

OR I & II

Inhalt der Vorlesungen von Prof. H.-J. Sebastian 2002/03

Lehrstuhl: www.dpor.rwth-aachen.de · Autor: urs.enke@web.de

Version: 24. Januar 2007

OR I & II beschäftigen sich meist mit der Lösung von Optimierungsproblemen, d.h. der Max- oder Minimierung von Zielfunktionen unter Einhaltung gewisser Restriktionen. 1 *Lineare Optimierung* beschränkt sich auf Probleme mit linear formulierten Ziel- und Restriktionsfkt. 3 *Transportprobleme* sind zwar i.d.R. linear, jedoch komplexitätsbedingt besser mit heuristischen anstatt exakten Algorithmen anzugehen. In 4 *Lagerhaltung* wird die Linearität verlassen, die gezeigten Lösungsansätze sind aber auf der Lagerhaltung ähnliche mehrstufige Entscheidungsprozesse beschränkt, im Ggs. zu den allgemeineren Ansätzen von 5 *Nichtlineare Optimierung*. Aus dem Rahmen fällt die 2 *Projektplanung* aus OR I, welche sich mit der Erstellung von Ablaufplänen und der Bestimmung darin enthaltener zeitlicher Spielräume beschäftigt.

1 Lineare Optimierung

Normalform, Basis, Basislösung

(Formalisierung eines umgangssprachl. formulierten Problems)

Sind Ziel- und Restriktionsfkt. eines Optimierungsproblems linear formuliert, läßt es sich in folgender *Normalform* darstellen:

$\max c^T x$, $(A, I)x = b$, $x, b \geq 0$, $x, c \in \mathbb{R}^{n+m}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

Minimiert die Zielfkt., sind *Entscheidungsvariablen* x_i oder Komponenten des *Kapazitätsvektors* b negativ oder ist die Koeffizientenmatrix nicht von der Form (A, I) , sind folgende Umformungsschritte nötig:

- Minimierung in Maximierung wandeln: $\min c^T x$ zu $\max -c^T x$
 - Zeilen mit negativem b_i mit -1 multiplizieren
 - Ungleichungen $\sum_{j=1}^n a_{ij} x_j \leq b_i$ durch Einführung sog. *Schlupfvariablen* $x_s \geq 0$ in Gleichungen $\sum_{i=1}^n a_{ij} x_i \pm x_s = b_j$ überführen
 - Unrestringierte Variablen x_i durch Differenz zweier positiver $(x_{i1} - x_{i2})$ darstellen
 - Umnummerierung der x_i zwecks Herstellung der Form (A, I) in der Koeffizientenmatrix. Stehen für I benötigte Spalten nicht zur Verfügung, werden diese 'künstlich' erzeugt und wird die Zielfkt. mittels $-Mx_t$ um die so eingeführten Entscheidungsvar. ergänzt, wobei $M \gg 0$.
- In der Normalform läßt sich bereits eine die Restriktionen $(A, I)x = b$ erfüllende Belegung der Entscheidungsvar. ablesen: Deren mit den Spalten der Teilmatrix I korrespondierende Komponenten werden mit den jeweiligen Werten des Kapazitätsvektors belegt, der Rest mit Nullen, salopp $x^1 = (0, b)$. Es handelt sich um eine *Basislösung* zur *Basis* I : $B \in \mathbb{R}^{m \times m}$ heißt Basis $\Leftrightarrow B$ Teilmatrix von (A, I) , und eine *Lösung* ist eine Variablenbelegung, ohne Rücksicht auf Restriktionen. Daher heißt x Basislösung $\Leftrightarrow (A, I)x = b \wedge x = (x_B, 0)$, $x_B \in \mathbb{R}^m$.

Simplextheorem

Die Einhaltung der Restriktionen macht die Lsg. *zulässig*, entsprechend heißt B zulässig $\Leftrightarrow B^{-1}b \geq 0$. Die unter den zulässigen Lsg. optimale ist laut *Simplextheorem* eine Basislsg.:

x^0 zul. Lsg. $\wedge \forall$ zul. Lsg. x : $c^T x^0 \geq c^T x \Rightarrow x^0$ Basislsg.

Unbewiesen bleibt hier, daß die Basislsg. den Ecken des konvexen Lösungsraums entsprechen (welcher vorliegt, da die Restriktionen konvexe Halbräume definieren). Jede Lsg. x im Lösungsraum läßt sich als Konvexkombination von Ecken x^i darstellen ($x = \sum_i \lambda_i x^i$), aufgrund der Linearität der Zielfkt. $c^T x$ gilt dies auch für Zielfunktionswerte: $c^T \sum_i \lambda_i x^i = \sum_i \lambda_i (c^T x^i)$. Durch Konvexkombination der Zielfunktionswerte der Ecken läßt sich aber kein Wert erhalten, der höher ist als der höchste an der Kombination beteiligte, das Optimum ist daher an einer Ecke – und somit an einer Basislsg. – zu finden. Sei x^k die 'beste' Ecke, dann sind alle kombinierten Punkte 'schlechter': $\sum_i \lambda_i (c^T x^i) \leq c^T x^k \sum_i \lambda_i = c^T x^k$.

Primaler Simplex

Der (primale) Simplex-Alg. startet mit der aus der Normalform ablesbaren Lsg. und führt Basiswechsel durch, die den Wert der Zielfkt. nicht verschlechtern, im nichtentarteten Falle garantiert verbessern und somit in endl. vielen Schritten zum Optimum führen. Hierzu berechnet er

(*Aufnahmeregel*), die Erhöhung welches der x_j von 0 auf 1 die stärkste Erhöhung der Zielfkt. ergäbe:

Seien (B, N) eine Spaltenpermutation von (A, I) und x_B, x_N die entsprechenden Entscheidungsvar., dann gilt (*Basisdarstellung*): $(B, N)(x_B, x_N)^T = b \Leftrightarrow x_B = B^{-1}b - B^{-1}Nx_N$, bzw. komponentenweise (mit $b^* = B^{-1}b$ und $a^* = B^{-1}N$) $x_{B_i} = b_i^* - \sum_{j=m+1}^{m+n} a_{ij}^* x_j$

