dabei ist r_n ist der Ankunftszeitpunkt von Auftrag n .² Zu diesem wird die Summe der Bearbeitungszeiten von Auftrag n auf den $m = 1(1)M$ Maschinen addiert.

Es ist jedoch nicht möglich, jeden Auftrag an Position 1 der Auftragsfolge zu fertigen. Daher sollte ein zusätzlicher Zeitraum berücksichtigt werden. Es erscheint sinnvoll, diesen Zeitraum von der Struktur der Bearbeitungszeiten der $n = 1(1)N$ Aufträge abhängig zu machen. So soll verhindert werden, daß die Termine grundsätzlich zu großzügig oder zu knapp gesetzt werden. Würden die Fertigstellungstermine zu großzügig gesetzt,

¹Vgl. Taillard (1993, S.279-281).

²In dieser Arbeit gilt: $r_n = 0$; $n = 1(1)N$.

so kann es passieren, daß immer alle Aufträge, unabhängig von der Auftragsfolge, pünktlich fertiggestellt werden. Wird, wie in dieser Arbeit, die Minimierung der maximalen Terminüberschreitung betrachtet, so wäre es dann z.B nicht mehr möglich, den Einfluß unterschiedlicher Verfahrensparameter auf die Optimierungseigenschaft zu analysieren, da T_{max} für alle $N!$ Auftragsfolgen Null wäre.

Für jeden Auftrag eines Problems wird der Fertigstellungstermin nach folgender Gleichung bestimmt:

$$d_n = r_n + \sum_{m=1}^M p_{nm} + U[0; 1] \cdot LB^{Cmax} \quad (\text{B.2})$$

$U[0;1]$ steht dabei für eine gleichverteilte Zufallszahl zwischen Null und Eins. Zur Ermittlung der Zufallszahl wird der von Taillard (1993) beschriebene Zufallszahlengenerator verwendet.³ Initialisiert wird der Zufallszahlenstrom mit dem gleichen Startwert, welcher auch für die Generierung der Bearbeitungszeiten verwendet wird. Die Größe LB^{Cmax} bezeichnet die untere Grenze für die Zykluszeit. Ihre Berechnung wird wie in Taillard (1993) beschrieben vorgenommen.⁴ Durch die Verwendung von LB^{Cmax} soll erreicht werden, daß die zusätzlich zu $r_n + \sum_{m=1}^M p_{nm}$ zu berücksichtigende Zeitspanne in Abhängigkeit von der im Problem vorliegenden Struktur der Bearbeitungszeiten bestimmt wird.⁵

³Vgl. Taillard (1993, S.279).

⁴Vgl. Taillard (1993, S.281).

⁵In Holthaus (1996) werden weitere Methoden zur Vergabe von Fertigstellungsterminen vorgestellt (vgl. Holthaus (1996, S.67-70) sowie die dort angegebene Literatur).

Anhang C

Verwendete kleinste obere Schranken für C_{max} , F_{mid} , und T_{max}

In diesem Anhang werden die zur Beurteilung von Lösungen mit Hilfe des DUB^k -Maßes verwendeten kleinsten oberen Schranken der ersten 90 Taillardbenchmarks für die drei im Zielsystem enthaltenen Ziele angegeben.

Die kleinsten oberen Schranken für die drei Zielgrößen werden aus unterschiedlichen Quellen entnommen. Bei allen drei Zielgrößen wurden die Ergebnisse der MOGLS-Heuristik von Ishibuchi u. Murata (1998) und der GPWGA-Heuristik von Chang u. a. (2002) berücksichtigt.¹ Diese beiden Heuristiken wurden, wie in der Literatur angegeben, in C++ implementiert. Anschließend wurden mit jeder Heuristik für jedes der 90 Taillardbenchmarkprobleme 10 heuristisch effiziente Mengen bestimmt. Die Parameter der Heuristik wurden dabei gemäß Tabelle C.1 eingestellt.

Für die Ermittlung der kleinsten oberen Schranken der einzelnen Zielgrößen werden die von einer Heuristik für eine Probleminstanz jeweils kleinsten auftretenden Zielfunktionswerte in den 10 generierten heuristisch effizienten Fronten berücksichtigt.

Für die Ermittlung der kleinsten oberen Schranken der Zykluszeit werden neben den von MOGLS und GPWGA ermittelten Werten die in Tail-

¹Vgl. für die Heuristiken Ishibuchi u. Murata (1998, S.392-398) und Chang u. a. (2002, S.6-8).

