Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE

Ondřej Pavlíček
Řešení optimalizačních úloh s neklesajícími max-separabilními omezeními

Katedra aplikované matematiky
Vedoucí diplomové práce: Prof. RNDr. Karel Zimmermann, DrSc.
Studijní program: Informatika, Diskrétní modely a algoritmy

Děkuji vedoucímu diplomové práce profesoru Karlu Zimmermannovi za velkou pomoc a konzultace poskytnuté během vzniku této diplomové práce.

Prohlašuji, že jsem svou diplomovou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 12.12.2005 Ondřej Pavlíček

Obsah

Název práce: Řešení optimalizačních úloh s neklesajícími max-separabilními omezeními
Autor: Ondřej Pavlíček
Katedra (ústav): Katedra aplikované matematiky
Vedoucí diplomové práce: Prof. RNDr. Karel Zimmermann, DrSc.
e-mail vedoucího: karel.zimmermann@mff.cuni.cz

Abstrakt:

Obsahem této diplomové práce jsou navržené algoritmy řešící optimalizační úlohy s max-separabilní účelovou funkcí ve tvaru $f(x) = \max_{j\in J} f_j\ (x_j)$, kde f_j jsou spojité unimodální funkce. Omezení úlohy mají tvar soustavy max-separabilních rovnic a nerovností s proměnnými na obou stranách rovnic a nerovností, přičemž max-separabilní funkce vystupující v omezení úloh jsou spojité a neklesající. V kapitole 6 je rozšíření těchto algoritmů na úlohy s různými proměnnými na obou stranách. V kapitole 7 je rozšíření úlohy na úlohy s koeficienty ∞ a -∞.

Práce vychází z dříve publikovaných prací, v nichž bylo dokázáno, že množina přípustných řešení úlohy má v případě, že je neprázdná, vždy maximální prvek. Navrhované algoritmy vychází z tohoto maximálního prvku a postupně snižují hodnotu účelové funkce postupem, který je analogií metody přístupných směrů.

Součástí diplomové práce je důkaz správnosti zde navrhnutých algoritmů, odhad jejich časové náročnosti. Dále také vytvořený program pro počítání úloh s použitím zde navrhnutých algoritmů.

Klíčová slova: max-separabilní, unimodální funkce, optimalizace

Title: Solving optimization problems on two-sided linear systems in max-algebra
Author: Ondřej Pavlíček
Department: Department of Applied Mathematics
Supervisor: Prof. RNDr. Karel Zimmermann, DrSc.
Supervisor's e-mail address: karel.zimmermann@mff.cuni.cz

Abstract:

The content of this work is a presentation of algorithms solving optimization problems with a max-separable objective function of the form $f(x) = \max_{j\in J} f_j\ (x_j)$, where f_j are continuous unimodal functions. The optimization problems are solved under constraints, which are described by a system of (max,+)-linear equations and inequalities with variables occurring on both sides of the constraints. In Chapter 6, a modification of this problem with different variables on each side of the constraints is studied. Chapter 7 deals with problems, in which the constraint coefficients may be infinite.

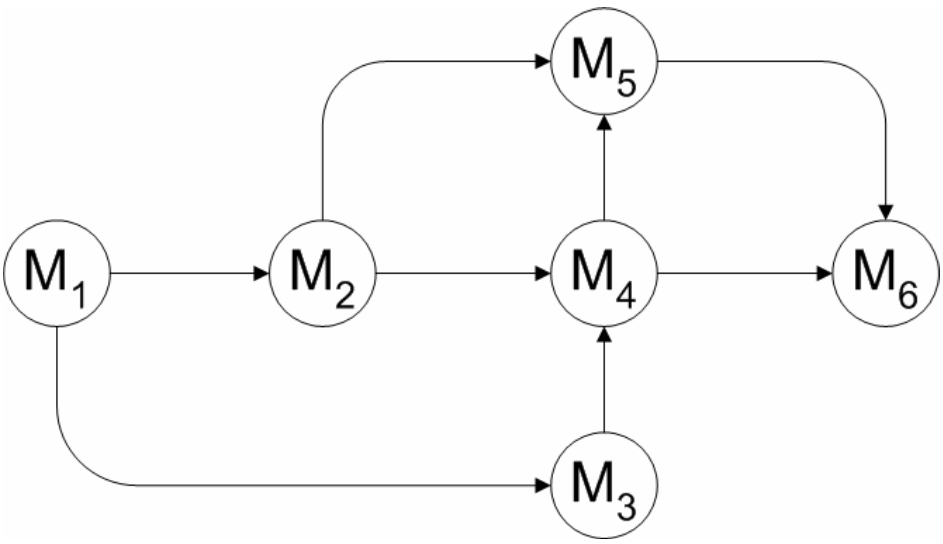
The work is based on results contained in previous publications, in which was shown that if the set of feasible solutions of the optimization problems considered here is nonempty, it has always the greatest element. The algorithms suggested in this work begin with this greatest element and decrease step by step the objective function without leaving the feasible set. The method proceeds in a certain sense by analogy with the method of feasible directions. Proofs of correctness of the algorithms and some results concerning computational complexity, as well as a computer implementation are included in the concluding part of the work.

Keywords: max-separable, unimodal function, optimization

Úvod a motivace

Tato diplomová práce se zabývá řešením úloh s max-separabilním omezeními. Pro ilustraci, k čemu úlohy s max-separabilními omezeními slouží si uvedeme příklad.

Uvažujme výrobní linku, se zdrojem M_1, stroji M_2, M_3, M_4, M_5 a koncovou stanicí M_6. Ze zdroje M_1 jde část materiálu do stroje M_2 a část do stroje M_3. Z M_2 do M_4 a M_5. A tak dále jak je znázorněno na diagramu.



Výrobní čas stroje M_i, provozní časy $M_i, i = 2,3,4,5$ a čas dokončení v M_6 označme $t_i, i = 1,2,3,4,5,6$. Dále čas potřebný pro převoz materiálu mezi jednotlivými stroji označme t_{ij} pro dobu převozu z M_i do M_j.

Celý systém pracuje v cyklech. Zajímá nás kdy bude vykonán *k*-tý cyklus. Označme $x_i(k)$ Nedřívější možný začátek operace M_i v k-tém cyklu. Každý stroj musí počkat, až dostane suroviny ze všech jemu předcházejících strojů. A samozřejmě nemůže začít následující cyklus, dokud nedokončil předchozí. Například stroj M_2 může začít $k+1$ cyklus poté co dokončí předchozí cyklus ($x_2(k)+t_2$) a dostane suroviny ze stroje M_1 ($x_1(k)+t_1+t_{12}$) Tím získáváme následující soustavu rovnic:

$$x_1(k+1) = x_1(k)+t_1$$
$$x_2(k+1) = \max\{x_1(k)+t_1+t_{12}; x_2(k)+t_2\}$$
$$x_3(k+1) = \max\{x_1(k)+t_1+t_{13}; x_3(k)+t_3\}$$
$$x_4(k+1) = \max\{x_2(k)+t_2+t_{24}; x_3(k)+t_3+t_{34}; x_4(k)+t_4\}$$
$$x_5(k+1) = \max\{x_2(k)+t_2+t_{25}; x_5(k)+t_5\}$$
$$x_4(k+1) = \max\{x_3(k)+t_3+t_{36}; x_4(k)+t_4+t_{46}; x_5(k)+t_5+t_{56}; x_6(k)+t_6\}$$

Tyto rovnice můžeme snadno upravit na maticový tvar. Dosaďme $a_{ii} = t_i$, pro $i = 1,...,6$ a $a_{ij} = t_i + t_{ij}$ pokud existuje v grafu šipka $i \rightarrow j$, jinak $a_{ij} = -\infty$. Tím dostáváme

$x(k+1) = A \otimes x(k)$, kde v operaci $\otimes$ násobení matice s vektorem použijeme $\wedge$(max) místo sčítání a +(sčítání) místo násobení.

Pokud máme dán čas, kdy by měl být k-tý produkční cyklus ukončen a zajímá nás, kdy celý proces má začít, dostáváme následující $(\oplus,\otimes)$-lineární rovnici:

$A \otimes x = b$, kde b je vektor určených časů dokončení.

Pokud bychom tento proces potřebovali synchronizovat s jiným procesem určeným rovnicí $B \otimes x = b$ dostáváme rovnici $A \otimes x = B \otimes x$, což je vlastně definicí problému kterým se zabývá tato diplomová práce.

Pozn.: *$A \otimes x = B \otimes x$ je zobecněním původní úlohy $A \otimes x = b$[1]. Nechť máme zadánu úlohu $A \otimes x = b$. Uvažujme nyní její rozšíření $A' \otimes \widetilde{x} = B' \otimes \widetilde{x}$, kde*

$$\widetilde{x} = (x, y) = (x_1, \ldots, x_n, y),\ A' = \left(\begin{array}{c|c} & -\infty \\ A & \vdots \\ & \infty \end{array}\right)\ a\ B' = \left(\begin{array}{ccc|c} -\infty & & & \\ & \ddots & & b \\ & & -\infty & \end{array}\right).$$

Potom je zřejmé že řešení této úlohy je zároveň řešením úlohy $A \otimes x = b$.

[1] viz [3]

1 Algoritmus řešící systém (max,+)-lineárních rovnic[2]

1.1 Formulace problému

Zaveďme následující značení:

$N = \{1,...,n\}, S = \{1,...,m\}, R = (-\infty, \infty),$
$R^n = R \times ... \times R$ (n - krát), $x = (x_1,...,x_n) \in R^n$,
$a_{ij}, b_{ij} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N}$ jsou dané,

$$a_i(x) \equiv \max_{j \in N}(a_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$b_i(x) \equiv \max_{j \in N}(b_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$F_i(x) = \{j \in N \mid a_{ij} + x_j = a_i(x)\} \ \forall \mathrm{i} \in \mathrm{S},$$
$$G_i(x) = \{j \in N \mid b_{ij} + x_j = b_i(x)\} \ \forall \mathrm{i} \in \mathrm{S}$$

Proměnné x_j, kde $j \in F_i(x)$ resp. $j \in G_i(x)$ budeme nazývat aktivní proměnné v bodě x pro $a_i(x)$, resp. $b_i(x)$.

Budeme uvažovat následující systém (max,+)-lineárních rovnic:

$$a_i(x) = b_i(x) \ \forall \mathrm{i} \in \mathrm{S} \tag{1}$$

Množinu všech řešení systému (1) budeme značit *M*. Dále si definujeme množinu $M(\bar{x})$ pro $\forall x \in M$ následovně:

$$M(\bar{x}) \equiv \{x \mid x \in M \ \& \ x \leq \bar{x}\} \tag{2}$$

Všimněme si, že funkce $a_i(x), b_i(x)$ jsou +-homogenní, tedy jestliže $\alpha \in R$, $x(\alpha) = (x_1 + \alpha,..., x_n + \alpha)$, potom $a_i(x(\alpha)) = a_i(x) + \alpha,\ b_i(x(\alpha)) = b_i(x) + \alpha$.
Z toho plyne, že pokud $x \in M$, potom $x(\alpha) \in M$ pro všechny $\alpha \in R$.

Lemma 1.1.1 Máme dáno $\bar{x} \in R.$ Potom $M \neq \emptyset \Leftrightarrow M(\bar{x}) \neq \emptyset.$

Důkaz:

Z toho, že $M(\bar{x}) \subseteq M$ rovnou plyne $M(\bar{x}) \neq \emptyset \Rightarrow M \neq \emptyset.$ Jestliže $M \neq \emptyset$ a $\tilde{x} \in M$, můžeme vždy nalézt takové $\alpha \in R$ aby $\tilde{x}(\alpha) \leq \bar{x}$. A jelikož $\tilde{x}(\alpha) \in M$ je také $\tilde{x}(\alpha) \in M(\bar{x})$. Máme tedy $M(\bar{x}) \neq \emptyset.$

---

[2] *převzato z [2]*

□

Definice 1.1.1 Nechť $L \subseteq R^n, \tilde{x} \in L$ a platí $x \in L \Rightarrow x \leq \tilde{x}$. Potom prvek $\tilde{x}$ se nazývá největší prvek množiny *L*.

Algoritmus uvedený v následující kapitole najde největší prvek množiny $M(\bar{x})$ nebo zjistí že je množina $M(\bar{x})$ pro dané matice *A,B* prázdná. Existence největšího prvku, v případě že množina $M(\bar{x})$ je neprázdná, je dokázána za obecnějších předpokladů v [1].

1.2 Algoritmus

Nyní se podíváme na samotný algoritmus pro nalezení největšího prvku množiny $M(\bar{x})$. Jestliže $\bar{x} \in M(\bar{x})$, potom je také řešením. Nechť tedy $\bar{x} \notin M(\bar{x})$. Bez újmy na obecnosti můžeme předpokládat, že

$$a_i(\bar{x}) \geq b_i(\bar{x}) \quad \mathrm{i} \in \mathrm{S} \tag{3}$$

Jestliže (3) neplatí stačí přejmenovat některé funkce a_i,b_i. Nechť $H(\bar{x}) \equiv \{i \in S \mid a_i(\bar{x}) > b_i(\bar{x})\}$. Jestliže $\bar{x} \notin M(\bar{x})$ potom také $H(\bar{x})$ je neprázdná. Dále platí, že $a_i(\bar{x}) = b_i(\bar{x})$ $i \in S \backslash H(\bar{x})$.

Abychom získali prvek z množiny $M(\bar{x})$, je potřeba zmenšit $a_i(\bar{x})$ pro všechna $i \in H(\bar{x})$. Abychom tohoto dosáhli, je třeba zmenšit všechny aktivní proměnné v $a_i(\bar{x}), i \in H(\bar{x})$. To znamená všechny proměnné $x_j, j \in A(\bar{x})$, které

$$A(\bar{x}) \equiv \bigcup_{i \in H(\bar{x})} F_i(\bar{x}) \tag{4}$$

Během procesu zmenšování $x_j, j \in A(\bar{x})$ je třeba se postarat o udržení rovností pro $i \in S \setminus H(\bar{x})$. Například pro rovnost $a_i(\bar{x}) = b_i(\bar{x})$ pro nějaké $i \in S \setminus H(\bar{x})$, $F_i(\bar{x}) \subseteq A(\bar{x})$ a $G_i(\bar{x}) \not\subset A(\bar{x})$ zmenšení x_j, $j \in A(\bar{x})$ naruší tuto rovnost. Abychom tomu zabránili, zahrneme do množiny snižovaných prvků další prvky, jmenovitě proměnné $x_j, j \in G_i(\bar{x}) \setminus A_i(\bar{x})$, kde $i \in S \setminus H(\bar{x})$. Nyní popíšeme konečný ALGORITMUS 1.1, který po maximálně *n-1* iteracích najde množinu $P(\bar{x})$ - proměnné, které potřebujeme zmenšit, jestliže mají být zmenšené proměnné x_j, $j \in A(\bar{x})$.

ALGORITMUS 1.1

1. $P(\bar{x}) := A(\bar{x})$;

2. $E_1 := \{i \in S \setminus H(\bar{x}) \mid F_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, G_i(\bar{x}) \not\subset P(\bar{x})\}$,
   $E_2 := \{i \in S \setminus H(\bar{x}) \mid F_i(\bar{x}) \not\subset P(\bar{x}) \,\&\, G_i(\bar{x}) \subseteq P(\bar{x})\}$;

3. Jestliže $E_1 \cup E_2 = \emptyset$, potom $P(\bar{x})$ je množina proměnných, které je potřeba snížit, KONEC.

4. $P(\bar{x}) := P(\bar{x}) \cup \bigcup_{i \in E_1} (G_i(\bar{x}) \setminus P(\bar{x})) \cup \bigcup_{i \in E_2} (F_i(\bar{x}) \setminus P(\bar{x}))$, přejdi na **2.**;

Protože $A(\bar{x})$ obsahuje alespoň jednu proměnnou a při každé iteraci se alespoň jedna proměnná přidá do $P(\bar{x})$, je ALGORITMUS 1.1 konečný a skončí po maximálně *n-1* iteracích.

Nyní popíšeme proces snižování proměnných $x_j, j \in P(\bar{x})$.

