

Ein gemischt-ganzzahliger Ansatz zur gleichmäßigen Verkehrsauslastung eines Ballungsraums mittels Verschiebung der Schulanfangszeiten

Yvonne Meier

BACHELORARBEIT

Ein gemischt-ganzzahliger Ansatz zur gleichmäßigen Verkehrsauslastung eines Ballungsraums mittels Verschiebung der Schulanfangszeiten

VORGELEGT VON:

YVONNE MEIER

STUDIENGANG: WIRTSCHAFTSMATHEMATIK

MATRIKELNUMMER: 3445456

16.MAI 2018

FAKULTÄT FÜR MATHEMATIK, NATURWISSENSCHAFTEN UND INFORMATIK,
FACHGEBIET INGENIEURMATHEMATIK UND NUMERIK DER OPTIMIERUNG

BETREUUNG: PROF. DR. RER. NAT. HABIL. ARMIN FÜGENSCHUH
ZWEITGUTACHTEN: PROF. DR. RER. NAT. HABIL. EKKEHARD KÖHLER

Inhaltsverzeichnis

Abbildungsverzeichnis	4
Tabellenverzeichnis	4
1. Einleitung	5
2. Forschungsstand	7
3. Grundlagen	11
3.1. Optimierungsrechnung	11
3.2. Gemischt-Ganzzahlige Optimierung	12
3.3. CPLEX	13
3.3.1. Presolving	13
3.3.2. Branch-and-Bound-Verfahren	14
3.3.3. Schnittebenenverfahren von Gomory	16
3.3.4. Parallel Solving	18
3.4. Multikriterielle Optimierung und Pareto-Optimierung	19
4. Modell	23
4.1. Vorgehensweise und Annahmen	23
4.2. Generierung von Instanzen	24
4.3. Modellbeschreibung	25
4.3.1. Modell 1	30
4.3.2. Modell 2	32
4.3.3. Modell 3	34
5. Ergebnisse und Analyse	35
5.1. Ergebnisse	35
5.2. Analyse der Ergebnisse	37
5.3. Analyse mittels Pareto-Optimierung	39
5.4. Vergleich mit dem Modell von Wenzel	44
6. Diskussion	49
7. Fazit	51
8. Literaturverzeichnis	52
A. Optimale Schulanfangszeiten	54
B. Hinweise zum digitalen Anhang	58

Abbildungsverzeichnis

3.1.	Problemverzweigung beim Branch&Bound-Verfahren	15
3.2.	Lösungsbeziehungen eines bikriteriellen Optimierungsproblems.	20
3.3.	Lösungsraum und Pareto-Front eines Minimierungsproblems mit zwei Ziel- funktionen.	21
4.1.	Aufteilungsschema einer Instanz	25
5.1.	Gesamtsystembelastung mit und ohne Verschiebung der Schulanfangszeiten	38
5.2.	Modell 1, Instanz 5: Pareto-Fronten bei variierender Pixelgröße	41
5.3.	Modell 2, Instanz 5: Pareto-Fronten bei variierender Pixelgröße	43

Tabellenverzeichnis

5.1.	Übersicht der Parameter der Instanzen.	35
5.2.	Problemgrößen der Instanzen.	36
5.3.	Laufzeiten und Zielfunktionswerte der Instanzen.	36
5.4.	Belastungswerte der Instanzen vor und nach Verschiebung der Schulan- fangszeiten.	39
5.5.	Modell 3: Problemgröße, Laufzeit und Zielfunktionswerte der Instanzen. . .	46
5.6.	Vergleich der Zielfunktionswerte für die verschiedenen Lösungen.	47
5.7.	Vergleich der Zielfunktionswerte für die verschiedenen Lösungen.	48
A.1.	Instanz 1: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	54
A.2.	Instanz 2: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	54
A.3.	Instanz 3: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	54
A.4.	Instanz 4: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	55
A.5.	Instanz 5: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	55
A.6.	Instanz 6: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	55
A.7.	Instanz 7: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	56
A.8.	Instanz 8: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.	57

1. Einleitung

Die vorliegende Arbeit beschäftigt sich mit der gleichmäßigen Auslastung eines Ballungsraums durch die Verschiebung der Schulanfangszeiten. Ein solcher Ballungsraum entsteht immer dann, wenn sich viele Menschen gleichzeitig, meist nah beieinander, innerhalb eines Systems bewegen. Insbesondere entsteht ein solcher Ballungsraum bei einem sogenannten *Schülertransportproblem*, d.h. der Aufgabe, Schüler innerhalb eines Systems zu einer bestimmten Zeit zu ihrer Schule zu transportieren. Die Staffelung der Schulanfangszeiten bietet eine Möglichkeit, dieses hohe Personenaufkommen gezielt zu kontrollieren und zu optimieren.

Die gleichmäßigere Auslastung eines Systems beeinflusst auch die Belastung im öffentlichen Nahverkehr. Im Gegensatz zur Schaffung neuer Alternativen können so die vorhandenen Kapazitäten effizienter eingesetzt werden, ohne das Angebot zu reduzieren. Durch das geringere punktuelle Aufkommen von Fahrgästen kann der zur Verfügung stehende Fuhrpark wirksamer genutzt werden. Zusätzlich machen die zu den Stoßzeiten entlasteten Busse und Bahnen den öffentlichen Personennahverkehr für andere Zielgruppen attraktiver.

Im Rahmen dieser Arbeit wurde ein gemischt-ganzzahliger Ansatz zur Realisation dieser Aufgabe entwickelt. In Kapitel 1 wird zunächst ein Überblick über verschiedene Betrachtungsweisen und Modelle für Schülertransporte und der Optimierung von Personenaufkommen gegeben, Kapitel 2 widmet sich den notwendigen mathematischen Grundlagen, die für den Ansatz wichtig sind. Im Anschluss werden die Modelle konzipiert und erläutert. Die Ergebnisse werden dargestellt und analysiert, worauf abschließend ein Ausblick über die behandelte Problematik folgt.

2. Forschungsstand

In der Literatur gibt es eine Reihe von Modellen und Ansätzen zur Optimierung von Schülertransporten. Dabei werden die Schulanfangszeiten jedoch oft nicht veränderbar, sondern als gegebene Parameter in das Modell eingearbeitet und andere Faktoren wie z.B. die Lage von Haltestellen oder die Route der Busse werden als Variablen genutzt. Bei einer solchen Betrachtung des Problems handelt es sich meist um ein sogenanntes *Fahrzeugzuteilungsproblem (vehicle routing problem)*, im Nachfolgenden VRP genannt. Dabei soll ein zur Verfügung stehender Fuhrpark (*vehicles*) effizient für den Transport der Kunden oder Produkte eingesetzt werden, um dadurch die Zielfunktion, z.B. die Betriebskosten oder die Gesamtfahrstrecke, zu minimieren. Gleichzeitig sind eine Reihe von Nebenbedingungen wie z.B. Anzahl der Fahrzeuge, Zeitbeschränkungen, Personalkapazität, Kundenzufriedenheit zu berücksichtigen [5, 10]. Genauer handelt es sich damit meist um ein *school bus routing problem (SBRP)*, eine spezielle Form eines VRP [4].

Da eine sehr detaillierte Betrachtung dieses Problems den Rahmen dieser Arbeit überschreiten würde, ist das hier beschriebene Modell vereinfacht und daher nicht direkt als VRP einzuordnen. Es bietet jedoch eine Basis, um zu einem solchen ergänzt zu werden. Die in der Literatur behandelten Modelle sind größtenteils so realitätsnah wie möglich gestaltet. Viele Störfaktoren, wie z.B. Verspätungen, Wenden des Busses, Ein- und Aussteigezeiten werden, berücksichtigt. Aufgrund der Komplexität des Problems werden dennoch in jedem Modell vereinfachende Annahmen getroffen. Trotz dieser Annahmen können häufig nur heuristische Verfahren herangezogen werden, um sich einer optimalen Lösung anzunähern [10, 16].

Erstmalig beschäftigen sich Dantzig und Ramser [7] neben der Theorie auch mit einer praktischen Anwendung eines Fahrzeugzuteilungsproblems, der Belieferung von Tankstellen mit Benzin, und geben ein erstes Lösungsverfahren an [10].

1967 bringt Vickrey [21] ein Konzept auf, um den Verkehr insbesondere im Bereich sogenannter *Bottlenecks*, also Engpässen eines Verkehrsnetzes, zu regulieren. Dazu schlägt er Gebühren auf vielbefahrenen Straßen vor, die im Verhältnis zur aktuellen Verkehrslage bzw. zu der Verspätung, die ein Fahrzeug den übrigen Verkehrsteilnehmern einbringt, wenn sich der Fahrer für diese Route entscheidet, richten. Die Gebühren sind zeit- bzw. zustandsabhängig, d.h. sie werden nur dann erhoben, wenn sich der Engpass gerade tatsächlich in einem kritischen Zustand befindet, und nicht jederzeit. Das Modell ist somit ein dynamisches Modell. Für Verkehrsabschnitte, die keinen punktuellen Engpass haben, son-

dern generell überlastet sind, verwendet Vickrey dieselbe Vorgehensweise. Abschließend diskutiert er die praktische Umsetzung der Gebühren. Die Einnahmen aus den Verkehrsüberlastungsgebühren könnten aber z.B. auch zur Senkung anderer Abgaben dienen, wie z.B. der Benzinsteuer.

In [8] beschäftigen sich De Palma und Lefèvre ebenfalls mit diesem Problem. Sie stellen ein stochastisch-dynamisches und ein deterministisch-dynamisches Modell vor. Zudem unterscheiden sie zwischen kleinen und großen Transportteilnehmern („*small and large transport agents*“ ([8], S.2)) mithilfe eines Markov-Modells, d.h. Transportteilnehmer, die Einfluss auf den Stau-Effekt (*large*) haben, und solche, die keinen Effekt darauf haben (*small*). Im Gegensatz zum deterministischen Modell garantiert das stochastische Modell nach De Palma und Lefèvre auch im Falle von Kombinationen aus kleinen und großen Fahrzeugen eine Lösung. Zudem ist es realitätsnäher und flexibler als das deterministische Modell.

Bögl et al. [3] stellen verschiedene Modelle vor, die auf unterschiedlichen Betrachtungen des Problems beruhen. Im Fokus der Arbeit steht besonders der Zusammenhang zwischen der Optimierung der Gesamtkosten und zulässigen Transfers, d.h. dem Umsteigen von Schülern. Die Ergebnisse des heuristisch gelösten Modells vergleichen sie abschließend mit den Ergebnissen von Modellen ohne Transfermöglichkeit. Dabei kommen sie zu dem Erkenntnis, dass das Modell mit Transfers bei vergleichbarer Fahrtzeit der Schüler die Betriebskosten signifikant niedriger hält als in den Modellen ohne Transfermöglichkeit. Um genauere Aussagen treffen zu können, sei jedoch ein mehrkriterieller Ansatz, d.h. ein Ansatz mit mehreren Zielfunktionen, nötig.

Mit der Optimierung für ländlichere Gegenden befassen sich Chen et al. [6]. Faktoren, die diese von größeren Städten unterscheiden, sind etwa die geringere Bevölkerungsdichte, größere Distanzen bei Routen, weniger Schüler pro Haltestelle oder die Tatsache, dass es kaum Einbahnstraßen gibt. Aufgrund dieser Faktoren sind SBRPs für ländliche und städtische Gegenden unterschiedlich zu modellieren. Das dort vorgestellte Lösungsverfahren besteht aus zwei Teilprogrammen: Einem sogenannten *MapInfo-Programm* sowie einem *Routing-Programm*. Zunächst werden für die gegebene Instanz beim MapInfo-Programm kürzeste Wege mithilfe des Dijkstra-Algorithmus berechnet. Diese werden dann im zweiten Schritt, dem Routing-Programm, kombiniert und optimiert. Bei Anwendung auf den ländlichen Schulbezirk Choctaw County konnte dieses Verfahren bei gleicher zurückgelegter Strecke den Bedarf an Bussen um 10% verringern.

Ein stärker vereinfachtes Modell betrachten Schittekat et al. [19]. Es beinhaltet lediglich eine Schule sowie einen Fahrzeugtyp für die Busse mit vorgeschriebener Kapazität. Ein weiterer Unterschied zu den meisten Fahrzeugzuordnungsproblemen ist, dass nicht alle gegebenen Haltestellen befahren werden müssen, sondern nur eine Auswahl aus potentiell-

len Haltestellen getroffen wird. Der entwickelte Instanzengenerator erzeugt einen Pool von potentiellen Haltestellen sowie Schüler mit dazugehörigem Wohnort. Aus den möglichen Haltestellen soll dann eine Teilmenge ausgewählt werden und jedem Schüler wird genau eine Haltestelle zugeordnet, falls ihm mehrere zur Auswahl stehen. Anschließend werden aus den ausgewählten Haltestellen Routen gebildet, sodass die Gesamtdistanz aller Touren minimal wird. Der exakte Algorithmus, welcher iterativ unzulässige Touren ausschließt, wird dann auf einige Instanzen mittlerer Größe angewandt, aufgrund der Laufzeit sind größere Instanzen unpraktikabel. Dieses SBRP wird erneut in [20] behandelt. Der Grundaufbau bleibt dabei derselbe. Die Instanzen werden darin jedoch mit einer Metaheuristik gelöst, um auch größere Instanzen (z.B. 80 mögliche Haltestellen, 800 Schüler) zuzulassen.

Ein Modell zur Optimierung von Fußgängeraufkommen bearbeiten Huang und Lam [13]. Sie betonen die Wichtigkeit der Anpassung eines Modells auf die geplante Einsatzregion. Empirische Untersuchungen ergaben, dass das Verhalten europäischer Fußgänger sich deutlich von dem asiatischer unterscheidet. Für das nichtlineare Modell wird die Stadt oder Region, hier Hong Kong, als Netzwerk mit Kapazitäten modelliert. Zusätzlich sind Matrizen gegeben, die in Abhängigkeit der Zeit für jedes Start-Ziel-Tupel die zugehörige Abreisezeit angeben. Dazu wird der zu untersuchende Zeithorizont in 6-Minuten-Intervalle, was den Abfahrtszeitintervallen der Bahnen in Hong Kong entspricht, eingeteilt. Das Modell bezieht sowohl die Routen- als auch die Zeitentscheidung (*route choice/departure time choice*) mit ein. Das nichtlineare Modell bringt mittels einer Heuristik eine optimale Lösung hervor.

Die Entlastung und Optimierung des Nahverkehrs muss jedoch nicht zwingend durch die Modellierung von Routen geschehen, das Problem lässt sich abstrahieren. Die Optimierung von Schulanfangszeiten hängt eng mit dem SBRP zusammen und sollte nach Möglichkeit mit in das Modell eingebunden werden [12]. Bisher taucht diese Möglichkeit jedoch nur in einer Minderheit der einschlägigen Literatur auf, eine Übersicht der Modelle und ihren Eigenschaften ist in [17] zu finden.

In [11] greift Fügenschuh dies auf und stellt ein gemischt-ganzzahliges lineares Modell vor, das die Schulanfangszeiten als Variablen statt als Parameter enthält. Busrouten und Haltestellen sind hingegen fest vorgegeben, Umsteigen von Schülern in andere Busse ist erlaubt. Zum Lösen des VRPTW (*vehicle routing problem with time windows*) am Beispiel einiger realer Instanzen werden unter anderem Branch-and-Cut Techniken angewandt. Im Vergleich zu den Ist-Zuständen reduziert sich die Anzahl benötigter Busse mit den damit erhaltenen Lösungen um bis zu 25%.

Wenzel [22] beschäftigt sich ebenfalls mit der Entlastung des Nahverkehrs durch die Anpassung von Schulanfangszeiten. Als Gütekriterium bzw. Maß führt er dazu eine Spannung zwischen je zwei Schulen ein, welche sich aus deren Distanz, dem Absolutbetrag des

zeitlichen Unterschiedes der Unterrichtsbeginnzeiten und der Anzahl der Schüler, die zu den betreffenden Schulen fahren, ergibt. Auf Einbringung von Busrouten oder Haltestellen verzichtet er. Anschließend stellt er drei Modelle vor und wendet diese auf Daten der Stadt Nürnberg an. Aus Laufzeitgründen verwendet er zur Lösung ein Dekompositionsverfahren, bei dem die Stadt in kreisförmige Teile zerlegt wird. Durch die Modelle können Reduzierungen der Spannung des Systems von bis zu 50% vermerkt werden.

Nicht nur die Entscheidung über Variablen und Parameter unterscheidet Modelle voneinander, auch die Wahl der Zielfunktion wirkt sich auf den Aufbau sowie die Lösung solcher aus. Die zu minimierenden Größen sind in vielen Modellen die Betriebskosten, die zurückgelegte Strecke bzw. die Gesamtfahrtzeit aller Busse oder die Anzahl der benötigten Busse [17]. Diese Kriterien liegen in erster Linie im Sinne der Busunternehmer. Mehrere Zielfunktionen und deren Zusammenspiel werden z.B. in [16] und [4] untersucht. Die Zielfunktionen können dabei im Widerspruch zueinander stehen, weshalb deren Gewichtung die Lösung maßgeblich beeinflusst (siehe Abschnitt 3.4). Keller und Müller [16] erarbeiten mehrere Modelle und insgesamt vier Zielfunktionen, die betrachtet werden: „Aus der Sicht der Gemeinde soll die gesamte Schulwegzeit der Schüler minimiert werden [...]. Die Schüler möchten so spät wie möglich zu Hause losgehen, sodass die gesamte Busfahrtzeit minimiert werden soll [...]. Als Ziel des Busunternehmens sollten die gesamten Fahrtkosten und damit die Streckenlänge minimiert werden [...]“([16], S.8). Eine Lösung, die für alle Kriterien optimal ist, existiert nicht. Infolge des hohen Rechenaufwandes für den verwendeten exakten Algorithmus wird auf Heuristiken verwiesen.

