

Eine Heuristik für quadratische 0-1-Probleme

Matthias Forster

Zusammenfassung

Eine neue Modellformulierung für kombinatorische Optimierungsprobleme wird vorgestellt. Auf der Grundlage des quadratischen 0-1-Problems ohne Nebenbedingungen wird ein vorteilhafter Typ kombinatorischer Optimierungsprobleme eingeführt. Eine einfache Heuristik, die sog. RII-Methode (randomized iterative improvement), erlaubt es, diesen Modelltyp effizient zu lösen. Die Flexibilität des Ansatzes wird an der Modellierung des K-Färbungsproblems gezeigt. Es werden Rechenergebnisse für Färbungsprobleme aus der Literatur präsentiert. Wendet man Randomized Iterative Improvement auf den neuen Modelltyp an, können diese Testprobleme effizient gelöst werden.

Abstract

A new model formulation for combinatorial optimization problems is presented. Based on the unconstrained quadratic program (binary quadratic program), a favourable type of combinatorial optimization problems is introduced. A simple heuristic method, i.e. randomized iterative improvement (RII), permits to solve this type of model efficiently. Modelling of the K-colouring problem shows the flexibility of the approach. Computational results on data sets from the literature about vertex colouring are reported. These benchmark problems are shown to be solved efficiently using randomized iterative improvement and the new type of model.

1 Einleitung

Kombinatorische Optimierungsprobleme findet man in fast allen Teilgebieten der Betriebswirtschaftslehre. Zu den Anwendungen zählen bestimmte Fragen der Produktionsplanung, komplizierte Fälle der Projektplanung, Fragestellungen aus dem Bereich Investition und Finanzierung, Finanzmanagement, viele Problemstellungen der Logistik, Stundenplanung usw. Zuerst wird der Begriff »Kombinatorische Optimierung« definiert und eine Lösungsmethode angegeben. Dann wird ein wichtiger Modelltyp der kombinatorischen Optimierung behandelt, das quadratische 0-1-Problem. Diese Modelle umfassen zahlreiche Probleme der betrieblichen Praxis. Am Beispiel des K-Färbungsproblems soll ein neuer, verbesserter Lösungsansatz entwickelt werden.

Bei einem kombinatorischen Optimierungsproblem (combinatorial optimization problem, COP) ist eine Funktion f gegeben, die jedem Element s einer Menge S eine reelle Zahl zuordnet. S ist eine endliche Menge. Wählt man die Formulierung als Minimierungsproblem, dann ist

$$(\text{COP}) \min f(s) : s \in S, \quad \text{wobei } |S| \in \mathbb{N}$$

S bezeichnet die Menge der zulässigen Lösungen des Optimierungsproblems. f ist die Zielfunktion. Die Aufgabe besteht darin, ein Minimum $f(s^*)$ der Funktion f zu finden unter der Bedingung, dass das Argument s^* in der Menge S enthalten ist. Die Aufgabe der Minimumsuche stellt keine Beschränkung der Allgemeinheit dar, da jedes Maximierungsproblem in ein äquivalentes Minimierungsproblem verwandelt werden kann. Das Problem scheint auf den ersten Blick trivial zu sein, weil S eine endliche Menge ist. Es ist aber zu bedenken, dass S bei realen Problemen i. d. R. sehr viele Elemente enthält (typisch 10^{50} und mehr). Daher scheidet die Untersuchung aller zulässigen Lösungen (Vollenumeration) aus. Man benötigt intelligente Suchverfahren, die den tatsächlichen Aufwand der Minimumsuche wesentlich reduzieren. In der Praxis ist oft sogar ein kleiner Anteil tatsächlich zu betrachtender Lösungsfälle so groß, dass sich das wahre Optimum nicht finden lässt. In diesem Falle muss man sich mit einer Näherungslösung zufrieden geben. Im besten Falle ist es dann möglich, eine Schranke für den Abstand vom wahren Minimum anzugeben. Ein Rechenverfahren, das mit vertretbarem Aufwand eine gute Lösung für ein schwieriges, kombinatorisches Optimierungsproblem liefert, nennt man Heuristik.