Definujme $x(t) = (x_1(t),\ldots,x_n(t))$ pro $t \geq 0$ následovně

$$\begin{aligned} x_j(t) &\equiv \bar{x}_j - t \text{ pokud } j \in P(\bar{x}), \\ x_j(t) &\equiv \bar{x}_j \quad \text{jinak} \end{aligned} \tag{5}$$

Jestliže zvýšíme parametr *t*, proměnné $x_j, j \in P(\bar{x})$ budou zmenšeny. (Tj. $x_j(0) = \bar{x}_j, x_j(t) < \bar{x}_j$ pro všechny $t>0$ a $j \in P(x)$.

Můžeme si všimnout, že je-li $t \in (0,\varepsilon)$, kde $\varepsilon > 0$ je dostatečně malé, potom pro všechna $i \in S$ taková, že $F_i(\bar{x}) \subseteq P(\bar{x})$ zůstává

$$F_i(x(t)) = F_i(\bar{x}),\ G_i(x(t)) = G_i(\bar{x}),$$

$$F_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow a_i(x(t)) = a_i(\bar{x}) - t,\quad G_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow b_i(x(t)) = b_i(\bar{x}) - t$$

a nové $H(x(t)) = H(\bar{x})$.

Předpokládejme, že $P(\bar{x}) \neq N$ (v [2] je dokázáno, že pokud $P(\bar{x}) = N$, potom $M(\bar{x}) = \emptyset$.)

Budeme zvyšovat *t*, dokud nenastane jedna z následujících podmínek.

(1) $F_i(x(t)) \neq F_i(\bar{x})$, tj. $a_i(x(t)) = \alpha_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})} (a_{ij} + \bar{x})$ pro nějaké $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x})$;

(2) $G_i(x(t)) \neq G_i(\bar{x})$, tj. $b_i(x(t)) = \beta_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})} (b_{ij} + \bar{x})$ pro nějaké $i \in S$ takové, že $G_i(\bar{x}) \subseteq P(\bar{x})$;

(3) $H(x(t)) \neq H(\bar{x})$, tj. $a_i(x(t)) = \beta_i(\bar{x})$ pro nějaké $i \in H(\bar{x})$.

Budeme definovat proměnné t_1, t_2 ,t_3 jako hodnoty, ve kterých nastaly předchozí podmínky.

Pokud $P(\bar{x}) \neq N$ (tj. $P(\bar{x}) \subset N$) a a_{ij}, b_{ij} jsou konečné, potom $\alpha_i(\bar{x}), \beta_i(\bar{x})$ jsou vždy konečné. Z předchozích podmínek můžeme dostat následující definice t_1, t_2, t_3:

$$t_1 = \min_{i \in L_1} (a_i(\bar{x}) - \alpha_i(\bar{x})),$$
$$\text{kde } L_1 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x})\}; \tag{6}$$

$$t_2 = \min_{i \in L_2} (b_i(\bar{x}) - \beta_i(\bar{x})),$$
$$\text{kde } L_2 \equiv \{i \in S \mid G_i(\bar{x}) \subseteq P(\bar{x})\}; \tag{7}$$

$$t_3 = \min_{i \in L_3} (a_i(\bar{x}) - \beta_i(\bar{x})),$$
$$\text{kde } L_3 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, i \in H(\bar{x})\}. \tag{8}$$

Nyní definujme $\tau \equiv \min(t_1, t_2, t_3)$, a $x(\tau)$ určíme jako novou horní mez. Jestliže $H(x(\tau)) \neq \emptyset$, začneme další iteraci s touto horní mezí, v opačném případě jsme nalezli řešení. Nyní již můžeme napsat ALGORITMUS 1.2 řešící celou naší úlohu.

ALGORITMUS 1.2

1. $\bar{y} := \bar{x}$
2. Jestliže $H(\bar{y}) = \emptyset$, potom $\bar{y}$ je největší prvek $M(\bar{x})$, KONEC
3. Najdi $P(\bar{y})$ pomocí ALGORITMU 1.1 (za $\bar{x}$ dosadíme $\bar{y}$). Jestliže $P(\bar{y}) = N$, potom $M(\bar{x}) = \emptyset$. KONEC
4. Najdi $x(t)$ pomocí (5) a t_1, t_2, t_3 pomocí (6),(7),(8) (za $\bar{x}$ dosadíme $\bar{y}$)
5. τ:=min(t_1, t_2, t_3), $\bar{y} := x(\tau)$, přejdi na **2.**

Důkaz správnosti kroku 2. a 3. je možné najít v článku [2].

2 Časová složitost ALGORITMU 1.2

V předchozí kapitole jsme popsali algoritmus na nalezení největšího prvku v množině omezené neklesajícími max-separabilními omezeními. Nyní se podíváme na časovou složitost tohoto algoritmů. V článku [2] je dokázáno, že časová složitost Algoritmu 1.2 je $q = O(m^4 n^4)$ a $C = O(m^4 n^4)$

2.3 Testování složitosti ALGORITMU 1.2

Nyní budeme sledovat dobu provádění ALGORITMU 1.2 v závislosti na počtu proměnných a počtu nerovností omezujících množinu řešení v praktickém provedení. Algoritmus byl testován na počítači s procesorem 1.5Ghz a pamětí 512MB. Výsledky jsou průměrovány vždy z 10 nezávislých náhodných zadání. Byla brána v úvahu pouze zadání, která měla řešení (tedy $M(\overline{x}) \neq \emptyset$).

První tabulka je pro n=80, tedy 80 proměnných, a mění se počet rovnic omezujících množinu řešení. Čas je uveden v milisekundách.

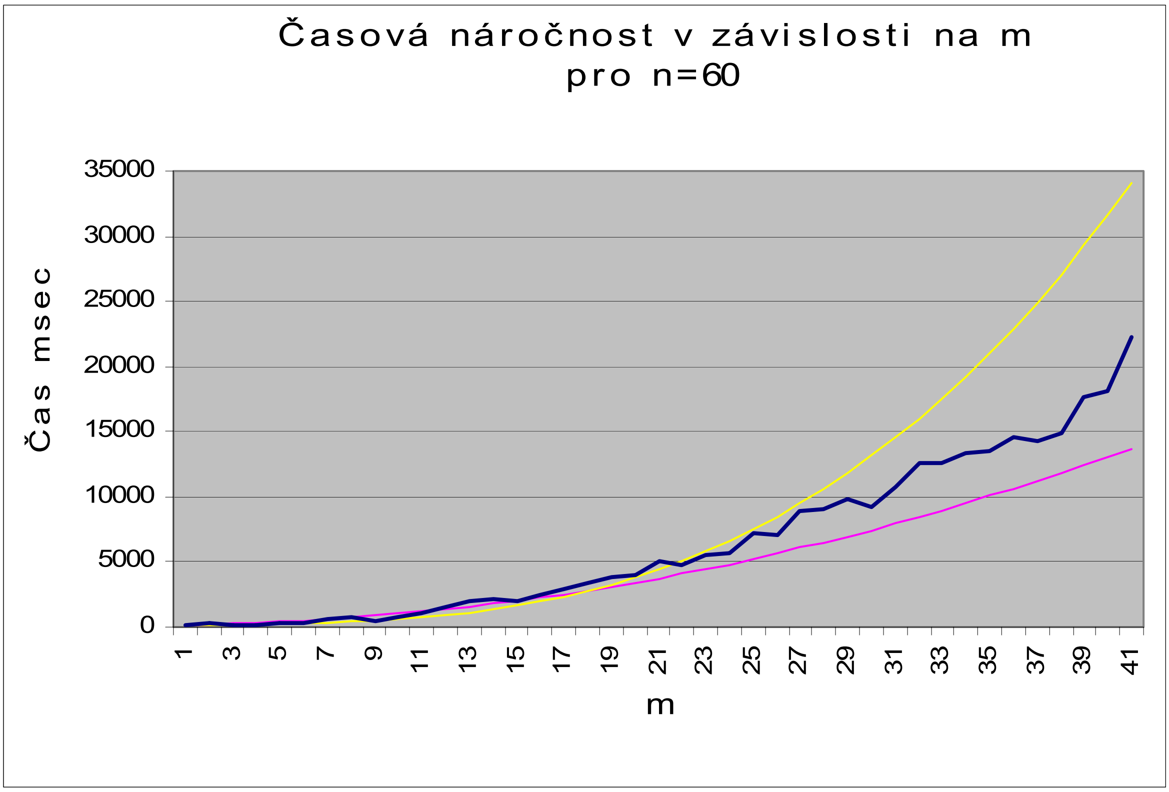
m	msec
1	206
2	253
3	145
4	177
5	269
6	308
7	547
8	711
9	519
10	794

m	msec
11	1086
12	1526
13	2061
14	2119
15	2009
16	2470
17	2908
18	3322
19	3855
20	4019

m	msec
21	5123
22	4686
23	5541
24	5610
25	7172
26	7044
27	8853
28	9088
29	9864
30	9231

m	msec
31	10736
32	12580
33	12561
34	13398
35	13508
36	14580
37	14242
38	14859
39	17621
40	18137

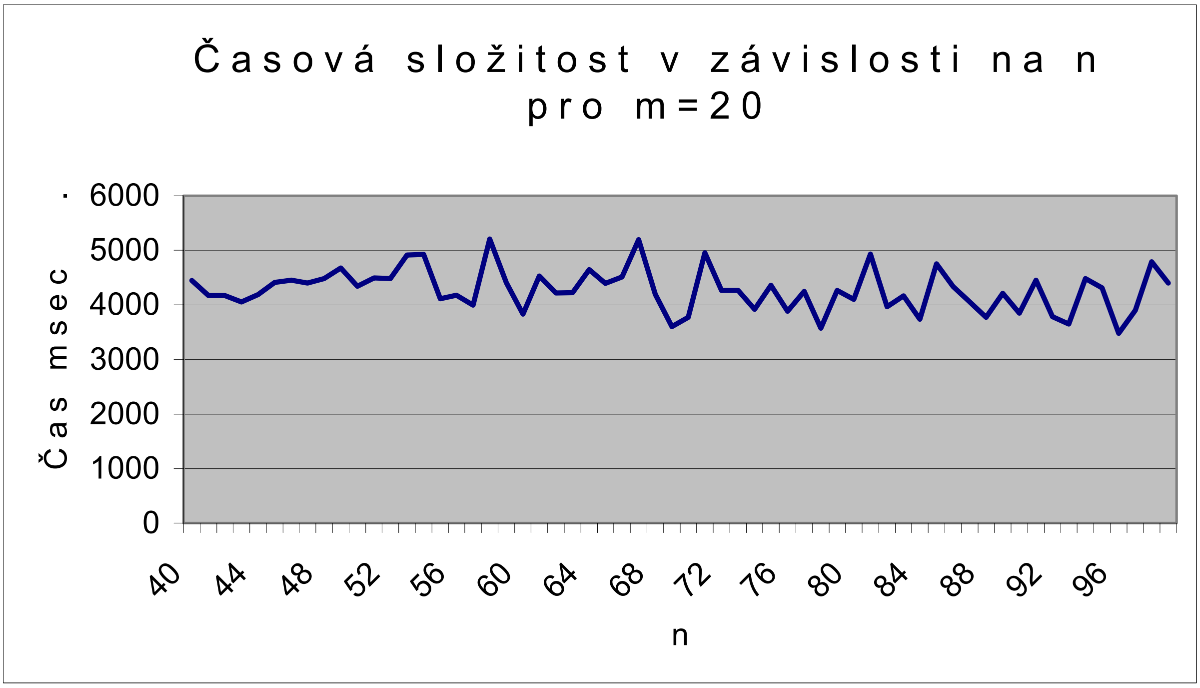
V následujícím grafu můžeme tyto hodnoty porovnat s průběhem funkcí m^2 a m^3:



Z grafu je vidět že pro reálná závislost časové složitosti na m je O(m^{α}), kde $2 \le \alpha \le 3$ **.**

Druhá tabulka s grafem jsou pro m=20, tedy v zadání je 20 rovnic určujících množinu přípustných řešení a mění se počet proměnných.

n	msec	n	msec	n	msec	n	msec
40	4450	**50**	4346	**60**	3831	**70**	3772
41	4172	**51**	4498	**61**	4528	**71**	4953
42	4174	**52**	4483	**62**	4219	**72**	4269
43	4053	**53**	4916	**63**	4223	**73**	4267
44	4192	**54**	4923	**64**	4650	**74**	3918
45	4415	**55**	4116	**65**	4395	**75**	4358
46	4453	**56**	4178	**66**	4513	**76**	3886
47	4401	**57**	3993	**67**	5192	**77**	4249
48	4487	**58**	5208	**68**	4197	**78**	3574
49	4675	**59**	4408	**69**	3602	**79**	4269



Na první pohled je vidět, že reálná časová závislost na *n* je konstantní.

3 Optimalizace s max-separabilní účelovou funkcí

3.1 Obecná úloha

Nyní se budeme zabývat řešením následující optimalizační úlohy.

Připomeňme si zavedené značení:

$N = \{1,...,n\}, S = \{1,...,m\}, R = (-\infty, \infty),$
$R^n = R \times ... \times R$ (n - krát), $x = (x_1,...,x_n) \in R^n$,
$a_{ij}, b_{ij} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N}$ jsou dané,

$$a_i(x) \equiv \max_{j \in N}(a_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$b_i(x) \equiv \max_{j \in N}(b_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$F_i(x) = \{j \in N \mid a_{ij} + x_j = a_i(x)\} \; \forall \mathrm{i} \in \mathrm{S},$$
$$G_i(x) = \{j \in N \mid b_{ij} + x_j = b_i(x)\} \; \forall \mathrm{i} \in \mathrm{S}$$

Proměnné x_j, kde $j \in F_i(x)$ resp. $j \in G_i(x)$ budeme nazývat aktivní proměnné v bodě x pro $a_i(x)$, resp. $b_i(x)$.

Opět uvažujeme následující systém (max,+)-lineárních rovnic:

$$a_i(x) = b_i(x) \; \forall \mathrm{i} \in \mathrm{S} \tag{1}$$

Množinu všech řešení systému (1) budeme značit *M*. Protože nyní budeme chtít minimalizovat nějakou účelovou funkci, musí být množina přípustných řešení oboustraně omezena. Tedy si dále definujme množinu $M(\underline{x}, \overline{x})$ pro $\forall x \in M$ následovně:

$$M(\underline{x}, \overline{x}) \equiv \{x \mid x \in M \;\&\; \underline{x} \le x \le \overline{x}\} \tag{2}$$

Naším úkolem bude minimalizovat max-separabilní účelovou funkci

$$f(x) = \max_{j \in J} f_j(x_j), \text{ kde } J \subseteq \{1,...,N\} \tag{3}$$

Tedy hledáme takové $x^{opt} \in M(\underline{x}, \overline{x})$, aby pro $\forall x \in M(\underline{x}, \overline{x})$ platilo $f(x) \ge f(x^{opt})$.

Označme si

$$J(x) = \{j \in J \mid f(x) = f_j(x_j)\} \tag{4}$$

Proměnné x_j, kde $j \in J(x)$ budeme nazývat aktivní proměnné v bodě x v účelové funkci *f*(*x*).

Funkce f_j jsou obecně libovolné funkce jedné proměnné, ale pro další práci s úlohou budeme od nich vyžadovat některé vlastnosti (např. monotónnost, unimodálnost).

Úlohu tedy můžeme napsat takto

$$\min_{x \in M(\underline{x},\overline{x})} f(x)$$

$$M(\underline{x},\overline{x}) \equiv \{x \mid x \in M \ \& \ \underline{x} \le x \le \overline{x}\}$$
$$M = \{x \mid a_i(x) = b_i(x) \ \forall \mathrm{i} \in \mathrm{S}\} \tag{5}$$

Pro řešení úlohy budeme potřebovat najít prvek $x \in M(\underline{x},\overline{x})$. V předchozí kapitole máme algoritmus na nalezení prvku $x \in M(\overline{x})$, a dokonce prvku největšího. Bude nám tedy stačit upravit tento algoritmus, aby pracoval s dolní mezí.

Nejdříve si musíme uvědomit, že $M(\underline{x},\overline{x}) \subseteq M(\overline{x})$. Tedy pokud ALGORITMUS 1.2.2 zjistí, že $M(\overline{x}) = \emptyset$, potom také $M(\underline{x},\overline{x}) = \emptyset$. Co v případě, kdy ALGORITMUS 1.2.2 najde nějaké řešení?

Pozn.: *Označme* $x < y$ *jestliže* $x \le y \,\&\, x \ne y$**.**

Lemma 3.1.1 Nechť $x^{\max}$ je prvek nalezený ALGORITMEM 1.2. Potom

(i) Nechť $x^{\max} \ge \underline{x}$. Potom $x^{\max} \in M(\underline{x},\overline{x})$ a dokonce je největším prvkem.

