Eine Heuristik basierend auf der ~~A heuristic based on the~~ Lagrange Relaxation für das Index Fonds Problem

$$Z := \max \sum_{i=1}^{n} \sum_{j=1}^{n} \rho_{ij} x_{ij}$$

u.d.Nb.

$$\sum_{j=1}^{n} y_j = q$$

(Anmerkung: ursprünglich $\sum_{j=1}^{n} y_j \leq q$)

(P)

$$x_{ij} \leq y_j \quad \forall\, i,j \in \overline{1,n}$$

relaxiert →

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \forall\, i \in \overline{1,n}$$

$$x_{ij}, y_j \in \{0,1\} \quad \forall\, i \in \overline{1,n}$$

Anmerkung: Das Problem löst sich mit allgemeinen Solvern (Excel, Matlab CVX) oder mit kommerziellen Solvern (CPLEX, GUROBI) nur für relativ kleine Werte von n.

Hauptschwierigkeit: Hohe Anzahl an binären Variablen n^2+n sowie Restriktionen n^2+n+1.

Standard Solver haben schon Probleme für $N=100$!

In praktischen Instanzen des Problems, die in Clustering-Anwendungen in Datenanalyse vorkommen, gilt oft $N \approx 10^5$.

Bei solchen Dimensionen des Problems gibt man sich auch mit heuristischen Lösungen zufrieden.

die Lagrange Relaxation des Problems (P)
mit Lagrange-Multiplikatoren $u \in \mathbb{R}^n$:

$$L(u) := \max_{x,y} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} x_{ij} + \sum_{i=1}^{n} u_i \left(1 - \sum_{j=1}^{n} x_{ij}\right)$$

u.d.Nb

$$(\overline{P}) \qquad \sum_{j=1}^{n} y_j = q$$

$$x_{ij} \le y_j \qquad \forall i,j \in \overline{1,n}$$

$$x_{ij}, y_j \in \{0,1\}, \quad \forall \; i,j \in \overline{1,n}$$

Eigenschaft 1
Aus Proposition 4.3 folgt $L(u) \geq Z$ (Achtung wir maximieren!)

Eigenschaft 2 Für ein fixes $u \in \mathbb{R}^n$ lässt sich $\overline{P}$ leicht lösen, wie folgt.

Reformuliere die Zielfunktion als $\to 0 \Rightarrow x_{ij} := y_j$

$$L(u) := \max \sum_{i=1}^{n} \sum_{j=1}^{n} (f_{ij} - u_i) x_{ij} + \sum_{i=1}^{n} u_i$$

Für ein gegebenes $y \in \mathbb{R}^n$ x_{ij} sollte entweder auf die obere Schranke y_j oder auf die untere Schranke 0 fixiert werden, je nach dem ob $f_{ij} - u_i > 0$ oder $f_{ij} - u_i \le 0$ (tie breaking arbitrarily).

Also lässt sich das Problem $\overline{P}$ erneut reformulieren:

$$(\overline{P}) \qquad \max \sum_{j=1}^{n} c_j y_j + \sum_{i=1}^{n} u_i \qquad \text{gleich geblieben}$$

u.d.Nb

$$\sum_{j=1}^{n} y_j = q$$

$$y_j \in \{0,1\} \quad \forall j \in \overline{1,n},$$

wobei $c_j := \sum_{i=1}^{n} \max\{0, s_{ij} - u_i\}$

Beobachtung $(\overline{P})$ is trivial lösbar:

Sortiere $(c_j)_{j \in \overline{1,n}}$ absteigend, d.h. $c_{j_1} \geq c_{j_2} \geq \dots \geq c_{j_n}$

und setze $y^*_{j_1} = y^*_{j_2} = \dots = y^*_{j_q} = 1$ und $y^*_j = 0, \forall j \neq j_\ell$, $\ell \in \overline{1,q}$

Dann gilt $L(u) = \sum_{\ell=1}^{q} c_{j_\ell} + \sum_{i=1}^{n} u_i$

Für Opt. lösung von $\overline{P}$ setze $x_{ij} = y_j \iff s_{ij} - u_j > 0$, sonst $x_{ij} = 0$.