• *Aufnahmeregel*: $z = c^T x = c_B^T x_B + c_N^T x_N$
 $= c_B^T (B^{-1}b - B^{-1}Nx_N) + c_N^T x_N = c_B^T B^{-1}b - (c_B^T B^{-1}N - c_N^T) x_N$
 $= \sum_{i=1}^m c_i b_i^* - \sum_{j=m+1}^{m+n} (\sum_{i=1}^m c_i a_{ij}^* - c_j) x_j = z_{alt} - \sum_j \Delta z_j x_j$
Es gilt also, zwecks Erhöhung von z das zum 'negativsten' der Δz_j gehörige der x_j in die Basis aufzunehmen, also x_k mit $k = \arg \min_j \{\Delta z_j\}$. Die Δz_j sind wie folgt nachvollziehbar: Zum einen wird wegen $x_j = 1$ der Wert der Zielfkt. um c_j erhöht, zum anderen erniedrigt er sich aufgrund der restriktionsbedingten Wertänderung der bisherigen Basisvar. um $\sum_{i=1}^m c_i a_{ij}^*$. (Steht kein negatives Δz_j zur Wahl, ist die optimale Lösung erreicht: Keine marginale Erhöhung einer Nichtbasisvar. brächte einen Vorteil.)

Unklar bleibt, auf welchen Wert x_k gesetzt werden darf und welches x_i die Basis verläßt. Beides wird simultan durch die *Eliminationsregel* geklärt, die bestimmt, welches x_i bei einer Erhöhung des x_k über 0 hinaus zuerst an seine Zulässigkeitsgrenze stieße, also negativ würde. x_k wird auf den maximal möglichen Wert gesetzt, besagtes x_i sinkt auf 0 und fällt so aus der Basis heraus:

• *Eliminationsregel*: Ist $a_{ik}^* \leq 0$, sinkt x_i für wachsendes x_k nicht und wird daher 'kein Problem' darstellen. Betrachtet werden also i mit $a_{ik}^* > 0$. (Gibt es keine, bricht das Verfahren ab, da beliebig gute Lsg. möglich sind.) Zulässigkeit erfordert:

$$\forall i \ x_{B_i} = b_i^* - a_{ik}^* x_k \geq 0 \Leftrightarrow \forall i \ x_k \leq \frac{b_i^*}{a_{ik}^*} \Rightarrow x_k = \min_i \frac{b_i^*}{a_{ik}^*}$$

Für den Simplex ist aber nur die Wahl der zu eliminierenden Variable relevant, da so die neue Basis bekannt ist und das neu formulierte Problem (Tableau) durch Matrixmultiplikation des Ausgangsproblems mit der Basisinversen erhalten wird. Gibt es mehrere Eliminationskandidaten, wird das Problem als 'entartet' bezeichnet.

Dualer Simplex

Gilt nicht $b \geq 0$, jedoch $\forall j \ \Delta z_j \geq 0$, läßt sich der *Duale Simplex-Alg.* anwenden, um b unter Beibehaltung der $\Delta z_j \geq 0$ nichtnegativ zu machen, also eine optimale Lösung zu finden.

• *Eliminationsregel*: Aus der Basis entfernt wird heuristisch die Var. mit dem negativsten b_i .

In sie aufgenommen wird jene, die aufgenommen werden muß, um $\Delta z_j \geq 0$ zu garantieren.

• *Aufnahmeregel*: Seien x_k zu eliminieren und x_l aufzunehmen, dann würde die zu x_k gehörige Zeile beim Basiswechsel mit $(a_{kl}^*)^{-1}$ multipliziert werden. Um das negative Vorzeichen derer rechten Seite 'loszuwerden', muß a_{kl}^* negativ sein. Die neuen $\Delta z_j'$ berechnen sich als $\Delta z_j' = \Delta z_j - \frac{a_{kj}^*}{a_{kl}^*} \Delta z_l$; ihre Nichtnegativität ist nur für $a_{kj}^* \leq 0$ gefährdet. Betrachtet werden also alle j und l mit $a_{kj}^*, a_{kl}^* \leq 0$. $\Delta z_j' \geq 0 \Leftrightarrow \frac{\Delta z_j}{a_{kj}^*} \leq \frac{\Delta z_l}{a_{kl}^*}$

ist nur zu garantieren, wenn das aufzunehmende x_l so gewählt wird, daß $l = \arg \max_j \{ \frac{\Delta z_j}{a_{kj}^*} \mid a_{kj}^* \leq 0 \}$.

Gomory

Gilt $x \in \mathbb{Z}^{m+n}$, so ist der Simplex zunächst unter Ignorieren der Ganzzahligkeitsbed. durchzuführen. Für jede nicht ganzzahlige Var. kann nun eine neue Restriktion hinzugefügt werden, deren Erfüllung (via dualer Simplex-Schritte) anschl. Ganzzahligkeit garantiert. Herleitung: Seien $b_i^* = \lfloor b_i^* \rfloor + h_i$ und $a_{ij}^* = \lfloor a_{ij}^* \rfloor + h_{ij}$, dann ist $x_{B_i} = b_i^* - \sum_{j=m+1}^{m+n} a_{ij}^* x_j = \lfloor b_i^* \rfloor - \sum_{j=m+1}^{m+n} \lfloor a_{ij}^* \rfloor x_j + h_i - \sum_{j=m+1}^{m+n} h_{ij} x_j$. Ob $x_{B_i} \in \mathbb{Z}$ hängt von den hinteren beiden Summanden ab, für die mit $x_j, h_{ij} \geq 0$ gilt: $h_i - \sum_{j=m+1}^{m+n} h_{ij} x_j \leq h_i < 1$. Die nächste ganze Zahl unter 1 ist 0, also $x_{B_i} \in \mathbb{Z} \Rightarrow h_i - \sum_{j=m+1}^{m+n} h_{ij} x_j \leq 0 \Leftrightarrow -\sum_{j=m+1}^{m+n} h_{ij} x_j + x_G = -h_i$, wobei die letzte Form eine neue Zeile des Simplex-Tableaus und $x_G \geq 0$ eine neue Entscheidungsvar. darstellt, die automatisch Teil der Basis ist. (Gezeigt wurden notwendige Folgen von Ganzzahligkeit. Wieso können diese als hinreichende Bedingung benutzt werden?)