Aufgrund des unterschiedlichen Komplexitätsgrades bzw. der abweichenden Problemformulierung der in der Literatur vorhandenen Modelle, kann aus diesen kein Lösungsverfahren übertragen werden. Insbesondere dienen die Schulanfangszeiten in den meisten Modellen nicht als Variablen, sondern als Parameter. Ein vergleichbarer Detailgrad der Modelle in [22] erlaubt eine Gegenüberstellung der Ergebnisse und Laufzeiten mit den hier vorgestellten Modellen, siehe dazu Abschnitt 5.4.

3. Grundlagen

Die Erstellung eines Modells für das Problem basiert auf der Theorie der Optimierungsrechnung bzw. des Operations Research. Die für das hier behandelte Problem relevanten Teilgebiete werden im Folgenden grob erläutert. Im Anschluss widmet sich Abschnitt 3.2 der Problemklasse der gemischt-ganzzahligen Optimierungsprobleme, da zur Optimierung der Auslastung innerhalb eines Ballungsgebietes im Rahmen dieser Arbeit ein solcher Ansatz entwickelt wurde. Zur Erstellung und Lösung des Modells wird der Solver IBM ILOG CPLEX Optimization Studio 12.7.1 verwendet, welcher bestimmte Lösungsverfahren und Algorithmen anwendet, die in Abschnitt 3.3 vorgestellt werden.

3.1. Optimierungsrechnung

Die Optimierungsrechnung dient dazu, für ein gegebenes System eine Verbesserung, im besten Fall das Optimum (lat. *optimus* - das Beste) zu finden. Das Optimum bezieht sich dabei auf die verwendete Zielfunktion z , diese ist je nach Modell zu minimieren oder zu maximieren unter Beachtung der Nebenbedingungen. Optimierungsfähige Situationen treten in nahezu allen Bereichen auf, sei es die Wegminimierung eines Kuriers, die Gewinnmaximierung eines Unternehmens oder die Kostenminimierung innerhalb eines Herstellungsprozesses. Ein Minimierungsproblem lässt sich ganz allgemein wie folgt beschreiben:

Definition 3.1.

$$(P) \quad \min z = f(x), \tag{3.1}$$

$$\text{bzgl. } x = (x_1, x_2, \dots, x_n)^T \in G, \tag{3.2}$$

wobei $f : \mathbb{R}^n \rightarrow \mathbb{R}$ die Zielfunktion des Problems ist, und $G \subseteq \mathbb{R}^n$ den zulässigen Bereich beschreibt. [9]

Definition 3.2. Ein Punkt $x \in G$ heißt *zulässiger Punkt* des Optimierungsproblems (P).

Definition 3.3. Ein Punkt $x^* \in G$ heißt *Optimalpunkt* oder *Lösung* von (P), falls $f(x^*) \leq f(x)$ für alle $x \in G$.

Definition 3.4. Ein Problem (P') : $\min g(x)$ bzgl. $x \in G' \subseteq \mathbb{R}^n$ heißt *Relaxation zum Problem* (P) : $\min f(x)$ bzgl. $x \in G \subseteq \mathbb{R}^n$, falls gilt:

$$(i) \quad G \subseteq G', \quad (3.3)$$

$$(ii) \quad g(x) \leq f(x), \quad \forall x \in G. \quad (3.4)$$

Je nach Gestalt der Zielfunktion und der Nebenbedingungen bzw. des zulässigen Bereichs ergibt sich die Klasse, der ein Optimierungsproblem zugeordnet werden kann. Klassen von Optimierungsproblemen sind z.B. lineare Optimierung, ganzzahlige Optimierung, gemischt-ganzzahlige Optimierung, quadratische Optimierung oder konvexe Optimierung [9].

3.2. Gemischt-Ganzzahlige Optimierung

Bei der gemischt-ganzzahligen Optimierung (*mixed integer programming (MIP)*), um die es sich bei der Betrachtung des Problems auf die hier vorgestellte Weise handelt, wird im Gegensatz zur ganzzahligen Optimierung die Ganzzahligkeit nicht für alle Variablen, sondern nur für einen Teil dieser gefordert. Für die optimale Beschreibung von Entscheidungsvariablen, also Variablen, die nicht eine Menge (Produktionsmenge, Schadstoffausstoß etc.), sondern eine Entscheidung (Eignung eines Standorts, Linienzuordnung von Bussen, Herstellung eines Produktes) repräsentieren, reicht die Einschränkung auf die Ganzzahligkeit oft nicht aus, sodass Binärvariablen nötig sind. Im Allgemeinen hat ein gemischt-ganzzahliges Minimierungsproblem die folgende Gestalt:

Definition 3.5.

$$(P) \quad \min \quad z = c^T x, \quad (3.5)$$

$$\text{bzgl.} \quad Ax = b, \quad (3.6)$$

$$x \geq 0, \quad (3.7)$$

$$x_1, \dots, x_k \in \mathbb{R}, \quad (3.8)$$

$$x_{k+1}, \dots, x_l \in \mathbb{Z}, \quad (3.9)$$

$$x_{l+1}, \dots, x_n \in \mathbb{B}, \quad (3.10)$$

$$x = (x_1, x_2, \dots, x_n)^T, \quad (3.11)$$

wobei diese Form gegebenenfalls durch Hinzufügen von Schlupf- bzw. Scheinvariablen sowie Umnummerieren erreicht wird. Es gilt $k, l, m, n \in \mathbb{N} : 1 \leq k \leq l \leq n$, $A \in \mathbb{R}^{m,n}$.

Der zulässige Bereich $G \subseteq \mathbb{R}^n$ ergibt sich aus den Nebenbedingungen (3.6) sowie den Bedingungen an die Variablen (3.7)-(3.10) [23].

Die Lösung (gemischt-)ganzzahliger Optimierungsprobleme erfolgt entweder durch Heuristiken, welche eine näherungsweise optimale Lösung bestimmen oder durch exakte Lösungsverfahren. Da gemischt-ganzzahlige Optimierungsprobleme i.A. jedoch zu den NP-schweren Problemen gehören, sind keine effizienten exakten Algorithmen bekannt [23]. Im Folgenden werden die Verfahren, die der verwendete Solver CPLEX zum Lösen einsetzt, erläutert.

3.3. CPLEX

Zur Lösung von Optimierungsproblemen stehen CPLEX eine Reihe von Methoden zur Verfügung, darunter der *Simplex-Algorithmus*, *Branch&Bound-Verfahren*, *Presolving*, *Schnittebenenverfahren*, *Probing*, *Parallel Solving* sowie *Branch&Cut-Methoden* [14]. Je nach Problemstellung wählt der Solver daraus die geeigneten Verfahrensweisen aus. Zum Lösen von gemischt-ganzzahligen Problemen sind dabei besonders Branch&Bound-Verfahren, Schnittebenenverfahren bzw. Branch&Cut-Verfahren sowie das Parallel Solving von Bedeutung. Auch wenn der Solver eine modifizierte, anspruchsvollere Variante dieser Lösungsstrategien verwendet [18], wird im Folgenden grob das Prinzip und die Vorgehensweise dieser erläutert.

3.3.1. Presolving

Bevor der Solver mit dem Lösen der Probleme beginnt, versucht CPLEX durch Einbinden von Vor- oder Unterroutrinen wie *Presolving* bessere Voraussetzungen zum Lösen des Problems zu schaffen. Dabei wird versucht, das Problem vor Beginn des eigentlichen Lösungsverfahrens geschickt zu modifizieren, um den darauf folgenden Rechenaufwand zu verringern [18, 23]. Dies geschieht zum Beispiel durch Streichen von überflüssigen Nebenbedingungen, das heißt Nebenbedingungen, die ohnehin erfüllt sind. Seien beispielsweise x_1 und x_2 Binärvariablen, d.h. $x_1, x_2 \in \{0, 1\}$, so ist die Bedingung

$$x_1 + x_2 \leq 3 \tag{3.12}$$

erfüllt, egal welchen Wert (0 oder 1) die beiden Variablen erhalten, und kann somit aus dem Problem entfernt werden.

Auch Nebenbedingungen, die bereits durch eine weitere Nebenbedingung erfüllt sind, können gestrichen werden. Betrachtet man beispielsweise das folgende Ungleichungssystem für $x_1, x_2, x_3 \in \{0, 1\}$:

$$x_1 + x_2 + x_3 \leq 2, \quad (3.13)$$

$$x_1 + x_2 + x_3 \leq 1, \quad (3.14)$$

so kann (3.13) gestrichen werden, da Bedingung (3.14) diese impliziert. Weiterhin werden die Schranken der Ungleichungen durch Einsetzen gegebener Schranken für die Variablen verschärft und logische Ungleichungen, sogenannte *clique-Ungleichungen*, zum Problem hinzugefügt [15].

3.3.2. Branch-and-Bound-Verfahren

Branch&Bound-Verfahren bestehen im Wesentlichen aus zwei Teilschritten: dem Verzweigen des Problems in Teilprobleme bzw. Knoten (*branch*) und dem Bilden von Schranken (*bound*). Beim Verzweigen werden stufenweise neue (verschärfte) Probleme (P_s) gebildet. Zur Initialisierung wird für das Problem (P) gemäß Definition 3.4 zunächst das kontinuierlich relaxierte Problem ($(P_{rel}) = (P_0)$) gebildet, d.h. die Bedingungen (3.8)-(3.10) werden ersetzt durch $x \in \mathbb{R}^n$, wodurch sich der zulässige Bereich zu G_{rel} vergrößert. Dieses wird dann z.B. mit dem Simplex-Algorithmus gelöst. Nach Definition 3.4 gilt für die zulässigen Bereiche des Ausgangsproblems und des relaxierten Problems die Beziehung $G \subseteq G_{rel}$ sowie die Ungleichung (ii). Dadurch erhält man eine erste untere Schranke für den Zielfunktionswert des Ausgangsproblems. Die obere Schranke wird zunächst auf $+\infty$ gesetzt. Anschließend werden schrittweise Verzweigungen gebildet, z.B. nach einer Variablen x_j , $j \in \{k+1, k+2, \dots, n\}$, die in der Lösung von (P_i) die Ganzzahligkeits- bzw. Binärforderung verletzt. Dazu werden zusätzliche Bedingungen in das Problem eingebaut. Verletzt ein x_j , $j \in \{k+1, k+2, \dots, l\}$ die Ganzzahligkeitsbedingung, d.h. der Wert dieser Variablen $x_{i,j}^*$ in der aktuellen Lösung des Problems (P_i) ist nicht ganzzahlig, so wird das Problem in 2 Teilprobleme ($P_{i,1}$) und ($P_{i,2}$) geteilt. Für die Variable x_j werden dabei neue Nebenbedingungen gefordert:

$$x_j \leq \lfloor x_{i,j} \rfloor, \quad (3.15)$$

$$x_j \geq \lceil x_{i,j} \rceil, \quad (3.16)$$

wobei $\lfloor x_{i,j} \rfloor = \max\{x \in \mathbb{Z} \mid x \leq x_{i,j}\}$ den abgerundeten ganzzahligen Anteil von $x_{i,j}$ und $\lceil x_{i,j} \rceil = \min\{x \in \mathbb{Z} \mid x \geq x_{i,j}\}$ den aufgerundeten ganzzahligen Anteil von $x_{i,j}$ beschreibt. Die Teilprobleme entsprechen bis auf eine Nebenbedingung dem Problem

(P_i). In ($P_{i,1}$) wird die zusätzliche Nebenbedingung (3.15) gefordert, während in Problem ($P_{i,2}$) Bedingung (3.16) zu (P_i) hinzugefügt wird. Verletzt hingegen eine Binärvariable x_j , $j \in \{l+1, l+2, \dots, n\}$ die Binärforderung, so wird das Problem in zwei Teilprobleme mit der zusätzlichen Nebenbedingung:

$$x_j = 0, \text{ bzw.} \quad (3.17)$$

$$x_j = 1, \quad (3.18)$$

aufgeteilt. Beim Verzweigen wird der zulässige Bereich des Problems verkleinert, dabei werden nur (für das Ausgangsproblem) unzulässige Punkte bzw. Bereiche abgeschnitten. Wird ein Problem P_s in m Teilprobleme $P_{s,1}, P_{s,2}, \dots, P_{s,m}$ aufgeteilt, so gilt für die zugehörigen zulässigen Bereiche $G_{s,i}$, $i = 1, \dots, m$ der dadurch entstandenen Probleme: $G_{s,i} \subset G \quad \forall i \in \{1, \dots, m\}$ sowie $G_{s,i} \cap G_{s,j} = \emptyset \quad \forall i, j = 1, \dots, m, i \neq j$. Durch diese stufenweise Verzweigung entsteht ein Baum. [18, 23]

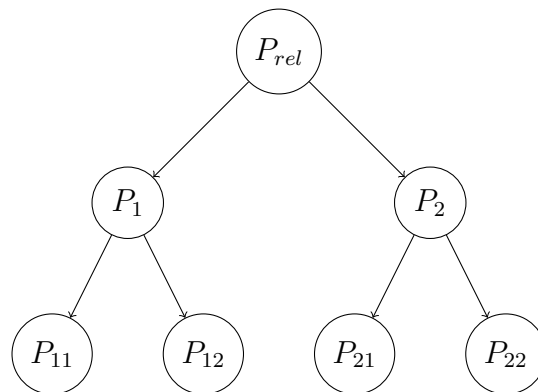


Abbildung 3.1.: Problemverzweigung (branch) beim Branch&Bound-Verfahren.

Die Vorgehensweise zur Bildung der Teilprobleme bzw. zur Untersuchung der gebildeten Probleme bestimmt die gewählte Knotenauswahlregel (z.B. lexikografisch oder schrankenbezogen) verknüpft mit einer Suchstrategie (z.B. Breiten- bzw. Tiefensuche). Bei der Untersuchung eines Teilproblems können nun verschiedene Fälle eintreten: (1) der zulässige Bereich des Teilproblems ist leer, (2) der Knoten liefert eine Lösung, deren Zielfunktionswert jedoch größer als die aktuelle obere Schranke ist (keine Verbesserung), (3) der Knoten liefert eine Lösung, deren Zielfunktionswert die aktuelle obere Schranke unterbietet. In Fall 1 und 2 muss der Knoten nicht mehr weiter verzweigt werden, er ist *ausgelotet*. Erfüllt die gefundene Lösung in Fall 3 die Bedingungen (3.7)-(3.10), so wird die aktuelle obere Schranke durch den erhaltenen Zielfunktionswert ersetzt. Verletzt sie jedoch eine oder mehrere dieser Bedingungen, so muss zunächst erneut gemäß Teilschritt 1 verzweigt werden, bevor neue Schranken gesetzt werden können. Durch dieses Verfahren verkleinert sich die Differenz zwischen oberer und unterer Schranke schrittweise, bis sie schließlich

im Optimalpunkt des Ausgangsproblems, vorausgesetzt es ist lösbar, identisch sind und dem Zielfunktionswert der Lösung entsprechen. Sobald neue Schranken gefunden werden, gibt CPLEX diese aus. Zusätzlich wird die momentane *optimality gap* (og), die sich aus diesen ergibt, angegeben. Die optimality gap ist ein relatives Maß für die Spanne zwischen der oberen und der unteren Schranke. Im Optimalpunkt ist diese Null. Sie wird folgendermaßen gebildet:

$$\text{og} = \frac{\text{aktuelle obere Schranke} - \text{aktuelle untere Schranke}}{\text{aktuelle obere Schranke}}. \quad (3.19)$$

Je nach Problem und Verzweigungs- und Knotenwahlstrategie können im Branch&Bound Verfahren Auslotungsfälle aus der Berechnung ausgenommen werden, um eine vollständige Enumeration zu verhindern und an eine Optimallösung für das Ausgangsproblem zu gelangen. Oft müssen dennoch viele Knoten überprüft werden. Die Effizienz dieses Verfahrens kann durch Hinzufügen von weiteren Schnitten bzw. Ungleichungen verbessert werden. Dazu wird das Verfahren mit dem Schnittebenenverfahren von Gomory kombiniert.

3.3.3. Schnittebenenverfahren von Gomory

Das Schnittebenenverfahren von Gomory basiert, wie das Branch&Bound-Verfahren, auf einer stufenweisen Reduzierung des zulässigen Bereichs mittels Herausschneidens unzulässiger Punkte. Dazu wird in jeder Stufe s das Problem (P_s) mittels Simplex-Algorithmus gelöst. Zur Initialisierung (Stufe 0) wird auch hier das stetig relaxierte Problem verwendet, d.h. $(P_0) = (P_{rel})$. Eine genaue Beschreibung des Simplex-Algorithmus ist in [23], Kapitel 3, zu finden.