Eigenschaft 3

Basierend auf der Optimallösung y^* von $\overline{P}$ kann eine heuristische Lösung (x^*, y^*) von P konstruiert werden. Darüber hinaus lässt sich auch eine Abschätzung für die Qualität dieser Lösung ermitteln.:

$\forall i \in \overline{1,n}$, sei $j(i) := \underset{j \in \overline{1,n} : y_j = 1}{\arg\max} s_{ij}$.

Für jedes i wird also jenes im Fonds eingewichte Asset $j(i)$ das die größte Ähnlichkeit zum i-ten Asset hat.

Setze dann $\forall i,j \in \overline{1,n}$, $x^*_{ij} = \begin{cases} 1 & \text{für } j = j(i) \\ 0 & \text{sonst} \end{cases}$

(x^*, y^*) ist somit eine zulässige Lösung für P

mit Zielfunktionswert $z^* = \sum_{i,j=1}^{n} s_{ij} x^*_{ij}$.

Für diese Lösung gilt $z^* \leq z \leq L(u)$ $\Rightarrow z - z^* \leq L(u) - z^*$

(für jede andere zulässige Lösung von P).

D.h. Wenn $L(u) - z^*$ "klein", so ist auch $z - z^*$ klein

In diesem Sinne kann man mit Hilfe von $L(u)$ die Güte der Lösung (x^*, y^*) bewerten.

Eigenschaft 4

Um die beste obere Schranke $L(u)$ und die dazugehörige heuristische Lösung (ermittelt wie in Eigenschaft 3) zu berechnen wird

$$\min_{u \in \mathbb{R}^n} L(u) \qquad \text{gelöst.}$$

Das ist ein unrestringiertes Minimierungsproblem mit einer konvexen, i.a. nicht differenzierbaren, Zielfunktion. So ein Problem lässt sich mit Hilfe eines Subgradienten-Verfahrens, eine Verallgemeinerung des Newton-Verfahrens für die unrestringierte Optimierung mit differenzierbarer Zielfunktion

Betrachten wir nun allgemein ein $f: \mathbb{R}^n \to \mathbb{R}$ konvex. und ein Optimierungsproblem (unrestringiert)

$$\min_{x \in \mathbb{R}^n} f(x)$$

Def: Ein $g \in \mathbb{R}^n$ heißt Subgradient von f in $x_0 \in \mathbb{R}^n$ falls $\forall\, y \in \mathbb{R}^n$

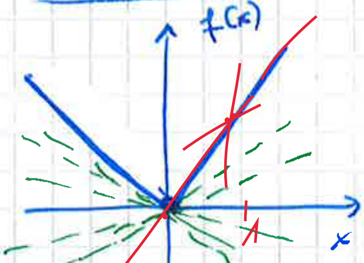
$$f(y) - f(x_0) \geq g^t (y - x_0) \quad \text{gilt.}$$

Das Subdifferential von f in x_0, bezeichnet mit $\partial f(x_0)$, ist die Menge aller Subgradienten von f in x_0:

$$\partial f(x_0) := \{ g \in \mathbb{R}^n : g \text{ ist Subgradient von } f \text{ in } x_0 \}$$

Das Subdifferential einer konvexen Funktion ist nicht leer in jedem Punkt $x_0 \in \mathbb{R}^n$, d.h. $\partial f(x_0) \neq \emptyset$.