Postoptimale Modifikation

Bei nachträgl. Änderung gewisser c_i oder Spalten der ursprüngl. Restriktionsmatrix (deren Aussehen im Endtableau man durch Multiplikation mit der kumulierten Basiswechsellmatrix erhält) müssen im Endtableau die betroffenen Δz_j neu berechnet und ggf. weitere Simplex-Schritte durchgeführt werden. Gleiches gilt für eine nachträgl. geänderte

rechte Seite. Wird eine neue Zeile samt neuer Entscheidungsvar. hinzugefügt, so daß letztere im Ausgangstableau Bestandteil der Basis ist, ergibt sich die Änderung des Endtableaus durch Ergänzung einer Eins und Nullen in den Basisspalten und Berechnung des verbleibenden Koeffizientenvektors durch Multiplikation des negativen alten Koeffizientenvektors mit der alten Basisinversen. Im resultierenden Tableau ist die neue Var. ebenfalls in der Basis enthalten, weitere Simplex-Schritte sind also ggf. erforderlich.

Durch Ersetzen von Elementen des Endtableaus durch Variablen kann z.B. untersucht werden, in welchen Grenzen sie variieren dürfen, ohne eine andere Basis optimal zu machen (*Sensitivitätsanalyse*). Statt einzelner Variablen können auch Ausdrücke der Form $c + \lambda v$ verwendet werden (insbes. an mehreren Stellen mit identischen λ , genannt *Parametrische Programmierung*).

Dualität

Zu jedem LOP läßt sich sein sog. *duales Problem (dP)* formulieren:

pP: $\max c^T x, Ax \leq b, x \geq 0, x, c \in \mathbb{R}^{m+n}, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$

dP: $\min b^T y, A^T y \geq c, y \geq 0, c \in \mathbb{R}^{m+n}, A \in \mathbb{R}^{m \times n}, y, b \in \mathbb{R}^m$

Durch die Transponierung der Restriktionsmatrix wird u.U. eine Reduktion des Lösungsraums erreicht. Ist das zu dualisierende Problem in der Normalform gegeben, bietet sich zur Lösung des dualisierten die Anwendung des (daraus seinen Namen beziehenden) dualen Simplexalg. an. (Umwandlungsregeln für allgemeiner formulierte pP)

Für pP und dP gilt:

- Das duale Problem des dualen ergibt wieder das primale.
- *Schwache Dualität*: Seien x, y zul. Lsg. ihrer jeweiligen Probleme, dann gilt $c^T x \leq b^T y$.
- *Starke Dualität*: Seien x, y zul. Lsg. ihrer jeweiligen Probleme, dann gilt $c^T x = b^T y \Leftrightarrow x, y$ opt. Lsg. ihrer jew. Probleme.
- Hat das pP eine opt. Lsg., so auch das dP (mit gleichem Zielfunktionswert). Hat das pP keine zulässige, hat das dP daher auch keine optimale.

(Beweise für obige Sätze)

Konstruktionsbedingt entspricht jedes y_j einer Restriktion des pP; es gibt jeweils an, um welchen Betrag sich der Zielfunktionswert verbesserte, würde die betr. Restriktion marginal um 1 relaxiert. Ein Durchlaufen der Ecken des Lösungsraumes des dP kann daher verstanden werden als Durchlaufen von Teilmengen von Restriktionen des pP: Im Optimum des dP sind dann genau jene y_j positiv, deren zugehörige Restriktion im Optimum des pP aktiv sind und deren Relaxierung somit einen höheren Zielfunktionswert verspricht.

Dieser Zshg. im Optimum wird beschrieben durch den *Satz des komplementären Schlupfes*: \forall zul. Lsg. $(x, x_s), (y, y_s)$ ihrer jew. Probleme (wobei x_s sowie y_s die Schlupf- und x bzw. y die verbleibenden, sog. *Strukturvariablen* seien) gilt, daß sie genau dann optimale Lösungen sind, wenn $x^T y_s = y^T x_s = 0$. Ferner gilt: Die Δz_j der primalen Strukturvar. [Schlupfvar.] sind gleich den Werten der dualen Schlupfvar. [Strukturvar.]. (Diese Zuordnung gilt für die zu Anfang genannten Formen von pP und dP. Dualisiert man das pP erst nach der Einführung von Schlupfvar., ist der Satz nicht direkt anwendbar.)

Die dualen Strukturvar. geben den Gradienten der Zielfkt. bei Relaxierung der zugehörigen Restriktion an, genannt deren *Schattenpreis* oder *Opportunitätskosten*. Der Satz des kompl. Schlupfes läßt sich also dahingehend interpretieren, daß entweder die Schlupfvar. einer Restriktion Null sein muß oder ihr Schattenpreis.

2 Projektplanung

Projekt

Projekt: Vorhaben mit def. Anfang und Ende, charakterisiert durch Einmaligkeit seiner Bed. (bzgl. Ziel, Restriktionen, Organisation) in ihrer Gesamtheit

Projektplanung: Planung von Teilzielen, des Aufwandes, der Organisation, der Dokumentation

Gantt-Chart: Tabellarische Darst. mit Teilaufgaben in den Zeilen und Zeitachse in den Spalten, auf welcher der Zeitraum der jew. Aufgabe durch einen Balken abgedeckt wird. (+) simpel, ermöglicht Soll-Ist-Vergl.; (-) Sachlich bedingte Reihenfolge unbestimmt; unflexibel bzgl. Änderung/Verbesserung

Netzplantechnik

Projekte lassen sich als Netze (= zyklensfreie gerichtete Graphen mit je einer Quelle und Senke) darstellen. Die resultierenden Netzpläne lassen

sich klassifizieren nach Pfeil- (Knoten = Ereignisse = Zeitpunkte) und Kreisdarstellung (Knoten = Vorgänge = zeitbeanspruchende Elemente) sowie nach Vorgangs- (Ereignisse def. Anfang/Ende der Vorgänge) und Ereignisorientiertheit (Ereignisse sind 'Kontrollpunkte', Vorgänge 'undefiniert'). Beim Netzplan-Entwurf wird i.d.R. ein Graph unter Beachtung gewisser Regeln aus einer Vorgangsliste erstellt (*Strukturanalyse*); es folgen ggf. iteriert *Zeitanalyse* (z.B. Pufferberechnung), *Kapazitätsanalyse* (Optimierung der Reihenfolge bzgl. der Ressourcen) und *Kostenanalyse* (Opt. der Vorgangsdauern bzgl. der Kosten). Die Optimierung erfolgt nur im Rahmen bei Struktur- und Zeitanalyse getroffener Annahmen. (Regeln des Netzplan-Entwurfs)