Das Endtableau der s -ten Stufe hat die folgende Gestalt:

$$\begin{array}{c|cc|c} & x_B^T & x_N^T & \\ \hline x_B & I & D & p \\ \hline -z & 0^T & h^T & -z_0 \end{array} \quad (3.20)$$

Dabei bezeichnet $J_B \subset \{1, \dots, n + s\}$ die Menge der Indizes der Variablen, die zur Basis gehören, x_B die Basisvariablen d.h. $x_B = \{x_i \mid i \in J_B\}$, $x_N = \{x_i \mid i \notin J_B\}$ die Nichtbasis-Variablen, I die Einheitsmatrix und $D = (d_{ij})$ die Matrix der Nichtbasis-Koeffizienten des Simplex-Tableaus. Erfüllt die Lösung x_{rel}^* des relaxierten Problems bereits die Bedingungen (3.7)-(3.10) aus Definition 3.5, so ist kein weiterer Schritt notwendig. Andernfalls wird eine gültige Ungleichung zum stetig relaxierten System hinzugefügt. Gültig bedeutet, dass alle Punkte des zulässigen Bereiches des Ausgangsproblems (P) diese erfüllen und dass so-

mit nur für (P) unzulässige Bereiche abgeschnitten werden. Der aktuelle Lösungspunkt x_s^* wird mit Einfügen dieses sog. *Gomory-Schnitts* abgeschnitten, sodass im nächsten Schritt eine neue Lösung gefunden wird. Zur Ermittlung des Gomory-Schnittes sind folgende Schritte zu befolgen: Zunächst wird die Schnittzeile i ausfindig gemacht. Dazu werden die Reste r_i für alle Variablen $x_{B,i}$, die eine der Bedingungen (3.7)-(3.10) verletzen, berechnet:

$$r_i = \max\{p_k - \lfloor p_k \rfloor\}, \quad (3.21)$$

Die Schnittzeile ist nun die Zeile mit dem kleinsten Index i , für die (3.21) gilt. Für diese Zeile werden anschließend die für die Gomory-Ungleichung nötigen Koeffizienten r_{ij} für die Indizes j bestimmt, die zu den Nichtbasis-Indizes gehören:

$$r_{ij} = d_{ij} - \lfloor d_{ij} \rfloor, \quad \forall j \notin J_B. \quad (3.22)$$

Die Gomory-Schnittungleichung ist dann gegeben durch:

$$- \sum_{j \notin J_B} r_{ij} \cdot x_j \leq -r_i. \quad (3.23)$$

Durch Hinzufügen der Ungleichung ändert sich der zulässige Bereich des Problems, woraus das Problem der nächsten Stufe (P_{s+1}) entsteht. Wird der zulässige Bereich leer (beispielsweise wenn sich die Nebenbedingungen widersprechen), so gibt es keine Lösung für das Ausgangsproblem, die alle Bedingungen erfüllt. Ist dieser nicht leer, so wird ein dualer Simplex-Schritt (siehe dazu [9], Abschnitt 2.4.2.1) durchgeführt und anschließend das Verfahren so lange wiederholt, bis die aktuelle Lösung x_s^* alle Bedingungen des Ausgangsproblems (P) erfüllt [23].

3.3.4. Parallel Solving

Die Effizienz von CPLEX zum Lösen von Problemen wird einerseits durch Kombination verschiedener Lösungsstrategien und andererseits durch den optimalen Einsatz der vorhandenen Kapazitäten garantiert.

Beim *Parallel Solving*, also gleichzeitigem Lösen, werden die Prozessoren des Rechners separat für verschiedene Programmunterabläufe verwendet. Voraussetzung dafür ist die Existenz mehrerer Prozessoren, die im Stande sind, einzelne Programmabläufe auszuführen und untereinander zu kommunizieren. Wie stark diese Kommunikation sein soll, ist abzuwägen. Auf der einen Seite bringt die Kommunikation zwischen den Prozessoren Vorteile, so können z.B. durch Kommunizieren von Schranken die Knoten der übrigen Prozessoren schneller ausgelotet werden, auf der anderen Seite kostet sie Laufzeit. Je nach Strategie und Zuordnung der Prozessoren kann ein Prozessor dann beispielsweise einen ganzen Baum, einen Unterbaum, oder nur einen einzigen Knoten untersuchen. Ein Parallelalgorithmus lässt sich in drei Phasen aufteilen:

1. *Ramp-up-phase*: In dieser Phase wird der Algorithmus in Teilaufgaben zerlegt und auf die zur Verfügung stehenden Prozessoren verteilt.
2. *Primary phase*: Während dieser Phase arbeitet der Algorithmus bzw. alle Prozessoren stetig, d.h. kein Prozessor hat bereits seinen Teil terminiert.
3. *Ramp-down-phase*: In der letzten Phase werden abschließende Prozesse erledigt und die Resultate dargestellt.

Der Übergang der zweiten zur dritten Phase ist schwer zu definieren, jedoch kann man von Beginn der dritten Phase sprechen, sobald der erste Prozessor sich in der Endphase seines Teilalgorithmus befindet. Die Methodik des Parallel Solvings ist besonders für Baum- bzw. Graphenalgorithmien und somit auch für den Branch&Bound-Algorithmus geeignet [18], weshalb es zum Lösen gemischt-ganzzahliger Modelle mit CPLEX von Relevanz ist.

3.4. Multikriterielle Optimierung und Pareto-Optimierung

Ein Optimierungsproblem, und insbesondere auch die Optimierung der Schulanfangszeiten, lässt sich aus unterschiedlichen Blickwinkeln betrachten [16], wodurch sich entsprechend die Zielfunktion ändert. Möchte man jedoch mehrere oder alle dieser Zielfunktionen gleichzeitig mit in das Problem einbeziehen, so handelt es sich um eine sogenannte *Multikriterielle Optimierung*. Für gewöhnlich widersprechen sich die verschiedenen Optimierungskriterien, wodurch ein Optimum, das alle Faktoren zur vollsten Güte erfüllt, meist nicht existiert [1]. Eine multikriterielle Betrachtung eines Modells ermöglicht eine Analyse der Lösungen und des Modells, d.h. wie realistisch ist ein Modell, das auf die Einbeziehung dieser zusätzlichen Zielfunktionen verzichtet? Wie sehr beeinträchtigen Vorgaben und Einschränkungen durch andere Faktoren eine Lösung? Bleibt die Lösung ohnehin dieselbe bzw. wie weit kann man die Vorgaben verschärfen, ohne dass sich die Lösung signifikant ändert, oder wie sensitiv reagiert das Modell auf Änderungen?

Zur Realisation einer multikriteriellen Betrachtung kann man wie folgt vorgehen: Eine Zielfunktion wird als Zielfunktion des Modells verwendet, während die anderen Zielfunktionen mithilfe einer oberen Schranke (bei Maximierungsproblemem untere Schranke) als Nebenbedingungen in das Modell eingebaut werden.

Definition 3.6. Seien z_1, \dots, z_k die zu minimierenden Zielfunktionen und $x = (x_1, \dots, x_n)^T$ der Vektor der Variablen. Dann wird das multikriterielle Minimierungsproblem für ein beliebiges, aber festes i , $i \in \{1, 2, \dots, k\}$ beschrieben durch:

$$(P_{mult,i}) \quad \min \quad z_i(x), \quad (3.24)$$

$$\text{bzgl.} \quad z_j(x) \leq c_j, \quad j = 1, \dots, k, j \neq i, \quad (3.25)$$

$$g_j(x) \leq 0 \quad j = 1, \dots, l, \quad (3.26)$$

$$u \leq x \leq o, \quad (3.27)$$

wobei $j, k, l \in \mathbb{N}$ und $u, o \in \mathbb{R}^n$ die unteren bzw. oberen Schranken für den Vektor x beschreiben.

Eine Lösung x^* des multikriteriellen Problems liefert für jede der betrachteten Zielfunktionen einen Wert, d.h. $z(x^*) = (z_1(x^*), z_2(x^*), \dots, z_k(x^*))^T$. Durch Variation der Konstanten c_j in (3.25) erhält man verschiedene Lösungen des Problems. Um alle (relevanten) Lösungen zu berechnen, sind aufgrund der Vielzahl der Möglichkeiten im Regelfall Algorithmen notwendig. Die damit erhaltenen Lösungen sind jedoch bezüglich ihrer Güte nicht alle äquivalent, wie die folgende Definition zeigt.

Definition 3.7. Seien x_1, x_2 Lösungen eines multikriteriellen Minimierungsproblems gemäß Definition 3.6 mit den dazugehörigen Zielfunktionsvektoren

$z(x_1) = (z_1(x_1), z_2(x_1), \dots, z_k(x_1))^T$, $z(x_2) = (z_1(x_2), \dots, z_k(x_2))^T$. Dann gilt:

$$\begin{aligned} z(x_1) \text{ dominiert } z(x_2) &\Leftrightarrow z_i(x_1) \leq z_i(x_2) \quad \forall i \in \{1, \dots, k\} \\ &\quad \wedge \exists j \in \{1, \dots, k\} : z_j(x_1) < z_j(x_2) \end{aligned}$$

Man sagt auch: Lösung x_1 dominiert Lösung x_2 . Dominiert keine der beiden Lösungen die andere, so nennt man deren Beziehung *indifferent*.

Dieser Zusammenhang lässt sich gut an einem Beispiel eines Optimierungsproblems mit zwei Zielfunktionen nachvollziehen:

Beispiel 3.8. z_1 und z_2 sind die zu minimierenden Zielfunktionen, x_1, x_2, x_3 Lösungen des bikriteriellen Optimierungsproblems.

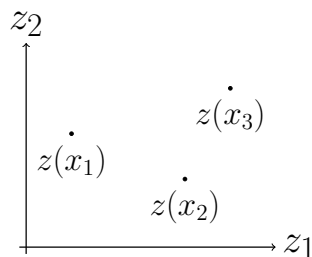


Abbildung 3.2.: Lösungsbeziehungen eines bikriteriellen Optimierungsproblems.

Wie man leicht einsieht, wird Lösung x_3 von Lösung x_1 dominiert, da x_1 sowohl in der ersten Zielfunktion als auch in der zweiten einen kleineren Wert als x_3 liefert. Selbiges gilt für x_2 und x_3 (x_2 dominiert x_3). x_1 ist indifferent gegenüber x_2 , da jede Lösung der anderen in einem Kriterium überlegen ist.

Ordnet man den Zielfunktionen zunächst keine Gewichtung zu, erhält man die im Allgemeinen überlegenen Lösungen. Diese Menge ist die sogenannte *Pareto-Menge*[1, 2].

Definition 3.9. Die Pareto-Menge ist die Menge der Lösungen, die von keinen anderen Lösungen aus dem Lösungsraum dominiert werden. Die dazugehörigen Punkte der Zielfunktionsvektoren bilden die Pareto-Front.

Aus der Definition folgt direkt, dass die Lösungen aus der Pareto-Menge alle paarweise indifferent zueinander sind. Abbildung 3.3 zeigt den typischen Aufbau des Lösungsraums und der Paretofront eines Minimierungsproblems mit zwei Zielfunktionen.

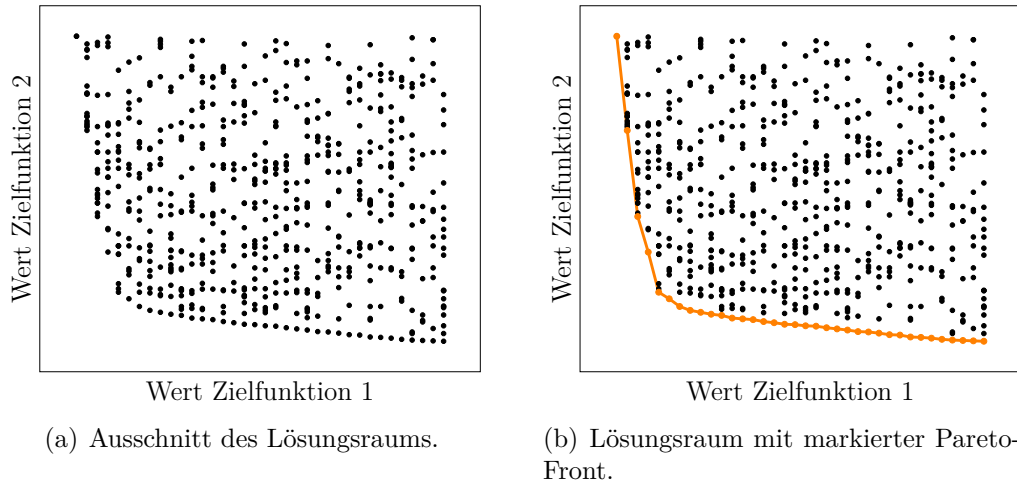


Abbildung 3.3.: Lösungsraum und Pareto-Front eines Minimierungsproblems mit zwei Zielfunktionen.

Berechnet man die Pareto-Front anstelle einer Lösung, welche nur eine Zielfunktion berücksichtigt, so kann der Anwender auch im Nachhinein noch seine Präferenzen anpassen und eine passende Vorgehensweise (z.B. Produktionsplan, Fahrtroute) wählen. Zudem können mittels der Pareto-Front im Voraus Abwägungen getätigt werden, beispielsweise unter welchen Bedingungen sich ein Projekt lohnt oder mit welchen Einbußen zu rechnen ist, falls sich die Rahmenbedingungen (gegeben durch zusätzliche Zielfunktionen) kurzfristig verändern.

Je höher die Anzahl der einzubringenden Zielfunktionen (k) ist, desto höher ist auch die Dimension des Pareto-Sets ($k - 1$), wodurch bei mehr als zwei Zielfunktionen die Anschaulichkeit deutlich abnimmt [1].

4. Modell

Im Folgenden soll nun der Modellaufbau zur Optimierung des Schülertransportes erläutert werden. Die grundlegende Idee dahinter ist, die Schulanfangszeiten geschickt zu verschieben, sodass starke Ballungen, die verhindert werden können, nicht zustande kommen. Das Personenaufkommen soll somit gleichmäßiger über einen bestimmten Zeitraum verteilt werden. Da die Modellierung des Problems unter Berücksichtigung sämtlicher Kriterien wie Haltestellen, Busrouten, Einhaltung der Personalkapazitäten, Umsteigen der Schüler den Rahmen dieser Arbeit übersteigen würde, wurde das Problem auf kompaktere Modelle transferiert, wofür einige Annahmen und Einschränkungen getätigt werden müssen. Die Modelle beinhalten keine Bushaltestellen, Busrouten oder Fahrzeuge, die Optimierung soll mithilfe der Minderung der Belastung eines Systems, welche im folgenden Kapitel erläutert wird, erfolgen. Mit den somit erhaltenen Lösungen lässt sich ein erster Eindruck gewinnen, wie hilfreich die Einbringung der Schulanfangszeiten als Variablen eines Modells sein kann.

Die Effekte der Belastungsminderung können u.U. auch auf ein System mit Bussen übertragen werden. Beginnen alle Schulen zur gleichen Zeit, so entstehen meist mehrere Ballungsgebiete, die nicht von einzelnen Bussen bewältigt werden können, wodurch mehrere Busse für eine Route eingesetzt werden müssen. Wird die Belastung des Systems so reguliert, dass sich über den gesamten Zeitraum eine mehr oder weniger ausgeglichene Belastung vermerken lässt, so können dadurch Fahrzeuge eingespart werden bzw. der vorhandene Fuhrpark effizienter genutzt werden [12].

4.1. Vorgehensweise und Annahmen

Für die Anwendung des Modells auf eine reale Instanz müssten die Wohnorte aller Schüler vorliegen, was sich sowohl aus datenschutztechnischen Gründen als auch aus Aufwandsicht als kompliziert erweisen kann. Alternativ könnte man die Schüler einer Schule innerhalb eines bestimmten Radius als gleichverteilt annehmen und bräuchte damit nur noch die Schulkoordinaten sowie die Anzahl der Schüler, die die Schule mit dem Nahverkehr erreichen. Diese Annahme könnte das Ergebnis jedoch so verändern, dass es bei der Anwendung auf die reale Instanz nicht mehr die gewünschten Effekte erzielen könnte.

Daher wurde für das Modell mittels der Programmiersprache *Python* (Version 2.7) ein Instanzengenerator erstellt, um es auf die damit erstellten Instanzen anzuwenden. Um die Instanzen, die Schülerbewegungen sowie den Belastungsgrad zu veranschaulichen, werden diese mithilfe der Programmbibliothek *Matplotlib* grafisch dargestellt. Nachdem eine Problem Instanz mittels CPLEX gelöst wurde, ermöglicht dies einen guten Vergleich des Ist- und des Optimalzustandes. Auch die für die Modelle nötigen Parameter werden vorab mittels Python berechnet und an den Solver zur Erstellung und Lösung des Problems übergeben. Für die Dateien zur Instanzengenerierung und Anwendung der Modelle sowie sonstige Dateien sei auf den digitalen Anhang verwiesen.

Die Komplexität des Problems erfordert vereinfachende Annahmen für das Modell. Es wird angenommen, dass Schüler, die entsprechend nah an der Schule wohnen, den Nahverkehr nicht nutzen, um zur Schule zu kommen (vgl. Kapitel 4.2). Weiterhin sollen der Einfachheit halber alle übrigen Schüler einer Schule den Nahverkehr nutzen. Alle Schüler erreichen exakt zu Beginn des Unterrichts die Schule, eventuelle Wartezeiten oder früheres Ankommen werden somit vernachlässigt. Da das Modell keine Modellierung von Haltestellen und Routen beinhaltet, wird davon ausgegangen, dass die Schüler sich mit einer Geschwindigkeit von 3 km/h über den kürzesten Weg, im Sinne der euklidischen Distanz, der Schule nähern.

Für die praktische Umsetzung ergibt eine Verschiebung der Anfangszeiten auf die Minute genau keinen Sinn, weshalb nur Zeiten im Abstand von fünf Minuten zur Auswahl stehen, d.h. eine Schule kann um 7:30 Uhr, 7:35 Uhr etc. beginnen. Auch eine zu große Änderung ist in der Praxis einerseits nicht praktikabel und teilweise nicht mit gesetzlichen Vorgaben in Einklang zu bringen, weshalb alle Schulanfangszeiten innerhalb eines Zeitfensters von einer Stunde, zwischen 7:30 Uhr und 8:30 Uhr, angesiedelt sein sollten.