Parameter	GPWGA	MOGLS
Anzahl Elitelösungen (N_{elite}):	3	3
Populationsgröße (N_{pop}):	20	20
Crossoverwahrscheinlichkeit (P_C):	90%	90%
Mutationswahrscheinlichkeit (P_M):	30%	30%
Tiefe Lokale Suche:	2	2
Abbruchbedingung	5000 Generationen	Gleiche Rechenzeit wie bei GPWGA

Tabelle C.1: Parametereinstellungen MOGLS und GPWGA

lard (1993) angegebenen kleinsten oberen Schranken berücksichtigt.² Bei der mittleren Durchlaufzeit werden zusätzlich die von Rajendran u. Ziegler (2004) veröffentlichten Ergebnisse der MSACO- und PACO-Heuristik, sowie die Ergebnisse von Liu u. Reeves (2001) berücksichtigt.³ Die kleinsten oberen Schranken für die maximale Terminüberschreitung werden allein aus den Ergebnissen der GPWGA- und MOGLS-Heuristik bestimmt.

Die ermittelten Werte der kleinsten oberen Schranken sind in den Tabellen C.2 bis C.4 angegeben. Hinter den Werten ist zusätzlich angegeben, aus welcher Quelle oder Heuristik der Wert entnommen bzw. ermittelt wurde. Die Zuordnung der Buchstaben zur Quelle bzw. Heuristik wird wie folgt vorgenommen:

Buchstabe	Quelle bzw. Heuristik
A	Taillard (1993)
B	MOGLS, Ishibuchi u. Murata (1998)
C	GPWGA, Chang u. a. (2002)
D	MSACO, Rajendran u. Ziegler (2004)
E	PACO, Rajendran u. Ziegler (2004)
F	Liu u. Reeves (2001)

²Vgl. Taillard (1993, S.280).

³Vgl. Rajendran u. Ziegler (2004, S.436) und Liu u. Reeves (2001, S.449-451).

Nr.	N	M	Seed	$z_{C_{max}}^{UB}$		$z_{F_{mid}}^{UB}$		$z_{T_{max}}^{UB}$	
1	20	5	873653221	1,278	A	702,05	B	549	B
2	20	5	379008056	1.359	A,B	757,55	D	498	B
3	20	5	1866992158	1.081	A	670,65	E	677	C
4	20	5	216771124	1.293	A	774,30	D	543	B
5	20	5	495070989	1.235	A	676,45	D,E	524	C
6	20	5	402959317	1.195	A	656,95	D	656	C
7	20	5	1369363414	1.239	A	677,95	D	587	B
8	20	5	2021925980	1.206	A	698,40	D	652	C
9	20	5	573109518	1.230	A	715,85	D	697	B
10	20	5	88325120	1.108	A	648,40	D	573	B
11	20	10	587595453	1.582	A	1.047,90	E	861	B
12	20	10	1401007982	1.659	A	1.122,00	D	892	C
13	20	10	873136276	1.496	A	991,65	D	529	C
14	20	10	268827376	1.377	A	936,20	D	659	B
15	20	10	1634173168	1.419	A	932,20	D	929	C
16	20	10	691823909	1.397	A	962,25	D,E	1.018	C
17	20	10	73807235	1.484	A	918,80	D	1.059	B
18	20	10	1273398721	1.538	A	1.012,05	D	657	B
19	20	10	2065119309	1.593	A	1.016,50	D	814	B
20	20	10	1672900551	1.591	A	1.066,00	D	746	B
21	20	20	479340445	2.297	A	1.681,15	D,E	1.257	B
22	20	20	268827376	2.099	A	1.580,20	D	826	B
23	20	20	1958948863	2.326	A	1.696,00	D	1.153	B
24	20	20	918272953	2.223	A	1.584,90	D	926	B
25	20	20	555010963	2.291	A	1.729,65	D,E	394	C
26	20	20	2010851491	2.226	A	1.631,85	D	1.046	B
27	20	20	1519833303	2.273	A	1.651,90	D	810	B
28	20	20	1748670931	2.200	A	1.622,20	D	843	B
29	20	20	1923497586	2.237	A	1.681,25	D	686	B
30	20	20	1829909967	2.178	A	1.615,75	E	763	B