CPM = Critical Path Method

Die CPM bildet aus Vorgangsdauern ein *Vorgangspfeilnetzwerk* (Netzwerk = Netz mit bewerteten Kanten) und errechnet zu diesem folgende zur Analyse dienliche Größen:

1. frühestmögliche Zeitpunkte FZ_i jedes Ereignisses, def. als längster Weg (Summe der Dauern D_{ij}) zur Quelle des Graphen.
2. spätestmögliche Zeitpunkte SZ_i als FZ der Senke des Graphen minus dem längsten Weg (Summe der Dauern D_{ij} dorthin).
3. frühestmöglicher/spätesterlaubter (F/S) Anfangs-/Endzeitpunkt (A/E) jedes Vorgangs: $FA_{ij} = FZ_i, FE_{ij} = FZ_j + D_{ij}, SA_{ij} = SZ_j - D_{ij}, SE_{ij} = SZ_j$.
4. Puffer jedes Vorgangs:

- *Gesamtpuffer* $GP = SE - FA - D$ (Projekt endet plangemäß, wenn alle Vorgänger und Nachfolger sich an ihre Dauern halten)
- *Freie Puffer* $FP_{ij} = FZ_j - FZ_i - D_{ij}$ (unmittelbarer Nachfolger kann plangemäß beginnen, wenn sich alle Vorgänger an ihre Dauern halten)
- *Unabh. Puffer* $UP_{ij} = \max(FZ_j - SZ_i - D_{ij}, 0)$ (unmittelbarer Nachfolger kann plangemäß beginnen, selbst wenn der Vorgänger seinen Gesamtpuffer ausnutzt)

Vorgänge mit $GP = 0$ heißen *kritisch*, eine ununterbrochene Folge solcher vom Beginn- zum Endereignis *kritischer Pfad*.

PERT = Program Evaluation & Review Technique

Hier werden Vorgangsdauern durch Zufallsvar. modelliert: Einfach parametrisierbar, linksschief, endl. realisiert und somit geeignet ist die β -Verteilung mit Dichtefkt. $k(t-a)^\alpha(b-t)^\beta$ (wenn $t \in [a; b]$, sonst 0) für $b > a, \alpha, \beta > -1, k = [(b-a)^{\alpha+\beta+1} B(\alpha+1, \beta+1)]^{-1}$. Sie erreicht ihr Maximum bei $m = \frac{\beta a + \alpha b}{\alpha + \beta}$, unter der vereinfachenden Annahme $\alpha + \beta = 4$ liegen ihr Erwartungswert bei $t_e = \frac{1}{6}(a + 4m + b)$ und ihre Varianz bei $\sigma^2 = \frac{1}{36}(b-a)^2$.

Schätzt man zu jedem Vorgang a, b und m als opti-, pessimistische bzw. wahrscheinl. Dauer, lassen sich in einem *Ereignisknotennetz* Erwartungswerte und Varianzen für die frühestmöglichen und spätesten Zeitpunkte der Ereignisse durch Vorwärts- und Rückwärtsrechnung à la CPM aus jenen der Vorgangsdauern berechnen. Die Differenz jener Zeitpunkte eines Ereignisses kann als Schlupfzeit T_{Schlupf} interpretiert werden, deren Varianz sich aus der Summe der Varianzen der beteiligten Zeitpunkte berechnet. Die Wahrscheinlichk. für solche Schlupfe läßt sich aufgrund des Zentralen Grenzwertsatzes durch Einsetzen von $z = \frac{T_{\text{Schlupf}}}{\sigma_{T_{\text{Schlupf}}}}$ in $e^{-\frac{1}{2}z^2}$ berechnen (da sich viele unabh. β -Verteilungen summieren); entspr. auch die des Einhaltens eines Zeitlimits als Wahrscheinlichk. eines Schlupfes zw. erwartetem und Soll-Zeitpunkt ($\sigma_{\text{Soll}}^2 = 0$).

3 Transportprobleme

Zuordnungsproblem

$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \forall i \sum_j x_{ij} = 1, \forall j \sum_i x_{ij} = 1, x_{ij} \in \{0; 1\}$

(Jeder Zeile von x wird genau eine Spalte zugeordnet.)

Greedy-Heur.: Immer das kleinstmögl. verbleibende Element belegen.

Ungarische Meth.: 1. Zeilen- und Spaltenreduktion der Kostenmatrix: Subtraktion des min. Elements, bis in jeder Zeile/Spalte eine 0 existiert. Im Folgenden Abbruch, wenn komplette Zuordnung vorhanden, sonst Iteration: 2. 'Auszeichnung' von Nullen ähnl. Greedy. 3. Tauschen der Ausz. auf allen alternierenden Wegen (= nicht erweiterbarer, zyklensfreier Weg durch sich abwechselnde nicht und ausgezeichneten Nullen, beginnend bei einer nicht ausgez. 0 in einer Zeile ohne Ausz., endend bei einer nicht ausgez. 0). 4. Streichung der Spalten mit ausgez. Nullen aller sog. Sackgassen (= Pfade, die bis auf ausgez. Ende oder fehlende Zyklensfreiheit alternierende Wege sind) und der Zeilen anschließend noch

ungestrichener ausgez. Nullen. Subtraktion des min. Wertes nicht gestrichener Einträge von diesen sowie Addition desselben auf doppelt gestrichenen Einträgen garantiert Existenz mindestens einer neuen 0 (ggf. unter Verlust alter Nullen).

Transportproblem

$\min \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}, \forall i \sum_j x_{ij} = a_i, \forall j \sum_i x_{ij} = b_j, x_{ij} \geq 0,$
o.B.d.A. $\sum_i a_i = \sum_j b_j$

(Zuordnungen entsprechen nun gerichteten Lieferungen mit vorgegebenem Aufkommen a der Zeilen und Bedarf b der Spalten.)

Greedy: Das kleinstmögliche verbleibende Element maximal belegen.

NW-Ecken-Regel: Linksobertes Element max. belegen, dann 'in Treppenstufen' weiter: Wenn Lager leer, nächstes Lager, sonst nächster Kunde. Falls beides leer, eine 0 explizit als Basis-Null markieren.

Vogel'sche Approximation: Zu jeder Zeile und Spalte Differenz zwischen den zweit- und niedrigsten Kosten (*Opportunitätskosten*) bilden und nun in der Zeile oder Spalte mit max. Diff. das minimale Element belegen, anschl. Rekursion auf Restmatrix. Im Falle der *dynamischen VA-Meth.* Neuberechnung der Diff. nach jeder Belegung.