4.2. Generierung von Instanzen

Zur Generierung von Instanzen für das Problem sind vom Nutzer vorab einige Parameter vorzugeben: die Anzahl der zu erzeugenden Schulen n ($S = \{1, 2, \dots, n\}$), die Größe bzw. Seitenlängen der Stadt in Metern (x_{max}, y_{max}) und die maximalen Einzugsradien der Schulen in Metern ($r_{max,i}$, $i = 1, 2, 3$), abhängig vom Schultyp (Grundschulen, weiterführende Schulen, sonstige Schulen). Auch der minimale Abstand, den jeder erzeugte Schüler von der Schule haben soll, wird vom Benutzer vorgegeben (r_{min}). So werden Schüler, die sehr nah an der Schule wohnen, direkt aus der Berechnung ausgenommen, da sie keinen Einfluss auf die Belastung des Nahverkehrs im betrachteten Sinne haben. Für jeden Schultyp können außerdem noch obere und untere Grenzen für die Schülerkapazitäten gesetzt

werden. Der Generator erzeugt Schulen, deren Einzugsbereiche sich immer mit dem Einzugsbereich von mindestens einer anderen Schule überschneidet, isolierte Schulen gibt es also nicht. Das hat den Hintergrund, dass solche Schulen für die Optimierung außer Acht gelassen werden können, da die Verschiebung der Anfangszeit einer solchen Schule keinen Einfluss auf die Minderung der Systembelastung hätte.

Zuerst wird die Schulart der n Schulen zufällig ausgewählt. Innerhalb der vorgegebenen Grenzen wird dann zu jeder Schule ein Radius ($r_i, i = 1, \dots, n$) und die Anzahl der Schüler ($schueleranzahl_i, i = 1, \dots, n$) gewählt. Dann werden die x - und y -Koordinaten (x_i, y_i) jeder Schule so generiert, dass $0 \leq x_i - r_i \wedge x_i + r_i \leq xMax$ und $0 \leq y_i - r_i \wedge y_i + r_i \leq yMax$ gilt, d.h. die Schule und ihr gesamter Einzugsbereich befinden sich innerhalb der Grenzen der Instanz. Zuletzt werden die Koordinaten bzw. Wohnorte der Schüler zufällig gleichverteilt innerhalb des Einzugsradius der Schule generiert.

4.3. Modellbeschreibung

Rasteraufteilung

Zur Beschreibung der Systembelastung innerhalb einer Instanz wird ein Maß benötigt. Dazu wird die Instanz in gleichgroße Quadrate, im Nachfolgenden Pixel genannt, zerlegt. Die Seiten der Stadt werden in z bzw. s Teile aufgeteilt, für den Fall $x_{max} = y_{max}$ gilt $z = s$. Daraus ergeben sich $z \cdot s$ Pixel $p_{ij}, i \in I = \{0, 1, \dots, z - 1\}, j \in J = \{0, 1, \dots, s - 1\}$.

p_{30}	p_{31}	p_{32}	p_{33}
p_{20}	p_{21}	p_{22}	p_{23}
p_{10}	p_{11}	p_{12}	p_{13}
p_{00}	p_{01}	p_{02}	p_{03}

Abbildung 4.1.: Aufteilungsschema einer Instanz ($z=s=4$).

Um auszuschließen, dass Schüler bei der Zuordnung mehreren Pixeln gleichzeitig zugeordnet werden, falls sie auf einer Kante stehen, sind die Pixel halboffen. Dabei sind die linke und die untere Kante eines Pixels abgeschlossen, während die anderen beiden Ränder offen sind. Die Randpixel am oberen und rechten Rand der Stadt sind auch dort abgeschlossen. Nun stellt sich die Frage, wie groß die Pixel bzw. die Feinheit des Rasters gewählt werden sollte, um ein realistisches Modell zu erhalten. Wählt man das Raster zu grob, wird das Modell unrealistisch. Schüler, die mehrere Quadratkilometer voneinander entfernt sind, sollten nicht das selbe Pixel belasten. Wählt man das Raster hingegen zu klein, sind durch das Modell keine Verbesserungen möglich, da jedes Pixel im schlimmsten Fall von nur einem oder keinem Schüler durchlaufen wird, egal wie man die Anfangszeiten der Schulen verschiebt. Da zu dieser Vorgehensweise keine Daten oder Studien existieren, wird die Annahme getätigt, dass eine realistische Größe für die Seitenlänge l_{pixel} eines Pixels 50 Meter sind. So können zwei Schüler nur dann das gleiche Pixel beeinflussen, wenn sie maximal 70,71 Meter ($\sqrt{50^2 + 50^2}$) voneinander entfernt sind. Die Pixelgröße kann vor Beginn der Berechnung ausgewählt werden.

Schulanfangszeiten

Als mögliche Schulanfangszeiten mit den bereits beschriebenen Vorgaben ergeben sich 13 mögliche Startzeitpunkte und damit die Menge:

$$B := \{450, 455, 460, 465, 470, 475, 480, 485, 490, 495, 500, 505, 510\}, \quad (4.1)$$

die diese in Minuteneinheiten enthält (d.h. 450 entspricht 7:30 Uhr, 455 entspricht 7:35 Uhr usw.). Obwohl im Folgenden die zeitliche Einheit Sekunden sind, werden die Schulanfangszeiten der Übersichtlichkeit halber in Minuten angegeben. Die Schulanfangszeiten sind variabel und können jeden in der Menge B enthaltenen Wert annehmen. Somit ergeben sich die $n \cdot |B|$ Binärvariablen, die angeben, ob eine Schule s ihren Unterricht zur Zeit t beginnt, oder nicht:

$$x_{s,t} = \begin{cases} 1 & , \text{ falls Schule } s \text{ ihren Unterricht zur Zeit } t \text{ beginnt} \\ 0 & , \text{ falls nicht} \end{cases} \quad \forall s \in S, t \in B. \quad (4.2)$$

Dass jede Schule ihren Unterricht nur zu genau einer Zeit beginnen kann, wird durch die folgende Bedingung gewährleistet:

$$\sum_{t \in B} x_{s,t} = 1, \quad \forall s \in S. \quad (4.3)$$

Da die Entscheidungsvariablen über die Anfangszeiten binär sind und somit nur den Wert 0 oder 1 annehmen können, wird die Gleichung in (4.3) genau dann erfüllt, wenn für jede Schule exakt eine dieser Variablen den Wert 1 annimmt, woraus resultiert, dass die Schule ihren Unterricht zu dieser und zu keiner anderen Zeit beginnt.

Systembelastung

Die momentane Gesamtsystembelastung der Instanz ist erklärt als die Anzahl sämtlicher Schüler, die sich zu diesem Zeitpunkt aktiv im System bewegen, d.h. nicht mehr Zuhause sind und auch noch nicht ihre Destination erreicht haben. Dabei handelt es sich um eine Momentaufnahme der Situation, die Positionen der Schüler werden in einem entsprechenden Takt aktualisiert. Im Folgenden wird dafür die Zeit in Sekunden nach Tagesbeginn (0:00 Uhr) aufgeführt. Befindet sich ein Schüler zu einer Zeit in einem Pixel, so erhöht er die lokale Belastung in diesem Pixel zu dieser Zeit um den Wert 1. Die Gesamtbelastung zu einer bestimmten Zeit ergibt sich durch Summieren der lokalen Belastungen über alle Pixel.

Die Taktung der Belastungsmessung des Systems ist von der Pixelgröße abhängig zu machen. Sind die Pixel sehr groß, so muss nicht etwa alle 30 Sekunden oder öfter der Belastungszustand der Instanz erfasst werden, da sich die Schüler dann deutlich länger als 30 Sekunden in einem Pixel aufhalten. Auf der anderen Seite muss verhindert werden, dass Schüler innerhalb einer Zeitperiode mehrere Pixel durchlaufen und diese dann in der Berechnung nicht berücksichtigt werden können, da die Taktung zu grob gewählt wurde. Die Seitenlänge der Pixel bestimmt das Intervall der Taktung. So soll ein Schüler, der sich parallel zu einer Kante des Pixels bewegt, innerhalb eines Messintervalls maximal ein Pixel vollständig durchlaufen können. Bei einer Bewegungsgeschwindigkeit von 3 km/h wird dazu die Intervalllänge θ (in Sekunden) zwischen den Aktualisierungen bei einer Pixelseitenlänge von l_{pixel} (in Metern) festgelegt auf:

$$\theta := \lfloor l_{pixel} \cdot \frac{3600}{3000} \rfloor. \quad (4.4)$$

Ein Schüler, der sich nicht parallel zu einem Rand eines Pixels bewegt, und zwischen zwei Momentaufnahmen ein drittes Pixel berührt, wird in dessen Belastung nicht berücksichtigt, da sonst die Intervalle zwischen der Belastungsmessung beliebig klein werden müssten (im Grenzfall gegen Null gehen), wodurch keine allgemeine Aussage über die Feinheit der Taktung möglich ist. Was innerhalb eines Zeitintervalls zwischen einer Momentaufnahme und der nächsten geschieht, bleibt also aus der Berechnung ausgenommen. Es werden lediglich der Startpunkt des Schülers und seine Position am Ende des Intervalls (bzw.

zum Startzeitpunkt des nächsten Intervalls) berücksichtigt. Bei der Wahl einer geeigneten Aktualisierungsrate ist lediglich die *Abgeschlossenheit* des Schulweges bezüglich der *Nachbarschaft* eines Pixels im folgenden Sinne erforderlich:

Definition 4.1. Die *Nachbarschaft* $N(p_{ij})$ eines Pixels p_{ij} sei definiert als das Pixel selbst sowie alle acht umliegenden Pixel, d.h.

$$N(p_{ij}) := \{p_{kl} \mid \max\{0, i-1\} \leq k \leq \min\{i+1, z-1\}, \\ \max\{0, j-1\} \leq l \leq \min\{j+1, s-1\}\}. \quad (4.5)$$

Definition 4.2. Sei $P_{s,t} = (Px_{s,t}, Py_{s,t})$ die Position von Schüler s zur Zeit t , bestehend aus einer x - und einer y -Koordinate. Die Positionen der Schüler werden alle θ Sekunden aktualisiert. $P_{s,t} \in p_{ij}$ bedeutet, Schüler s befindet sich zur Zeit t in Pixel p_{ij} . Der Schulweg eines Schülers heißt *abgeschlossen*, wenn für diesen Schüler zu jeder Zeit gilt: Befindet sich der Schüler zur Zeit t in Pixel p_{ij} , so befindet sich der Schüler zur Zeit $t + \theta$ in der Nachbarschaft von Pixel p_{ij} , d.h.:

$$P_{s,t} \in p_{ij} \Rightarrow P_{s,t+\theta} \in p_{kl} \in N(p_{ij}). \quad (4.6)$$

Satz 4.1. Bei einer Intervalllänge von $\theta = \lfloor l_{pixel} \cdot \frac{3600}{3000} \rfloor$ Sekunden zwischen den Aktualisierungen ist der Weg eines jeden Schülers zu seiner Schule abgeschlossen.

Für den Beweis sind die zwei folgenden Definitionen notwendig.

Definition 4.3. Seien $x = (x_1, x_2) \in \mathbb{R}^2$ und $y = (y_1, y_2) \in \mathbb{R}^2$ zwei beliebige Punkte. Der Abstand $dist(x, y)$ dieser Punkte ist definiert als:

$$dist(x, y) := \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2}. \quad (4.7)$$

Definition 4.4. Die abgeschlossene Kreisscheibe $K_r(M)$ vom Radius $r \in \mathbb{R}$ um den Mittelpunkt $M = (M_x, M_y) \in \mathbb{R}^2$ ist definiert als:

$$K_r(M) := \{P = (P_x, P_y) \in \mathbb{R}^2 \mid dist(P, M) \leq r\}. \quad (4.8)$$

Beweis von Satz 4.1. Um die Abgeschlossenheit des gesamten Schulweges eines beliebigen Schülers zu beweisen, genügt es, (4.6) für eine beliebige Zeit und ein beliebiges Pixel zu zeigen, was dann die Aussage impliziert. Die Schüler bewegen sich mit einer Geschwindigkeit von $3 \frac{km}{h}$, bzw. $\frac{3000}{3600} \cdot \theta \frac{m}{\theta s} \leq l_{pixel} \frac{m}{\theta s}$. Sei die Zeit t beliebig, es gelte $P_{s,t} \in p_{ij}$ beliebig. Dann befindet sich der Schüler zur Zeit $t + \theta$ innerhalb oder am Rand der abgeschlossenen Kreisscheibe $K_{l_{pixel}}(P_{s,t})$ vom Radius l_{pixel} um den Mittelpunkt $P_{s,t}$. Ungeachtet der

genauen Position $P_{s,t}$ des Schülers zur Zeit t , ist der Abstand zum Rand der Nachbarschaft gemäß (4.6) mindestens l_{pixel} Meter. Damit liegt die Kreisscheibe vollständig in der Nachbarschaft des Pixels, d.h. $K_{l_{pixel}}(P_{s,t}) \subset N(p_{ij})$, was den Satz impliziert. \square

Bei einer Seitenlänge von 50 Metern müssen die Positionen der Schüler und die Belastung des Systems somit alle 60 Sekunden überprüft werden. Da die Schulanfangszeiten variabel sind, ist die Belastung zunächst in Relation zum Schulbeginn anzugeben. Für jeden Schüler werden dazu die schrittweisen Koordinaten, ebenfalls in Relation zum Unterrichtsbeginn der Schule, berechnet. Zudem werden für jeden Schüler p einer Schule s die Anzahl der Schritte ($\sigma_{s,p}$) ermittelt, die er mit der Aktualisierungsrate θ zu seiner Schule benötigt. Für jede Schule s ergibt sich dann der Wert σ_s , der den Maximalwert für diese Schule enthält. Das Maximum über alle Schulen liefert das Systemmaximum σ . Daraus ergibt sich die maximale relative Zeit vor Schulbeginn $\tau_{max} = \sigma \cdot \theta$. Die Menge R aller relativen Zeitpunkte (in Sekunden vor Schulbeginn) bei einer Pixelseitenlänge von 50 Metern ist somit gegeben durch:

$$R := \{\tau \in [0, \tau_{max}] \mid \tau = 60 \cdot k, k \in \mathbb{N}_0\}. \quad (4.9)$$

Damit kann bestimmt werden, wie viele Schüler einer bestimmten Schule sich zeitlich relativ zum Schulbeginn in einem Pixel aufhalten, was durch den Parameter $\delta_{s,i,j,\tau}$ beschrieben wird.

$$\delta_{s,i,j,\tau} := \begin{cases} c \in \mathbb{N}_0 \text{ beliebig} & , \text{ für } \tau \in [0, \tau_{max,s}] \\ 0 & , \text{ sonst} \end{cases} \quad , \forall s \in S, i \in I, j \in J, \tau \in R \quad (4.10)$$

D.h. $\delta_{2,1,1,120} = 200$ bedeutet, 200 Schüler der Schule 2 halten sich 120 Sekunden vor Unterrichtsbeginn im Pixel (1,1) auf. Für die Optimierung ist jedoch die Belastung des Systems zu absoluten Zeiten relevant. Die Menge der absoluten Zeiten soll alle möglichen Fälle abdecken. Die früheste mögliche Zeit tritt dann ein, wenn die Schule des Schülers, der den längsten Schulweg hat, bereits um 7:30 Uhr beginnt. Als obere Grenze für t ergibt sich 8:30 Uhr, da alle Schulen spätestens um 8:30 Uhr beginnen müssen und ab diesem Zeitpunkt keine Belastung mehr messbar wäre. Somit ergibt sich die Menge T aller absoluten Zeiten:

$$T = \{t \in [450 \cdot 60 - \tau_{max}, 510 \cdot 60] \mid t = 60 \cdot k, k \in \mathbb{N}_0\} \quad (4.11)$$

Da die Belastungen der Pixel zu absoluten Zeiten von den Schulanfangszeiten abhängen, sind sie variabel. Mithilfe der Entscheidungsvariablen der Schulanfangszeiten lassen sich diese berechnen:

$$d_{i,j,t} = \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} \cdot \delta_{s,i,j,(60 \cdot \tau - t)}, \quad \forall i \in I, j \in J, t \in T. \quad (4.12)$$

Befindet sich ein Schüler (aktiv) zum Zeitpunkt t in einem Pixel (i, j) , so erhöht er den temporären Wert der Belastung $d_{i,j,t}$ in diesem Pixel um den Wert 1.

4.3.1. Modell 1

Für das erste Modell soll die Belastung eines Pixels nach Verschiebung der Anfangszeiten mit dem pixelindividuellen Maximalwert vor der Verschiebung (Ist-Zustand) verglichen werden. Der Ist-Zustand ist der Zustand des Systems für den Fall, dass sämtliche Schulen innerhalb der Instanz ihren Unterricht zur selben Zeit (7:30 Uhr) beginnen. Dafür wird vorab für jedes Pixel der Vergleichswert $\Delta_{i,j}$ berechnet. Das ist der maximale Wert, den die Systembelastung innerhalb dieses Pixels im Gesamtzeitraum annimmt, d.h.:

$$\Delta_{i,j} = \max_{t \in T} \Delta_{i,j,t}, \quad \forall i \in I, j \in J, \text{ wobei} \quad (4.13)$$

$$\Delta_{i,j,t} = \sum_{s \in S} \delta_{s,i,j,(60 \cdot 450 - t)}, \quad \forall t \in T, i \in I, j \in J. \quad (4.14)$$

Für jedes Pixel lässt sich, abhängig von der Zeit, die Differenz aus Soll-Wert und maximalem Ist-Wert der Belastung berechnen:

$$d_{i,j,t} - \Delta_{i,j} = D_{i,j,t}, \quad \forall i \in I, j \in J, t \in T. \quad (4.15)$$

Für den gesamten Betrachtungszeitraum ergibt sich also für jedes Pixel eine obere Schranke $D_{i,j} = \max_{t \in T} D_{i,j,t}$, d.h. ein Wert, um den sich die Belastung dieses Pixels im Vergleich zum Maximalwert der Ausgangssituation mindestens verringert, sodass gilt:

$$d_{i,j,t} - \Delta_{i,j} \leq D_{i,j}, \quad \forall i \in I, j \in J, t \in T. \quad (4.16)$$

Zusätzlich soll gewährleistet sein, dass sich in jedem Pixel die Belastung im Vergleich zum Maximalwert des Ist-Zustandes nicht verschlechtert (höher wird) bzw. einen bestimmten Schwellenwert nicht überschreitet. Das heißt, in jedem Pixel (i, j) soll die Belastung im Vergleich zum Wert $\Delta_{i,j}$ möglichst sinken, jedoch maximal den Wert $\Delta_{i,j}$ annehmen, oder im Falle eines kleineren Wertes bei einer Pixelgröße von 50 Metern maximal den Wert 100 annehmen. Da die Modelle Grundlage für die Optimierung des Schülernahverkehrs ist,

orientiert sich die obere Schranke von 100 an den Sitz- und Stehplatzkapazitäten üblicher Linienbusse. Ungleichung (4.17) beschreibt diese Bedingung.

$$d_{i,j,t} \leq \max\{\Delta_{i,j}, 100\}, \quad \forall i \in I, j \in J, t \in T. \quad (4.17)$$

Das Modell wird durch diese Bedingung nicht unzulässig, da der Momentanzustand immer einer Lösung für das Modell entspricht, die alle Bedingungen trivialerweise erfüllt.