Beispiel $f(x) = |x|$, $f: \mathbb{R} \to \mathbb{R}$



$$\partial f(x_0) = \begin{cases} 1 & x_0 > 0 \\ -1 & x_0 < 0 \\ [-1,1] & x_0 = 0 \end{cases}$$

$\partial f(0)$ besteht aus den Steigungen der grünen strichlierten Geraden

$x_0 = 0$ — $|x| \geq g \cdot x \ \forall x \in \mathbb{R} \Rightarrow g \in [-1, 1]$ $\quad$ $x_0 = 1$ $\quad$ $|x| - 1 \geq g(x-1) \ \forall x \in \mathbb{R} \overset{!}{\Rightarrow} g = 1$

Anmerkung: falls f in x_0 differenzierbar, dann gilt $\partial f(x_0) = \{\nabla f(0)\}$
→ Gradient for f in x_0

In diesem Sinne sind Subgradienten eine Verallgemeinerung der Gradienten.

Algorithmus: Subgradienten-Verfahren (generisch)

Input: $f: \mathbb{R}^n \to \mathbb{R}$, convex

Output: eine heuristische Lösung (approximative Lösung) für $\min_{x \in \mathbb{R}^n} f(x)$

(1) Wähle $x_0 \in \mathbb{R}^n$

(2) for $i = \cancel{0}1, 2, \ldots$ do /* bis ein Stoppkriterium erfüllt wird */
  - wähle $g_i \in \partial f(x_i)$
  - Wähle $\alpha_i > 0$ /* Schrittweite */
  - setze $x_{i+1} := x_i - \alpha_i g_i$

 end for

(3) Output the current x_i

Es gibt einige Fragen bzgl. der Implementierung dieses Algorithmus.

a) Wahl der Startlösung x_0

b) Wahl der Schrittlänge α_i in jeder Iteration.

Ad b) Backtracking Ansätze wie etwa der Armijo-Goldstein Ansatz für den Algorithmus des steilsten Abstiegs funktionieren nicht im Fall von nicht differenzierbaren Funktionen.

Es gibt viele andere Ansätze in der Literatur zB ein sehr einfacher Ansatz wäre $\alpha_i = \alpha > 0 \ \forall i$

Ein weiterer einfacher Ansatz: wähle α_i monoton fallend und so dass $\sum_{i=0}^{\infty} \alpha_i^2 < \infty$ und $\sum_{i=0}^{\infty} \alpha_i = \infty$

Exakte Algorithmen zur Lösung gemischt-ganzzahliger linearer Optimierungsprobleme

- Branch and Bound (B&B) (Land and Doig 1960)
- Cutting Planes (CP) (Dantzig 1954 für das Rundreiseproblem) (Gomory 1958, 1960 für allgemeine MILP)
- Kombiniertes Verfahren: Branch and Cut (B&C) (Padberg and Rinaldi 1987)