MODI (Modified Distribution Method): Wahl einer Anfangslsg., anschl. iterierte Berechnung der Δz_{ij} mit Basiswechsel, bis alle $\Delta z_{ij} \geq 0$. Grundlage: Dualisieren und Einteilen der y_i bzgl. Angebots- (u_i) und Bedarfsrestr. (v_j) führt zu $\max \sum_i a_i u_i + \sum_j b_j v_j, \forall i, j u_i + v_j \leq c_{ij}$. Komplementärer Schlupf $\Rightarrow x_{ij}(c_{ij} - u_i - v_j) = 0$ (primale Strukturale duale Schlupfvar.), für Basisvar. $x_{ij} (> 0)$ einer Anfangslösung folgt also $c_{ij} = u_i + v_j$. Dieses System ist unterbestimmt, daher z.B. Setzung $u_0 := 0$ möglich.

Vorgehen: a) Bestimmung der restlichen u_i und v_j . b) Errechnen der Δz_{ij} aus $c_{ij} - u_i - v_j = \Delta z_{ij}$ (duale Schlupfvar. gleich primalen Δz_{ij}). c) Aufnahme des zum negativsten Δz_{ij} gehörigen x_{ij} in die Basis. Dessen neuer Wert sowie jene der anderen Var. bestimmt sich aus dem *Stepping-Stone-Pfad*, dem eindeutig bestimmten Zyklus im Tableau, der die neue Basisvar. derart mit einer Teilmenge der bisherigen verbindet, daß der Pfad an ihnen allen 90°-Richtungswechsel vornimmt. Ist er gefunden, wird die neue Basisvar. mit einem '+' markiert und alle übrigen 'Stationen' des Pfades abwechselnd mit '-' und '+'. Der Wert der kleinsten mit einem '-' markierten alten Basisvar. wird nun von jeder mit '-' markierten Variable subtrahiert und zu jeder mit '+' markierten addiert. Die resultierenden Werte stellen die nächste Variablenbelegung dar, insbesondere fällt also jene Variable aus der Basis heraus, deren Wert subtrahiert bzw. addiert wurde. Der *Stepping-Stone-Pfad* läßt sich manuell finden durch Überlegungen des Typs: "Wenn der Wert der neuen Basisvar. steigen soll, muß jener einer anderen in ihrer Spalte sinken. Dann aber muß in der Zeile jener anderen Variable eine weitere im Wert steigen, in deren Spalte wiederum die Änderung kompensiert werden" usw., bis ein Pfad gefunden ist, der wieder bei der neuen Basisvar. endet.

Umladeproblem

$\min \sum_{i,j} c_{ij} x_{ij}, 0 \leq x_{ij} \leq \kappa_{ij} \geq 0,$

$\forall i \sum_{j \in \text{succ}(i)} x_{ij} - \sum_{k \in \text{pred}(i)} x_{ki} = a_i, \text{ o.B.d.A. } \sum_i a_i = 0.$

Spalten und Zeilen stehen nun für dieselben Orte, zw. denen im Rahmen der Kapazitäten κ Transporte erfolgen können, um die vorgeschriebenen Warensaldi a zu erfüllen. Es kann zusätzl. gefordert werden, daß sich die Orte in solche reiner Produktion ($a_i > 0$) nahe der Quelle, reinen Konsums ($a_i < 0$) nahe der Senke sowie reinen Umladens ($a_i = 0$) partitionieren lassen.

Rucksackproblem

$\max \sum_i c_i x_i, \sum_i a_i x_i \leq b, x_i \in \{0, 1\}$

Greedy: Sortieren nach Rentabilität $\frac{c_i}{a_i}$

Branch&Bound I: Wurzel: leerer Rucksack; Branching durch Hinzufügen bzw. explizites Ausschließen gewisser x_i , Betrachtungsreihenfolge der x_i nach Rentabilität; Bound: theoretisch erreichbarer Gewinn (= Wert aller nicht ausgeschlossenen x_i); Wahl des nächsten Knotens: best first

Branch&Bound II: Anwendung des Verfahrens von Dakin (s.u.); plus ggf. frühzeitiger Ausschluß nicht mehr in den Rucksack passender Elemente.

Travelling-Salesman-Problem

Das TSP ist ein um gewisse Bed. erweitertes Zuordnungsproblem:

$c_{ij} \geq 0, (i, j) \notin E \Rightarrow c_{ij} = \infty, \forall W \neq \emptyset \sum_{i \in W, j \notin W} x_{ij} \geq 1$

Nearest-Neighbour-Heuristik: Wähle laufend die Kante zum nächsten nicht gewählten Kunden, $O(|V|^2)$; anschl. schließe den Kreis; ggf. Backtracking, falls verbleibende Kunden nicht von aktuellen Tourenenden aus erreichbar

Branch&Bound: Wurzel: auf Zuordnungsprob. relaxiertes Prob.; Branching nach den Kanten des kleinsten Teilzyklus: Verbiete ($c_{ij} := \infty$) jeweils die betr. Kante sowie die von den Startknoten der bereits abgehandelten Kanten in den Restgraphen außerhalb des Teilzyklus führenden; Bound: Kosten der Lsg. des aktuellen Zuordnungsprob.

Vehicle-Routing-Problem

$V = \text{Depots} \cup \text{Kunden}$: Besuche jeden Kunden genau einmal mit einem Fahrzeug, welches anschl. zum Depot zurückkehrt.

Varianten: E un-/gerichtet, Kosten C symmetr./euklid., variable Anz. Fahrzeuge (pro Depot), Fixkosten pro Fahrzeug/Depot, Kapazitäten der Fahrzeuge, Restriktionen bzgl. Anz. Kunden pro Route, Reise-/Wartezeit, Zeitfenster/Reihenfolge der Kunden

Savings-Verfahren (Eröffnungsheur.): Start mit Pendeltouren. Zu jedem Kundenpaar (i, j) Berechnung der Einsparung, wenn Touren kombiniert würden ($s_{ij} = d_{0i} + d_{j0} - d_{ij}$), Realisierung der Tourfusionen nach absteigendem s_{ij} , wenn $s_{ij} \geq 0, i, j$ aus versch. Touren, dort jew. erster bzw. letzter, und erlaubte Kapazität der komb. Tour nicht überschritten.