Als Zielfunktion des Modells soll die Summe der oberen Schranken $D_{i,j}$ der Belastungsminderung über alle Pixel dienen:

$$z = \sum_{i \in I} \sum_{j \in J} D_{i,j}. \quad (4.18)$$

Die Ungleichung in (4.16) ergibt für ein $D_{i,j}$ im Falle einer Belastungsminderung einen negativen Wert, kann die Belastungsminderung nicht verringert werden oder steigt sie sogar, so ist die Schranke positiv bzw. gleich null. Je größer die Belastungsminderung, desto besser. Die Zielfunktion soll also minimiert werden.

Aus den beschriebenen Nebenbedingungen, Variablen und der Zielfunktion ergibt sich insgesamt das folgende Modell:

$$\min \quad \sum_{i \in I} \sum_{j \in J} D_{i,j}, \quad (4.19)$$

$$\text{bzgl.} \quad \sum_{t \in B} x_{s,t} = 1, \quad \forall s \in S, \quad (4.20)$$

$$d_{i,j,t} = \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} \cdot \delta_{s,i,j,(60 \cdot \tau - t)}, \quad \forall i \in I, j \in J, t \in T, \quad (4.21)$$

$$d_{i,j,t} - \Delta_{i,j} \leq D_{i,j}, \quad \forall i \in I, j \in J, t \in T, \quad (4.22)$$

$$d_{i,j,t} \leq \max\{\Delta_{i,j}, 100\}, \quad \forall i \in I, j \in J, t \in T, \quad (4.23)$$

$$x_{s,t} \in \{0, 1\}, \quad \forall s \in S, t \in B, \quad (4.24)$$

$$d_{i,j,t} \in \mathbb{R}, \quad \forall i \in I, j \in J, t \in T, \quad (4.25)$$

$$D_{i,j} \in \mathbb{R}, \quad \forall i \in I, j \in J. \quad (4.26)$$

Zum einen soll jede Schule zu genau einem Zeitpunkt ihren Unterricht beginnen, was durch (4.20) garantiert wird. Die temporäre Systembelastung eines jeden Pixels wird in (4.21) bestimmt. (4.22) dient der Festlegung der pixelindividuellen Obergrenzen für die Belastungsminderung, die maximale Verschlechterung der temporären Systembelastung der Pixel wird in (4.23) gesetzt. In (4.24) erfolgt die Zuordnung der Variablen x für die

Schulanfangszeiten als binär. Die Festlegung der Variablen $d_{i,j,t}$ sowie $D_{i,j}$ als reellwertig in (4.25) bzw. (4.26) lässt den Ansatz zu einem gemischt-ganzzahligen Modell werden. Faktisch ergeben sich für diese Variablen ohnehin ganzzahlige bzw. positiv ganzzahlige Werte, wodurch für das Modell ebenso ein ganzzahliger Ansatz gewählt werden könnte. Aus Laufzeitgründen wird jedoch dieser gemischt-ganzzahlige Ansatz gewählt.

4.3.2. Modell 2

Auch in Modell 2 wird ein gemischt-ganzzahliger Ansatz gewählt. Hier sollen die pixelindividuellen Belastungswerte mit einem optimalen Szenario verglichen werden. Das bedeutet, für jedes Pixel wird der Wert berechnet, der bei absoluter Entkoppelung der Schülerbewegungen der einzelnen Schulen, welche i.A. bei einem Zeitrahmen der Verschiebungen von einer Stunde nicht möglich sein muss, maximal erreicht wird. Die Vergleichsparameter ergeben sich also zu:

$$\Delta'_{i,j} = \max_{s \in S} (\max_{\tau \in R} \delta_{s,i,j,\tau}), \quad \forall i \in I, j \in J. \quad (4.27)$$

Auch in diesem Modell soll gemäß (4.23) zu jeder Zeit eine obere Schranke an die Pixelbelastungen gesetzt sein. Weiterhin stellt das Maximum der Belastung eines Pixels nach Verschiebung der Anfangszeiten ($d_{i,j}$) eine obere Schranke für alle Zeitpunkte dar, d.h.

$$d_{i,j,t} \leq d_{i,j}, \quad \forall i \in I, j \in J, t \in T. \quad (4.28)$$

Diese pixelbezogenen Maximalwerte werden dann mit den Maximalwerten des optimalen Szenarios verglichen.

$$d_{i,j} - \Delta'_{i,j} = m_{i,j}, \quad \forall i \in I, j \in J. \quad (4.29)$$

Als Zielfunktion soll nun das Maximum der Werte $m_{i,j}$ dienen. Dieses ergibt sich durch die folgende Nebenbedingung:

$$m_{i,j} \leq M, \quad \forall i \in I, j \in J. \quad (4.30)$$

Zusammengefasst ergeben (4.29) und (4.30) die folgende Nebenbedingung:

$$d_{i,j} - \Delta'_{i,j} \leq M, \quad \forall i \in I, j \in J. \quad (4.31)$$

Die absolute Belastung wird wie im vorherigen Modell mithilfe der relativen Belastungswerte berechnet. Jede Schule soll ihren Unterricht auch in diesem Modell zu genau einer Zeit beginnen. Insgesamt ergibt sich damit das folgende Modell:

$$\min \quad M, \quad (4.32)$$

$$\text{bzgl.} \quad \sum_{t \in B} x_{s,t} = 1, \quad \forall s \in S, \quad (4.33)$$

$$d_{i,j,t} = \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} \cdot \delta_{s,i,j,(60 \cdot \tau - t)}, \quad \forall i \in I, j \in J, t \in T, \quad (4.34)$$

$$d_{i,j,t} \leq \max\{\Delta_{i,j}, 100\}, \quad \forall i \in I, j \in J, t \in T, \quad (4.35)$$

$$d_{i,j,t} \leq d_{i,j}, \quad \forall i \in I, j \in J, t \in T, \quad (4.36)$$

$$d_{i,j} - \Delta'_{i,j} \leq M, \quad \forall i \in I, j \in J, \quad (4.37)$$

$$x_{s,t} \in \{0, 1\}, \quad \forall s \in S, t \in B, \quad (4.38)$$

$$d_{i,j,t} \in \mathbb{R}, \quad \forall i \in I, j \in J, t \in T, \quad (4.39)$$

$$d_{i,j} \in \mathbb{R}, \quad \forall i \in I, j \in J, \quad (4.40)$$

$$M \in \mathbb{R}. \quad (4.41)$$

Der Zielfunktionswert dieses Modells ermöglicht eine bessere Interpretation als in Modell 1. Er gibt den maximalen Zuwachs der Belastung (in Schülern) im Vergleich zum Optimalzenario innerhalb des Systems an, welcher zu mindestens einem Zeitpunkt in mindestens einem Pixel zu finden ist.

4.3.3. Modell 3

In Modell 3 soll das Optimierungskriterium die Gesamtsystembelastung C , d.h. die aufsummierte Belastung aller Pixel zu bestimmten Zeiten, sein. Zu jeder Zeit $t \in T$ soll der Wert der Gesamtsystembelastung so gering wie möglich gehalten werden. Ziel ist dann, den maximalen Wert über alle Zeiten zu minimieren. Die sonstigen Nebenbedingungen können aus Modell 1 bzw. 2 übertragen werden. Schließlich ergibt sich das folgende Modell:

$$\min \quad C, \quad (4.42)$$

$$\text{bzgl.} \quad \sum_{t \in B} x_{s,t} = 1, \quad \forall s \in S, \quad (4.43)$$

$$d_{i,j,t} = \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} \cdot \delta_{s,i,j,(60 \cdot \tau - t)}, \quad \forall i \in I, j \in J, t \in T, \quad (4.44)$$

$$d_{i,j,t} \leq \max\{\Delta_{i,j}, 100\}, \quad \forall i \in I, j \in J, t \in T, \quad (4.45)$$

$$\sum_{i \in I, j \in J} d_{i,j,t} \leq C, \quad \forall t \in T, \quad (4.46)$$

$$x_{s,t} \in \{0, 1\}, \quad \forall s \in S, t \in B, \quad (4.47)$$

$$d_{i,j,t} \in \mathbb{R}, \quad \forall i \in I, j \in J, t \in T, \quad (4.48)$$

$$C \in \mathbb{R}. \quad (4.49)$$

Auch in diesem Modell soll jede Schule zu genau einer Zeit ihren Unterricht beginnen (4.43). Die Berechnung der absoluten Belastungswerte (4.44) bzw. der oberen Schranke an die Belastungsänderung (4.45) sind ebenfalls dieselben wie für die anderen Modelle. Bedingung (4.46) zwingt C auf den Wert der maximalen Systembelastung nach Verschiebung der Schulanfangszeiten, welcher gleichzeitig für die Zielfunktion (4.42) verwendet wird. Die Variablenzuordnung (4.47)-(4.49) ergibt auch in diesem Modell einen gemischt-ganzzahligen Ansatz.

5. Ergebnisse und Analyse

Die Berechnungen der folgenden Ergebnisse wurden auf einem Rechner mit dem Betriebssystem Linux, 256 GB RAM und Intel Xeon E5-2667 Multicore-Prozessoren (16· 8 Prozessorkerne) mit 3,3 GHz durchgeführt.

Die entwickelten Optimierungsmodelle wurden auf die folgenden Instanzen angewandt, welche mittels Instanzengenerator erstellt wurden. Die Seitenlänge der Pixel wurde zur Optimierung aller Instanzen, soweit nicht anders vermerkt, auf 50 Meter gesetzt. Tabelle 5.1 zeigt die Parameter zu diesen Instanzen.

Instanz	Größe ($x_{\max} \times y_{\max}$)	Anzahl Schulen	Anzahl Schüler
1	6 km x 4 km	5	2.297
2	5 km x 5 km	4	2.472
3	5 km x 8 km	8	3.719
4	6 km x 6 km	6	3.299
5	4 km x 4 km	10	6.061
6	5 km x 5 km	15	6.197
7	5 km x 5 km	24	13.900
8	7 km x 6 km	40	26.643

Tabelle 5.1.: Übersicht der Parameter der Instanzen.

5.1. Ergebnisse

Bevor CPLEX mit dem Lösen beginnt, wird durch bereits beschriebene presolve-Verfahren versucht, die Größe des ursprünglichen Problems (MIP) auf ein kleineres Problem (*reduced MIP*) zu überführen. Nachfolgend sind die Anzahl der Nebenbedingungen $|NB|$, der Variablen $|V|$ und die Anzahl der darin enthaltenen Binärvariablen $|B|$ der Probleme nach Anwendung der presolve-Verfahren für die Modelle aufgelistet.

Instanz	Modell 1			Modell 2			Modell 3		
	NB	V	B	NB	V	B	NB	V	B
1	58.407	3.451	65	1.405	66	65	80	56	55
2	8.845	736	235	3.537	62	52	105	49	48
3	89.030	2.952	104	4.500	105	104	90	91	90
4	131.529	3.279	78	8.816	66	65	129	111	76
5	99.843	3.199	139	9.928	131	130	242	264	127
6	255.089	5.631	213	47.004	267	195	326	373	193
7	374.991	7.367	312	105.913	1.454	312	1.617	760	312
8	-	-	-	183.360	1.494	520	2.029	1.373	513

Tabelle 5.2.: Problemgrößen der Instanzen.

Die mit den Modellen 2 und 3 erzeugten Probleme beinhalten deutlich weniger Nebenbedingungen und Variablen als die Probleme, die mit Modell 1 erzeugt wurden. Je nach Problemgröße benötigt der Solver unterschiedlich lange, um ein Optimum für die Instanzen zu finden. Die berechneten optimalen Schulanfangszeiten der Schulen sind in Anhang A aufgelistet. Tabelle 5.3 zeigt den Zielfunktionswert des jeweiligen Modells vor der Optimierung sowie im Lösungsvektor des Modells. Die jeweils letzte Spalte zeigt die vom Solver benötigte Laufzeit zur Berechnung dieser Lösung in Sekunden gerundet auf zwei Nachkommastellen.

Instanz	Modell 1			Modell 2			Modell 3		
	z_{ist}	$z(x_1^*)$	t (in s)	z_{ist}	$z(x_2^*)$	t (in s)	z_{ist}	$z(x_3^*)$	t (in s)
1	0	-161	8,53	6	0	13,33	2.297	806	2,13
2	0	-1.266	63,45	42	0	10,87	2.472	822	3,09
3	0	-833	1.184,65	22	0	15,02	3.719	1.161	3,94
4	0	-1.670	12.387,88	26	0	19,83	3.299	1.143	4,61
5	0	-2.299	1.915,51	39	0	10,44	6.061	1.434	9,83
6	0	-6.370	80.910,82	58	2	265,54	6.197	1.774	9,99
7	0	-	-	301	7	10.523,48	13.900	4.026	58,25
8	0	-	-	239	7	13.559,16	26.643	7.608	815,82

Tabelle 5.3.: Laufzeiten und Zielfunktionswerte der Instanzen.

Zu sehen ist, dass die Laufzeit des Solvers bei Modell 1 mit der Größe der Instanzen und wachsender Problemgröße (Tabelle 5.2) deutlich steigt. Da die Berechnung von Instanz 6 mit Modell 1 bereits über 22 Stunden in Anspruch nimmt, wurde sie für Instanz 7 nach knapp 47 Stunden bei einer *optimality gap* von 348% abgebrochen. Die Schranken für den Zielfunktionswert der Lösung befanden sich zu diesem Zeitpunkt bei -16.832 bzw. -75.407. Modell 2 hingegen zeigt bei allen Instanzen eine geringere Laufzeit, da auch die Probleme von deutlich geringerer Größe sind als bei Modell 1. Auf die Problemgröße wirken sich neben der Modellformulierung sowohl die Anzahl der Schulen, die Anzahl der

Schüler, als auch die Fläche der Instanz bzw. die betrachtete Rasterfeinheit aus. Der Zielfunktionswert in Modell 2 ist für die Instanzen 1-5 null. Das heißt, die Schülerbewegungen der einzelnen Schulen konnten in diesen Beispielen vollständig entkoppelt werden. Dies ist möglich, da die getesteten Instanzen relativ klein sind bzw. wenige Schulen beinhalten. Bei größeren Instanzen wie z.B. den Instanzen 6, 7 oder 8 ist eine vollständige Entkopplung bereits nicht mehr möglich, dennoch konnte der Zielfunktionswert durch die Optimierung in diesen Instanzen um mehr als 95% gesenkt werden. Die geringste Laufzeit weist Modell 3 auf. Zwar enthalten die Probleme für Modell 3 eine nahezu identische Anzahl an (Binär)Variablen wie die Probleme für Modell 2, die Anzahl der Nebenbedingungen ist jedoch deutlich geringer, woraus eine geringere Laufzeit resultiert. Der Zielfunktionswert wurde im Vergleich zur Momentansituation ebenfalls deutlich gesenkt. Zu beachten ist allerdings, dass der Zielfunktionswert in Modell 3 der maximalen Gesamtsystembelastung entspricht und somit niemals null werden kann, er ist nach unten beschränkt durch die Schüleranzahl der größten Schule der Instanz.

Setzt man die Lösung eines Modells in die Zielfunktion des jeweils anderen Modells ein, so erhält man für kleine Instanzen mit den verwendeten Parametern denselben Zielfunktionswert wie im Optimum des Modells selbst. Die Modelle berechnen jedoch nicht generell äquivalente Lösungen, auch das ist der Größe der Instanzen bzw. der Wahl der Parameter geschuldet. Bei größeren Instanzen wie z.B. Instanz 6 stimmen diese Werte nicht mehr überein.

5.2. Analyse der Ergebnisse

In Modell 1 erlaubt die Zielfunktion keine weitere Interpretation der tatsächlichen Situationsverbesserung. Bedingung (4.23) garantiert lediglich, dass sich die Belastung jedes Pixels zu jedem Zeitpunkt im Vergleich zum Maximalwert vor der Optimierung nicht verschlechtert, bzw. den Wert 100 nicht übersteigt. Wie sich die Belastung im Vergleich zur zeitspezifischen Situation vor der Optimierung verhält, ist am Zielfunktionswert nicht abzulesen. Der Zielfunktionswert aus Modell 2 entspricht dem maximalen Zuwachs der Belastung im Vergleich zum Maximalwert eines optimalen Szenarios, der innerhalb des gesamten betrachteten Zeitraums in mindestens einem Pixel auftritt. Zur Bewertung der Ergebnisse ist eine Gegenüberstellung der Ist- und Soll-Werte der Gesamtsystembelastung zu den absoluten Zeiten wie in Modell 3 aussagekräftiger. Die Gesamtsystembelastung B_t ergibt sich wie folgt:

$$B_t = \sum_{i \in I, j \in J} d_{i,j,t}, \quad \forall t \in T. \quad (5.1)$$

Abbildung 5.1 zeigt diesen Vergleich beispielhaft für die Instanzen 1 und 5.