(ii) Nechť $x^{\max} \ngeq \underline{x}$. Potom $M(\underline{x},\overline{x}) = \emptyset$.

Důkaz:

(i)

$x^{\max} \ge \underline{x}$. Dle ALGORTIMU 1.2 je $x^{\max} \in M(\overline{x})$ a tedy $x^{\max} \le \overline{x}$. Z toho plyne, že $x^{\max} \in M(\underline{x},\overline{x})$.

To že je největší prvek dokážeme sporem. Nechť existuje $\tilde{x} \in M(\underline{x},\overline{x})$, $\tilde{x} > x^{\max}$. Tedy $\tilde{x} \in M(\overline{x})$ a $\tilde{x} > x^{\max}$, což je spor se správností ALGORITMU 1.2.

(ii)

Sporem. Nechť existuje $\tilde{x} \in M(\underline{x},\overline{x})$. Jelikož $M(\underline{x},\overline{x}) \subseteq M(\overline{x})$ máme $\tilde{x} \in M(\overline{x})$. Máme $x^{\max} \ngeq \underline{x}$ a $\underline{x} \le \tilde{x}$. Víme, že $x^{\max} \ne \tilde{x}$. Nechť tedy

$x^{\max} \le \tilde{x}$. Potom máme spor se správností ALGORITMU 1.3.

$x^{\max} \ge \tilde{x}$. Potom máme $x^{\max} \ge \tilde{x} \ge \underline{x}$ což je spor s předpokladem.

$x^{\max} \ngeq \tilde{x}$ a $x^{\max} \nleq \tilde{x}$. Protože $\tilde{x} \in M(\overline{x})$ a $x^{\max} \in M(\overline{x})$ máme spor, protože pokud je $M(\overline{x})$ neprázdná, má největší prvek a to je podle správnosti ALGORITMU 1.2 prvek $x^{\max}$.

□

Můžeme tedy nechat proběhnout celý algoritmus a na konci zkontrolovat, jestli $x^{\max} \ge \underline{x}$. To by ale mohlo být velmi neefektivní pro případ kdy $x^{\max} \ngeq \underline{x}$. Je tedy lepší kontrolovat tuto podmínku při každé iteraci algoritmu. Upravený algoritmus bude tedy vypadat následovně:

ALGORITMUS 3.1

1. $\bar{y} := \bar{x}$

2. Jestliže $\bar{y} \ngeq \underline{x}$ potom $M(\bar{x}) = \emptyset$. KONEC.

3. Jestliže $H(\bar{y}) = \emptyset$, potom $\bar{y}$ je největší prvek $M(\bar{x})$, KONEC.

4. Najdi $P(\bar{y})$ pomocí ALGORITMU 1.1 (za $\bar{x}$ dosadíme $\bar{y}$). Jestliže $P(\bar{y}) = N$, potom $M(\bar{x}) = \emptyset$. KONEC.

5. Najdi $x(t)$ pomocí (5) a t_1, t_2, t_3 pomocí (6),(7),(8) (za $\bar{x}$ dosadíme $\bar{y}$)

6. τ:=min(t_1, t_2, t_3), $\bar{y} := x(\tau)$, přejdi na **2.**

Díky tomu, že při běhu programu se $\bar{y}$ pouze snižuje, máme zajištěno že pokud je v libovolné iteraci algoritmu $\bar{y} < \underline{x}$, pro výsledek původního algoritmu $\tilde{x}$ by platilo $\tilde{x} \leq \bar{y} < \underline{x}$, a tedy dle lemma 3.1.1 $M(\underline{x}, \bar{x}) = \emptyset$.

Definice 3.1.1 $\hat{x}$ nazveme přípustným řešením, pokud $\hat{x} \in M(\underline{x}, \bar{x})$.
$\hat{x}$ nazveme největším přípustným řešením pokud $\hat{x}$ je přípustné řešení a pro $\forall x \in M(\underline{x}, \bar{x}),\ x \leq \hat{x}$.

Pozn.: *$\hat{x}$ je největší přípustné řešení právě tehdy, když ALGORITMUS 3.1 vrátí $\hat{x}$ jako řešení.*

Definice 3.1.2 $\tilde{x}$ je optimální řešení úlohy (5) právě tehdy, když $\tilde{x} \in M(\underline{x}, \bar{x})$ a $\forall x \in M(\underline{x}, \bar{x}), f(x) \geq f(\tilde{x})$.

Definice 3.1.3 $x' \in M(\underline{x}, \bar{x})$ je ε-optimální řešení úlohy (5) pro $\varepsilon > 0$, pokud $\tilde{x} \in M(\underline{x}, \bar{x})$ je libovolné optimální řešení a $|f(x') - f(\tilde{x})| < \varepsilon$.

3.2 Monotónní účelová funkce

Pro tuto kapitolu budeme předpokládat, že $f_j, j \in J$ jsou spojité, monotónní funkce. Budeme také požadovat, abychom pro každou $f_j, j \in J$ znali také f_j^{-1}. Úkolem bude najít optimální, nebo ε-optimální řešení úlohy (5).

Pozn.: *Požadavek inverzní funkce f_j^{-1} znamená, že funkce f_j musí být ryze monotónní. Pro nerostoucí, neklesající, popř. konstantní funkce inverzní funkce neexistuje. Nadále tedy předpokládejme pouze ryze monotónní, tedy rostoucí nebo klesající funkce.*

První krok algoritmu bude najít největší prvek množiny $M(\underline{x}, \overline{x})$. Nyní si jako $\overline{x}$ označíme tento největší prvek.

Pokud je $f_j, j \in J$ rostoucí (resp. klesající), potom $f_j(\overline{x}_j)$ je horní mez (resp. dolní mez) hodnot, kterou funkce f_j může nabývat na množině $M(\overline{x}, \underline{x})$. Stejně tak pokud je $f_j, j \in J$ rostoucí (resp. klesající), potom $f_j(\underline{x}_j)$ je dolní mez (resp. morní mez) hodnot, kterou funkce f_j může nabývat na množině $M(\overline{x}, \underline{x})$.ˇ

Definujme

$$\begin{aligned} \bar{f}(\underline{x}, \overline{x}) &= \max_{j \in J}(\max(f_j(\overline{x}_j), f_j(\underline{x}_j)) \\ \underline{f}(\underline{x}, \overline{x}) &= \max_{j \in J}(\min(f_j(\overline{x}_j), f_j(\underline{x}_j)) \end{aligned} \tag{6}$$

Z předchozího odstavce plyne

$$\underline{f}(\underline{x}, \overline{x}) \le f(x) \le \bar{f}(\underline{x}, \overline{x}), \text{pro } \forall x \in M(\underline{x}, \overline{x}) \tag{7}$$

Tedy $\underline{f}(\underline{x}, \overline{x})$ (resp. $\bar{f}(\underline{x}, \overline{x})$) je dolní mez (resp. horní mez) funkčních hodnot $f(x)$ pro $\forall x \in M(\underline{x}, \overline{x})$.

Máme tedy také omezeno optimální řešení $\tilde{x}$. Algoritmus bude používat půlení intervalu $< \underline{x}, \overline{x} >$.

Vezmeme tedy $\tilde{f} = \dfrac{\underline{f}(\underline{x}, \overline{x}) + \bar{f}(\underline{x}, \overline{x})}{2}$ a najdeme hodnoty

$$\begin{aligned} &\left.\begin{aligned} \underline{x}'_j &= \max(\underline{x}_j, f_j^{-1}(\tilde{f})) \\ \overline{x}'_j &= \overline{x}_j \end{aligned}\right\} \text{pro } f_j \text{ klesající} \\ &\left.\begin{aligned} \underline{x}'_j &= \underline{x}_j \\ \overline{x}'_j &= \min(\overline{x}_j, f_j^{-1}(\tilde{f})) \end{aligned}\right\} \text{pro } f_j \text{ rostoucí} \end{aligned} \tag{8}$$

Pokud $(M(\underline{x}', \overline{x}')) \ne \emptyset$, potom dosadíme $\underline{x} = \underline{x}', \overline{x} = \overline{x}''$, kde $\overline{x}''$ je největší prvek množiny $(M(\underline{x}', \overline{x}'))$, a znovu půlíme interval $< \overline{x}, \underline{x} >$.

Pokud $(M(\underline{x}', \overline{x}')) = \emptyset$, znamená to že hodnota optimálního řešení je větší nežli $\tilde{f}$.

Dosadíme tedy $\underline{f}(\underline{x},\overline{x}) = \widetilde{f}$ a znovu půlíme interval $< \overline{x}, \underline{x} >$.

Musíme ještě kontrolovat, jestli jsme nenašli již optimální řešení a zajistit konečnost algoritmu. Nechť tedy máme $\overline{x}, \underline{x}$ a $(M(\underline{x},\overline{x})) \neq \emptyset$. Dříve než-li začneme dělit interval $< \overline{x}, \underline{x} >$ spočítáme $J(\overline{x})$ dle (4) a pro každé $j \in J(\overline{x})$ platí, že jestliže f_j je klesající, potom $\overline{x}$ je optimální řešení.

Tedy pokud

$$\exists j \in J(\overline{x}), f_j(\overline{x}_j) \leq f_j(\underline{x}_j) \tag{9}$$

$\overline{x}$ je optimální.

Vzhledem k tomu, že ani tyto dvě podmínky nám nezaručí konečnost algoritmu, je třeba zadat konstantu ε, která nám bude určovat přesnost, s jakou budeme hledat řešení. Tedy jestliže $\underline{f}(\underline{x},\overline{x}) - \bar{f}(\underline{x},\overline{x}) < \varepsilon$ a $\widetilde{x}$ je optimální řešení, potom $f(\overline{x}) - f(\widetilde{x}) < \varepsilon$ a $\overline{x}$ je tedy ε-optimální řešení.

ALGORITMUS 3.2

1. Použij Algoritmus 3.1 na spočítání $\overline{y}$ - největšího prvku množiny $M(\underline{x},\overline{x})$, dosaď $\underline{y} = \underline{x}$. Pokud $\overline{y}$ neexistuje, potom $M(\underline{x},\overline{x}) = \emptyset$ a optimální řešení neexistuje. KONEC

2. Dle (4) urči $J(\overline{y})$. Pokud je podmínka (9) splněna (y všude dosaď za x), potom $\overline{y}$ je optimální řešení. KONEC.

3. Spočti $\bar{f}(\underline{y},\overline{y})$ a $\underline{f}(\underline{y},\overline{y})$.

4. Pokud $\bar{f}(\underline{y},\overline{y}) - \underline{f}(\underline{y},\overline{y}) < \varepsilon$, potom $\overline{y}$ je ε-optimální řešení. KONEC.

5. $\widetilde{f} = \dfrac{\underline{f}(\underline{y},\overline{y}) + \bar{f}(\underline{y},\overline{y})}{2}$. Urči $\overline{y}', \underline{y}'$ dle (8) (všude dosaď y za x).

6. Pomocí Algoritmu 3.1 spočti $\overline{y}''$ - největší prvek množiny $M(\underline{y}',\overline{y}')$.

7. Pokud $\overline{y}''$ neexistuje, tedy $M(\underline{y}',\overline{y}')$ je prázdná, dosaď $\underline{f}(\underline{y},\overline{y}) := \widetilde{f}$. Jdi na 4.

8. Dosaď $\underline{y} = \underline{y}', \overline{y} = \overline{y}''$. Jdi na 2.

3.3 Důkaz správnosti a konečnosti ALGORITMU 3.2

Správnost kroku 1. Je dána správností Algoritmu 3.1.

Ověřme krok 2. Nechť je splněna podmínka (9). Tedy máme $j' \in J(\bar{x}), f_{j'}(\bar{x}_{j'}) \le f_{j'}(\underline{x}_{j'})$, $\bar{x}_{j'} \ge \underline{x}_{j'}$. Z definice množiny $M(\underline{x},\bar{x})$ plyne, že pro $\forall x \in M(\underline{x},\bar{x})$, platí $x_{j'} \le \bar{x}_{j'}$. A díky monotónnosti funkcí f_j máme $f_{j'}(\bar{x}_{j'}) \le f_{j'}(x_{j'})$. Z definice $f(x)$ máme $f(x) \ge f_{j'}(x_{j'})$. Protože $j' \in J(\bar{x})$, platí $f_{j'}(\bar{x}_{j'}) = f(\bar{x})$. Tedy
$\forall x \in M(\underline{x},\bar{x}), f(\bar{x}) = f_{j'}(\bar{x}_{j'}) \le f_{j'}(x_{j'}) \le f(x)$. Tedy $\bar{x}$ je optimální řešení.

Nyní musíme ověřit správnost iteračního kroku.

Věta 3.3.1: Nechť $M(\underline{x},\bar{x})$ je neprázdná, $\underline{f}(\underline{x},\bar{x}), \bar{f}(\underline{x},\bar{x})$ je dolní (resp. horní) mez funkčních hodnot $f(x)$ pro $\forall x \in M(\underline{x},\bar{x})$. Dále nechť platí $\underline{f}(\underline{x},\bar{x}) \le \tilde{f} \le \bar{f}(\underline{x},\bar{x})$ a $\underline{x}',\bar{x}'$ jsou definovány jako v (8). Potom

(i) pro $\forall x \in M(\underline{x},\bar{x})$, $f(x) \le \tilde{f}$ platí $x \in M(\underline{x}',\bar{x}')$.

(ii) jestliže $M(\underline{x}',\bar{x}')$ je prázdná, potom pro $\forall x \in M(\underline{x},\bar{x})$, $f(x) > \tilde{f}$.

Důkaz:

(i) Nechť existuje $x \in M(\underline{x},\bar{x})$ $f(x) \le \tilde{f}$, takové, že $x \notin M(\underline{x}',\bar{x}')$. Tedy $\exists p$ takové, že buď $x_p > \bar{x}'_p$ nebo $x_p < \underline{x}'_p$.

Nechť tedy $x_p > \bar{x}'_p$. Za využití $x \in M(\underline{x},\bar{x})$ a (8) zjistíme, že $x_p > f_p^{-1}(\tilde{f})$ a f_p je rostoucí.

Tedy $f_p(x_p) > f_p(f_p^{-1}(\tilde{f}))$.

Naopak pokud $x_p < \underline{x}'_p$. Za využití $x \in M(\underline{x},\bar{x})$ a (8) zjistíme, že $x_p < f_p^{-1}(\tilde{f})$ a f_p je klesající. Tedy $f_p(x_p) > f_p(f_p^{-1}(\tilde{f}))$.

Z obou variant plyne že $f_p(x_p) > \tilde{f}$. Tím jsme získali $f(x) \ge f_p(x_p) > \tilde{f}$ což je spor s předpokladem.

(ii) triv.

Nechť $M(\underline{x}',\bar{x}')$ je prázdná a existuje $x \in M(\underline{x},\bar{x}), f(x) > \tilde{f}$. Potom z existence $x \in M(\underline{x},\bar{x}), f(x) > \tilde{f}$ a (i) platí $x \in M(\underline{x}',\bar{x}')$, což je spor s předpokladem.

□

Nyní můžeme tuto větu využít k důkazu iteračního kroku.

Jestliže v kroku 7. neexistuje $\bar{y}''$ (maximální prvek množiny $M(\underline{x}', \bar{x}')$), tedy $M(\underline{x}', \bar{x}')$ je prázdná a dle Věty 3.3.1 (ii) máme $\forall x \in M(\underline{x}, \bar{x})$, $f(x) > \tilde{f}$. Tedy $\tilde{f}$ je dolní mez funkčních hodnot $f(x)$ pro $\forall x \in M(\underline{x}, \bar{x})$ a tedy můžeme dosadit $\underline{f}(\underline{y}, \bar{y}) := \tilde{f}$ a znovu dělit.

Nechť tedy $M(\underline{x}', \bar{x}')$ je neprázdná.

Potom $\forall x \in M(\underline{x}', \bar{x}'), \forall x' \in M(\underline{x}, \bar{x}) \setminus M(\underline{x}', \bar{x}')$ $f(x) \le \tilde{f} < f(x')$. Tedy optimální řešení úlohy se nachází v $M(\underline{x}', \bar{x}')$ a stačí nadále procházet pouze tuto podmnožinu.