Sweep-Verfahren (Eröffnungsheur.): Polarkoordinaten der Kunden berechnen, einen auf 0° setzen, dann Tour mit steigendem Winkel erweitern bis Kapazität erschöpft. Wdh., bis alle Kunden bedient, dann TSP für jede Tour lösen.

Verbesserungsheur. (teilw. auch für TSP): k -opt 'tauscht' k Kanten (gemeint sind Ersetzungen der Form $(a, b), (c, d) \rightsquigarrow (a, d), (c, b)$), Or -opt beschränkt dies auf bis zu drei benachbarte, 2 -opt* auf den Tausch zweier Kanten verschiedener Touren. Ebenfalls zw. zwei Touren tauschen *Cross* (Bedingung: Tausch zweier Abschnitte) und λ -*Interchange* (Bed.: Tausch von Mengen von max. λ Knoten. (Nach welchen Kriterien getauscht wird, bleibt zunächst offen.)

Verfahren von Dakin

Lösung eines gemischt-ganzz. LOP mittels Branch&Bound: Um Ganzzahligkeitsbedingung relaxiertes Prob. als Wurzel; Branching für $x_i \notin \mathbb{Z}$, dabei pro Kindknoten eine neue Schranke $\leq [x_i]$ bzw. $\geq [x_i]$; Knotenwahl nach erwarteter Änderung des Zielfunktionswertes, Abweichung von \mathbb{Z} oder externe Prioritäten.

Branch & Bound

Lösung eines Optimierungsprob. durch Lösung einer Problemfolge.

Wurzel: Ausgangs- oder relaxiertes Prob., aktuelle Schranke $\pm \infty$ oder bekannter Zielfunktionswert

Branching: Verzweigung untersucher, nicht terminierter Teilmodelle (zu def.: wie, aber auch wo: best-/depth-first, dynamisch (depth \rightarrow best) oder extern bestimmt)

Bounding: Abbruch, wenn Lösungsmenge leer oder erwartbare Lsg. nicht besser als bekannte

4 Lagerhaltung

Harris-Wilson

Gegeben: Bestellfixe Kosten K , Preis/ME c , Lieferzeit λ , Lagerkosten/t/ME h , Abgangsrate μ

Gesucht: Kostenminimierende Bestellpolitik (Menge Q , Abstand $\tau = \frac{Q}{\mu}$), also $\min C(Q) = \tau^{-1}(K + cQ + \frac{1}{2}Qh\tau)$

$\rightsquigarrow Q^* = \sqrt{\frac{2K\mu}{h}}, \tau^* = \sqrt{\frac{2K}{\mu h}} \Rightarrow C(Q^*) = \sqrt{2K\mu h} + c\mu$

Modifikation: Lieferverzug erlaubt und als negativer Lagerbestand modelliert: Bei Bestellung sofortige Erfüllung des Verzuges, neuer Lagerbestand y ; Lagerbestand über τ' hinweg positiv, über τ'' negativ mit Verzugskosten/t/ME p

Gesucht: Bestellpolitik $(Q, y, \tau = \tau' + \tau'')$ zu $\min C(Q, y) = \tau^{-1}(K + cQ + \frac{1}{2}yh\tau' + \frac{1}{2}(Q - y)p\tau'')$

$\rightsquigarrow Q^* = \sqrt{\frac{2K\mu}{h}} \sqrt{\frac{h+p}{p}}, y^* = \sqrt{\frac{2K\mu}{h}} \sqrt{\frac{p}{h+p}}, \tau^* = \sqrt{\frac{2K}{\mu h}} \sqrt{\frac{h+p}{p}}$

$\Rightarrow C(Q^*, y^*) = \sqrt{2K\mu h} \sqrt{\frac{p}{h+p}} + c\mu$ (für $p \rightarrow \infty$ Annäherung an das Ausgangsmodell, da Lieferverzug de-facto nicht vorteilhaft)

Dynamisches Deterministisches Modell

Im Gegensatz zur Modellierung für Harris-Wilson wird nun der Warenabgang nicht mit einer konstanten Rate, sondern als Funktionenschar $\beta_i(t)$ für eine vorgegebene Menge von Zeitperioden $i \in \{1, \dots, n\}$ eben-

falls vorgegebener Länge τ definiert, die für ein $t \in [0; \tau]$ angibt, wieviel Ware in Periode i bis zum Zeitpunkt t abgegangen sein wird. Zu bestimmen ist eine kostenminimierende Bestellpolitik für jede Periode i , wobei die Bestellmengen (am Anfang von i) u_i heißen und sich die Lagerbestände x_i (am Ende von i) gemäß $u_i = x_i - x_{i-1} + z_i$, $x, u, z \geq 0$ aus altem Lagerbestand, Bestellmenge und Abgang $z_i := \beta_i(\tau)$ ergeben, wenn Anfangs- und Endbestand als $x_0 = x_n = 0$ festgelegt sind. (Da Lieferverzöger wieder nicht erlaubt ist, muß jede zulässige Bestellpolitik eine *mittlere Mindestreserve* $\xi_i = \tau^{-1} \int_0^\tau (z_i - \beta_i(t)) dt$ vorhalten.) Unter den genannten Restriktionen zu optimieren ist also das Problem:
 $\min C(u) = \sum_{i=1}^n [K \cdot \delta(u_i) + cu_i + h \int_0^\tau (x_i + z_i - \beta_i(t)) dt]$ ($\delta = \text{sgn}$)
 $\hat{=} \min C(u) = \sum_{i=1}^n [K \cdot \delta(u_i) + h\tau x_i] =: z(x, u)$ (Dyn. Opt'modell)

Bellmann'sches Optimalitätsprinzip

Das BO bezieht sich auf n-stufige Entscheidungsprozesse (wie dem DDM), bei denen eine Zustandstrajektorie x und ein Entscheidungsvektor y entsprechend $x_i = T_i(x_{i-1}, y_i)$ zusammenhängen und ein stufenbasiertes Zielkriterium $z(x, y) = F(c_1(x_0, y_1), \dots, c_n(x_{n-1}, y_n))$ (mit Kostenfkt. F und c_i) zu optimieren (minimieren) ist. Das Prinzip, daß die Optimalität eines Restprozesses nur vom Anfangszustand, nicht aber von den dafür verantwortlichen Entscheidungen abh. ist, ist z.B. für additive Zielfkt. $z(x, y) = \sum_i c_i(x_{i-1}, y_i)$ erfüllt. Hierzu lassen sich die Bellmann'schen Funktionalgleichungen aufstellen:

$f_{n+1}^*(x_n) = 0$, $f_i^*(x_{i-1}) = \min_{y_i} f_i(x_{i-1}, y_i)$,
 $f_i^*(x_{i-1}, y_i) = c_i(x_{i-1}, y_i) + f_{i+1}^*(x_i)$, $x \in X_i$, $y_i \in Y_i(x_{i-1})$
(X_i und $Y_i(x_{i-1})$ liefern zulässige Definitionsbereiche). Es gilt, daß $f_i^*(x_0) = \min_y z(x, y)$ und \exists opt. Entscheidungsfunktionen h_i^* mit $y_i^* = h_i^*(x_{i-1})$.