2 Ein Verbesserungsverfahren

Eine gebräuchliche Lösungsmethode für (COP) ist das RII-Verfahren (randomized iterative improvement). Eine detaillierte Darstellung findet der Leser in Hoos/Stütze [1], S. 72 ff.

Abb. 1 zeigt das Flussdiagramm dieses Verbesserungsverfahrens: Sei $S = \{1, \dots, K\}^n$ und $K, n \in \mathbb{N}$. Dann beginnt man mit einer Startlösung $s \in S$, die im einfachsten Falle zufällig gewählt werden kann. Oft lässt sich im konkreten Anwendungsfall eine gute Startlösung konstruieren. Es folgt die iterative Verbesserung der Startlösung, bis ein sinnvolles Endekriterium erfüllt ist (z. B. Ende nach 10^7 Iterationen oder Ende, wenn 10^4 Iterationen keine Verbesserung gebracht haben o. ä.). In jeder Iteration wird entweder ein Zufallsschritt (a) oder ein Verbesserungsschritt (b) durchgeführt:

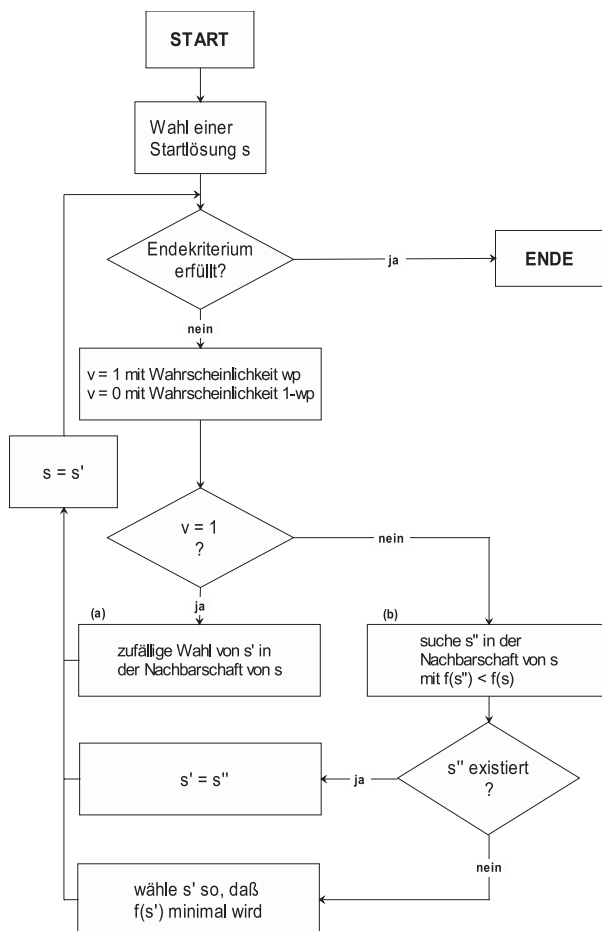


Abb. 1: Flussdiagramm

- (a) Einer Komponente von s zufällig einen neuen Wert geben (Komponente wählen, Wert wählen)
- (b) Suche einer besseren Lösung s'' in der Nachbarschaft von s .

Zur Nachbarschaft von s gehören alle Lösungen, die sich in genau einer Komponente von s unterscheiden. Es sind also nacheinander für s_1, s_2, \dots, s_n alle Elemente aus $\{1, \dots, K\}$ einzusetzen. Findet man so keine bessere Lösung als $f(s)$, dann nimmt man das Minimum dieser Versuche, das dann aber i. d. R. zu einer Verschlechterung führt, zumindest keine Verbesserung bewirkt. Der Zufallsschritt (a) bewirkt, dass der Algorithmus theoretisch jedes lokale Minimum wieder verlassen kann. Die Entscheidung, ob Zweig (a) oder (b) durchlaufen wird, erfolgt zufällig: Die Variable v wird mit Wahrscheinlichkeit w_p (walk probability) gleich 1 gesetzt. $v=1$ bedeutet, dass ein Zufallsschritt auszuführen ist. Ist $v=0$, dann wird ein Verbesserungsschritt ausgeführt. Jede Iteration erzeugt eine neue Lösung s' . Diese wird dann zur aktuellen Lösung $s=s'$. Wegen $0 < w_p < 1$ findet der RII-Algorithmus ggf. nach sehr vielen Iterationen mit jeder vorgegebenen Wahrscheinlichkeit die Optimallösung von (COP).