Tabelle C.2: Kleinste obere Schranken: TAIL001-TAIL030

Nr.	N	M	Seed	$z_{C_{max}}^{UB}$		$z_{F_{mid}}^{UB}$		$z_{T_{max}}^{UB}$	
31	50	5	1328042058	2.724	A	1.308,56	E	1.797	C
32	50	5	200382020	2.836	A	1.373,28	F	2.715	C
33	50	5	496319842	2.621	A	1.283,32	D	1.773	B
34	50	5	1203030903	2.751	A	1.382,26	D	1.867	B
35	50	5	1730708564	2.863	A	1.406,62	D	2.390	C
36	50	5	450926852	2.829	A	1.351,26	D	2.316	C
37	50	5	1303135678	2.725	A	1.340,28	D	1.394	B
38	50	5	1273398721	2.683	A	1.297,26	D	1.876	B
39	50	5	587288402	2.554	A	1.274,70	D	2.298	B
40	50	5	248421594	2.782	A,C	1.405,12	F	2.563	C
41	50	10	1958948863	3.037	A	1.775,40	D	2.202	B
42	50	10	575633267	2.911	A	1.672,24	E	2.167	B
43	50	10	655816003	2.871	A	1.629,28	E	1.856	B
44	50	10	1977864101	3.067	A	1.751,58	E	1.929	B
45	50	10	93805469	3.011	A	1.766,46	E	2.209	B
46	50	10	1803345551	3.021	A	1.760,22	E	1.905	B
47	50	10	49612559	3.124	A	1.799,12	E	1.947	B
48	50	10	1899802599	3.048	A	1.766,60	E	2.149	B
49	50	10	2013025619	2.910	A	1.739,50	D	2.086	B
50	50	10	578962478	3.100	A	1.786,64	E	2.344	B
51	50	20	1539989115	3.886	A	2.537,48	E	2.620	C
52	50	20	691823909	3.733	A	2.424,16	D	2.700	B
53	50	20	655816003	3.673	A	2.343,36	E	2.246	B
54	50	20	1315102446	3.755	A	2.458,12	E	2.853	B
55	50	20	1949668355	3.648	A	2.398,40	D	2.667	B
56	50	20	1923497586	3.719	A	2.447,38	E	2.203	B
57	50	20	1805594913	3.730	A	2.506,34	E	2.675	C
58	50	20	1861070898	3.737	A	2.490,86	D	2.613	B
59	50	20	715643788	3.772	A	2.476,84	E	1.839	B
60	50	20	464843328	3.791	A	2.524,06	E	2.169	C

Tabelle C.3: Kleinste obere Schranken: TAIL031-TAIL060

Nr.	N	M	Seed	$z_{C_{max}}^{UB}$		$z_{F_{mid}}^{UB}$		$z_{T_{max}}^{UB}$	
61	100	5	896678084	5.493	A	2.567,89	F	4.417	B
62	100	5	1179439976	5.274	A	2.456,09	F	4.459	B
63	100	5	1122278347	5.175	A	2.405,37	D	4.782	B
64	100	5	416756875	5.018	A	2.304,80	D	3.743	B
65	100	5	267829958	5.250	A	2.430,13	D	4.045	B
66	100	5	1835213917	5.135	A	2.357,93	F	4.485	B
67	100	5	1328833962	5.247	A	2.437,41	F	4.489	B
68	100	5	1418570761	5.094	A	2.344,53	E	4.049	B
69	100	5	161033112	5.448	A	2.512,91	F	4.770	B
70	100	5	304212574	5.328	A	2.462,61	D	4.026	B
71	100	10	1539989115	5.776	A	3.039,10	E	4.803	B
72	100	10	655816003	5.362	A	2.785,96	E	4.888	B
73	100	10	960914243	5.679	A	2.953,89	E	4.826	B
74	100	10	1915696806	5.820	A	3.068,50	E	5.091	B
75	100	10	2013025619	5.491	A	2.890,60	E	4.537	B
76	100	10	1168140026	5.308	A	2.753,84	E	4.714	C
77	100	10	192397586	5.600	A	2.858,58	E	5.039	B
78	100	10	167698528	5.640	A	2.961,30	F	4.518	B
79	100	10	1528387973	5.891	A	3.096,64	D	4.174	B
80	100	10	993794175	5.856	A	2.964,62	E	4.523	B
81	100	20	450926852	6.330	A	3.721,56	E	4.654	B
82	100	20	1462772409	6.320	A	3.806,44	E	5.003	C
83	100	20	1021685265	6.364	A	3.775,79	E	4.891	C
84	100	20	83696007	6.331	A	3.802,01	D	5.158	B
85	100	20	508154254	6.405	A	3.772,68	D	4.374	B
86	100	20	1861070898	6.487	A	3.788,57	E	4.641	B
87	100	20	26482542	6.379	A	3.815,31	E	5.133	B
88	100	20	444956424	6.514	A	3.926,97	E	5.364	B
89	100	20	2115448041	6.386	A	3.825,52	E	5.037	B
90	100	20	118254244	6.534	A	3.872,82	E	4.952	B

Tabelle C.4: Kleinste obere Schranken: TAIL061-TAIL090

Lebenslauf

Persönliche Daten:

Name: Christian Petri
Geburtstag: 10.03.1975
Geburtstort: Giessen

Schulbildung:

1981 - 1987 Grundschole und Förderstufe der
Ludwig-Uhland Schule Giessen

1987 - 1994 Gymnasium der Liebigschule Giessen

Wehrdienst:

07/1994-06/1995 Sanitätssoldat am Bundeswehrkrankenhaus in
Giessen

Ausbildung und Studium:

08/1995-06/1997 Ausbildung zum Industriekaufmann

10/1997-06/2002 Studium der Betriebswirtschaftslehre
an der Universität Passau

07/2002-12/2006 Promotion in Betriebswirtschaftslehre im
Fachbereich Produktion und Logistik
an der Universität Passau