Je potřeba ještě zkontrolovat konečnost ALGORITMU 3.2. V každé iteraci algoritmu zmenšíme $< \underline{f}(\underline{x}, \bar{x}), \bar{f}(\underline{x}, \bar{x}) >$, tedy interval možných hodnot účelové funkce, na polovinu. V nejhorším případě skončíme v okamžiku, kdy $\bar{f}(\underline{x}, \bar{x}) - \underline{f}(\underline{x}, \bar{x}) < \varepsilon$. Nechť $D_0 = \bar{f}(\underline{x}, \bar{x}) - \underline{f}(\underline{x}, \bar{x})$ na počátku algoritmu. Po k krocích je velikost intervalu možných hodnot účelové funkce

$$D_k = \frac{D_{k-1}}{2} = \frac{D_0}{2^k}.$$

Hledáme takové k, pro které $D_k < \varepsilon \le D_{k-1}$. Po úpravách dostaneme

$$\log_2\left(\frac{D_0}{\varepsilon}\right) < k \le \log_2\left(\frac{D_0}{\varepsilon}\right) + 1$$

Tedy $k = \left\lceil \log_2\left(\frac{\bar{f}(\underline{x}, \bar{x}) - \underline{f}(\underline{x}, \bar{x})}{\varepsilon}\right) \right\rceil$

3.4 Unimodální účelová funkce

Nyní budeme předpokládat, že $f_j, j \in J$ jsou spojité, unimodální funkce na intervalu $< \underline{x}_j, \bar{x}_j >$. Tedy funkce f_j je buď ryze monotónní na $< \underline{x}_j, \bar{x}_j >$, nebo $\exists \dot{x} \in < \underline{x}_j, \bar{x}_j >$, že

A f_j je klesající na $< \underline{x}_j, \dot{x}_j >$ a rostoucí na $< \dot{x}_j, \bar{x}_j >$

B f_j je rostoucí na $< \underline{x}_j, \dot{x}_j >$ a klesající na $< x_j, \bar{x}_j >$

Ke klesajícím (resp. rostoucím) funkcím budeme přistupovat jako k typu A (resp. B) pro $\dot{x} = \infty$.

Budeme také požadovat, abychom pro každou $f_j, j \in J$ znali také $\dot{x}_j$ a inverzní funkce

f_j' f_j'', kde

$$f_j' = f_j^{-1}(x_j) \text{ pro } x_j \in <\underline{x}_j, \dot{x}_j>$$

$$f_j'' = f_j^{-1}(x_j) \text{ pro } x_j \in <\dot{x}_j, \overline{x}_j>.$$

Úkolem bude opět najít optimální, nebo ε-optimální řešení úlohy (5).

Pozn.: *Podmínka ryzí monotónnosti je, jako v předchozí úloze, kvůli existenci inverzních funkcí.*

Pozn.: *Předpoklad že známe bod, ve kterém je lokální minimum, resp. maximum funkce, může být v některých úlohách příliš omezující. Místo toho je možné požadovat diferencovatelnost funkce f_j. Vzhledem k tomu že funkce je monotónní na intervalu platí, že je-li funkce typu A, potom pokud derivace f_j v bodě x je záporná potom $x < \dot{x}_j$. Jinak $x \geq \dot{x}_j$. Pro typ B platí obdobně.*

Pokud není známo, které ho je typu, spočítáme derivaci funkce f_j v bodě $\underline{x}_j$. Pokud je derivace v $\underline{x}_j$ záporná, budeme s ní pracovat jako s typem A, jinak jako s typem B. Nemusí to odpovídat skutečnému typu funkce, ale pro práci na intervalu $<\underline{x}_j, \overline{x}_j>$ to dostačuje.

Použijeme ideu Algoritmu 3.2, pouze upravíme pro unimodální funkce. První krok algoritmu bude opět najít největší prvek množiny $M(\overline{x}, \underline{x})$. Tento prvek si označíme jako $\overline{x}$.

Pro všechny funkce $f_j, j \in J$ definujme

$$\begin{aligned} \bar{f}_j(\underline{x}, \overline{x}) &= \begin{cases} \max(f_j(\underline{x}_j), f_j(\overline{x}_j), f_j(\dot{x}_j)) \text{ pokud } \dot{x}_j \in (\underline{x}_j, \overline{x}_j) \\ \max(f_j(\underline{x}_j), f_j(\overline{x}_j)) \text{ jinak} \end{cases} \\ \underline{f}_j(\underline{x}, \overline{x}) &= \begin{cases} \min(f_j(\underline{x}_j), f_j(\overline{x}_j), f_j(\dot{x}_j)) \text{ pokud } \dot{x}_j \in (\underline{x}_j, \overline{x}_j) \\ \min(f_j(\underline{x}_j), f_j(\overline{x}_j)) \text{ jinak} \end{cases} \end{aligned} \qquad (10)$$

Tedy $\bar{f}_j(\underline{x}, \overline{x})$ (resp. $\underline{f}_j(\underline{x}, \overline{x})$) je horní mez (resp. dolní mez) funkčních hodnot funkce f_j na intervalu $<\underline{x}_j, \overline{x}_j>$.

Dále musíme definovat

$$\begin{aligned} \bar{f}(\underline{x}, \overline{x}) &= \max_{j \in J}(\bar{f}_j(\underline{x}, \overline{x})) \\ \underline{f}(\underline{x}, \overline{x}) &= \max_{j \in J}(\underline{f}_j(\underline{x}, \overline{x})) \end{aligned} \qquad (11)$$

Platí $\bar{f}(\underline{x},\bar{x}) = \max_{j\in J}(\bar{f}_j(\underline{x},\bar{x})) \geq f(x) \geq \max_{j\in J}(\underline{f}_j(\underline{x},\bar{x})) = \underline{f}(\underline{x},\bar{x})$ pro $\forall x \in M(\underline{x},\bar{x})$.

Dosadíme $\widetilde{f} = \dfrac{\bar{f}(\underline{x},\bar{x}) + \underline{f}(\underline{x},\bar{x})}{2}$ a opět budeme dále zkoumat podmnožinu přípustných řešení jejichž hodnota optimalizační funkce je menší nežli $\widetilde{f}$. Budeme muset projít více možností. Nechť f_j je

typ A

1, Nechť $\dot{x}_j \leq \underline{x}_j$, potom

$$\begin{aligned} \underline{x}'_j &= \underline{x}_j \\ \bar{x}'_j &= \min(\bar{x}_j, f''_j(\widetilde{f})) \end{aligned} \tag{12}$$

2, Nechť $\dot{x}_j \geq \bar{x}_j$, potom

$$\begin{aligned} \underline{x}'_j &= \max(\underline{x}_j, f'_j(\widetilde{f})) \\ \bar{x}'_j &= \bar{x}_j \end{aligned} \tag{13}$$

3, Nechť $\underline{x}_j \leq \dot{x}_j \leq \bar{x}_j$, potom

$$\begin{aligned} \underline{x}'_j &= \max(\underline{x}_j, f'_j(\widetilde{f})) \\ \bar{x}'_j &= \min(\bar{x}_j, f''_j(\widetilde{f})) \end{aligned} \tag{14}$$

typ B

1, Nechť $\dot{x}_j \leq \underline{x}_j$, potom

$$\begin{aligned} \underline{x}'_j &= \max(\underline{x}_j, f''_j(\widetilde{f})) \\ \bar{x}'_j &= \bar{x}_j \end{aligned} \tag{15}$$

2, Nechť $\dot{x}_j \geq \bar{x}_j$, potom

$$\begin{aligned} \underline{x}'_j &= \underline{x}_j \\ \bar{x}'_j &= \min(\bar{x}_j, f'_j(\widetilde{f})) \end{aligned} \tag{16}$$

3, Nechť $\underline{x}_j \leq \dot{x}_j \leq \bar{x}_j$, potom pokud $\widetilde{f} > f_j(\dot{x}_j)$ dosadíme

$$\begin{aligned} \underline{x}'_j &= \underline{x}_j \\ \bar{x}'_j &= \bar{x}_j \end{aligned}$$

jinak se interval $< \underline{x}_j, \bar{x}_j >$ rozdělí na dvě množiny:

$$\begin{aligned} \underline{x}'_j &= \underline{x}_j & \qquad \underline{x}''_j &= f''_j(\widetilde{f}) \\ \bar{x}'_j &= f'_j(\widetilde{f}) & \qquad \bar{x}''_j &= \bar{x}_j \end{aligned} \tag{17}$$

Pozn.: *Pro jednu z množin může platit $\underline{x}'_j \geq \overline{x}'_j$. V tomto případě je množina přípustných řešení odpovídající intervalu $<\underline{x}'_j, \overline{x}'_j>$ prázdná.*

Pozn.: *Díky tomu, že při této variantě vznikají z intervalu $<\underline{x}_j, \overline{x}_j>$ intervaly dva, může se množina $M(\overline{x}, \underline{x})$ rozdělit až na 2^n množin. Ovšem tato varianta nastane pouze v případě, že všechny funkce jsou typu B a u všech dojde k dělení typu 3.*

Ještě se musíme podívat, kdy už nemůžeme dále snižovat hodnotu $f(x)$. Pro lineární rovnici stačila podmínka (9) $\exists j \in J(\overline{x}), f_j(\overline{x}_j) \leq f_j(\underline{x}_j)$

Nyní musíme ale brát do úvahy to že funkce může být unimodální. Upravíme tedy podmínku na

$$\exists j \in J(\overline{x}), (f_j(\overline{x}_j) \leq f_j(\underline{x}_j)) \,\&\, ((\underline{x}_j \leq \dot{x}_j \leq \overline{x}_j) \Rightarrow f_j(\overline{x}_j) \leq f_j(\dot{x})) \qquad (18)$$

Abychom mohli napsat algoritmus, zavedeme si následující značení. Podúloha p je dána čtveřicí $\underline{x}^p, \overline{x}^p, \underline{f}^p, \overline{f}^p$, kde $M(\underline{x}^p, \overline{x}^p)$ je množina přípustných řešení dané podúlohy a $\overline{f}^p, \underline{f}^p$ je horní a dolní mez hodnot účelové funkce na $M(\underline{x}^p, \overline{x}^p)$. P je množina všech podúloh.

Nyní můžeme napsat Algoritmus.

ALGORITMUS 3.3

1. Použij Algoritmus 3.1 na spočítání $\overline{y}$ - největšího prvku množiny $M(\underline{x}, \overline{x})$, $\underline{y} = \underline{x}$. Pokud $\overline{y}$ neexistuje, potom $M(\underline{x}, \overline{x}) = \emptyset$ a optimální řešení neexistuje. KONEC

2. Vlož do P podúlohu $(\underline{y}, \overline{y}, \underline{f}(\underline{y}, \overline{y}), \overline{f}(\underline{y}, \overline{y}))$, kde $\underline{f}(\underline{y}, \overline{y}), \overline{f}(\underline{y}, \overline{y})$ spočti dle (11).

3. Vyřaď z P všechny podúlohy r, pro které existuje $q \in P$ takové, že $\underline{f}^r \geq \overline{f}^q$.

4. Z P vyjmi podúlohu $p = (\underline{y}^p, \overline{y}^p, \underline{f}^p, \overline{f}^p)$ s nejmenší možnou hodnotou $\underline{f}^p$.

5. Jestliže $\underline{f}^p = \overline{f}^p$, potom $\overline{y}^p$ je optimální řešení. KONEC.

   Pokud $\overline{f}^p - \underline{f}^p < \varepsilon$ Potom $\overline{y}$ je ε-optimální řešení. KONEC.

6. Dle (4) urči $J(\overline{y}^p)$. Pokud je podmínka (18) splněna (y všude dosaď za x), potom do P přidej $p = (\overline{y}^p, \overline{y}^p, f(\overline{y}^p), f(\overline{y}^p))$. Jdi na 3.

7. $\widetilde{f} = \dfrac{\underline{f}^p + \bar{f}^p}{2}$.

Urči podle (12)-(17) všechny možné intervaly $< \bar{y}', \underline{y}' >$ a pro každý:

a) podle Algoritmu 3.1 spočítej $\bar{y}''$ jako největší prvek $M(\underline{y}', \bar{y}')$.

b) pokud existuje spočti $\underline{f}(\underline{y}', \bar{y}''), \bar{f}(\underline{y}', \bar{y}'')$ a vlož do P podúlohu $(\bar{y}', \underline{y}'', \underline{f}(\underline{y}', \bar{y}''), \bar{f}(\underline{y}', \bar{y}''))$.

8. Pokud v kroku 7. nebyla do P vložena žádná podúloha, vlož do P podúlohu $(\underline{y}^p, \bar{y}^p, \widetilde{f}, \overline{f}^p)$

Jdi na 3.

3.5 Důkaz správnosti a konečnosti ALGORITMU 3.3

Definice: $\widetilde{M}(P) = \bigcup_{r \in P} M(\underline{x}^r, \bar{x}^r)$

Správnost kroku 3.

Nechť existuje $p, r \in P, \underline{f}^p \geq \bar{f}^r$. Jelikož $\underline{f}^p$ je dolní mez funkčních hodnot, máme $\underline{f}^p \leq f(x'), x' \in M(\underline{x}^p, \bar{x}^p)$. Podobně $\bar{f}^r$ také platí $\bar{f}^r \geq f(x''), x'' \in M(\underline{x}^r, \bar{x}^r)$. Tedy máme

$$f(x') \geq \underline{f}^p \geq \bar{f}^r \geq f(x''), x' \in M(\underline{x}^p, \bar{x}^p), x'' \in M(\underline{x}^r, \bar{x}^r)$$

Protože $M(\underline{x}^r, \bar{x}^r) \neq \emptyset$ (např. $\bar{x}^r \in M(\underline{x}^r, \bar{x}^r)$), můžeme množinu $M(\underline{x}^p, \bar{x}^p)$ vypustit z P.

Správnost kroku 5.

Nechť $\underline{f}^p = \bar{f}^p$. Protože jsme vybrali $p \in P$, tak aby $\underline{f}^p$ bylo co nejmenší, tedy pro $\forall r \in P$ platí, že $\forall x \in M(\underline{x}^r, \bar{x}^r), f(x) \geq \underline{f}^r \geq \underline{f}^p = \overline{f}^p = f(\bar{x}^p)$. Tedy $\bar{x}^p$ je optimální řešení na $\widetilde{M}(P)$.

Nechť $\bar{f}^p - \underline{f}^p < \varepsilon$. Jako v předchozím máme $\forall r \in P$ platí $\forall x \in M(\underline{x}^r, \bar{x}^r)$ $f(x) \geq \underline{f}^r \geq \underline{f}^p$. Nechť x^{opt} je optimální řešení na $\widetilde{M}(P)$. Potom $x^{opt} \in \widetilde{M}(P)$ a tedy $f(x^{opt}) \geq \underline{f}^p$. Tedy pro $\bar{x}^p$ máme $f(\bar{x}^p) - f(x^{opt}) \leq \bar{f}^p - \underline{f}^p < \varepsilon$. Tedy $\bar{x}^p$ je ε-optimální řešení na $\widetilde{M}(P)$.

Později dokážeme, že pokud x^{opt} je optimální řešení na $\widetilde{M}$ $\Rightarrow$ x^{opt} je optimální řešení na

$M(\underline{x},\overline{x})$.

Správnost kroku 6.

Pokud je (9) splněna, potom $\overline{x}^p$ je optimální na $M(\underline{x}^p,\overline{x}^p)$. Tedy můžeme přestat uvažovat všechna přípustná řešení na $M(\underline{x}^p,\overline{x}^p)$ kromě $\overline{x}^p$. Přidáme tedy do P podúlohu `$(\overline{y}^p,\overline{y}^p,f(\overline{y}^p),f(\overline{y}^p))$, která obsahuje pouze toto přípustné řešení.

Musíme zkontrolovat správnost tohoto indukčního kroku. Nechť P' je množina podúloh na začátku tohoto kroku. P'' je nově vzniklá množina podúloh.

Nechť je splněna podmínka (9) tedy $\exists j \in J(\overline{x}), f_j(\overline{x}_j) \le f_j(\underline{x}_j)$. Tedy $\overline{x}^p$ je optimální na $M(\underline{x}^p,\overline{x}^p)$ a pro $\forall x \in M(\underline{x}^p,\overline{x}^p), f(x) \ge f(\overline{x}^p)$.