3 Das quadratische 0-1-Problem

Ein wichtiger Vertreter der kombinatorischen Optimierungsprobleme ist das quadratische 0-1-Problem ohne Nebenbedingungen (binary quadratic program oder unconstrained quadratic program, UQP):

$$(UQP) \min x^T Q x : x \in \{0,1\}^n$$

Gesucht ist ein n -dimensionaler Vektor x , der nur aus Nullen und Einsen besteht, sodass die quadratische Funktion $x^T Q x$ ein Minimum annimmt. Die Menge $\{0,1\}^n$ ist endlich, daher handelt es sich hier um ein (COP). Von Kochenberger/Glover et al. [3] wurde dieses Problem vorgeschlagen, um viele praktisch relevante Fragestellungen aus allen Bereichen der Betriebswirtschaftslehre und Logistik zu behandeln. Viele der in [3] aufgelisteten Probleme können jedoch vorteilhaft so formuliert werden:

$$(P) \min x^T Q x : Ax = \mathbf{1}, x \in \{0,1\}^n,$$

wobei $m, n \in \mathbb{N}$; $A \in \mathbb{R}^{m,n}$; $Q \in \mathbb{R}^{n,n}$ eine symmetrische Matrix ist. Die Matrix A hat in jeder Spalte genau eine 1, alle anderen Elemente jeder Spalte von A sind 0; $\mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^m$. Das Problem (UQP) ist allgemeiner als (P), da sich (P) in ein quadratisches 0-1-Problem ohne Nebenbedingungen transformieren lässt (siehe [3]). Unter Ausnutzung der Nebenbedingungen kann man (P) aber wesentlich schneller lösen als das quadratische 0-1-Problem ohne Nebenbedingungen. Im folgenden Abschnitt wird Randomized Iterative Improve-

ment unter Verwendung der Form (P) genutzt, um das K-Färbungsproblem zu lösen. Ähnlich lassen sich alle Probleme der Form (P) mit Randomized Iterative Improvement lösen.

4 Das K-Färbungsproblem

Sei $G=(V,E)$ ein ungerichteter Graph mit n Knoten und der Knotenmenge $V=\{1,2,\dots,n\}$. Die nichtleere Menge E ist die Kantenmenge, wobei $ij \in E$ genau dann gilt, wenn eine Verbindung zwischen $i \in V$ und $j \in V$ ($i \neq j$) existiert. Aus Gründen der Eindeutigkeit sei o. B. d. A. $i < j$ angenommen. Abb. 2 zeigt einen Graphen mit $n=5$ Knoten und 8 Kanten. Die Aufgabe besteht darin, jedem Knoten genau eine Farbe zuzuordnen. Dabei ist jedoch folgende Restriktion zu beachten: Zwei Knoten, die durch eine Kante verbunden sind, müssen unterschiedlich gefärbt sein. Abb. 3 zeigt eine zulässige Lösung mit $K=3$ Farben (B:blau, R:rot und G:grün). Der Knoten 2 ist in dieser Lösung blau, der Knoten 4 grün. Beide Knoten haben unterschiedliche Farben, da sie durch eine Kante verbunden sind. Knoten 2 und Knoten 5 haben die gleiche Farbe. Das ist zulässig, da sie nicht durch eine Kante verbunden sind. Bei einem K-Färbungsproblem sind die Anzahl K der erlaubten Farben und der Graph G vorgegeben. Gesucht ist eine zulässige Färbung bzw. das Er-