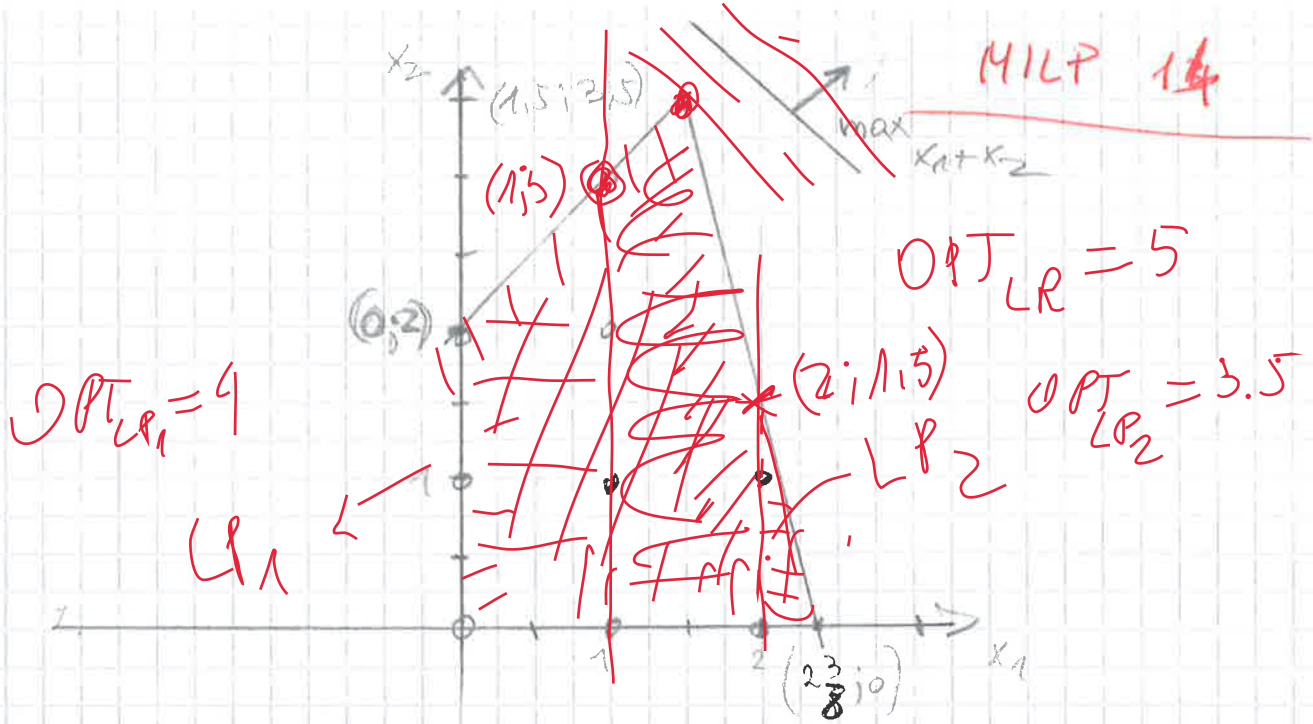
Example

$$\max \quad x_1 + x_2$$

u.d. N/B

$$(P) \qquad \begin{aligned} -x_1 + x_2 &\leq 2 \\ 8x_1 + 2x_2 &\leq 19 \\ x_1, x_2 &\geq 0 \\ x_1, x_2 &\in \mathbb{Z} \end{aligned}$$

MILP 14



Die lineare Relaxation

Das
$$(LP)\quad \max\ x_1+x_2 \quad \text{und NB}\quad -x_1+x_2\le 2,\quad 8x_1+2x_2\le 19,\quad x_1,x_2\ge 0$$

hat Opt.lösung $\bar{x}:=(1{,}5;3{,}5)$ mit Zielfunkt.wert 5.

Es gilt also $Opt_P \le 5$.

$\bar{x}$ ist nicht zulässig für P.

Wie können wir diese nicht-ganzzahlige Lösung des Problems ausschließen ohne ganzzahlige Lösungen, also zulässige Lösungen zu verlieren?

Branch! Erzeuge 2 neue LPs LP_1 und LP_2 mit zusätzlichen Restriktionen $x_1\le 1$ bzw. $x_1\ge 2$.

Jede zul. Lösung von P muss eine zulässige Lösung für eins der 2 LPs LP_1 und LP_2.

$$(LP_1)\quad \max\ x_1+x_2 \quad \text{und NB}\quad -x_1+x_2\le 2,\quad 8x_1+2x_2\le 19,\quad x_1\le 1,\quad x_1,x_2\in\mathbb{R}_+$$

$$(LP_2)\quad \max\ x_1+x_2 \quad \text{und NB}\quad -x_1+x_2\le 2,\quad 8x_1+2x_2\le 19,\quad x_1\ge 2,\quad x_1,x_2\in\mathbb{R}_+$$

HILP 15

Opt. Lösung von LP1 : $\bar{x} = (1;3)$ mit Zielfkt. wert 4.
Diese Lösung ist auch zulässig für P.
D.h. jetzt haben wir auch eine untere Schranke für Opt_P : $LB := 4 \le Opt_P \le 5 =: UB$