Dále nechť x^{opt} je optimální řešení na $\widetilde{M}(P'')$, potom pro $\forall x \in \widetilde{M}(P'')$ platí, že $f(x) \ge f(x^{opt})$. Dokážeme, že x^{opt} je optimální řešení na $\widetilde{M}(P')$.

Máme, že pro $\forall x \in \widetilde{M}(P') \cap \widetilde{M}(P'') = \widetilde{M}(P'')$ platí $f(x) \ge f(x^{opt})$.

Nechť tedy $x \in \widetilde{M}(P') \setminus \widetilde{M}(P'')$. Potom $f(x) \ge f(\overline{x}^p) \ge f(x^{opt})$.

Tedy $\forall x \in \widetilde{M}(P')\ f(x) \ge f(x^{opt})$, $x^{opt} \in \widetilde{M}(P'')$. Jelikož $\widetilde{M}(P') \subseteq \widetilde{M}(P'')$, platí $x^{opt} \in \widetilde{M}(P')$. Tedy x^{opt} je optimální řešení na $\widetilde{M}(P')$.

Správnost kroku 7 a 8.

Věta 3.5.1: Nechť $M(\underline{x},\overline{x})$ je neprázdná, $\underline{f}(\underline{x},\overline{x}), \bar{f}(\underline{x},\overline{x})$ je dolní (resp. horní) mez funkčních hodnot $f(x)$ pro $\forall x \in M(\underline{x},\overline{x})$. Dále nechť platí $\underline{f}(\underline{x},\overline{x}) \le \widetilde{f} \le \bar{f}(\underline{x},\overline{x})$ a $\underline{x}^i,\overline{x}^i, i \in 1,...,k$ jsou všechny $\underline{x}',\overline{x}'$ vzniklé použitím (12)-(17). Potom

(i) pro $\forall x \in M(\underline{x},\overline{x})$, $f(x) \le \widetilde{f}$ existuje $i \in 1,..,k$ pro které $x \in M(\underline{x}^i,\overline{x}^i)$.

(ii) jestliže $\forall i \in 1,...,k$ platí, že $M(\underline{x}^i,\overline{x}^i)$ je prázdná, potom pro $\forall x \in M(\underline{x},\overline{x})$, $f(x) > \widetilde{f}$.

Důkaz:

(i) Dokážeme sporem. Nechť existuje $x \in M(\underline{x},\overline{x})\ f(x) \le \widetilde{f}$, takový, že pro $\forall i \in 1,...,k$ platí $x \notin M(\underline{x}^i,\overline{x}^i)$. Tedy pro všechna *i* existuje *p* takový, že buď $x_p > \overline{x}^i{}_p$ nebo $x_p < \underline{x}^i{}_p$.

Vezměme libovolné pevné $i \in 1,...,k$ a k němu odpovídající *p*.

Nechť hodnoty $\underline{x}^i{}_p, \overline{x}^i{}_p$ vznikly například použitím (12).

Potom $\underline{x}^i{}_p = \underline{x}_p$ a $\overline{x}^i{}_p = \min(\overline{x}_p, f_p''(\widetilde{f}))$. Protože $\underline{x}^i{}_p = \underline{x}_p$, musí platit $\overline{x}_p \ge x_p > \overline{x}^i{}_p$. Tedy $\overline{x}^i{}_p = f_p''(\widetilde{f})$. Funkce f_p je na intervalu $< \underline{x}^i{}_p, \overline{x}^i{}_p >$ rostoucí. Tedy

$f(x) = f_p(x_p) > f_p(\overline{x}^i{}_p) = \widetilde{f}$. Tedy spor s předpokladem

Obdobně dokážeme pro (13)-(16)

Tedy pro všechna $i \in 1,...,k$ a k nim odpovídající p taková, že buď $x_p > \overline{x}^i{}_p$ nebo $x_p < \underline{x}^i{}_p$, platí, že hodnoty $\underline{x}^i{}_p, \overline{x}^i{}_p$ vznikly použitím (17).

Opět vezmeme libovolné pevné i a k němu odpovídající p. Víme, že hodnoty $\underline{x}^i{}_p, \overline{x}^i{}_p$ vznikly použitím (17).

Pokud by $\widetilde{f} > f_j(\dot{x}_j)$, tedy $\underline{x}^i{}_p = \underline{x}_p$ a $\overline{x}^i{}_p = \overline{x}_p$, potom by platilo $x_p > \overline{x}^i{}_p = \overline{x}_p$ nebo $x_p < \underline{x}^i{}_p = \underline{x}_p$, tedy $x \notin M(\underline{x}, \overline{x})$ což je spor. Museli jsme tedy použít

$$\begin{array}{l} \underline{x}'_p = \underline{x}_p \\ \overline{x}'_p = f'_p(\widetilde{f}) \end{array} \quad \text{nebo} \quad \begin{array}{l} \underline{x}''_p = f'_p(\widetilde{f}) \\ \overline{x}''_p = \overline{x}_p \end{array}.$$

Nechť tedy BÚNO $\begin{array}{l} \underline{x}^i{}_p = \underline{x}_p \\ \overline{x}^i{}_p = f'_p(\widetilde{f}) \end{array}$ a tedy existuje j takové, že $\begin{array}{l} \underline{x}^j{}_p = f'_p(\widetilde{f}) \\ \overline{x}^j{}_p = \overline{x}_p \end{array}$.

Protože $x \in M(\underline{x}, \overline{x})$, tedy $\underline{x}_p \le x_p \le \overline{x}_p$ a přitom $x \notin <\underline{x}^i{}_p, \overline{x}^i{}_p> \cup <\underline{x}^j{}_p, \overline{x}^j{}_p>$, máme $\overline{x}^i{}_p < x_p < \underline{x}^j{}_p$.

Nyní tedy jestliže $x_p < \dot{x}_p$, potom protože f_p je rostoucí na $<\overline{x}^i{}_p, \dot{x}_p>$, platí že $\widetilde{f} = f_p(\overline{x}^i{}_p) < f_p(x_p) = f(x)$, což je spor

Naopak pokud $x_p > \dot{x}_p$, potom jelikož f_p je klesající na $<\dot{x}_p, \underline{x}^j{}_p>$, platí $\widetilde{f} = f_p(\underline{x}^j{}_p) < f_p(x_p) = f(x)$, což je opět spor.

(ii)

Nechť $M(\underline{x}', \overline{x}')$ je prázdná pro $\forall i \in 1,...,k$ a existuje $x \in M(\underline{x}, \overline{x}), f(x) > \widetilde{f}$. Potom z existence $x \in M(\underline{x}, \overline{x}), f(x) > \widetilde{f}$ a (i) plyne, že existuje $i \in 1,..,k$ pro které $x \in M(\underline{x}^i, \overline{x}^i)$, což je spor s předpokladem.

□

Tím máme dokázanou správnost algoritmu 3.3. Nyní je potřeba zajistit jeho konečnost.

Jak jsme dokázali v kapitole 3.3, pokud bychom množinu $M(\underline{x}, \overline{x})$ nerozdělili na více ale pokračovali vždy s maximálně jednou její podmnožinou, provedl by algoritmus maximálně

$$k = \left\lceil \log_2\left(\frac{\bar{f}(\underline{x},\bar{x}) - \underline{f}(\underline{x},\bar{x})}{\varepsilon}\right) \right\rceil$$

kroků, než by nalezl řešení. V našem případě ale může dojít k rozdělení $M(\underline{x},\bar{x})$ na více podmnožin a v každé může být potřeba najít optimální řešení. Je třeba si uvědomit, že dle každé proměnné x_p můžeme dělit pouze jednou. Nejhorší složitost tedy dostaneme v případě, kdy všechny funkce jsou typu B a podle všech dělíme hned v první iteraci. Tedy získáme $2^{|J|}$ úloh a pro každou provedeme $\left\lceil \log_2\left(\frac{\bar{f}(\underline{x},\bar{x}) - \underline{f}(\underline{x},\bar{x})}{\varepsilon}\right) \right\rceil$ kroků. Tedy celkem provedeme maximálně $2^{|J|}\left\lceil \log_2\left(\frac{\bar{f}(\underline{x},\bar{x}) - \underline{f}(\underline{x},\bar{x})}{\varepsilon}\right) \right\rceil$ kroků.

Toto je velmi hrubý odhad, protože předpokládáme, že všechny funkce jsou typu B, u všech došlo k dělení, u všech v první iteraci a všechny podmnožiny jsou neprázdné. Vidíme tedy mnoho podmínek které musí být pro tuto složitost splněny. Navíc v kroku 4. vezmeme úlohu která má nejmenší $\underline{f}^p$ a neprocházíme už úlohy u kterých víme že optimální řešení neexistuje. Tedy průměrná výsledná složitost bude mnohem menší.

3.6 Lineární účelová funkce

Nyní budeme předpokládat, že $f_j, j \in J$ jsou lineární funkce. Tedy

$$f_j = k_j x_j + d_j \tag{18}$$

Algoritmus tentokrát v každém kroku sníží hodnotu $\bar{x} \in M(\underline{x},\bar{x})$ na $\bar{x}'$, tak aby stále $\bar{x} \in M(\underline{x},\bar{x})$ a při tom snížil hodnotu $f(x)$. Na konci algoritmu budeme mít optimální řešení úlohy.

Již jsme definovali $J(x) = \{j \in J \mid f(x) = f_j(x_j)\}$. Proměnné x_j, kde $j \in J(x)$ budeme nazývat aktivní proměnné v bodě x v účelové funkci *f*(*x*).

Nechť tedy máme $\bar{x}$, největší prvek množiny $M(\underline{x},\bar{x})$, nalezený algoritmem 3.1. Abychom snížili hodnotu $f(\bar{x})$, musíme snížit $f_j(\bar{x}_j)$ pro $j \in J(\bar{x})$. Jestliže tedy pro nějaké $j \in J(\bar{x})$ platí, že f_j je nerostoucí ($k_j \le 0$), potom $\bar{x}$ je optimální řešení úlohy.

Předpokládejme proto, že pro všechna $j \in J(\bar{x})$ platí $k_j > 0$.

Musíme snížit všechna $x_j, j \in J(\bar{x})$. Tedy

ALGORITMUS 3.4

1. $P(\bar{x}) := J(\bar{x})$

2. $E_1 := \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x})\ \&\ G_i(\bar{x}) \not\subset P(\bar{x})\}$,
$E_2 := \{i \in S \mid F_i(\bar{x}) \not\subset P(\bar{x})\ \&\ G_i(\bar{x}) \subseteq P(\bar{x})\}$;

3. Jestliže $E_1 \cup E_2 = \emptyset$, potom $P(\bar{x})$ je množina proměnných, které je potřeba snížit, KONEC.

4. $P(\bar{x}) := P(\bar{x}) \cup \bigcup_{i \in E_1} (G_i(\bar{x}) \setminus P(\bar{x})) \cup \bigcup_{i \in E_2} (F_i(\bar{x}) \setminus P(\bar{x}))$, přejdi na **2.**;

Nyní popíšeme proces snižování proměnných $x_j, j \in P(\bar{x})$.

Definujme opět $x(t) = (x_1(t),\ldots,x_n(t))$ pro $t \geq 0$ následovně

$$x_j(t) \equiv \bar{x}_j - t \text{ pokud } j \in P(\bar{x}),$$
$$x_j(t) \equiv \bar{x}_j \quad \text{jinak}$$

Jestliže zvýšíme parametr *t*, proměnné $x_j, j \in P(\bar{x})$ budou zmenšeny. (Tj. $x_j(0) = \bar{x}_j, x_j(t) < \bar{x}_j$ pro všechny t>0 a $j \in P(\bar{x})$).

Můžeme si všimnout, že pokud snížíme *t* na hodnotu ε>0, kde ε je dostatečně malé, potom pro všechna $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x})$ zůstává

$$F_i(x(t)) = F_i(\bar{x}),\ G_i(x(t)) = G_i(\bar{x}),$$
$$F_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow a_i(x(t)) = a_i(\bar{x}) - t,\quad G_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow b_i(x(t)) = b_i(\bar{x}) - t$$
$$\text{Také platí } \forall j \in J(\bar{x}), \forall l \in J \setminus J(\bar{x}), f_j(x_j(t)) > f_l(x_l(t))$$

Budeme zvyšovat *t*, dokud nenastane jedna z následujících podmínek.

(1) $F_i(x(t)) \neq F_i(\bar{x})$, tj. $a_i(x(t)) = \alpha_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})} (a_{ij} + \bar{x}_j)$ pro nějaké $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x})$;

(2) $G_i(x(t)) \neq G_i(\bar{x})$, tj. $b_i(x(t)) = \beta_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})} (b_{ij} + \bar{x}_j)$ pro nějaké $i \in S$ takové, že $G_i(\bar{x}) \subseteq P(\bar{x})$;

(3) $x_j(t) = \underline{x}_j$ pro nějaké $j \in P(\bar{x})$.

(4) $f_j(x_j(t)) = f_l(x_l(t))$, pro nějaké $j \in J(\bar{x}), l \in J \cap P(\bar{x}) \setminus J(\bar{x})$
nebo $f_j(x_j(t)) = f_l(\bar{x}_l)$, pro nějaké $j \in J(\bar{x}), l \in J \setminus P(\bar{x})$

Budeme definovat proměnné t_1, t_2, t_3, t_4 jako hodnoty, ve kterých nastaly předchozí podmínky.

Pokud $P(\bar{x}) = N$, potom $\alpha_i(\bar{x}), \beta_i(\bar{x})$ jsou rovny +∞. V tomto případě položíme $t_1 = t_2 = \infty$. Jinak můžeme z předchozích podmínek dostat následující definice t_1, t_2, t_3:

$$t_1 = \min_{i \in L_1}(a_i(\bar{x}) - \alpha_i(\bar{x})), \tag{19}$$

kde $L_1 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x})\};$

$$t_2 = \min_{i \in L_2}(b_i(\bar{x}) - \beta_i(\bar{x})), \tag{20}$$

kde $L_2 \equiv \{i \in S \mid G_i(\bar{x}) \subseteq P(\bar{x})\};$

$$t_3 = \min_{j \in P(\bar{x})}(\bar{x}_j - \underline{x}_j), \tag{21}$$

Nyní musíme ještě určit t_4. Uvažujme nejprve prvou variantu

$f_j(x_j(t)) = f_l(x_l(t))$, pro nějaké $j \in J(\bar{x}), l \in J \cap P(\bar{x}) \setminus J(\bar{x})$

$$k_j(x_j - t) + d_j = k_l(x_l - t) + d_l$$

tedy $$-k_j t + k_l t = d_l - d_j - k_j x_j + k_l x_l$$

$$t = \frac{d_l - d_j - k_j x_j + k_l x_l}{k_l - k_j}$$

pro druhou variantu obdobně

$f_j(x_j(t)) = f_l(\bar{x}_l)$, pro nějaké $j \in J(\bar{x}), l \in J \setminus P(\bar{x})$

$$k_j(x_j - t) + d_j = k_l x_l + d_l$$

tedy $$-k_j t = k_l x_l + d_l - k_j x_j - d_j$$

$$t = \frac{-d_l + d_j - k_l x_l + k_j x_j}{k_j}$$

tím získáváme

$$t_4 = \min_{l \in J \setminus J(\bar{x})}(r_{jl}) \tag{22}$$

kde $j \in J(\bar{x})$ a

$$r_{jl} = \begin{cases} \dfrac{d_l - d_j + k_l x_l - k_j x_j}{k_l - k_j} & \text{pro } l \in P(\overline{x}) \\ \dfrac{-d_l + d_j - k_l x_l + k_j x_j}{k_j} & \text{pro } l \notin P(\overline{x}) \end{cases}, \text{ kde } \; k_j = \max_{l \in J(\overline{x})} k_l .$$

Dále budeme uvažovat pouze r_{jl} větší nežli nula

Nyní definujme $\tau \equiv \min(t_1, t_2, t_3, t_4)$.

Kdybychom v tuto chvíli skončili a v každém kroku snižovali všechna $x_j, j \in P(\overline{x})$, nedostali bychom se k optimálnímu řešení. Proč tomu tak je uvidíme na následujícím příkladu:

Příklad:

$A = (0,-10,-10) \quad B = (-10,-10,0)$

$\overline{x} = (10,10,10) \qquad f_1(x_1) = x_1$

$\underline{x} = (0,0,0) \qquad f_2(x_2) = 2x_2 - 10$

$f_3(x_3) = -x_3 + 10$

Řešení:

$\overline{x}$	$J(\overline{x})$	$P(\overline{x})$	τ
(10,10,10)	1,2	1,2,3	$\frac{10}{3}$
$(\frac{20}{3},\frac{20}{3},\frac{20}{3})$	1	1,3	$\frac{10}{6}$
$(5,\frac{20}{3},5)$	1,3		

Dalším snižováním bychom zvětšili hodnotu účelové funkce – můžeme skončit. Řešením tedy je $x^{opt} = (5,\frac{20}{3},5)$.