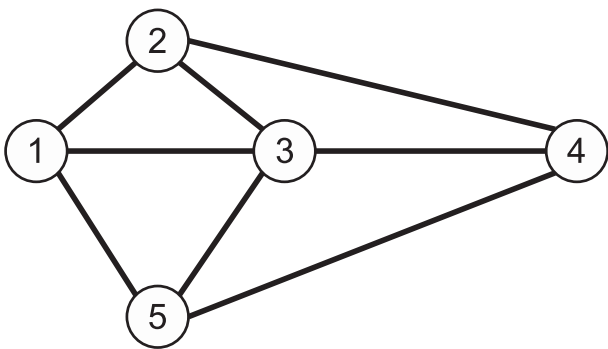


Abb. 2: Graph

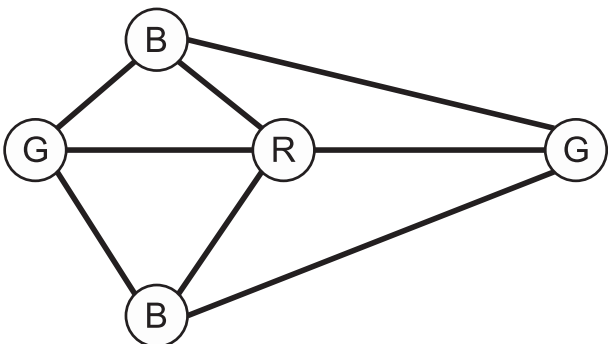


Abb. 3: zulässige Lösung

gebnis, dass eine Färbung mit K Farben nicht möglich ist. Es ist sinnvoll, die Farben zu nummerieren. Dann ist jedem Knoten eine Farbe aus der Menge $C=\{1,2,\dots,K\}$ der möglichen Farben zuzuordnen, wobei K eine vorgegebene natürliche Zahl ist. Die dem Knoten $i \in V$ zugeordnete Farbe sei mit $s_i \in C$ bezeichnet. Der Vektor s ist dann eine Färbung der Knotenmenge V . s ist zulässig, wenn für alle $i, j \in V$ mit $i < j$ folgende Bedingung zutrifft: Aus $ij \in E$ folgt $s_i \neq s_j$. Wird diese Bedingung für mindestens eine Kante $ij \in E$ verletzt, dann ist s unzulässig. Kurz: Die durch Kanten verbundenen Knoten sind unterschiedlich zu färben, wobei höchstens K verschiedene Farben verwendet werden dürfen. Graphenfärbungsprobleme findet man in der Verkehrslogistik, in der Stundenplanung und in komplizierteren Situationen der Projektplanung. Auch bei technischen Fragestellungen spielen Graphenfärbungsprobleme eine Rolle (Sendern Frequenzen zuweisen, Design elektronischer Schaltungen, Registeroptimierung im Zielcode).

Das K-Färbungsproblem kann man leicht in die Form (P) bringen: Die Variablen x_{ih} sind Binärvariablen für alle $i \in V$, $h \in C$, d. h. sie dürfen nur die Werte 0 oder 1 annehmen. Falls Knoten i mit Farbe h gefärbt wird, ist $x_{ih} = 1$, sonst $x_{ih} = 0$. Im Beispiel der Abb. 2, wo $K=3$ vorausgesetzt ist, haben wir $5 \cdot 3 = 15$ Variablen x_{ih} . $x_{12} = 1$ würde bedeuten, dass Knoten 1 die Farbe 2 erhält. $x_{12} = 0$ würde bedeuten, dass Knoten 1 nicht Farbe 2 bekommt, sondern eine andere. Da jeder Knoten zu färben ist, muss die Bedingung

$$(1) \sum_{h=1}^K x_{ih} = 1 \text{ für alle } i = 1, \dots, n \text{ erfüllt sein.}$$

Bedingung (1) sorgt dafür, dass jedem Knoten i genau eine Farbe zugewiesen wird. Denn für jedes i muss genau eine der Variablen x_{ih} gleich 1 sein, damit die Summe den Wert 1 annimmt.

