Ganzzahlige lineare Programmierung:

Anwendungen und Lösungsansätze

Jens Pickenpack[1]

[1] FH-Wedel - Studiengang Master Informatik

Seminar Algorithmik im SS 2013

Gliederung

Gliederung

Problemstellung

Produktionsplanung

- Welches Produktionsprogramm liefert mir einen maximalen Deckungsbeitrag unter meinen Restriktionen?

Tourenplanung

- Welche Route ist optimal?

Telekommunikationsnetze

- Wie kann es zu minimalen Kosten dimensioniert werden?

Problemstellung

Produktionsplanung

- Welches Produktionsprogramm liefert mir einen maximalen Deckungsbeitrag unter meinen Restriktionen?

Tourenplanung

- Welche Route ist optimal?

Telekommunikationsnetze

- Wie kann es zu minimalen Kosten dimensioniert werden?

Problemstellung

Produktionsplanung

- Welches Produktionsprogramm liefert mir einen maximalen Deckungsbeitrag unter meinen Restriktionen?

Tourenplanung

- Welche Route ist optimal?

Telekommunikationsnetze

- Wie kann es zu minimalen Kosten dimensioniert werden?

Praxisrelevanz

steigender Optimierungsbedarf

- Globalisierung
- Resourcenknappheit
- Umweltschutz

Probleme sind oftmals ganzzahlig

- Die Stückzahl von herzustellenden Produkten ist ganzzahlig
- Ein LKW/Bus fährt oder nicht
- Kapazitäten in Netzen werden geschaffen oder nicht

Praxisrelevanz

steigender Optimierungsbedarf

- Globalisierung
- Resourcenknappheit
- Umweltschutz

Probleme sind oftmals ganzzahlig

- Die Stückzahl von herzustellenden Produkten ist ganzzahlig
- Ein LKW/Bus fährt oder nicht
- Kapazitäten in Netzen werden geschaffen oder nicht

Praxisrelevanz (2)

Probleme der theoretischen Informatik / Kombinatorik

- Rucksackproblem
- Travelling Salesman Problem (TSP)
- Erfüllbarkeitsproblem

Praxisrelevanz (3)
Probleme

- Ganzzahligkeit macht das Problem NP-Vollständig
    - Exponentielle Laufzeit
    - optimale Lösung im Allgemeinen bei größeren Problemen nicht errechenbar
    - je nach Problem benötigt man also Heuristiken
    - Schwierigkeit eines Problems schwer Abzuschätzen

Praxisrelevanz (4)

Problemeinordnung

- Gute Lösungen reichen oft aus
    - es kann bestimmt werden wie weit man von der optimalen Lösung maximal entfernt ist
    - Lineare Programme mit mehreren 10000 Variablen werden optimal gelöst

Gliederung

wichtige Begriffe

Definition

Die Ganzzahligkeitsbedingungen eines GLP wegzulassen und es als LP zu lösen nennt man Relaxierung oder Relaxation.

Definition

Gemischt ganzzahlige lineare Programme enthalten auch Variablen ohne Ganzzahligkeitsbedingung

Definition

Binäre lineare Programme (BLP) enthalten nur Variablen mit den Werten Null oder Eins

mathematische Problemdefinition eines GLP

Definition

Ein GLP ist ein Problem folgender Art:
$\max; min \left\{ c^T x | Ax \leq ; \geqslant b, x \geq 0, x \in \mathbb{Z}^{\mathbb{N}} \right\}$

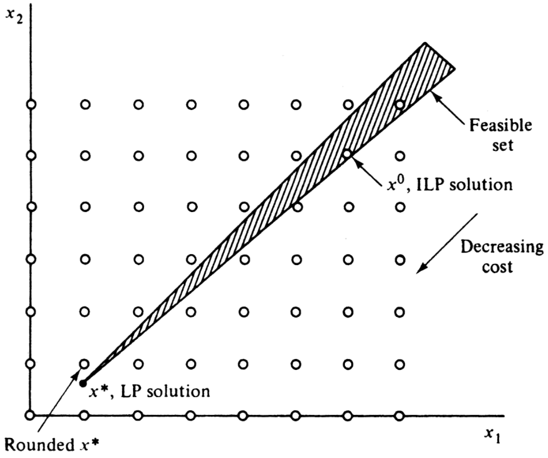
Beispiel

$\max \left\{ x_0 = 2x_1 + 5x_2 \right\}$
$2x_1 - x_2 \leq 9$
$2x_1 + 8x_2 \leq 31$
$x_1 x_2 \in \mathbb{N}$

warum Runden nicht genügt (1)