Nyní uvažujme bod $\widetilde{x} = (5,\frac{15}{2},5)$. Platí že $\widetilde{x} > x^{opt}$ a přitom $f(\widetilde{x}) = f(x^{opt})$. V prvním kroku řešení jsme snížili x_2 příliš a tím jsme přeskočili řešení $\widetilde{x} = (5,\frac{15}{2},5)$. V tomto případě jsme dostali řešení se stejnou funkční hodnotou, ale v jiných úlohách můžeme minout jediné optimální řešení.

Abychom zajistili, že opravdu najdeme optimální řešení, nebudeme snižovat všechna $x_j, j \in P(\bar{x})$, ale snížíme nejprve takové $x_{j'}, j' \in J(\bar{x})$, pro které platí $\forall j \in J(\bar{x}): k_j \geq k_{j'}$ a všechny $x_j, j \in N$ potřebné k zachování rovností. Použijeme algoritmus 3.4 s tím rozdílem, že v prvním kroku dosadíme $P(\bar{x}) := \{j'\}$, kde j' je libovolné z $\forall j \in J(\bar{x}): k_j \geq k_{j'}$. Výslednou množinu označíme $P'(\bar{x})$.

Definujme $x[t] = (x_1[t], \ldots, x_n[t])$ pro $t \geq 0$ následovně

$$x_j[t] \equiv \bar{x}_j - t \text{ pokud } j \in P'(\bar{x}),$$
$$x_j[t] \equiv \bar{x}_j \quad \text{jinak}$$

Ještě musíme dořešit o jakou hodnotu můžeme $x_j, j \in P'(\bar{x})$ snížit. Prozkoumejme, jak se změní $\tau \equiv \min(t_1, t_2, t_3, t_4)$ pokud místo $P(\bar{x})$ použijeme $P'(\bar{x})$.

1. pro (19) dokážeme, že $t_1 \leq t_1'$

$$t_1 = \min_{i \in L_1}(a_i(\bar{x}) - \alpha_i(\bar{x})), \text{ kde } L_1 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x})\};$$
$$t_1' = \min_{i \in L_1'}(a_i(\bar{x}) - \alpha_i'(\bar{x})), \text{ kde } L_1' \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P'(\bar{x})\};$$

- $i \in L_1' \Leftrightarrow F_i(\bar{x}) \subseteq P'(\bar{x}) \Rightarrow F_i(\bar{x}) \subseteq P(\bar{x}) \Leftrightarrow i \in L_1$ tedy $L_1' \subseteq L_1$ tím dostáváme $\min_{i \in L_1}(a_i(\bar{x}) - \alpha_i'(\bar{x})) \leq \min_{i \in L_1'}(a_i(\bar{x}) - \alpha_i'(\bar{x}))$
- $\alpha_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})}(a_{ij} + \bar{x}), \alpha_i'(\bar{x}) \equiv \max_{j \in N \setminus P'(\bar{x})}(a_{ij} + \bar{x})$.

$$P \subseteq P'$$
$$N \setminus P \supseteq N \setminus P'$$
$$\alpha_i(x) \geq \alpha'(x)_i$$

- $\min_{i \in L_1}(a_i(\bar{x}) - \alpha_i(\bar{x})) \leq \min_{i \in L_1}(a_i(\bar{x}) - \alpha_i'(\bar{x})) \leq \min_{i \in L_1'}(a_i(\bar{x}) - \alpha_i'(\bar{x}))$ a tedy $t_1 \leq t_1'$.

2. pro (20) stejným způsobem dostaneme $t_2 \leq t_2'$

3. pro (21) snadno ukážeme se $t_3 \leq t_3'$

$$t_3 = \min_{j \in P(\bar{x})}(\bar{x}_j - \underline{x}_j)$$
$$t_3' = \min_{j \in P'(\bar{x})}(\bar{x}_j - \underline{x}_j)$$
$$P'(\bar{x}) \subseteq P(\bar{x}) \Rightarrow t_3' \geq t_3$$

4. pro (22) toto neplatí. Musíme tedy ještě spočítat t_4'.

Nadefinujme tedy $\tau \equiv \min(t_1, t_2, t_3, t_4, t'_4)$

Pozn.: *Tím tedy pro libovolné* $t \in (0,\tau)$ *všechna* $i \in S$ *takové, že* $F_i(\bar{x}) \subseteq P'(\bar{x})$ *zůstává* $F_i(x[t]) = F_i(\bar{x}),\ G_i(x[t]) = G_i(\bar{x}),$ *a přitom* $\forall j \in J(\bar{x}), \forall k \in J \setminus J(\bar{x}), f_j(x_j[t]) > f_k(x_k[t])$

Jako novou horní mez si tedy označíme $x[\tau]$. Nyní již můžeme napsat ALGORITMUS 3.4 řešící celou naši úlohu.

ALGORITMUS 3.5

1. Pomocí algoritmu 3.1 najdeme $\bar{y}$ - největší prvek množiny $M(\underline{x},\bar{x})$

2. Jestliže $\exists j \in J(\bar{y})$, $k_j \le 0$, potom $\bar{y}$ je optimální řešení . KONEC

3. Najdi $P(\bar{y})$ a $P'(\bar{y})$ pomocí ALGORITMU 3.4 (za $\bar{x}$ dosadíme $\bar{y}$).

Jestliže $\exists j \in P(\bar{y})$, kde $\bar{y}_j = \underline{x}_j$, potom $\bar{y}$ je optimální řešení . KONEC

4. Urči t_1, t_2, t_3, t_4 z $P(\bar{y})$ pomocí (19)-(22) a t'_4 z $P'(\bar{y})$ pomocí (22) (za $\bar{x}$ dosadíme $\bar{y}$)

5. $\tau := \min(t_1, t_2, t_3, t_4, t'_4)$, $\bar{y} := x[\tau]$, přejdi na **2.**

3.7 Důkaz správnosti ALGORITMU 3.5

Věta 3.7.1: Nechť $\bar{y}$ je současná horní mez v ALGORITMU 3.5, $\bar{y} \in M(\underline{x},\bar{x})$ není optimální, $t \in (0,\tau)$, $\tilde{x} \le \bar{y}$ a $\tilde{x} \not\le x[t]$. Potom $\tilde{x} \notin M(\underline{x},\bar{x})$ nebo existuje $x' \le x[t]$, $x' \in M(\underline{x},\bar{x})$ a $f(x') < f(\tilde{x})$.

Důkaz:

Nejprve předpokládejme, že $\tilde{x}_{j'} > x_{j'}(t)$, kde $j' \in J(\bar{y})$ je vybrán v prvním kroku algoritmu 3.4. Protože $\bar{y}$ není optimální platí $\forall j \in J(\bar{y})$, f_j je rostoucí. Tedy také $f_{j'}$ je rostoucí. Protože $\tilde{x}_{j'} > x_{j'}(t)$, máme $f(\tilde{x}) \ge f_{j'}(\tilde{x}_{j'}) \ge f_{j'}(x_{j'}[t])$. Dosaďme $x' = x(t)$. Protože $P'(x) \subseteq P(x)$, platí $x(t) \le x[t]$.

Víme, že platí $\forall k \in J \setminus J(\bar{x})$ $f_{j'}(x_{j'}(t)) > f_k(x_k(t))$. Dále pro $\forall j \in J(\bar{x})$ platí $f_j(\bar{x}_j) = f_{j'}(\bar{x}_{j'})$. Protože $\forall j \in J(\bar{x}) : k_j \ge k_{j'}$, máme $f_j(x_j(t)) \le f_{j'}(x_{j'}(t))$ pro $\forall j \in J(\bar{x})$. A tedy také $\forall k \in J, f_k(x_k(t)) \le f_{j'}(x_{j'}(t))$. Tím jsme dostali $f(x(t)) = f_{j'}(x_{j'}(t))$.

Protože $j' \in P(\bar{x})$ i $j' \in P'(\bar{x})$ platí $x_{j'}[t] = x_{j'}(t)$. Takže nakonec

$f(\tilde{x}) \geq f_{j'}(\tilde{x}_{j'}) \geq f_{j'}(x_{j'}[t]) = f_{j'}(x_{j'}(t)) = f(x(t))$

Tím jsme tuto část dokázali.

Nechť tedy $\tilde{x}_{j'} \leq x_{j'}(t)$. Máme tedy

$\tilde{x}_r > x_r(t)$ pro $r \in N_1 \subseteq P(\bar{y}) \setminus \{j'\}$

$\tilde{x}_r \leq x_r(t)$ jinak

Nechť $p \in N_1$ je takový index, který se jako první z N_1 dostal v ALGORITMU 3.5 do $P(\bar{y})$. Tedy musela nastat situace $\exists i_0 \in S$

$$\begin{matrix} p \notin F_{i_0}(\bar{y}) \text{ a } p \in G_{i_0}(\bar{y}) \\ p \in F_{i_0}(\bar{y}) \text{ a } p \notin G_{i_0}(\bar{y}) \end{matrix} \quad \text{nebo}$$

Bez újmy na obecnosti nechť nastala první varianta. Jelikož $F_{i_0}(x(t)) = F_{i_0}(\bar{y})$ a $G_{i_0}(x(t)) = G_{i_0}(\bar{y})$, tedy $p \notin F_{i_0}(x(t))$ a $p \in G_{i_0}(x(t))$. Jelikož p je první index z N_1 v $P(\bar{y})$ je jasné, že $j \notin F_{i_0}(x(t)), \forall j \in N_1$. A tedy máme $a_{i_0 j} + \tilde{x}_j < a_{i_9}(x(t))$ pro $\forall j \in N_1$.

Jelikož pro všechny $\tilde{x}_j, j \in N \setminus N_1$ $\tilde{x}_j \leq x_j(t)$ máme

$$a_{i_0}(\tilde{x}) \leq a_{i_0}(x(t)) = b_i(x(t)) = b_{i_0 p} + x_p(t) < b_{i_0 p} + \tilde{x}_p \leq b_{i_0}(\tilde{x}).$$

Tedy $a_{i_0}(\tilde{x}) < b_{i_0}(\tilde{x})$, což znamená, že $\tilde{x} \notin M(\underline{x}, \bar{y})$

□

3.8 Konečnost algoritmu 3.5

V předchozí kapitole jsme provedli důkaz správnosti algoritmu 3.5. Důkaz konečnosti se provést nepodaří, protože algoritmus v některých případech konečný není. Ukážeme si to na následujícím příkladu.

Příklad:

$A = (0,5,0)$ $\quad B = (0,0,5)$

$\bar{x} = (10,10,10)$ $\quad f_1(x_1) = 2x_1$

$\underline{x} = (0,0,0)$ $\quad f_2(x_2) = 4x_2 - 20$

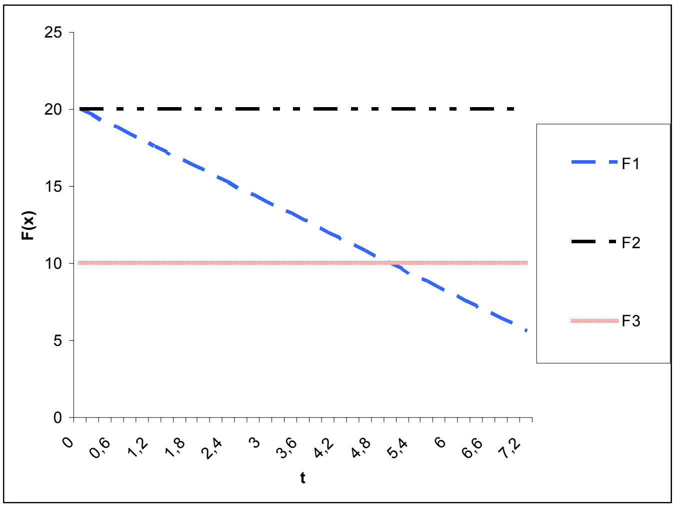
$f_1(x_1) = x_3$

Řešení:

1, V bodě (10,10,10) máme

$J(\bar{x}) = \{1,2\}$, $P(\bar{x}) = \{1,2,3\}$,

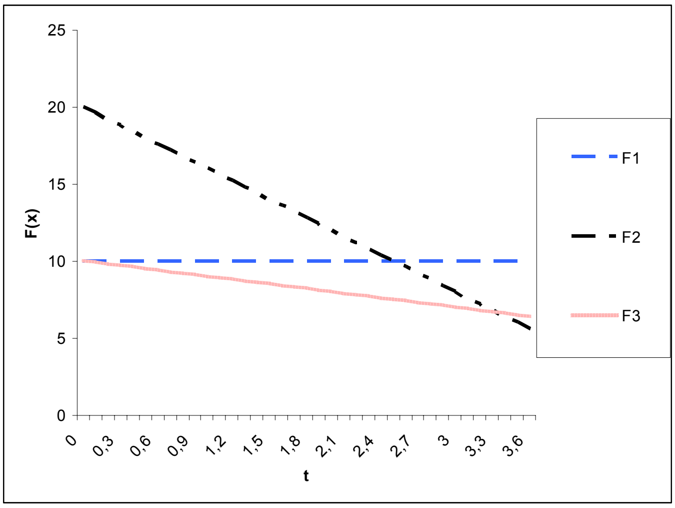
$J'(\bar{x}) = \{1\}$, $P'(\bar{x}) = \{1\}$.



Snadno zjistíme že $\tau = \min(t_1, t_2, t_3, t_4) = t_4 = 5$

2, Tím se dostáváme do bodu (5,10,10).

$J(\bar{x}) = J'(\bar{x}) = \{2\}$, $P(\bar{x}) = P'(\bar{x}) = \{2,3\}$,

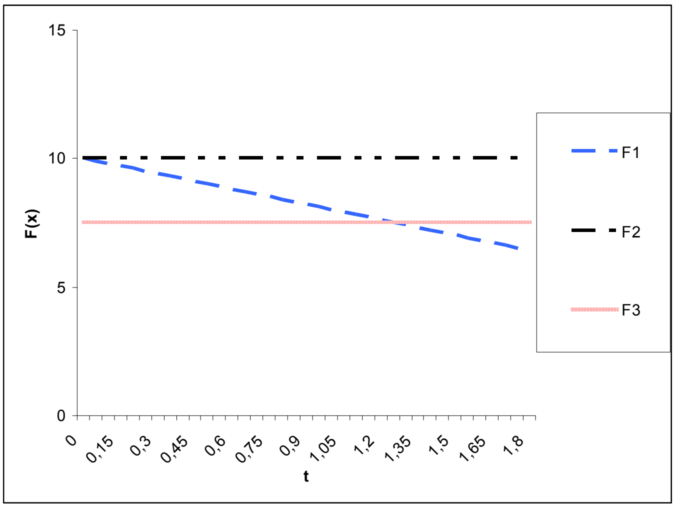


Opět dostaneme $\tau = \min(t_1, t_2, t_3, t_4) = t_4 = 2{,}5$

2, Dalším bodem je $(5, \frac{15}{2}, \frac{15}{2})$

$J(\bar{x}) = \{1,2\}$, $P(\bar{x}) = \{1,2,3\}$,

$J'(\bar{x}) = \{1\}$, $P'(\bar{x}) = \{1\}$.



Z obrázku je zřejmé, že jsme se dostali do obdobné stejné situace jako jsme byli v kroku 1. V následující tabulce vidíme, několik dalších kroků. Situace se stále opakuje a tím dochází k zacyklení algoritmu.