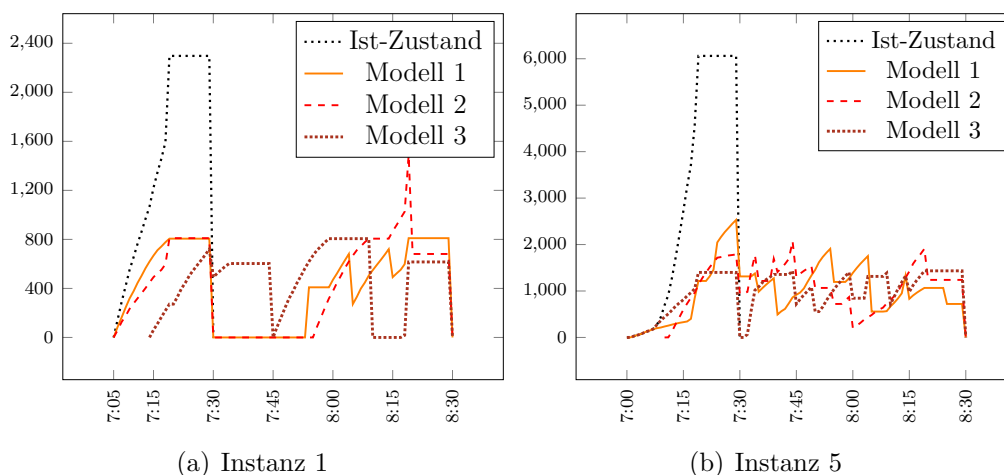


Abbildung 5.1.: Temporäre Gesamtsystembelastung mit und ohne Verschiebung der Schulanfangszeiten

Aus der Pixelgröße resultiert eine Taktung der Belastungsmessung von 60 Sekunden, d.h. jede Minute wird die Anzahl der aktiven Schüler überprüft. Im Gegensatz zum Ist-Zustand ist der Pegel der Belastung nach Verschiebung der Anfangszeiten in beiden Beispielen deutlich ausgeglichener. Es ist nun kein singulärer Spitzenwert mehr zu finden, stattdessen sind über den Zeitraum von ca. 1,5 Stunden mehrere Anstiege zu erkennen, die sich jedoch im Vergleich zur Ausgangssituation alle im Rahmen halten. Der Spiegel fällt immer dann rasant ab, wenn eine oder mehrere Schulen ihren Unterricht beginnen. Die Phasen bei Instanz 1, in denen die Belastung null ist, resultieren aus der fehlenden oberen Schranke an die Summe der Verschiebungen. In beiden Beispielen wird der vorgegebene Rahmen für die Schulanfangszeiten von einer Stunde voll ausgeschöpft. Tabelle 5.5 stellt die Maximalwerte $B_{max} = \max_{t \in T} B_t$ und Durchschnittswerte $B\emptyset$ der durch die Modelle erhaltenen Situationen den Werten des Momentanzustandes gegenüber. Zur Berechnung des Durchschnittswertes wird die minimale Zeit $t_{min} \in T$ und die maximale Zeit $t_{max} \in T$ benötigt, zu der Schüler im System aktiv sind, welche von der Lösung abhängen.

Definition 5.1. Sei x^* der Lösungsvektor einer Instanz mit n Schulen, d.h.

$$x^* = (x_{1,450}, \dots, x_{1,510}, \dots, x_{n,450}, \dots, x_{n,510}). \quad (5.2)$$

Dann sind t_{min} und t_{max} folgendermaßen definiert:

$$t_{min} = \min_{s \in S} \left(\sum_{t \in B} 60 \cdot x_{s,t} - 60 \cdot \sigma_s \right), \quad (5.3)$$

$$t_{max} = \max \{ t \mid x_{s,t} = 1, s \in S, t \in B \}. \quad (5.4)$$

Somit ergibt sich die Durchschnittsbelastung $B\emptyset$:

$$B\emptyset = \left[\sum_{\substack{t \in T \\ t \in [t_{min}, t_{max}]}} B_t \cdot \frac{60}{(t_{max} - t_{min})} \right] \quad (5.5)$$

Instanz	Ist-Zustand		Modell 1		Modell 2		Modell 3	
	B_{max}	$B\emptyset$	B_{max}	$B\emptyset$	B_{max}	$B\emptyset$	B_{max}	$B\emptyset$
1	2.297	1.526	810	436	1.487	436	806	489
2	2.472	1.479	1.036	540	1.036	611	822	504
3	3.719	1.804	1.554	771	1.770	771	1.161	804
4	3.299	1.364	1.864	700	1.864	725	1.143	694
5	6.061	3.075	2.530	1.002	2.073	1.117	1.434	1.002
6	6.197	2.843	2.533	1.312	2.419	1.312	1.774	1.264
7	13.900	6.400	-	-	5.607	3.266	4.026	2.909
8	26.643	11.115	-	-	12.445	5.855	7.608	5.511

Tabelle 5.4.: Belastungswerte der Instanzen vor und nach Verschiebung der Schulanfangszeiten.

Die durchschnittliche Systembelastung wurde bei allen Instanzen um 50-70% reduziert. Auch die Maximalbelastung, welche innerhalb des gesamten Zeitraumes auftritt, wird durch die Modelle in nahezu allen Beispielen um mehr als 50% verringert. Die Werte, die durch die Modelle erreicht werden, sind relativ ähnlich. Zwischen Modell 1 und 2 ist eine Tendenz, dass die Lösungen eines Modells die des anderen dominieren, nicht zu erkennen. Die Lösungen aus Modell 3 liefern jedoch in beinahe allen Instanzen eine geringere Durchschnitts- und Maximalbelastung als die Lösungen aus Modell 1 bzw. 2.

5.3. Analyse mittels Pareto-Optimierung

Die Ergebnisse haben gezeigt, dass in der Theorie deutliche Minderungen der Systembelastung möglich sind. In der Praxis sind jedoch noch einige andere Faktoren zu berücksichtigen, die die Lösung beeinflussen können. Da das Ziel des Modells eine Situationsverbesserung mittels Verschiebung der Schulanfangszeiten ist, hängt das Ergebnis signifikant davon ab, ob und in welchem Maße Verschiebungen der Anfangszeiten überhaupt zulässig sind. Sind keine Verschiebungen erlaubt, so lässt sich der Wert der Systembelastung nicht mit dieser Methode verringern. Für eine Analyse der Sensitivität der Lösung im Zusammenhang mit den zulässigen Verschiebungen eignet sich die Erstellung der Pareto-Front. Zunächst wird dazu das Modell gemäß Definition 3.6 formuliert, diesmal mit zwei Zielfunktionen: Zielfunktion 1 gemäß der bisherigen Definition der Zielfunktion und Zielfunktion 2 als Summe aller Verschiebungen vom Ist-Zustand in Minuten. Beide sind so gering wie

möglich zu halten. Dafür wird Zielfunktion 1 als Zielfunktion des Modells verwendet und Zielfunktion 2 wie eine Nebenbedingung in das Modell eingebaut und durch eine Konstante beschränkt. Durch dieses Vorgehen ergibt sich für Modell 1 folgendes erweitertes Modell:

$$\min \quad \sum_{i \in I} \sum_{j \in J} D_{i,j}, \quad (5.6)$$

$$\text{bzgl.} \quad \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} |t_{ist,s} - \tau| \leq c, \quad (5.7)$$

$$\sum_{t \in B} x_{s,t} = 1, \quad \forall s \in S, \quad (5.8)$$

$$d_{i,j,t} = \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} \cdot \delta_{s,i,j,(60 \cdot \tau - t)}, \quad \forall i \in I, j \in J, t \in T, \quad (5.9)$$

$$d_{i,j,t} - \Delta_{i,j} \leq D_{i,j}, \quad \forall i \in I, j \in J, t \in T, \quad (5.10)$$

$$d_{i,j,t} \leq \max\{100, \Delta_{i,j}\}, \quad \forall i \in I, j \in J, t \in T, \quad (5.11)$$

$$x_{s,t} \in \{0, 1\}, \quad \forall s \in S, t \in T, \quad (5.12)$$

$$d_{i,j,t} \in \mathbb{R}, \quad \forall i \in I, j \in J, t \in T, \quad (5.13)$$

$$D_{i,j} \in \mathbb{R}, \quad \forall i \in I, j \in J. \quad (5.14)$$

In (5.7) wird die zweite Zielfunktion durch eine Konstante (maximaler Wert der Summe aller Verschiebungen in Minuten) kontrolliert, wobei $t_{ist,s}$ die aktuelle Anfangszeit (i.A. 7:30 Uhr) einer jeweiligen Schule s beschreibt. Analog zu den zulässigen Verschiebungen der Anfangszeiten sollte auch c ein Vielfaches von fünf sein. Um nun die Pareto-Front zu erhalten, müssen alle möglichen Werte für c eingesetzt werden, woraus Tupel von Zielfunktionswerten,

$$(\text{Summe aller Verschiebungen in Minuten (Wert ZF 2), Wert ZF 1}), \quad (5.15)$$

entstehen. Die Summe aller Verschiebungen wird minimal, wenn bei keiner Schule die Anfangszeit verschoben wird, dann ist der Wert von ZF_2 Null, woraus folgt, dass c größer oder gleich Null sein muss. Der maximale Wert für c ergibt sich, wenn alle Schulen um die maximal zugelassene Zeit verschoben würden, also jeweils um eine Stunde, oder im Fall von individuellen Verschiebungsrahmen, das jeweilige Maximum. Für den allgemeinen Fall ergibt sich eine obere Schranke für c : $60 \cdot |S| = 60 \cdot n$. Die Berechnung kann u.U. jedoch schon früher abgebrochen werden, genau dann, wenn der Zielfunktionswert von ZF_1 bereits denselben Wert hat wie das unikriterielle Problem, da dieser immer kleiner oder gleich dem Zielfunktionswert ist, der beim multikriteriellen Problem erreicht wird.

Ist also x^* die Lösung des Problems ohne (5.7), so ergibt sich für c :

$$0 \leq c = 5k \leq \sum_{s \in S} \sum_{\tau \in T} x_{s,\tau}^* \cdot |t_{ist} - \tau|, \quad k \in \mathbb{N}_0. \quad (5.16)$$

Da die Lösung von der Wahl des Rasters abhängt, hängt auch die Gestalt der Pareto-Front davon ab. Abbildung 5.2 zeigt einige mit Modell 1 berechneten Pareto-Fronten für Instanz 5. Aufgrund des hohen Rechenaufwands wurde die Pixelgröße schrittweise verkleinert. Auch die Optimallösungen der unikriteriellen Probleme sind in der Abbildung zu finden. Zu lösen sind maximal 120 Teilprobleme, ab Erreichen des optimalen Zielfunktionswertes der unrestringierten Aufgabe kann die Berechnung jedoch abgebrochen werden. Der Algorithmus erstellt und löst bei jeder Veränderung der Konstante c ein neues Problem. Ist der Wert von Zielfunktion 1 für einige Probleme derselbe, so ist lediglich die Lösung mit dem geringsten Wert der Summe aller Verschiebungen Teil des Pareto-Sets, da diese die übrigen Lösungen dominiert.

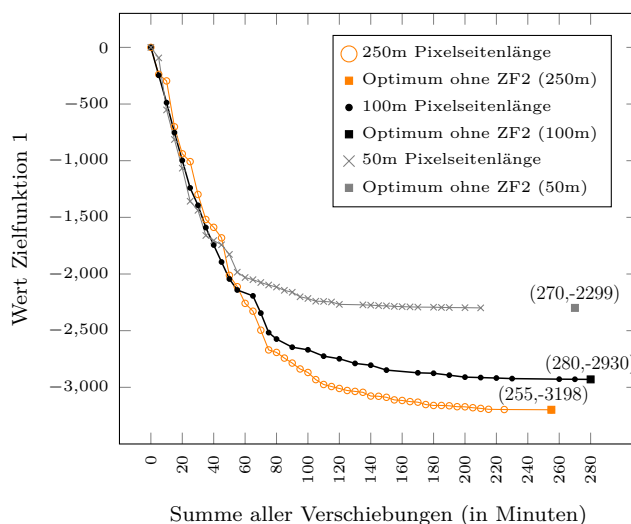


Abbildung 5.2.: Modell 1: Pareto-Fronten bei variierender Pixelgröße.

Zu sehen ist, wie der Wert der ersten Zielfunktion von den zulässigen Gesamtverschiebungen abhängt. Schränkt man diese ein, so muss man mit Einbußen in Form einer Vergrößerung des Zielfunktionswertes in Zielfunktion 1 rechnen, da die beiden Zielfunktionen konfliktär sind. Analog gilt dieser Zusammenhang auch umgekehrt. Beide Kriterien können relativ gut am „Ausgleichspunkt“ erfüllt werden, dabei sind insgesamt Verschiebungen von ungefähr 40-60 Minuten notwendig, was im Vergleich zum Optimalpunkt des unbeschränkten Systems nur einem kleinen Bruchteil der dort notwendigen Verschiebungen entspricht. Gleichzeitig verschlechtert sich der Wert der ersten Zielfunktion nur um etwa 20%. Für ein reales System mit zusätzlichen Bedingungen an die möglichen Verschiebungen kann eine solche Optimierung also dennoch durchaus rentabel sein, solange die

Einschränkungen nicht zu scharf werden. Bei einer Pixelseitenlänge von 50 Metern ist zu erkennen, dass die ohne Einschränkung der Verschiebungen berechnete Lösung nicht die Beste mit diesem Zielfunktionswert sein muss. Das heißt, diese Lösung ist nicht immer eine Lösung aus dem Pareto-Set. Bereits mit Lösungen, die Gesamtverschiebungen von 210 Minuten verlangen, lässt sich der Optimalwert -2.299 erreichen. Das uniobjektive Modell errechnet eine optimale Lösung, welche Verschiebungen von 270 Minuten verlangt.

Analog ergibt sich für Modell 2 das erweiterte Modell:

$$\min \quad M, \quad (5.17)$$

$$\text{bzgl.} \quad \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} |t_{ist,s} - \tau| \leq c, \quad (5.18)$$

$$\sum_{t \in B} x_{s,t} = 1, \quad \forall s \in S, \quad (5.19)$$

$$d_{i,j,t} = \sum_{s \in S} \sum_{\tau \in B} x_{s,\tau} \cdot \delta_{s,i,j,(60 \cdot \tau - t)}, \quad \forall i \in I, j \in J, t \in T, \quad (5.20)$$

$$d_{i,j,t} \leq \max\{\Delta_{i,j}, 100\}, \quad \forall i \in I, j \in J, t \in T, \quad (5.21)$$

$$d_{i,j,t} \leq d_{i,j}, \quad \forall i \in I, j \in J, t \in T, \quad (5.22)$$

$$d_{i,j} - \Delta'_{i,j} \leq M, \quad \forall i \in I, j \in J, \quad (5.23)$$

$$x_{s,t} \in \{0, 1\}, \quad \forall s \in S, t \in B, \quad (5.24)$$

$$d_{i,j,t} \in \mathbb{R}, \quad \forall i \in I, j \in J, t \in T, \quad (5.25)$$

$$d_{i,j} \in \mathbb{R}, \quad \forall i \in I, j \in J, \quad (5.26)$$

$$M \in \mathbb{R}. \quad (5.27)$$

Das folgende Diagramm zeigt die mit Modell 2 errechneten Pareto-Fronten für Instanz 5. Hierbei wurde die Summe aller Verschiebungen ebenfalls schrittweise um den Wert 5 verringert.

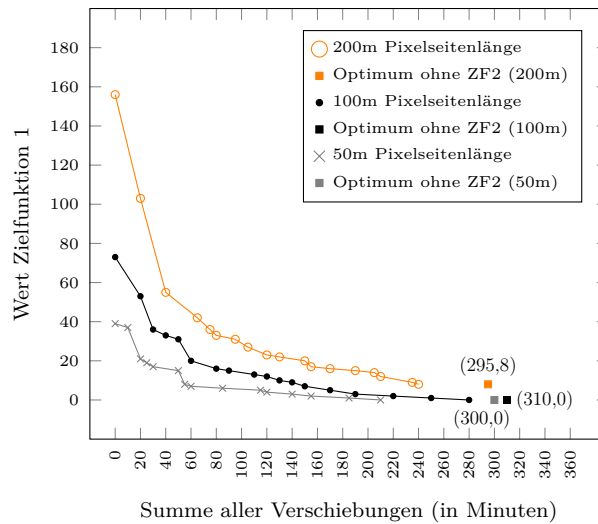


Abbildung 5.3.: Modell 2: Pareto-Fronten bei variierender Pixelgröße.

Die Pareto-Fronten für Modell 2 sind im Vergleich zu Modell 1 weniger dicht besiedelt. Um eine bessere Lösung zu erhalten, muss in diesem Modell also auch die Schranke der zulässigen Verschiebungen entsprechend erhöht werden. Auch hier sind die Optima der monokriteriellen Probleme, bei denen für eine Pixelseitenlänge von 50 Metern 300 Minuten, bei 100 Metern Pixellänge 310 Minuten und bei 200 Metern 295 Minuten Verschiebungen nötig sind, nicht Teil der Pareto-Menge.

Zu bemerken ist jedoch der hohe Rechenaufwand für die Erstellung einer Pareto-Front. Die Anzahl der zugelassenen Verschiebungen in (5.7) wird schrittweise, ausgehend von den nötigen Verschiebungen des monokriteriellen Problems, um 5 Minuten verringert. Für den Fall, dass für alle Schulen dieselben Obergrenzen gültig sind, müssen bei einer Anzahl von n Schulen bereits maximal $n \cdot 12$ Modelle erstellt und gelöst werden. Haben die Schulen individuelle Anforderungen an die zulässigen Verschiebungen, kann die Anzahl der maximal nötigen Modelle entsprechend steigen bzw. sinken. Aus diesem Grund wurde auf die Berechnung der Pareto-Fronten für Modell 3 verzichtet.