Das BO gilt allgemein für *monoton separable* Zielfkt.:

z separabel $\Leftrightarrow \exists$ Verknüpfungen \circ_i und Zerlegungsfunktionen g_i so daß $z(x, y) = g_1(c_1, \dots, c_n)$, $g_i(c_i, \dots, c_n) = c_i \circ_i g_{i+1}(c_{i+1}, \dots, c_n)$, $g_n(c_n) = c_n$, wobei c_i für $c_i(x_{i-1}, y_i)$ steht. Ist z separabel und sind die g_i für alle c_i monoton wachsend in g_{i+1} , heißt z monoton separabel, und es ergibt sich (analog Bellmann):

$$\min g_i(c_i, g_{i+1}) = \min g_i(c_i, \min g_{i+1}) = \min(c_i \circ_i \min g_{i+1})$$

Wagner-Whitin

Die Rechtsbündigkeit der Bellmann'schen Rekursion legt eine Lösung des Problems 'von hinten' nahe, die Dyn. Opt. läßt sich aber auch 'vorwärts' formulieren: $C_0^*(x_0) = 0$, $C_i^*(x_i) = \min[K \cdot \delta(u_i) + h\tau x_i + C_{i-1}^*(x_{i-1})]$. Dann ist $C_n^*(x_n) = 0 = \min z(x, u)$ die optimale Lsg.

Sei nun (x_{i-1}^*, u_i^*) ein opt. Lösungspaar, dann erlauben die Eigenschaften $\delta(u_i^*) \oplus \delta(x_{i-1}^*) = 1$ sowie $u_i^* \in \{0; z_i; z_i + z_{i+1}; \dots; z_i + \dots + z_n\}$ folgende vereinfachte Darst.: $C_0^* = 0$, $C_1^* = K$,

$$C_j^* = \min_{k \in \{1, \dots, j\}} p_j(k), \quad k_j^* = \arg \min_{k \in \{1, \dots, j\}} p_j(k),$$

$$p_j(k) = K + h\tau \sum_{i=k}^j \sum_{\gamma=i+1}^j z_\gamma + C_{k-1}^*$$

Gemäß dem Verfahren von Wagner-Whitin führt dies zu folgender opt. Bestellpolitik: $u_{k_n}^* = \sum_{i=k_n}^n z_i$, $u_{k_n+1}^* = \dots = u_n = 0$, die restlichen u_i erhält man durch iterative Substitution von n mit $k_n^* - 1$.

Die Gesamtkosten betragen so $C^* = C_n^* + c \sum_{i=1}^n z_i + h\tau \sum_{i=1}^n \xi_i$.

Weitere Vereinfachung ergibt sich aus der Monotonie der $\{k_j^*\}$: C_j^* und k_j^* sind effizienter berechenbar unter Berücksichtigung, daß $k \in \{k_{j-1}^*, \dots, j\}$. Außerdem ist folgende Rekursion möglich:

$$p_j(k) = p_{j-1}(k) + (j-k)h\tau z_j, \quad p_j(j) = K + C_{j-1}^*$$

5 Nichtlineare Optimierung

Kuhn-Tucker-Bedingungen

Sei das Problem $\min f(x)$, $g_i(x) \leq 0$ gegeben und $f, g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ stetig diff'bar. Ein x ist optimal, wenn alle verbessernden Richtungen aus dem zulässigen Bereich hinaus zeigen. Verbessernd ist eine Richtung, wenn sie $< 90^\circ$ vom Vektor des umgekehrten Gradienten der Zielfkt. (steilster Abstieg) abweicht. Unzulässig ist sie, wenn sie $< 90^\circ$ vom Vektor des Gradienten einer Restriktionsfkt. abweicht, die an x 'aktiv' ist (d.h. bereits 0 ist). Läßt sich für ein x der Vektor des steilsten Abstiegs als Linearkomb. (Koeff. ≥ 0) aus den Gradientenvekt. der aktiven Restriktionen darstellen, so gibt es keine verbessernde und zulässige Richtung, x ist lokales Optimum. Dies wird durch die KTB formalisiert:

$\exists u \nabla f(x^*) + \sum_{i \in I} u_i \nabla g_i(x^*) = 0$, $\forall i \ g_i(x^*) \leq 0 \wedge u_i \geq 0$, wobei $I = \{i | g_i(x^*) = 0\}$ die Indexmenge der aktiven Restr. ist.

Notw. Bed.: x^* lok. Min. $\wedge \forall i \in I \ \nabla g_i(x^*)$ lin. unabh. \Rightarrow [KTB erfüllt]

Hintr. Bed.: [KTB erf.] $\Rightarrow x^*$ lokales Min. (mit f, g konvex \Rightarrow glob. Min.)

Modifikation: $I := \{1, \dots, n\}$, $u_i g_i(x^*) = 0$ (entweder ist die Restriktion aktiv, oder $u_i = 0$ 'kann bei der Konvexkomb. nicht helfen')

Die KTB testen also auf Optimalität, sind aber auch zur konstruktiven Bestimmung von Lsg. verwendbar (s. *Quadrat. Programmierung*).

Lagrange-Version der KTB

(Folgendes gilt für $x, x^0, u, u^0 \geq 0$.) Zu o.g. Modell ist die *Lagrange-Fkt.* def. als $L(x, u) = f(x) + u^T g(x)$.

(x^0, u^0) heißt *Sattelpunkt* von $L(x, u) \Leftrightarrow \forall x, u \geq 0 \ [L(x^0, u) \leq L(x^0, u^0) \leq L(x, u^0)]$, x^0 ist dann optimal (minimiert f).