$\overline{x}$	$f(\overline{x})$	$J(\overline{x})$	$P(\overline{x})$	$J'(\overline{x})$	$P'(\overline{x})$	τ
(10,10,10)	20	1,2	1,2,3	1	1	5
(5,10,10)	20	2	2,3	2	2,3	$\frac{5}{2}$
$(5,\frac{15}{2},\frac{15}{2})$	10	1,2	1,2,3	1	1	$\frac{5}{4}$
$(\frac{15}{4},\frac{15}{2},\frac{15}{2})$	10	2	2,3	2	2,3	$\frac{5}{8}$
$(\frac{15}{4},\frac{55}{8},\frac{55}{8})$	$(\frac{7}{2})$	1,2	1,2,3	1	1	$\frac{5}{16}$
$(\frac{55}{16},\frac{55}{8},\frac{55}{8})$	$(\frac{7}{2})$	2	2,3	2	2,3	$\frac{5}{32}$

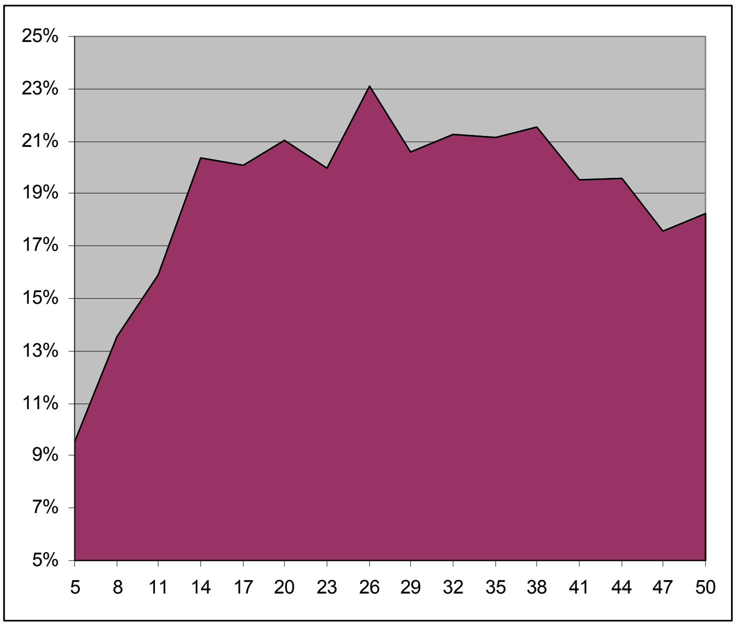
Z toho je zřejmé, že se bude τ neustále zmenšovat a hodnota $\overline{x}$ se bude neustále přibližovat k $(\frac{10}{3},\frac{20}{3},\frac{20}{3})$, ale nikdy této hodnoty nedosáhne.

Během výpočtu je tedy třeba kontrolovat hodnotu τ a pokud se zmenší pod určitou mez musíme výpočet přerušit. Tuto mez můžeme stanovit v závislosti na přesnosti s jakou jsme schopni počítat. Pokud dojde k tomuto zacyklení, musíme pro výpočet použít algoritmus 3.3.

Je dobré se podívat v kolika případech k tomuto zacyklení dojde a jestli má vůbec cenu se o tento výpočet pokoušet.

Test byl proveden na různě velkých příkladech. Pro každou velikost matice bylo otestováno tisíc náhodně vygenerovaných příkladů. V úvahu byly brány pouze úlohy mající řešení.

(n,m)	Neúspěšné
(5,2)	9,5%
(8,3)	13,5%
(11,4)	15,9%
(14,5)	20,4%
(17,6)	20,1%
(20,7)	21,1%
(23,8)	20,0%
(26,9)	23,1%
(29,10)	20,6%
(32,11)	21,3%
(35,12)	21,2%
(38,13)	21,5%
(41,14)	19,5%
(44,15)	19,6%
(47,16)	17,6%
(50,17)	18,2%



Vidíme, že pokud zanedbáme úlohy s n<14, dostaneme, že nezávisle na velikosti úlohy se úspěšnost algoritmu na náhodných datech pohybuje mezi 75-85%.

3.9 Dělená lineární účelová funkce

Úkolem tohoto odstavce bude rozšířit předchozí algoritmus na případ, kdy pro každou $f_j, j \in J$ existuje hodnota $\dot{x}_j$ a platí:

$$f_j = \begin{cases} k_j^1 x_j + d_j^1 \text{ pro } x_j \le \dot{x}_j \\ k_j^2 x_j + d_j^2 \text{ pro } x_j > \dot{x}_j \end{cases}$$

Také platí $k_j^1 \dot{x}_j + d_j^1 = k_j^2 \dot{x}_j + d_j^2$, tedy f_j je spojitá na *R*.

Pozn.: k_j^1 a k_j^2 *nemusí mít různá znaménka. Funkce může být například na obou intervalech rostoucí s jiným sklonem.*

Předchozí algoritmus upravíme aby rozlišoval s kterou částí funkce má právě pracovat. Dále při snižování *x(t)* se zastaví i v bodech $\dot{x}_j, j \in J(\overline{x})$ a pustíme předchozí algoritmus na tuto úlohu. Algoritmus může skončit ve dvou místech.

V kroku 3, kdy pro nějaké $\overline{y}_j, j \in P(\overline{y})$ platí, že $\overline{y}_j = \underline{x}_j$. V tomto kroku není možné dále postupovat a $\overline{y}$ je optimální řešení.

V kroku 2, kdy $\exists j \in J(\overline{y})$, $k_j \leq 0$. V tomto případě to nemusí znamenat konec. Jestliže $\dot{x}_j < \overline{y}_j$ a $k_j^1 > 0$ můžeme najít lepší řešení. Současné $\overline{y}$ si uložíme jako možné řešení a spustíme znovu celý algoritmus na množině $M(\underline{x}, \overline{x}')$, kde

$$\overline{x}'_j = \left\langle \begin{array}{l} \dot{x}_j \text{ pro } j \in J(\overline{y}), k_j \leq 0 \\ \quad \overline{y}_j \text{ jinak} \end{array} \right.$$

Pozn.: *Není třeba si pamatovat všechna možná řešení. Pokud budeme ukládat nové možné řešení, porovnáme ho s již uloženým a necháme pouze lepší.*

Pozn.: *Tímto postupem jsme vlastně rozdělili $M(\underline{x}, \overline{x})$ na dvě podmnožiny M_1 pro $x_j \leq \dot{x}_j$ a M_2 pro $x_j > \dot{x}_j$. Díky tomu, že $M_1 \cup M_2 = M(\underline{x}, \overline{x})$ máme jistotu, že jsme žádné řešení nevynechali. Nedělíme ovšem podle všech $j \in J$, ale pouze tam, kde je možnost najít lepší řešení. Protože podle každého $j \in J$ můžeme dělit pouze jednou, bude složitost, maximálně n-krát horší, nežli složitost algoritmu 3.5.*

Pozn.: *Obdobně je možné algoritmus rozšířit na úlohu, kdy pro každou funkci $f_j, j \in J$ existují body $\dot{x}_j^1, ..., \dot{x}_j^k$, funkce f_j je lineární na $< \dot{x}_j^i, \dot{x}_j^{i+1} >$ a spojitá na R.*

4 Algoritmus řešící systém (max,min) - lineárních rovnic

4.1 Formulace problému

Další obměnou bude nahrazení operátoru + operátorem $\wedge$(minimum). Zopakujme si značení:

$N = \{1,...,n\}, S = \{1,...,m\}, R = (-\infty, \infty),$
$R^n = R \times ... \times R$ (n - krát), $x = (x_1,...,x_n) \in R^n$,
$a_{ij}, b_{ij} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N}$ jsou dané,

$$a_i(x) \equiv \max_{j \in N}(a_{ij} \wedge x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$b_i(x) \equiv \max_{j \in N}(b_{ij} \wedge x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

$$F_i^+(x) = \{j \in N \mid a_{ij} = a_i(x), a_i(x) < x_j\} \quad \forall \mathrm{i} \in \mathrm{S}$$
$$F_i(x) = \{j \in N \mid x_j = a_i(x)\} \quad \forall \mathrm{i} \in \mathrm{S}$$
$$G_i^+(x) = \{j \in N \mid b_{ij} = b_i(x), b_i(x) < x_j\} \quad \forall \mathrm{i} \in \mathrm{S}$$
$$G_i(x) = \{j \in N \mid x_j = b_i(x)\} \quad \forall \mathrm{i} \in \mathrm{S}$$

Proměnné x_j, kde $j \in F_i(x)$ resp. $j \in G_i(x)$ budeme nazývat aktivní proměnné v bodě x pro $a_i(x)$, resp. $b_i(x)$. Proměnné x_j, kde $j \in F_i^+(x)$ resp. $j \in G_i^+(x)$ budeme nazývat +aktivní proměnné v bodě x pro $a_i(x)$, resp. $b_i(x)$.

Budeme uvažovat následující systém $(\max, \wedge)$ -lineárních rovnic:

$$a_i(x) = b_i(x) \ \forall \mathrm{i} \in \mathrm{S} \tag{1}$$

Množinu všech řešení systému (1) budeme značit *M*. Dále si definujeme množinu $M(\bar{x})$ pro $\forall x \in M$ následovně:

$$M(\bar{x}) \equiv \{x \mid x \in M \ \& \ x \le \bar{x}\} \tag{2}$$

Funkce $a_i(x), b_i(x)$ v tomto případě nejsou $\wedge$-homogenní.
Definujme následující hodnoty $\bar{a}_j = \min_{i \in S}(a_{ij}), \bar{b}_j = \min_{i \in S}(b_{ij})$,

$$\tilde{x}_j = \min(\bar{a}_j, \bar{b}_j) \tag{3}$$

Všimněme si, že pro $\forall x \le \tilde{x}$ je soustava (1) splněna. A tedy $M \neq \emptyset$ a pro $\forall \bar{x}, M(\bar{x}) \neq \emptyset$.

Připomeňme si definici 1.1.1: Nechť $L \subseteq R^n, \widetilde{x} \in L$ a platí $x \in L \Rightarrow x \leq \widetilde{x}$. Potom prvek $\widetilde{x}$ se nazývá největší prvek množiny *L*.

Algoritmus uvedený v následující kapitole najde největší prvek množiny $M(\bar{x})$. Existence největšího prvku je dokázána za obecnějších předpokladů v [1].

4.2 Algoritmus

Nyní se podíváme na samotný algoritmus pro nalezení největšího prvku množiny $M(\bar{x})$. Jestliže $\bar{x} \in M(\bar{x})$, potom je také řešením. Nechť tedy $\bar{x} \notin M(\bar{x})$. Bez újmy na obecnosti můžeme předpokládat, že

$$a_i(\bar{x}) \geq b_i(\bar{x}) \ \ \mathrm{i} \in \mathrm{S} \tag{3}$$

Jestliže (3) neplatí, stačí přejmenovat některé funkce a_i, b_i. Nechť tedy $H(\bar{x}) \equiv \{i \in S \mid a_i(\bar{x}) > b_i(\bar{x})\}$. Jestliže $\bar{x} \notin M(\bar{x})$ potom také $H(\bar{x})$ je neprázdná. Dále platí, že $a_i(\bar{x}) = b_i(\bar{x}) \ \ \mathrm{i} \in \mathrm{S} \setminus \mathrm{H}(\mathrm{x})$.

Obdobně jako u algoritmu 1.2 budeme snižovat prvek $\bar{x}$, tak abychom našli největší prvek $M(\bar{x})$. Je tedy třeba zmenšit $a_i(\bar{x})$ pro všechna $i \in H(\bar{x})$. Toho dosáhneme zmenšením všech aktivních a +aktivních proměnných v $a_i(\bar{x}), i \in H(\bar{x})$. Pokud jsou pro $a_i(x)$ v $\bar{x}$ pouze aktivní proměnné, tak jejich snižováním snižujeme také $a_i(\bar{x})$. Pokud jsou pro $a_i(x)$ v $\bar{x}$ také +aktivní proměnné, potom při dostatečně malém *t* zůstane hodnota $a_i(x(t)) = a_i(\bar{x})$. Musíme tedy nejprve snížit +aktivní proměnné, a teprve v tehdy jestliže nejsou žádné +aktivní proměnné v $a_i(\bar{x}), i \in H(\bar{x})$ můžeme snižovat aktivní proměnné. Definujme tedy

$$\begin{aligned} A(\bar{x}) &\equiv \bigcup_{i \in H(\bar{x})} F_i(\bar{x}) \\ A^+(\bar{x}) &\equiv \bigcup_{i \in H(\bar{x})} F^+{}_i(\bar{x}) \end{aligned} \tag{4}$$

V algoritmu budeme snižovat nejprve proměnné z $A^+(\bar{x})$. Pokud $A^+(\bar{x}) = \emptyset$ budeme snižovat $x_j, j \in A(\bar{x})$.

Během procesu zmenšování $x_j, j \in A^+(\bar{x})$ (resp. $j \in A(\bar{x})$) je znovu třeba udržovat rovnosti pro $i \in S \setminus H(\bar{x})$. Narozdíl od algoritmu 1.1 to nebude vždy možné a budeme muset nejprve snížit jiné proměnné. Vezměme například rovnost $a_i(\bar{x}) = b_i(\bar{x})$ pro nějaké $i \in S \setminus H(\bar{x})$, $F_i(\bar{x}) \subseteq A(\bar{x}), F_i^+(\bar{x}) = \emptyset$ a $G_i^+(\bar{x}) \neq \emptyset$. V tomto případě nestačí pouze stejně zmenšit i prvky z $G_i^+(\bar{x})$ protože při malém zmenšení se $a_i(\bar{x})$ také zmenší, ale $b_i(\bar{x})$

zůstane nezměněné. Je tedy potřeba zmenšit nejprve proměnné z $G_i^+(\bar{x})$.

Jestliže si zmenšování některé proměnné vyžádá dřívější zmenšení jiné proměnné, vyvstává zde otázka, jestli nemůže dojít k zablokování algoritmu. K tomu by došlo tím že snížení kterékoliv proměnné si vyžádá předchozí snížení jiné proměnné. A v tom případě nebudeme schopni snížit jakoukoliv proměnnou bez narušení rovností. Uvažujme tedy případ kdy snižování proměnné *j* si vyžádá snížení proměnné *k*. Tedy bez újmy na obecnosti nechť $j \in F_i(\bar{x}), F_i^+(\bar{x}) = \emptyset$ a $k \in G_i^+(\bar{x})$. Potom $c_{ij} \wedge x_j = c_{ik} \wedge x_k$, $x_j < c_{ij}$ a naopak $x_k > c_{ik}$. Tedy máme $x_j = c_{ik} < x_k$. Tedy snižování proměnné x_j si může vyžádat snížení pouze proměnné x_k, kde $x_k > x_j$. Tedy vždy najdeme takovou proměnnou, kterou je možno snížit. Je také zřejmé, že vždy začneme nejprve největší proměnou, protože nám u ní nastane nejméně závislostí.

Nyní tedy můžeme napsat algoritmus.

ALGORITMUS 4.1

1. Pokud $A^+(\bar{x}) \neq \emptyset$, potom $P(\bar{x}) = \{j\}$, kde $j \in A^+(\bar{x})$ a $x_j = \max\limits_{k \in A^+(\bar{x})} x_k$.
   Jinak $P(\bar{x}) := A(\bar{x})$;

2. $E_1 := \{i \in S \setminus H(\bar{x}) \mid F_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, G_i(\bar{x}) \not\subset P(\bar{x}) \,\&\, F_i^+(\bar{x}) = \emptyset \,\&\, G_i^+(\bar{x}) = \emptyset\}$,
   $E_2 := \{i \in S \setminus H(\bar{x}) \mid F_i(\bar{x}) \not\subset P(\bar{x}) \,\&\, G_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, F_i^+(\bar{x}) = \emptyset \,\&\, G_i^+(\bar{x}) = \emptyset\}$;
   $E_1^+ := \{i \in S \setminus H(\bar{x}) \mid F_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, F^+{}_i(\bar{x}) = \emptyset \,\&\, G^+{}_i(\bar{x}) \neq \emptyset\}$,
   $E_2^+ := \{i \in S \setminus H(\bar{x}) \mid G_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, F^+{}_i(\bar{x}) \neq \emptyset \,\&\, G^+{}_i(\bar{x}) = \emptyset\}$;

3. Jestliže $E_1^+ \cup E_2^+ \neq \emptyset$, potom $P(\bar{x}) = \{j\}$, kde hodnota x_j, $j \in \bigcup\limits_{\mathrm{i} \in \mathrm{E}_1^+} (G_i^+(\bar{x})) \cup \bigcup\limits_{\mathrm{i} \in \mathrm{E}_2^+} (F_i^+(\bar{x}))$ je největší možná.
   Jdi na 2.