5.4. Vergleich mit dem Modell von Wenzel

Für einen sinnvollen Vergleich der Ergebnisse verschiedener Modelle ist zunächst der Detailgrad der Modelle entscheidend. Dieser ist bei den Modellen von Wenzel [22] den hier entwickelten Modellen ähnlich. Auch er verzichtet auf die spezielle Optimierung von Routen oder die Einbringung von Haltestellen, stattdessen wird über die Entfernung der Schulen und deren Schülerkapazitäten eine Art Spannungsnetzwerk aufgebaut. Diese Spannung soll dann minimiert werden. Eines seiner Modelle behandelt ebenfalls einen gemischt-ganzzahligen Ansatz. Die Ergebnisse dieses Modells, welches im Folgenden gemäß [22] reproduziert wird, sollen anschließend mit den durch Modell 1, 2 und 3 erhaltenen Ergebnissen verglichen werden.

Für Wenzels Modell, im Weiteren Modell 4 genannt, sind zunächst folgende Variablen und Parameter nötig: Die Menge S aller Schulen einer Instanz ist identisch mit der hier verwendeten Menge. Als Parameter dienen die Anzahl der Schüler, die eine jeweilige Schule besuchen sowie die Koordinaten der Schulen. Zur Darstellung der relativen zeitlichen Differenz zweier Schulanfangszeiten dient die Menge Z , die 25 mögliche 5-Minuten-Abstände enthält:

$$Z := \{-12, -11, \dots, 11, 12\}. \quad (5.28)$$

Das heißt, ist der Wert der Zeitdifferenz zwischen Schule s_1 und Schule s_2 3, so beginnt der Unterricht in Schule s_2 3·5 Minuten nach Schule s_1 . Die ganzzahlige Variable u_s beschreibt die Schulanfangszeit jeder Schule, sie wird ebenfalls zu 5-Minuten-Werten interpretiert:

$$u_s \in \{0, 1, \dots, 12\}, \quad \forall s \in S. \quad (5.29)$$

Es besteht folgender Zusammenhang mit der in den hier vorgestellten Modellen verwendeten Binärvariablen $x_{s,t}$:

$$u_s = \beta \in \{0, 1, \dots, 12\} \text{ beliebig} \Leftrightarrow x_{s,450+5\cdot\beta} = 1, \quad \forall s \in S. \quad (5.30)$$

Aufbauend auf der Menge Z ist für je zwei Schulen eine Variable D_{s_1, s_2} nötig, die den Zeitunterschied der Schulanfangszeiten im 5-Minuten-Maß beschreibt. Da dieser Zusammenhang im Absolutbetrag symmetrisch ist, ist diese Variable nur $\frac{n(n-1)}{2}$ mal nötig.

$$D_{s_1, s_2} \in \{-12, -11, \dots, 11, 12\}, \quad \forall s_1, s_2 \in S : s_1 < s_2. \quad (5.31)$$

Schließlich gibt es für je zwei Schulen die Binärvariable $x_{s_1, s_2, t}$, die beschreibt, ob die Anfangszeiten zweier Schulen s_1 und s_2 den Zeitunterschied t haben.

$$x_{s_1, s_2, t} = \begin{cases} 1, & \text{zwischen Schule } s_1 \text{ und Schule } s_2 \\ & \text{besteht der Zeitunterschied } t \quad \forall t \in Z, s_1, s_2 \in S : s_1 < s_2. \\ 0, & \text{sonst} \end{cases}$$

Die Nebenbedingungen ergeben sich durch die Zusammenhänge zwischen den Variablen. Der Zeitunterschied D_{s_1, s_2} ergibt sich aus der Differenz der Unterrichtsbeginnzeiten der Schulen:

$$D_{s_1, s_2} = u_{s_1} - u_{s_2}, \quad \forall s_1, s_2 \in S : s_1 < s_2. \quad (5.32)$$

Es muss zudem gewährleistet sein, dass jede Schule nur zu genau einer Uhrzeit beginnt, und somit der Zeitabstand zwischen zwei Schulen ebenfalls eindeutig sein muss, was durch die Bedingung (5.33) erfolgt:

$$\sum_{t \in Z} x_{s_1, s_2, t} = 1, \quad \forall s_1, s_2 \in S : s_1 < s_2. \quad (5.33)$$

Schließlich wird der Zusammenhang zwischen D_{s_1, s_2} und $x_{s_1, s_2, t}$ hergestellt:

$$\sum_{t \in Z} x_{s_1, s_2, t} \cdot t = D_{s_1, s_2}, \quad \forall s_1, s_2 \in S : s_1 < s_2. \quad (5.34)$$

Die Spannung zwischen jeweils zwei Schulen, die für die Zielfunktion verwendet wird, wird über eine Formel mithilfe der Exponentialfunktion beschrieben. Da in dieser Arbeit die Maßeinheit der Distanz $\text{dist}(s_1, s_2)$ zwischen zwei Schulen s_1 und s_2 nicht in Kilometern wie in Wenzels Modell, sondern in Metern aufgeführt wird, müssen die Koeffizienten der Formel für die Spannung angepasst werden. Ab einer Distanz von 7000 Metern soll die Exponentialfunktion nur noch einen Einfluss von 0.1 auf die Spannung haben [22]. Diese Bedingung führt auf den Koeffizienten $a=0,000329$. Koeffizient $b=0,2$ bleibt derselbe.

Somit ergibt sich die Formel für die Berechnung der Spannung zwischen zwei Schulen:

$$c_{s_1, s_2, t} = e^{-3,29 \cdot 10^{-4} \cdot \text{dist}(s_1, s_2)} \cdot e^{-0,2|t|} \cdot (a_{s_1} + a_{s_2}), \quad \forall t \in Z, s_1, s_2 \in S, \quad (5.35)$$

wobei a_s die Anzahl der Schüler beschreibt, die die Schule s besuchen, d.h. $a_s = \text{schueleranzahl}_s$.

Insgesamt ergibt sich das Modell:

$$\min \sum_{\substack{s_1, s_2 \in S: \\ s_1 < s_2}} x_{s_1, s_2, t} \cdot c_{s_1, s_2, t}, \quad (5.36)$$

$$\text{bzgl. } u_{s_1} - u_{s_2} = D_{s_1, s_2}, \quad \forall s_1, s_2 \in S : s_1 < s_2, \quad (5.37)$$

$$\sum_{t \in Z} x_{s_1, s_2, t} = 1, \quad \forall s_1, s_2 \in S : s_1 < s_2, \quad (5.38)$$

$$\sum_{t \in Z} x_{s_1, s_2, t} \cdot t = D_{s_1, s_2}, \quad \forall s_1, s_2 \in S : s_1 < s_2, \quad (5.39)$$

$$x_{s_1, s_2, t} \in \{0, 1\}, \quad \forall s_1, s_2 \in S, t \in Z, \quad (5.40)$$

$$D_{s_1, s_2} \in \{-12, -11, \dots, 11, 12\}, \quad \forall s_1, s_2 \in S : s_1 < s_2, \quad (5.41)$$

$$u_s \in \{0, 1, \dots, 12\}, \quad \forall s \in S. \quad (5.42)$$

Modell 4 wurde auf dieselben Instanzen wie die anderen Modelle angewandt. Die Lösungen dieses Modells sind ebenfalls in Anhang A zu finden. Tabelle 5.5 zeigt die resultierende Problemgröße, den errechneten Zielfunktionswert z sowie die Laufzeit t , die CPLEX zum Errechnen der optimalen Lösung benötigt.

Instanz	reduziertes MIP			t (in s)	z
	NB	V	B		
1	20	255	250	61	1.623
2	12	154	150	61	1.712
3	56	708	700	2955	4.680
4	30	381	375	138	3.511
5	90	1135	1125	-	-
6	210	2.640	2.625	-	-
7	552	6.924	6.900	-	-
8	1.560	19.540	19.500	-	-

Tabelle 5.5.: Modell 3: Problemgröße, Laufzeit und Zielfunktionswerte der Instanzen.

Die Berechnung von Instanz 5 wurde nach 100 Stunden bei einer *optimality gap* von 20,00% abgebrochen. Zu dieser Zeit hatte die obere Schranke für den Zielfunktionswert den Wert 14.404, die untere Schranke befand sich bei 11.495. Da die Instanzen 6 und 7 größer als Instanz 5 sind, wurde die Berechnung dieser beiden Instanzen nach 3 Stunden abgebrochen, da auch hier kein Ergebnis in akzeptabler Zeit zu erwarten war. Bei Abbruch nach 3 Stunden betrug die *optimality gap* 35,28% (Instanz 6) bzw. 49,40% (Instanz 7). Der Zielfunktionswert für Instanz 6 lag zu dieser Zeit zwischen 16.141 und 24.943, für Instanz 7 waren die aktuellen Schranken 52.373 bzw. 103.497.

Um die Ergebnisse mit den Ergebnissen aus den Modellen 1, 2 und 3 vergleichen zu können, bietet es sich auch hier an, den Lösungsvektor eines Modells in die Zielfunktion eines anderen Modells einzusetzen. Setzt man einen Lösungsvektor aus den Modellen 1-3 in Modell 4 ein, so sind alle Nebenbedingungen trivialerweise erfüllt, da die Nebenbedingungen aus Modell 4 direkt oder indirekt auch Bedingungen dieser Modelle sind. Tabelle 5.6 zeigt die Werte der Zielfunktion aus Modell 4 ohne Verschiebung der Anfangszeiten z_{ist} sowie durch Einsetzen der in den Modellen M_i , $i = 1, 2, 3$, erhaltenen Lösungen $x_{M_i}^*$.

Instanz	Zielfunktion Modell 3 (gerundet auf 0 Nachkommastellen)				
	z_{ist}	$z(x_{M_4}^*)$	$z(x_{M_1}^*)$	$z(x_{M_2}^*)$	$z(x_{M_3}^*)$
1	5.451	1.623	1.816	1.976	1.822
2	5.723	1.712	2.099	1.936	1.725
3	13.702	4.681	5.228	5.339	5.171
4	10.288	3.511	4.337	4.568	3.894
5	37.449	-	15.596	16.151	15.006
6	61.727	-	26.312	26.347	25.774
7	239.843	-	-	109.026	105.366
8	576.247	-	-	274.665	258.688

Tabelle 5.6.: Vergleich der Zielfunktionswerte für die verschiedenen Lösungen.

Bei allen Instanzen reduziert sich mit jeder der erhaltenen Lösungen der Zielfunktionswert des Modells im Vergleich zum Ist-Zustand deutlich. Die Zielfunktionswerte der Lösungen aus den Modellen 1-3 sind zwar geringfügig höher als die aus Modell 4, die Differenz zum Zielfunktionswert der optimalen Lösung aus Modell 4 ist jedoch gering. Modell 3 liefert im Sinne der Zielfunktion aus Modell 4 bessere Lösungen als die Modelle 1 und 2 für alle Instanzen.

Setzt man umgekehrt einen Lösungsvektor aus Wenzels Modell in Modell 1, 2 oder 3 ein, so muss Bedingung (4.23) bzw. (4.35) nicht zwingend erfüllt sein. Die folgende Tabelle zeigt die Ergebnisse dieses Vorgehens.

Instanz	Zielfunktion Modell 1		Zielfunktion Modell 2		Zielfunktion Modell 3	
	$z(\mathbf{x}_{M_1}^*)$	$z(\mathbf{x}_{M_4}^*)$	$z(\mathbf{x}_{M_2}^*)$	$z(\mathbf{x}_{M_4}^*)$	$z(\mathbf{x}_{M_3}^*)$	$z(\mathbf{x}_{M_4}^*)$
1	-161	-161	0	0	806	1.039
2	-1.266	-1.266	0	5	822	884
3	-833	-775	0	5	1.161	1.548
4	-1.670	-1.446	0	9	1.143	1.585
5	-2.299	-	0	-	1.434	-
6	-6.370	-	2	-	1.774	-
7	-	-	7	-	4.026	-
8	-	-	7	-	7.608	-

Tabelle 5.7.: Vergleich der Zielfunktionswerte für die verschiedenen Lösungen.

Auch dieser Fall verhält sich ähnlich. Die Unterschiede zwischen den Zielfunktionswerten der Lösungen sind relativ gering (im Vergleich zum Ist-Zustand). Diesmal liefern Modell 1 und 4 sowohl für Instanz 1 als auch für Instanz 2 eine äquivalente Lösung. Äquivalente Lösungen erzeugen auch Modell 2 und 4 für Instanz 1. Das heißt, die Modelle liefern in diesen Fällen Lösungen, die bezüglich ihrer Güte (im Sinne der verwendeten Zielfunktion) äquivalent sind. Die Lösungen sind zwar nicht identisch, sie liefern dennoch denselben Zielfunktionswert. Um zu entscheiden, welche Lösung zu bevorzugen ist, sind somit zusätzliche Kriterien zu berücksichtigen (vgl. Abschnitt 5.3).

6. Diskussion

Die Ergebnisse der Modelle konnten für die verwendeten Instanzen deutliche Entlastungen der durchschnittlichen und maximalen Systembelastung im Vergleich zur Momentansituation durch die Verschiebung der Schulanfangszeiten bewirken. Auch die Betrachtung der Pareto-Front konnte hervorbringen, dass die Optimierung mit den hier vorgestellten Ansätzen auch unter Berücksichtigung weiterer Vorgaben erfolgsversprechend sein kann, sofern diese nicht zu einschränkend sind. Um Leerlaufphasen, d.h. Phasen, in denen die Belastung null ist, zu verhindern, könnte in den Modellen eine obere Schranke an die Summe aller Verschiebungen gefordert werden. Weiterhin sind die behandelten Instanzen alle relativ klein, sodass die Schülerbewegungen in diesen Beispielen noch vollständig entkoppelt werden konnten. Für deutlich größere Beispiele muss dies nicht mehr der Fall sein.

Im Bezug auf die Ergebnisse ist außerdem zu berücksichtigen, dass der Ansatz sehr abstrahiert ist und viele wichtige Kriterien eines realen Sachverhalts nicht berücksichtigt. Dazu zählen zum Beispiel das Einbeziehen von Haltestellen, Zeitverzögerungen durch hohes Verkehrsaufkommen, Zuordnen der Schüler zu Haltestellen, Umsteigen sowie Warte- und Wendezeiten. Die potenziellen Einsparungen, die für die Systembelastung möglich sind, können daher nicht uneingeschränkt auf eine reale Situation übertragen werden.

Verbesserungswürdig ist auch der hohe Rechenaufwand. Vor allem die Berechnung der nötigen Parameter und Vergleichswerte ist sehr aufwendig. Zudem steigt die Laufzeit des Solvers mit steigender Anzahl an Schulen, Schülern, bzw. der Fläche der Instanz enorm. Da es sich um einen gemischt-ganzzahligen Ansatz handelt, ist dies jedoch zu erwarten.

Die Gegenüberstellung der Lösungen mit denen aus Modell 4 konnte die Ergebnisse bestätigen. Die Zielfunktionswerte waren für die Instanzen meist sehr ähnlich und teilweise sogar identisch. Zu den Ergebnissen ist jedoch anzumerken, dass die in Modell 4 berechneten Lösungen die Nebenbedingung, die den Zuwachs der Belastung innerhalb eines Pixels beschränkt, nicht zwangsläufig erfüllen. Das heißt, die Lösungen aus Modell 4 liefern zwar einen ähnlichen Zielfunktionswert, an die Belastung einzelner Pixel ist jedoch keine obere Schranke gesetzt, lediglich die Gesamtbelastung bzw. Spannung wird gesenkt. Bei identischem Zielfunktionswert ist daher eine Lösung aus den Modellen 1-3 zu bevorzugen, um auch die Belastung der einzelnen Pixel zu berücksichtigen. Für kleine Instanzen ist aufgrund des hohen Rechenaufwands zur Berechnung der für die in den Modellen 1-3 nötigen Parameter (aktive Schüler relativ zum Schulanfang, maximale Belastungswerte Ist-Situation bzw. Optimalsituation) dennoch Modell 4 vorzuziehen. Für größere Instan-

zen stößt Modell 4 jedoch schnell an seine Grenzen, da die Zahl der ganzzahligen bzw. binären Variablen und damit auch die Größe des Branch-Baumes signifikant wächst. Selbst nach langer Laufzeit konnte Modell 4 bereits für Instanz 5, welche 10 Schulen beinhaltet, keine optimale Lösung erzeugen. Die Modelle 1-3 enthalten deutlich weniger ganzzahlige bzw. binäre Variablen, woraus eine geringere Laufzeit resultiert. Modell 2 und insbesondere Modell 3 konnten auch größere Instanzen in akzeptabler Zeit lösen.

7. Fazit

Die durch die gemischt-ganzzahligen Ansätze berechneten Lösungen konnten die Auslastung innerhalb der Ballungsgebiete deutlich gleichmäßiger gestalten. Dabei sind, zumindest theoretisch, Entlastungen der durchschnittlichen Gesamtbelastung eines Systems von bis zu 70% möglich. Auch die maximale Gesamtbelastung konnte in fast allen Instanzen durch die Modelle um mehr als 50% reduziert werden. Dennoch ist zu berücksichtigen, dass die Ergebnisse aufgrund des Abstraktionsgrades der Modelle nicht uneingeschränkt auf eine reale Situation übertragbar sind. Bei einer so deutlichen Minderung der Belastung, wie sie durch die Modelle entstand, ist jedoch auch eine Besserung für reale Situationen denkbar.

Zur Verbesserung der Laufzeit könnte ein Dekompositionsverfahren verwendet werden, so dass auch größere Instanzen in kürzerer Zeit berechenbar sind. Die Laufzeit kann jedoch bis zu einem gewissen Grad damit gerechtfertigt werden, dass eine solche Berechnung in der Regel nur ungefähr einmal jährlich oder seltener durch die Einschulung neuer Schüler bzw. Schulwechsel anfällt, da die Positionen der Schulen und die Wohnorte der Schüler sich für gewöhnlich nicht ändern bzw. kleine Änderungen zu vernachlässigen sind.