Seien die partiellen Abl. benannt $L_x(x, u) = \nabla f(x) + \sum_i u_i \nabla g_i(x)$ und $L_u(x, u) = g(x)$, dann lassen sich die KTB formulieren als:

$L_x(x^0, u^0) \geq 0$, $L_u(x^0, u^0) \leq 0$, $x^{0T} L_x(x^0, u^0) = 0$, $u^{0T} L_u(x^0, u^0) = 0$
Notw. Bed.: (x^0, u^0) Sattelpkt. \Rightarrow [L-KTB erf.]

Hintr. Bed.: [L-KTB erf.], L konvex in x , konkav in $u \Rightarrow (x^0, u^0)$ Sattelp. Slater-Bed.: Existiert ein (somit innerer) Punkt p , für den alle nichtlinearen Restr. mit $g(p) < 0$ erfüllt sind, so erfüllt p die Slater-Bed. Es gilt: [L-KTB und Slater erf.] $\wedge L$ konvex in $x \Leftrightarrow x^0$ optimal

Quadratische Programmierung

Für Probleme der Form $\min f(x) = c^T x + \frac{1}{2} x^T Q x$, $Ax \leq b$, $x \geq 0$, bei denen Q symmetr. und positiv semidefinit (Eigenwerte ≥ 0) sind ($\Rightarrow f$ konvex), ist $L(x, u) = c^T x + \frac{1}{2} x^T Q x + u^T (Ax - b)$ konvex in x und konkav in u , daher sind die [L-KTB] Optimalitätsbed., die hier lauten:

a) $L_x^0(x^0, u^0) = c + Qx^0 + A^T u^0 \geq 0$, b) $L_u(x^0, u^0) = Ax^0 - b \leq 0$,

c) $x^{0T} L_x(x^0, u^0) = 0$, d) $u^{0T} L_u(x^0, u^0) = 0$.

Der *Algorithmus von Wolfe* minimiert f unter den Restr. a) und b) mittels Simplex mit beschränktem Basiseintritt: Ist x_i in der Basis, wird die zugehörige Schlupfvar. aus a) nicht aufgenommen (und umgekehrt), gleiches gilt für u_i und die Schlupfvar. aus b). Von einer nur aus Schlupf- und Hilfsvar. bestehenden Basis ausgehend bleiben daher c) und d) erfüllt.

Unrestringierte Optimierungsprobleme

Zwecks $\min f(x)$ gilt es, von einem Wert x^k zu einem besseren nächsten zu gelangen, $x^{k+1} = x^k + \mu^k d^k$, wobei $\mu^k > 0$ und d^k Verbesserungsrichtung. Die Wahl des Startwertes x^0 kann relevant sein.

Das *Gradientenverf.* setzt schlicht $d^k := -\nabla f(x^k)$, bestimmt den minimierendsten Koeff. μ^k und bricht bei $|d^k| < \varepsilon$ ab. Eine Verbesserung ist garantiert, wird aber häufig auf einem Zick-Zack-Kurs erreicht.

Das *mehrdim. Newtonverf.* setzt voraus, daß f zweimal stetig diff'bar ist: Eine lokale quadrat. Approximation q (Taylor-Entw. zweiten Grades an x^k) wird benutzt, um die 'nächstbeste' Stelle zu finden. Sei $H(x)$ Hesse-Matrix von f (= Matrix der zweiten partiellen Abl.), dann gilt $q(x) := f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2} (x - x^k)^T H(x^k) (x - x^k)$.

q hat Optima bei $\nabla q(x) = 0 \Leftrightarrow \nabla f(x^k) + H(x^k)(x - x^k) = 0 \Leftrightarrow x^{k+1} := x^k - H^{-1}(x^k) \nabla f(x^k)$. (+) Quadratische Konvergenz: $\|x^{k+1} - x^k\| < \gamma \|x^k - x^*\|^2$, $\gamma \in]0; 1[$, Schrittweitenbestimmung unnötig; (-) braucht guten Start, da nur lokal optimal, Berechnung von $H(x)$ (bei *Quasi-Newton* nur in erster Iteration, später approx. Anpassung) und Inversion aufwendig/unmöglich.

Restringierte Optimierungsprobleme

Das *Verfahren zulässiger Richtungen* (Verf. von Zoutendijk) fordert von verbessernden Richtungen zusätzl. Zulässigkeit: d ist verbessernde & zul. Richtung in $x \Leftrightarrow \nabla f(x)^T d < 0 \wedge \nabla g_{i \in I}(x)^T d < 0$, wobei wieder $I = \{i | g_i(x) = 0\}$. Zu lösen ist (mittels Simplex mit z und den Komponenten d_j von d als Entscheidungsvar.):

$\min z$, $\nabla f(x)^T d - z \leq 0$, $\nabla g_{i \in I}(x)^T d - z \leq 0$, $\forall j \ d_j \in [-1; 1]$, $z \leq 0$. Ist $z = 0$, ist das Optimum erreicht (x ist *Kuhn-Tucker-Punkt*), sonst ist d verbessernde & zul. Richtung.

Umwandlung restringierter in unrestringierte Probleme

Zulässiger Bereich: $Z = \{x | g_i(x) \leq 0\}$

Das *Penalty-Verf.* integriert die Restriktionen in Form von Strafkosten in die Zielfkt. und nähert sich Z so i.d.R. von außen. Def. wird eine Folge $\{\mu_r S_r(x)\}_{r>0}$ mit $\mu_r > 0$, $\forall x \notin Z \ (S_r(x) > 0 \wedge \forall r \ \mu_{r+1} S_{r+1}(x) > \mu_r S_r(x) \wedge (r \rightarrow \infty \Rightarrow \mu_r S_r(x) \rightarrow \infty))$, $\forall x \in Z \ S_r(x) = 0$, $S_r(x)$ stetig, mit der das neue Problem $\min(f(x) + \mu S(x))$ formuliert wird. μ sollte wachsen, z.B. $\mu_{r+1} = \beta \mu_r$, $\beta > 1$. Bsp. für eine Strafkostenfkt. im Falle von Restr. $g_i(x) \leq 0$, $h_j(x) = 0$ ist

$$S(x) = \sum_i [\max(0, g_i(x))]^p + \sum_j |h_j(x)|^p, \quad p > 0.$$