Außerdem müssen je zwei Knoten i und j , die durch eine Kante verbunden sind, unterschiedliche Farben bekommen: Der Ausdruck $x_{ih} \cdot x_{jh}$ nimmt dann und nur dann den Wert 1 an, wenn beide Variablen gleich 1 sind. In allen anderen Fällen ist mindestens eine der beiden Variablen gleich 0 und somit das Produkt $x_{ih} \cdot x_{jh}$ ebenfalls gleich 0. Das Produkt wird genau dann gleich 1 sein, wenn sowohl Knoten i als auch Knoten j die Farbe h erhalten. In der Gleichung

$$(2) z = \sum_{ij \in E} x_{ih} \cdot x_{jh}$$

gibt z die Häufigkeit unzulässiger Farbkombinationen an. In jeder zulässigen Färbung des Graphen G ist $z=0$.

Gleichung (2) stellt die quadratische Zielfunktion des Problems (P) dar, wobei z der Zielfunktionswert ist. Gleichung (1) entspricht den Nebenbedingungen des Problems (P). Das K-Färbungsproblem ist daher

$$(P') \min z = \sum_{ij \in E} x_{ih} x_{jh} : \sum_{h=1}^K x_{ih} = 1 \text{ für alle } i \in V, \\ x_{ih} \in \{0,1\} \text{ für alle } i \in V \text{ für alle } h \in C$$

Löst man (P') mit dem Ergebnis $z=0$, dann hat man eine zulässige Färbung gefunden. Das K-Färbungsproblem ist dann für den gegebenen Graphen G lösbar. Ist im Optimum $z>0$, dann existiert keine Färbung des Graphen G mit maximal K Farben. Im nächsten Abschnitt wird die neue Lösungsmethode für das K-Färbungsproblem auf der Basis des Modells (P) bzw. (P') mit Hilfe von Randomized Iterative Improvement (RII) vorgestellt.

5 Ein neuer Lösungsansatz für das K-Färbungsproblem

Der in Abschnitt 2 eingeführte Lösungsvektor s lautet im Falle des K-Färbungsproblems $s = (s_i)_{i=1..n}$ mit $s_i \in C = \{1, \dots, K\}$ für alle $i = 1, \dots, n$. Für jeden Knoten i des Graphen G hat s eine Komponente, welche die Farbe des Knotens i angibt. s kann zulässig oder unzulässig sein. Die Gleichung (1), nach der jeder Knoten genau eine Farbe bekommt, ist erfüllt, da s_i ein eindeutiger Wert ist. Für alle Knoten i besteht der Zusammenhang: $s_i = h$ bedeutet $x_{ih} = 1$ und $x_{ik} = 0$ für alle $k \neq h$. Somit ist $f(s) = z$ entsprechend Gleichung (2) zu berechnen. Die Lösungsmethode wurde nach dem Flussdiagramm der Abb. 1 in der Programmiersprache Delphi implementiert (Win32-Compiler Delphi 2006). Delphi ist eine Weiterentwicklung der bekannten Sprache Pascal. Sie unterstützt die objektorientierte Programmierung. Das Programm, mit dem die Tests durchgeführt wurden, hat eine Länge von 350 Zeilen. Wegen der Nutzung objektorientierter Programmieretechniken kann es an alle Modelle angepasst werden, die sich in die Form $(P) \min x^T Q x : Ax = \mathbf{1}, x \in \{0,1\}^n$ bringen lassen. Für die Tests wurde der Code aus Gründen der Effizienz für die sehr schnelle Evaluierung von $x^T Q x$ optimiert. Der Hauptanteil der Rechenzeit geht tatsächlich in die Bestimmung der Zielfunktionswerte. Daneben hat der Parameter w_p (walk probability) großen Einfluss auf die Rechenzeit. In den Tests wurde $w_p = 0,2$ verwendet, da dieser Wert eine recht gute Performance liefert. In 20 % aller Iterationen wird daher ein Zufallsschritt