Eine Weiterentwicklung auf Grundlage des entstandenen Ansatzes ist denkbar. Dazu könnte man künftig Haltestellen und Busrouten sowie andere Kriterien miteinbeziehen, um das Modell realitätsgetreuer zu gestalten. Auch kann das Modell für die Optimierung des Schülerverkehrs am Nachmittag angepasst und angewandt werden. Das punktuelle Aufkommen ist hier jedoch meist ohnehin geringer, da der zeitliche Rahmen, in dem die Schulen enden, sich über einen größeren Zeitraum als eine Stunde erstreckt.

Abschließend ist festzuhalten, dass die Einbringung der Schulanfangszeiten als Variablen bei der Entwicklung eines Optimierungsansatzes für den Schülertransport sehr erfolgversprechend ist. Da die Literatur im Bezug auf Modelle ohne die Möglichkeit der Verschiebung der Anfangszeiten schon ausgereifter ist, könnten die vorhandenen Modelle ebenfalls dahingehend erweitert werden, Verschiebungen zuzulassen, um so die Optimierung des Schülerverkehrs noch weiter voranzutreiben.

8. Literaturverzeichnis

- [1] N. BEUME, B. NAUJOKS, J. RUDOLPH, “Mehrkriterielle Optimierung durch Evolutionäre Algorithmen mit S-Metrik-Selektion”, *Conf. Proc. 16th GMA Workshop Computational Intelligence*, 2006.
- [2] S. BLUM, “Mehrzieloptimierung durch Evolutionäre Algorithmen”, *Weimarer Optimierungs- und Stochastiktag 2004*, 2004.
- [3] M. BÖGL, K.F. DOERNER, S.N. PARRAGH, “The school bus routing and scheduling problem with transfers”, *Networks*, Bd. 65, Nr. 2, S. 180-203, 2015.
- [4] R. BOWERMAN, B. HALL, P. CALAMAI, “A multi-objective optimization approach to urban school bus routing: Formulation and solution method”, *Transportation Research Part A: Policy and Practice*, Bd. 29, Nr. 2, S. 107-123, 1995.
- [5] J. BRACA, J. BRAMEL, B. POSNER, D. SIMCHI-LEVI, “A computerized approach to the New York Cityschool bus routing problem”, *IIR Transactions*, Bd. 29, Nr. 8, S. 693-702, 1997.
- [6] D.-S. CHEN, H.A. KALISEN, H.-C. CHEN, V.-C. TSENG, “A bus routing system for rural school districts”, *Computers & Industrial Engineering*, Bd. 19, Nr. 1-4, S. 322-325, 1990.
- [7] G.B. DANTZIG, J.H. RAMSER, “The Truck Dispatching Problem”, *Management Science*, Bd. 6, Nr. 1, S. 80-91, 1959.
- [8] A. DE PALMA, C. LEFÈVRE, “Bottleneck Models and Departure Time Problems”, *HAL*, 2018.
- [9] W. DOMSCHKE, A. DREXL, R. KLEIN, A. SCHOLL, “Einführung in Operations Research”, 9. Auflage, Gabler Verlag, 2015.
- [10] M. FISHER, “Chapter 1 Vehicle routing”, *Handbooks in Operations Research and Management Science*, Bd. 8, S. 1-33, in *Network Routing*, Elsevier, 1995.
- [11] A. FÜGENSCHUH, “Solving a school bus scheduling problem with integer programming”, *European Journal of Operational Research*, Bd. 193, Nr. 3, S. 867-884, 2009.
- [12] A. FÜGENSCHUH, A. MARTIN, C. MEHLERT, P. STÖVEKEN, “Ein Planungstool zur Schulzeitstaffelung”, in *Supply Chain Management und Logistik*, Physica-Verlag Heidelberg, S.419-436, 2005.
- [13] H.-J. HUANG, W.H.K. LAM, “Quasi-Continuous Dynamic Equilibrium Assignment with Departure Time Choice in Congested Unidirectional Pedestrian Networks”, *Journal of the Operational Research Society*, Bd. 53, Nr. 1, S.97-107, 2002.

-
- [14] IBM (01.10.2004), IBM Information describing the algorithms CPLEX uses, <https://www-304.ibm.com/support/docview.wss?uid=swg21400019>, (besucht am 06.04.2018).
- [15] J. KALLRATH, “Gemischt-ganzzahlige Optimierung: Modellierung in der Praxis; mit Fallstudien aus Chemie, Energiewirtschaft, Papierindustrie, Metallgewerbe, Produktion und Logistik”, 2.Auflage, Springer Spektrum Wiesbaden, S. 158-166, 2013.
- [16] H. KELLER, W. MÜLLER, “Optimierung des Schülerverkehrs durch gemischt-ganzzahlige Programmierung”, *Zeitschrift für Operations Research*, Bd. 23, Nr. 4, S. 105-122, 1979.
- [17] J. PARK, B.-I. KIM, “The school bus routing problem: A review”, *European Journal of Operational Research*, Bd. 202, Nr. 2, S. 311-319, 2010.
- [18] T. RALPHS, Y. SHINANO, T. BERTHOLD, T. KOCH, “Parallel Solvers for Mixed Integer Linear Programming”, *ZIB Report 16-74*, 2016.
- [19] P. SCHITTEKAT, J. KINABLE, K. SÖRENSEN, “A Mathematical Formulation for a School Bus Routing Problem”, *International Conference on Service Systems and Service Management 2006*, Bd. 2, S. 1552-1557, 2006.
- [20] P. SCHITTEKAT, J. KINABLE, K. SÖRENSEN, M. SEVAUX, F. SPIEKSMASMA, “A metaheuristic for the school bus routing problem with bus stop selection”, *European Journal of Operational Research*, Bd. 229, Nr. 2, S. 518-528, 2013.
- [21] W. VICKREY, “Optimization of Traffic and Facilities”, *Journal of Transport Economics and Policy*, Bd. 1, Nr. 2, S. 123-136, 1967.
- [22] C. WENZEL, “Optimale Schulanfangszeiten zur Entlastung des Nahverkehrs in der Stadt Nürnberg”, Bachelorthesis AMOS 42, Helmut-Schmidt-Universität Hamburg, 2016.
- [23] H.-J. ZIMMERMANN, “Operations Research: Methoden und Modelle ; für Wirtschaftsingenieure, Betriebswirte, Informatiker, 2. aktualisierte Auflage, Vieweg Wiesbaden, 2008.

Anhang A.

Optimale Schulanfangszeiten

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	8:30	7:30	8:30	8:30
2	7:30	7:30	8:20	8:10	7:30
3	7:30	8:30	7:30	7:45	7:50
4	7:30	8:05	8:30	8:30	8:30
5	7:30	8:15	8:30	7:30	8:10

Tabelle A.1.: Instanz 1: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	8:30	7:30	8:30	8:30
2	7:30	7:35	8:00	8:15	8:10
3	7:30	8:05	8:30	7:30	7:30
4	7:30	8:25	7:35	7:50	7:45

Tabelle A.2.: Instanz 2: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	7:35	7:35	8:30	7:30
2	7:30	8:05	8:30	8:30	7:30
3	7:30	7:55	8:30	8:10	7:40
4	7:30	8:30	7:30	7:30	8:30
5	7:30	7:40	8:10	8:05	8:10
6	7:30	7:30	8:30	7:45	7:30
7	7:30	7:45	8:15	8:30	8:25
8	7:30	8:25	7:30	7:50	7:55

Tabelle A.3.: Instanz 3: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	7:45	7:45	7:50	8:25
2	7:30	7:30	7:35	7:30	7:35
3	7:30	8:10	8:10	7:55	7:30
4	7:30	8:05	8:05	8:30	8:10
5	7:30	8:30	8:30	8:30	7:55
6	7:30	8:00	8:00	8:15	8:30

Tabelle A.4.: Instanz 4: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	8:15	7:35	8:00	-
2	7:30	8:25	7:55	7:30	-
3	7:30	7:35	7:50	8:15	-
4	7:30	7:30	7:40	7:30	-
5	7:30	8:05	8:20	7:30	-
6	7:30	8:05	8:30	7:50	-
7	7:30	7:40	7:30	8:10	-
8	7:30	8:30	8:00	8:30	-
9	7:30	7:55	8:30	8:30	-
10	7:30	7:30	7:45	7:45	-

Tabelle A.5.: Instanz 5: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	8:25	7:50	8:00	-
2	7:30	8:30	8:30	7:45	-
3	7:30	7:35	7:35	8:30	-
4	7:30	8:20	7:55	7:45	-
5	7:30	8:10	8:20	7:30	-
6	7:30	7:55	8:30	7:30	-
7	7:30	8:05	8:15	8:30	-
8	7:30	7:30	7:30	7:35	-
9	7:30	7:45	8:30	8:05	-
10	7:30	8:00	8:20	8:15	-
11	7:30	7:40	7:40	8:10	-
12	7:30	7:30	7:30	7:30	-
13	7:30	8:20	8:10	8:15	-
14	7:30	8:30	7:45	7:50	-
15	7:30	8:30	7:50	8:30	-

Tabelle A.6.: Instanz 6: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	-	7:35	7:50	-
2	7:30	-	8:05	8:15	-
3	7:30	-	8:30	8:05	-
4	7:30	-	7:40	7:50	-
5	7:30	-	7:30	7:45	-
6	7:30	-	8:20	8:30	-
7	7:30	-	8:30	7:40	-
8	7:30	-	7:50	7:55	-
9	7:30	-	7:50	7:40	-
10	7:30	-	8:10	7:45	-
11	7:30	-	8:30	8:10	-
12	7:30	-	8:25	8:30	-
13	7:30	-	8:20	8:30	-
14	7:30	-	8:00	8:10	-
15	7:30	-	7:30	7:30	-
16	7:30	-	8:05	8:30	-
17	7:30	-	7:35	7:30	-
18	7:30	-	7:55	8:05	-
19	7:30	-	7:45	7:30	-
20	7:30	-	8:00	8:30	-
21	7:30	-	8:10	7:30	-
22	7:30	-	8:20	7:30	-
23	7:30	-	8:20	8:30	-
24	7:30	-	8:30	8:00	-

Tabelle A.7.: Instanz 7: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Schule	Anfangszeit				
	Ist	Modell 1	Modell 2	Modell 3	Modell 4
1	7:30	-	8:30	7:30	-
2	7:30	-	7:35	8:00	-
3	7:30	-	7:30	8:30	-
4	7:30	-	7:40	7:55	-
5	7:30	-	8:25	8:30	-
6	7:30	-	7:35	7:30	-
7	7:30	-	8:15	8:30	-
8	7:30	-	8:20	8:10	-
9	7:30	-	8:10	8:15	-
10	7:30	-	8:00	7:45	-
11	7:30	-	8:15	8:10	-
12	7:30	-	8:00	8:30	-
13	7:30	-	8:15	8:30	-
14	7:30	-	8:05	7:45	-
15	7:30	-	7:40	7:30	-
16	7:30	-	7:50	8:05	-
17	7:30	-	8:25	7:45	-
18	7:30	-	8:00	7:45	-
19	7:30	-	7:35	7:40	-
20	7:30	-	7:30	7:30	-
21	7:30	-	8:30	7:45	-
22	7:30	-	7:50	7:50	-
23	7:30	-	8:00	7:45	-
24	7:30	-	8:20	7:50	-
25	7:30	-	7:40	8:30	-
26	7:30	-	8:05	7:30	-
27	7:30	-	8:00	7:30	-
28	7:30	-	8:30	8:30	-
29	7:30	-	8:30	7:55	-
30	7:30	-	8:30	8:15	-
31	7:30	-	7:55	7:35	-
32	7:30	-	7:50	8:00	-
33	7:30	-	8:05	8:30	-
34	7:30	-	7:55	8:05	-
35	7:30	-	8:15	7:35	-
36	7:30	-	8:10	7:50	-
37	7:30	-	8:30	8:30	-
38	7:30	-	7:40	7:30	-
39	7:30	-	7:30	7:30	-
40	7:30	-	8:05	8:30	-

Tabelle A.8.: Instanz 8: Vergleich der Schulanfangszeiten Ist- und Optimalwerte.

Anhang B.

Hinweise zum digitalen Anhang

Der digitale Anhang enthält alle Dateien, die zur Berechnung der Ergebnisse der Arbeit verwendet wurden. Für die Ausführung des Python-Programmes des jeweiligen Modells bzw. zur Berechnung der Pareto-Front ist die Installation von *IBM ILOG CPLEX Optimization Studio* sowie *Python 2.7* inkl. der Programmbibliothek *Matplotlib* zur Darstellung der Ballungsbewegungen notwendig. Der Instanzengenerator, die Programmdateien für Berechnungen mit den Modellen, die Programmdatei zur Berechnung einer Pareto-Front sowie die Parameterdatei sind im Ordner **Programme** zu finden. Zusätzlich enthält der Reiter **Ergebnisse** die Ergebnisse, Inputdateien und Ausgabedateien der in der Arbeit vorgestellten Instanzen, untergliedert in die jeweiligen Instanzen und Modelle.

Um eine Instanz zu generieren, müssen vorerst in der Datei **Parameter.py** die Parameter für die Instanz vorgegeben werden. In der Datei ist vermerkt, welche Daten vor und welche nach der Instanzenerzeugung noch verändert werden können. Anschließend kann die Programmdatei des Instanzengenerators (**Instanzenerzeugung.py**) aufgerufen werden, dabei muss die Parameterdatei mit übergeben werden. Der Instanzengenerator erzeugt dann die Instanz, welche in einem separaten Ordner unter dem Namen der aktuellen Zeit abgespeichert wird. In diesem Ordner befinden sich alle Dateien, die zur anschließenden Berechnung benötigt werden (*Arrays.py*, *Schuelerkoordinaten.txt*, *Parameter.py*).

Für die erstellten Instanzen können dann die Programme zur Berechnung der Modelle 1 und 2 (**Modell1.py**, **Modell2.py**) bzw. zur Berechnung der Pareto-Front (**Paretofront_Modell1.py**, **Paretofront_Modell2.py**) aufgerufen werden. Dazu muss sich die Programmdatei im Ordner der Instanz befinden (nach Generierung von Instanzen automatisch). Vor Aufruf der Programme können nochmals einige Parameter verändert werden: die Geschwindigkeit der Schüler, der Zeitrahmen, in dem sich die Schulanfangszeiten bewegen sollen, sowie die gewünschte Pixelgröße. Beim Aufruf der Modell-Programmdateien und der Pareto-Front muss ebenfalls die Parameterdatei übergeben werden. Alle Ergebnisdateien sowie die lp-Datei des Problems werden automatisch im Ordner der Instanz bzw. in entsprechend beschrifteten Unterordnern gespeichert.

Zur Berechnung von Modell 3 dient die Datei **modell_wenzel.zpl**, welche die Parameter aus der Datei *Schulkoordinaten.dat* einliest, die der Instanzengenerator erstellt. Die zpl-Datei muss zunächst in eine lp-Datei übersetzt werden (z.B. mit Zimpl) und kann anschließend von CPLEX gelöst werden.

Nachfolgend werden einige im Anhang befindlichen bzw. während der Berechnung erstellten Dateien noch genauer erläutert:

- **Arrays.txt** enthält zeilenweise für jede Schule deren x- und y-Koordinate, deren Einzugsradius, die Anzahl der Schüler, die diese Schule besuchen, sowie die aktuelle Anfangszeit der Schule
- **Belastungskurve.txt** enthält die Systembelastung vor und nach Verschiebung der Schulanfangszeiten zu den absoluten Zeiten
- **output.txt** (nur bei den Ergebnissen der vorgestellten Instanzen) enthält die Terminalausgabe des Rechenprozesses
- **solution.txt** enthält die Lösung bzw. die Werte der Variablen nach Lösen des Problems
- **modell_vergleich_wenzel.lp** ist identisch mit der Datei *modell.lp*, jedoch ohne die Bedingungen (4.23) in Modell 1 bzw. (4.35) in Modell 2. Sie dient zum Ermitteln des Zielfunktionswertes der Modelle 1 und 2 bei Einsetzen einer Lösung aus Modell 3. Da die Lösungen aus Modell 3 diese Bedingungen nicht zwangsläufig erfüllen, würde der Solver beim Einsetzen in die *modell.lp* Datei das Modell für unlösbar erklären, da einige Nebenbedingungen nicht erfüllt wären.

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Wörtlich und inhaltlich verwendete Quellen wurden entsprechend den anerkannten Regeln wissenschaftlichen Arbeitens zitiert. Die Arbeit ist nicht in gleicher oder vergleichbarer Form, auch nicht auszugsweise im Rahmen einer anderen Prüfung bei einer anderen Hochschule vorgelegt worden. Sie wurde bisher auch nicht veröffentlicht. Ich erkläre mich damit einverstanden, dass die Arbeit mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate überprüft wird.

Ort, Datum

Unterschrift

IMPRESSUM

Brandenburgische Technische Universität Cottbus-Senftenberg
Fakultät 1 | MINT - Mathematik, Informatik, Physik, Elektro- und Informationstechnik
Institut für Mathematik
Platz der Deutschen Einheit 1
D-03046 Cottbus

Professur für Ingenieurmathematik und Numerik der Optimierung
Professor Dr. rer. nat. Armin Fügenschuh

E fuegenschuh@b-tu.de
T +49 (0)355 69 3127
F +49 (0)355 69 2307

Cottbus Mathematical Preprints (COMP), ISSN (Print) 2627-4019
Cottbus Mathematical Preprints (COMP), ISSN (Online) 2627-6100

www.b-tu.de/cottbus-mathematical-preprints
cottbus-mathematical-preprints@b-tu.de
doi.org/10.26127/btuopen-4802