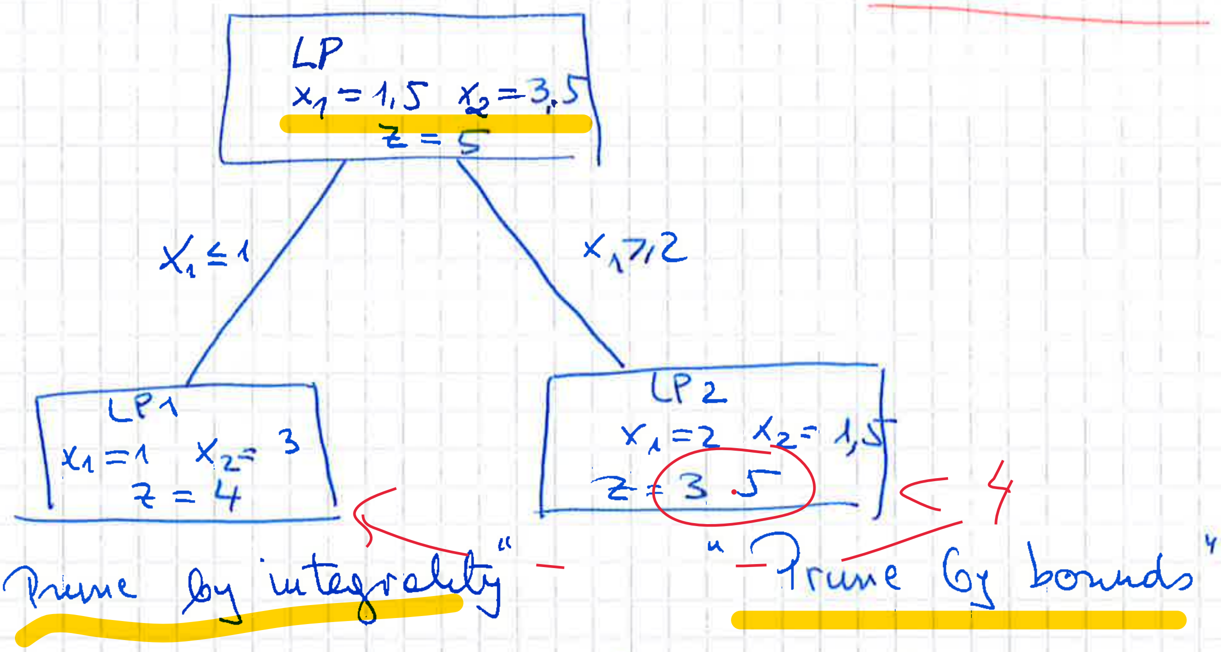
Opt. Lösung von LP2 : $\bar{x} = (2; 1,5)$ mit Zielfkt. wert $= 3,5$

Achtung: 3,5 ist kleiner als die vorhandene untere Schranke $LB = 4$ für Opt_P, und im zulässigen Bereich des LP2 können keine Lösungen von P mit einem besseren Zielfkt. Wert als 3,5 vorkommen. Also braucht man nicht weiter nach der Opt. Lösung von P im zulässigen Bereich von LP1 zu finden. (LP2) kann eliminiert werden ("fathom LP2" oder "prune LP2")

Im zulässigen Bereich von LP1 kann auch keine bessere Lösung als (1,3) gefunden werden

$\Longrightarrow$ (1,3) ist eine Opt. Lösung von P.

Wir mussten 3 LPs lösen; diese lassen sich in einem sogenannten Branch and Bound Baum (B&B-Baum) organisieren, in dem jeder Knoten einem dieser gelösten LPs entspricht.



Die Suche nach Lösungen in einem Knoten des B&B-Baum kann ~~aus~~ 3 Gründen unterbrochen werden

(i) Ganzzahligkeit der Optimallösung der LP-Relaxation in diesem Knoten
(prune by integrality)

(ii) Der optimale Zielfunktionswert der LP-Relax. in diesem Knoten is schlechter als ~~der~~ Zielfkt.-Wert einer bekannten zulässigen Lösung des ursprünglichen Problems P
(prune by bound)

(iii) Die lineare Relaxation in diesem Knoten ist unzulässig
(Prune by infeasibility)