ausgeführt, in 80 % der Iterationen erfolgt ein Verbesserungsschritt. Folgende Tabelle enthält die Ergebnisse für zehn Testprobleme P1 bis P10 aus der Literatur. Die 2./3. Spalte gibt die Anzahl der Knoten/Kanten des Problems an. In Spalte 4 sind die Rechenzeiten aus [2] zusammengestellt. Für mehrere dieser Probleme wurde in [2] nur angegeben: Zeit $< 1,0$ s. In Spalte 5 sind die Rechenzeiten des neuen Lösungsansatzes zusammengestellt. Die Probleme wurden auf einem 3,19-GHz-PC gelöst. (Die Angaben in Spalte 4 wurden umgerechnet, sodass sie mit Spalte 5 vergleichbar sind.) Für die Probleme P1 bis P3 ist die Rechenzeit praktisch 0; die Dauer war für die Zeitmessung zu kurz. Die anderen Zeitangaben sind die gerundeten Werte von 20 Versuchen:

Name	Knoten	Kanten	Rechenzeit TS mit (UQP) [s]	Rechenzeit RII mit (P) [s]
P1	25	160	< 1,0	< 0,1
P2	36	290	< 1,0	< 0,1
P3	49	476	< 1,0	< 0,1
P4	64	728	< 1,0	2,9
P5	74	301	1,1	0,5
P6	80	254	< 1,0	2,4
P7	87	406	38,9	4,6
P8	95	755	< 1,0	2,8
P9	138	493	25,1	6,4
P10	191	2360	< 1,0	15,2

Zu Spalte 4 der Tabelle: Kochenberger/Glover et al. [2,3] lösen das Problem (UQP) mit einer aufwändigen Heuristik. Sie verwenden eine anspruchsvolle Implementation des Tabu-Search-Verfahrens (TS). Die Rechenzeiten sind in der Tabelle in Sekunden angegeben. Zu Spalte 5 der Tabelle: Unter Ausnutzung der Modellformulierung (P') lassen sich K-Färbungsprobleme aus [2] oft in wesentlich kürzeren Rechenzeiten mit Hilfe der recht einfachen RII-Heuristik lösen. Bei P4, P6 und P8 ist die RII-Heuristik schlechter, aber vergleichbar mit TS. P10 ist das einzige Problem in der Tabelle mit wesentlich längerer Rechenzeit des RII-Verfahrens. Ergebnis: In Anwendungen kann oft die Modellformulierung (P) genutzt werden. Die Tests zeigen, dass Modell (P) der Formulierung (UQP) überlegen ist, denn Tabu Search (TS) ist wesentlich leistungsfähiger als Randomized Iterative Improvement (RII). Mit Tabu Search wären die Ergebnisse in der 5. Tabellenspalte sicher besser als mit RII.

6 Ausblick

Viele kombinatorische Optimierungsprobleme, die sich als quadratische 0-1-Probleme ohne Nebenbedingungen formulieren lassen, sollte man unter algorithmischen Aspekten besser in die Form des hier vorgestellten Modells bringen. Selbst eine einfache Heuristik auf der Basis der hier vorgeschlagenen Problemformulierung liefert oft bessere Ergebnisse als ein aufwändiges Verfahren für das quadratische 0-1-Problem ohne Nebenbedingungen. Anspruchsvolle Heuristiken, wie z. B. Tabu Search oder Scatter Search, sollten daher weit bessere Ergebnisse liefern. Auf die effiziente Berechnung von $x^T Q x$ ist stets zu achten, da die Zielfunktion sehr oft zu evaluieren ist.

Literatur

- [1] Hoos, H./Stützle, Th. (2005): Stochastic Local Search: Foundations and Applications, San Francisco: Elsevier/Morgan Kaufmann.
- [2] Kochenberger, G./Glover, F./Alidaee, B./Rego, C. (2003): An unconstrained quadratic binary programming approach to the vertex coloring problem, working paper, University of Colorado at Denver.
- [3] Kochenberger, G./Glover, F./Alidaee, B./Rego, C. (2004): An unified modeling and solution framework for combinatorial optimization problems, OR Spectrum 26, pp. 237-250.

Autor

Prof. Dr. rer. pol. Matthias Forster
Technische Fachhochschule Wildau
Fachbereich Betriebswirtschaft/Wirtschaftsinformatik
Tel. +49 3375 508-529
matthias.forster@tfh-wildau.de