- Man könnte zunächst auf die Idee kommen, dass der Forderung nach Ganzzahligkeit einfach durch Runden entsprochen werden könnte.
- Dies ist nicht der Fall. Die diskrete Optimallösung kann beliebig weit von der linearen entfernt sein.
- Es kann auch sein, dass keine diskrete Lösung existiert.

warum Runden nicht genügt (2)



Gliederung

Branch and Bound Verfahren

Idee

- Es handelt sich um ein rekursives Verfahren, bei dem ein Entscheidungsbaum aufgebaut wird.
- es stammt aus den 1950/1960er Jahren

Branch and Bound Verfahren (2)

Algorithmus

- Lösen der Relaxation. Wenn die Variablen ganzzahlig sind, sind wir fertig.
- Zerlegen des Problems in Teilmengen (Branches), die optimale Lösungen enthalten können oder nicht
- Berechnung von oberen/unteren Schranken (Bound)
- eine Auswahlregel zur Wahl der nächsten betrachteten Branches (LIFO oder best bound)
- Lösung mit dem Branch and Bound Verfahren

Schnittebenenverfahren

Idee

- Es handelt sich um ein Verfahren, bei dem zusätzliche Restriktionen eingeführt werden, die den Lösungsbereich so lange einschränken bis alle Variablen ganzzahlig sind
- es stammt aus den 1950/1960er Jahren

Algorithmus

- Lösen der Relaxation. Wenn die Variablen ganzzahlig sind, sind wir fertig.
- solange nicht alle Variablen ganzzahlig:
  - Einführung einer zusätzlichen Bedingung (Cut), die keine potentiell zulässigen Lösungen eliminiert
  - Erneute Lösung

Schnittebenenverfahren

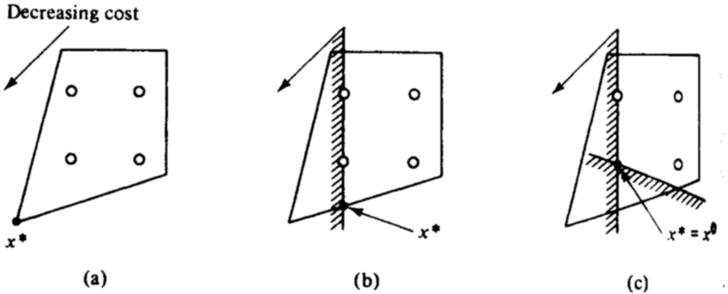
Idee

- Es handelt sich um ein Verfahren, bei dem zusätzliche Restriktionen eingeführt werden, die den Lösungsbereich so lange einschränken bis alle Variablen ganzzahlig sind
- es stammt aus den 1950/1960er Jahren

Algorithmus

- Lösen der Relaxation. Wenn die Variablen ganzzahlig sind, sind wir fertig.
- solange nicht alle Variablen ganzzahlig:
    - Einführung einer zusätzlichen Bedingung (Cut), die keine potentiell zulässigen Lösungen eliminiert
    - Erneute Lösung

Schnittebenenverfahren (2)



Schnittebenenverfahren (3)

- In späteren Schritten werden die abgeschnittenen Bereiche immer kleiner.
- Die Tabelle wird immer größer, somit dauert die Berechnung späterer Schritte länger
- Es können ganz unterschiedliche Schnitte durchgeführt werden z.B. Gomory Cuts
- Man kann zeigen, dass diese Verfahren in endlich vielen Schritten terminieren.

Schnittebenenverfahren (4)

Gomory Cut

- Idee: Aufteilung einer Zeile in ganzzahligen und gebrochenen Teil.
- Beispiel für die Trennung:
  - 2,7 -> 2 und 0,7
  - -3,7 -> -4 und 0,3
- für den gebrochenen Teil gilt: $\sum f_{ij} * x_j \geqslant f_{i0}$
- Es wird zum Tableau folgende Zeile hinzugefügt:$-\sum f_{ij} * x_j + s = -f_{i0}$

Schnittebenenverfahren (5)

Gomory Cut

Ausschnitt aus einem Tableau:

	K1	K2	K3	K4	S5	S6	S7	S8	S9	RS
K1	1	1/5	0	0	0	0	0	-1/20	0	3,5

1*K1 + 1/5*K2 - 1/20*S8 = 3,5

zusätzliche Schnittbedingung also:

-1/5*K2 - 19/20*S8 + C1 = -0,5

Es ergibt sich ein neues Tableau mit einer Zeile und einer Spalte mehr

Branch and Cut Verfahren

Idee

- Es handelt sich um eine Kombination aus Branch and Bound und Schnittebenen Verfahren und stammt aus den 1980-1990er Jahren
- vor dem Branch werden Cut Vorgänge durchgeführt, was sich als sehr effektiv herausgestellt hat

Gliederung

Heuristische Verfahren

- Verzicht auf sichere Optimalität der Lösung - Abschätzung der Qualität aber möglich
- Problemspezifische Ansätze, daher sind die Algorithmen nicht generisch
- einzige Möglichkeit mit Sicherheit bei komplexen GLP-Problemen schnelle Lösungen zu erhalten

Heuristische Verfahren

- Verzicht auf sichere Optimalität der Lösung - Abschätzung der Qualität aber möglich
- Problemspezifische Ansätze, daher sind die Algorithmen nicht generisch
- einzige Möglichkeit mit Sicherheit bei komplexen GLP-Problemen schnelle Lösungen zu erhalten

Heuristische Verfahren

- Verzicht auf sichere Optimalität der Lösung - Abschätzung der Qualität aber möglich
- Problemspezifische Ansätze, daher sind die Algorithmen nicht generisch
- einzige Möglichkeit mit Sicherheit bei komplexen GLP-Problemen schnelle Lösungen zu erhalten

Gliederung

Bewertung

Algorithmus	Prozessor	Speicher	Qualität	Zeit
Branch and Bound	o	-	++	-
Schnittebenen	-	o	++	-
Branch and Cut	o	o	++	o
Heuristiken	(+)	(+)	(o)	++

Einsatz

- Varianten von Branch and Cut sind State-of-the-Art
- sie kommen auch in freien Solvern wie GLPK zum Einsatz
- GLPK ist einfach konfigurierbar was die eingesetzten Cut und Branchverfahren betrifft
- verwenden Sie z.B. Sage (mathematische Bibliothek/Präcompiler für Python) welches standardmäßig GLPK verwendet
- für sehr große Probleme (etwa mit zehntausenden Variablen) sind kommerzielle Lösungen wie CPLEX empfehlenswert

Gliederung

Problemstellung 1

Problemdefinition

- zu produzierende Kunststoffe K1 .. K4, Integer und >=0
- zu verbrauchende Ressourcen R1 .. R5
- zu minimierende Kostenfunktion: Kges = 5*K1+ 2*K2 + 1*K3 + 1*K4
- Benötigte Ressourcen pro produziertem Kunststoff

Problem Tableau

	K1	K2	K3	K4	zu verbrauchende Mengeneinheiten
R1	15	4	6	7	<=100
R2	0	3	5	1	<=50
R3	20	0	0	0	<=80
R4	20	4	0	0	>=70
R5	10	35	50	60	>=300
Kosten	5	2	1	1	

Gliederung

Lösung der Relaxierung

Benötigt zwei Simplexschritte und ergibt die Lösung:
Kosten = 21.92
K1: 3.5
K2: 0.0
K3: 0.0
K4: 4.42

Lösen per GLPK

Kosten = 24.0
K1: 3.0
K2: 3.0
K3: 1.0
K4: 2.0

Gliederung

Problemstellung 2

Problemdefinition

- Aussagenlogische Formel auf Erfüllbarkeit prüfen
- a v b ^ ~a v b

Problemmodellierung

- Maximiere a + b
  - a + b >= 1
  - (1-a) + b >= 1 <=> -a + b >= 0
  - a,b binär

Result : 2.0 a: 1.0 b: 1.0

Zusammenfassung

- die Lösung von diskreten linearen Problemen ist deutlich schwieriger als die Lösung von linearen Problemen - nämlich NP vollständig
- mit GLP können viel komplexere Sachverhalte modelliert werden als mit LP
- Tools wie GLPK (frei) oder CPLEX (kommerziell) sind unbedingt empfehlenswert

Quellen I

- C. Papadimitriou K.Steiglitz. , Combinatorial Optimization , Dover Publications, 1982
- P.Murzel R.Weiskircher, Optimizing over all combinatorial embeddings of a planar Graph aus Integer Programming and Combinatorial Optimization, Springer, 1999
- P. Brucker , Ganzzahlige lineare Programmierung mit ökonomischen Anwendungen , Verlag Anton Hain, 1975
- Sandor Nevelö , Ganzzahlige lineare Programmierung mit Hilfe von Branch & Bound , Grin Verlag , 2002