4. Jestliže $E_1 \cup E_2 = \emptyset$, potom $P(\bar{x})$ je množina proměnných, které je potřeba snížit, KONEC.

5. $P(\bar{x}) := P(\bar{x}) \cup \bigcup\limits_{\mathrm{i} \in \mathrm{E}_1} (G_i(\bar{x}) \setminus P(\bar{x})) \cup \bigcup\limits_{\mathrm{i} \in \mathrm{E}_2} (F_i(\bar{x}) \setminus P(\bar{x}))$, přejdi na 2.;

Nyní se podíváme na počet iterací, které algoritmus provede. Na začátku algoritmu $P(\bar{x})$ obsahuje alespoň jednu proměnnou *j*.

Při každé iteraci nastane jedna ze dvou variant.

Na konci iterace do $P(\bar{x})$ přidáváme alespoň jeden index. Nechť búno je přidána díky E_1. Tedy přidávaný index je aktivní v $G_i(\bar{x})$ a nějaká $x_k, k \in P(\bar{x})$ je aktivní v $F_i(\bar{x})$. Potom $x_i = x_k$.

Nebo se na konci iterace inicializuje $P'(\bar{x})$ novou proměnnou *k*, kde, jak jsme již ukázali, $k > j$ pro nějaké $j \in P(\bar{x})$.

Algoritmus provede maximálně *n* iterací, kdy přidá index některé proměnné x_p do $P(\bar{x})$ a skončí, nebo inicializuje novou $P(\bar{x})$. V tomto případě je inicializovaná jedinou proměnnou *i*. Od tohoto okamžiku v každé iteraci přidává buď minimálně jedno *k*, kde $x_i = x_k$, nebo inicializuje novou $P(\bar{x}) = \{l\}$, kde $x_l > x_i$. Tedy každou proměnou přidá maximálně jednou a algoritmus provede maximálně $n + n$ iterací. Tedy je konečný.

Nyní popíšeme proces snižování proměnných $x_j, j \in P(\bar{x})$.

Definujme $x(t) = (x_1(t), \ldots, x_n(t))$ pro $t \geq 0$ následovně

$$\begin{aligned} &x_j(t) \equiv \bar{x}_j - t \text{ pokud } j \in P(\bar{x}), \\ &x_j(t) \equiv \bar{x}_j \quad \text{jinak} \end{aligned} \tag{5}$$

Jestliže zvýšíme parametr *t*, proměnné $x_j, j \in P(\bar{x})$ budou zmenšeny. (Tj. $x_j(0) = \bar{x}_j, x_j(t) < \bar{x}_j$ pro všechny t>0 a $j \in P(x)$).

Můžeme si všimnout, že pokud snížíme *t* na hodnotu ε>0, kde ε je dostatečně malé, potom pro všechna $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x})$ zůstává

$$F_i(x(t)) = F_i(\bar{x}),\ G_i(x(t)) = G_i(\bar{x}),$$
$$F_i^+(x(t)) = F_i^+(\bar{x}),\ G_i^+(x(t)) = G_i^+(\bar{x}).$$

Dále pro takové *t* platí, že

jestliže $F_i(\bar{x}) \subseteq P(\bar{x}), F_i^+(\bar{x}) = \emptyset$, potom $a_i(x(t)) = a_i(\bar{x}) - t$ jinak $a_i(x(t)) = a_i(\bar{x})$,

jestliže $G_i(\bar{x}) \subseteq P(\bar{x}), G_i^+(\bar{x}) = \emptyset$, potom $b_i(x(t)) = b_i(\bar{x}) - t$ jinak $b_i(x(t)) = b_i(\bar{x})$

a nové $H(x(t)) = H(\bar{x})$.

Budeme zvyšovat *t*, dokud nenastane jedna z následujících podmínek.

(1) $F_i(x(t)) \neq F_i(\bar{x})$, tj. $a_i(x(t)) = \alpha_i(\bar{x}) \equiv \max\limits_{j \in N \setminus P(\bar{x})} (a_{ij} \wedge \bar{x})$ pro nějaké $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x}), F_i^+(\bar{x}) = \emptyset;$

(2) $G_i(x(t)) \neq G_i(\bar{x})$, tj. $b_i(x(t)) = \beta_i(\bar{x}) \equiv \max\limits_{j \in N \setminus P(\bar{x})} (b_{ij} \wedge \bar{x})$ pro nějaké $i \in S$ takové, že $G_i(\bar{x}) \subseteq P(\bar{x}), G_i^+(\bar{x});$

(3) $F_i^+(x(t)) \neq F_i^+(\bar{x})$ pro nějaké $i \in S$.

(4) $G_i^+(x(t)) \neq G_i^+(\bar{x})$ pro nějaké $i \in S$.

(5) $H(x(t)) \neq H(\bar{x})$, tj. $a_i(x(t)) = \beta_i(\bar{x})$ pro nějaké $i \in H(\bar{x})$.

Budeme definovat proměnné t_1, t_2 ,t_3 jako hodnoty, ve kterých nastaly předchozí podmínky.

Pokud $P(\bar{x}) \neq N$ (tj. $P(\bar{x}) \subset N$) a a_{ij}, b_{ij} jsou konečné, potom $\alpha_i(\bar{x}), \beta_i(\bar{x})$ jsou vždy konečné. Z předchozích podmínek můžeme dostat následující definice t_1, t_2, t_3:

$$t_1 = \min_{i \in L_1}(a_i(\bar{x}) - \alpha_i(\bar{x})), \tag{6}$$

kde $L_1 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x}) \;\&\; F_i^+(\bar{x}) = \emptyset\}$;

$$t_2 = \min_{i \in L_2}(b_i(\bar{x}) - \beta_i(\bar{x})), \tag{7}$$

kde $L_2 \equiv \{i \in S \mid G_i(\bar{x}) \subseteq P(\bar{x}) \;\&\; G_i^+(\bar{x}) = \emptyset\}$;

$$t_3 = \min_{i \in S} \min_{j \in P(\bar{x}),\, \text{kde}\, x_j > a_{ij}} (x_j - a_{ij}) \tag{8}$$

$$t_4 = \min_{i \in S} \min_{j \in P(\bar{x}),\, \text{kde}\, x_j > b_{ij}} (x_j - b_{ij}) \tag{9}$$

$$t_5 = \min_{i \in L_3}(a_i(\bar{x}) - \beta_i(\bar{x})), \tag{10}$$

kde $L_3 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x}) \;\&\; i \in H(\bar{x})\}$.

Nyní definujme $\tau \equiv \min(t_1, t_2, t_3, t_4, t_5)$, a $x(\tau)$ určíme jako novou horní mez. Jestliže $H(x(\tau)) \neq \emptyset$, začneme další iteraci s touto horní mezí, v opačném případě jsme nalezli řešení. Nyní již můžeme napsat ALGORITMUS 4.2 řešící celou úlohu.

ALGORITMUS 4.2

1. $\bar{y} := \bar{x}$

2. Jestliže $H(\bar{y}) = \emptyset$, potom $\bar{y}$ je největší prvek $M(\bar{x})$, KONEC

3. Najdi $P(\bar{y})$ pomocí ALGORITMU 4.1 (za $\bar{x}$ dosadíme $\bar{y}$).

4. Najdi $x(t)$ pomocí (5) a t_1, t_2, t_3, t_4, t_5 pomocí (6)-(10) (za $\bar{x}$ dosadíme $\bar{y}$)

5. $\tau := \min(t_1, t_2, t_3, t_4, t_5)$, $\bar{y} := x(\tau)$, přejdi na **2.**

4.3 Důkaz správnosti a konečnosti ALGORITMU 4.2

Věta 4.3.1: Nechť $\bar{y}$ je současná horní mez v Algoritmu 4.2, $\bar{y} \notin M(\bar{y})$, $t \in (0, \tau)$, $\tilde{x} \le \bar{y}$ a $\tilde{x} \not\le x(t)$. Potom $\tilde{x} \notin M(\bar{y})$.

Důkaz:

Nejprve nechť $\tilde{x}_p > x_p(t)$ pro nějaké *p*, kterým byla množina $P(\bar{y})$ inicializována.

(i) Nechť $A^+(\bar{y}) \ne \emptyset$ a $p \in A(\bar{y}) = \bigcup_{i \in H(\bar{y})} F_i(\bar{y})$. Potom pro nějaké $k \in H(\bar{y})$ platí $p \in F_k(\bar{y})$, a $F_k^{\,+}(\bar{y}) = \emptyset, G_k^{\,+}(\bar{y}) = \emptyset$.

Máme $\bar{y}_p \ge \tilde{x}_p > x_p(t) \ge x_p(\tau)$, takže existuje $\tilde{t} \in (0, t)$ takový, že $\tilde{x}_p = \bar{y}_p - \tilde{t} = x_p(\tilde{t})$ a $a_k(x(\tilde{t})) = a_k(\bar{y}) - \tilde{t}$.

Nejprve předpokládejme $G_k(\bar{y}) \not\subseteq P(\bar{y})$. Potom $b_k(x(\tilde{t})) = b_k(\bar{y})$, $b_k(\bar{y}) \ge b_k(\tilde{x})$. A protože $F_k(\bar{y}) = F_k(x(\tilde{t})) = F_k(x(t)), a_k(x(\tilde{t})) = a_{kp} \wedge (\bar{y}_p - \tilde{t}) = \bar{y}_p - \tilde{t}$ a tedy máme $a_k(\bar{y}) \ge a_k(\tilde{x}) \ge a_{kp} \wedge \bar{y}_p - \tilde{t} = a_k(x(\tilde{t})) > b_k(\bar{y}) \ge b_k(\tilde{x})$ a tedy $\tilde{x} \notin M(\bar{y})$.

Pokud $G_k(\bar{y}) \subseteq P(\bar{y})$, potom $b_k(x(t)) = b_k(\bar{y}) - t$ a existuje $t^* \in (0, t)$ takový, že $b_k(\tilde{x}) = b_k(\bar{y}) - t^*$. Jelikož $p \in F_k(\bar{y}) = F_k(x(t^*))$ máme:

$a_k(\tilde{x}) \ge a_{kp} \wedge x_p(t^*) = a_{kp} \wedge (\bar{y}_p - t^*) = \bar{y}_p - t^* = a_k(\bar{y}) - t^* > b_k(\bar{y}) - t^* = b_k(\tilde{x})$ a tedy opět $\tilde{x} \notin M(\bar{y})$.

(ii) Nechť $p \in A^+(\bar{y})$. Tedy existuje $k \in H(\bar{y})$ takové, že platí $p \in F_k^{\,+}(\bar{y})$. Tedy $a_k(\bar{y}) = a_k(x(t))$. Protože $\tilde{x} \le \bar{y}$ máme $a_k(\tilde{x}) \le a_k(\bar{y})$ a $b_k(\tilde{x}) \le b_k(\bar{y})$. Zároveň protože $\tilde{x}_p > x_p(t)$ a $p \in F_k^{\,+}(\bar{y}) = F_k^{\,+}(x(t))$ máme $a_k(\tilde{x}) \ge a_k(x(t))$. Z toho plyne $a_k(\tilde{x}) = a_k(\bar{y}) > b_k(\bar{y}) \ge b_k(\tilde{x})$. A tedy $\tilde{x} \notin M(\bar{y})$.

(iii) Množina $P(\bar{y})$ byla inicializována během běhu algoritmu 4.1. Budeme pouze předpokládat, že $\tilde{x}_p \ge x_p(t)$.

Existuje předchozí množina $P'(\bar{y})$ a $k \in S \setminus H(\bar{y})$ pro které $p \in F_k^+(\bar{y})$, $G_k^+(\bar{y}) = \emptyset$, $G_k(\bar{y}) \subseteq P'(\bar{y})$ nebo $p \in G_k^+(\bar{y})$, $F_k^+(\bar{y}) = \emptyset$, $F_k(\bar{y}) \subseteq P'(\bar{y})$. Búno nechť nastala první možnost. Protože $\bar{y}_p \ge \tilde{x}_p > x_p(t)$ a $p \in F_k^+(\bar{y}) = F_k^+(x(t))$ máme $a_k(\tilde{x}) = a_k(\bar{y}) = b_k(\bar{y})$. Pokud má být $\tilde{x} \in M(\bar{y})$, musí $b_k(\bar{y}) = b_k(\tilde{x})$. Tedy existuje $r \in G_k(\bar{y})$, $\tilde{x}_r = \bar{y}_r$. Protože $r \in G_k(\bar{y})$ tedy také $r \in P^1(\bar{y})$. Z postupu tvoření $P^1(\bar{y})$ plyne, že existuje $p^1 \in P^1(\bar{y})$, kterým byla $P^1(\bar{y})$ inicializovaná a pro který platí $x_{p^1} = \bar{y}_{p^1}$. Jinak by došlo k porušení některé rovnosti a tedy $\widetilde{x} \notin M(\overline{y})$. Máme $x_{p^1} = \overline{y}_{p^1} \geq x_{p^1}(t)$.

Pokud $P^1(\overline{y})$ byla inicializována během běhu algoritmu 4.1., máme p^1 takový, že $\widetilde{x}_{p^1} \geq x_{p^1}(t)$, a tedy můžeme indukčně pokračovat, dokud nezískáme množinu $P^k(\overline{y})$, která byla inicializována na počátku algoritmu. Tedy máme nějaký prvek p^k, $p^k \in P^k(\overline{y})$, pro který platí $\widetilde{x}_{p^k} = \overline{y}_{p^k}$ a $p^k \in A(\overline{y})$ nebo $p^k \in A^+(y)$. Tedy existuje $i \in H(\overline{y})$, kde $p^k \in F_i(\overline{y})$ nebo $p^k \in F_i^+(\overline{y})$.Z $\widetilde{x}_{p^k} = \overline{y}_{p^k}$ a $\widetilde{x} \leq \overline{y}$ plyne $a_i(\widetilde{x}) = a_i(\overline{y})$. Z $\widetilde{x} \leq \overline{y}$ také plyne $b_i(\overline{y}) \geq b_i(\widetilde{x})$. Tedy máme:

$a_i(\widetilde{x}) = a_i(\overline{y}) > b_i(\overline{y}) \geq b_i(\widetilde{x})$. Z čehož plyne, že $\widetilde{x} \notin M(\overline{y})$.

Nechť tedy $\widetilde{x}_p > x_p(t)$ platí pro nějaké *p*, které bylo do $P(\overline{y})$ přidáno v běhu algoritmu. Označme si množinu $N_1 \equiv \{j \mid x_j(t) < \widetilde{x}_j \leq \overline{y}_j\}$. Nechť $p \in N_1$ je prvek, který do $P(\overline{y})$ vstoupí jako první. Tedy existuje index $i_0 \in S \setminus H(\overline{y})$, pro který platí $F_{i_0}^{\ +} = \emptyset$, $G_{i_0}^{\ +} = \emptyset$ a buď $p \notin F_{i_0}(\overline{y})$ a $p \in G_{i_0}(\overline{y})$ nebo $p \in F_{i_0}(\overline{y})$ a $p \notin G_{i_0}(\overline{y})$. Búno předpokládejme, že nastal první případ. Jelikož $F_{i_0}(\overline{y}) = F_{i_0}(x(t))$ a $G_{i_0}(\overline{y}) = G_{i_0}(x(t))$ máme také, že $p \notin F_{i_0}(x(t))$ a $p \in G_{i_0}(x(t))$. Protože p je první vložený do $P(\overline{y})$, je zřejmé, že pro všechna $j \in N_1$ platí $j \notin F_{i_0}(x(t))$. A tedy máme:

$$a_{i_0 j} \wedge \widetilde{x}_j < a_{i_o}(x(t)) \quad \text{pro všechna } j \in N_1.$$

Protože pro všechna ostatní $\widetilde{x}_j, j \in N \setminus N_1$ platí $\widetilde{x}_j \leq x_j(t)$ dostali jsme

$$a_{i_0}(\widetilde{x}) < a_{i_0}(x(t)) = b_{i_0}(x(t)) = b_{i_0 p} \wedge x_p(t)$$

Protože $p \in G_{i_0}(x(t))$ máme

$$a_{i_0}(\widetilde{x}) < a_{i_0}(x(t)) = b_{i_0}(x(t)) = b_{i_0 p} \wedge x_p(t)$$

a tedy $\widetilde{x} \notin M(\overline{y})$.

□

4.4 Optimalizace s max-separabilní účelovou funkcí.

Opět budeme řešit optimalizační úlohu pro max-separabilní účelovou funkci. Budeme tedy minimalizovat funkci

$$f(x) = \max_j f_j(x_j), \text{kde } j \in J \subseteq \{1,\dots,N\} \tag{11}$$

na množině

$$M(\underline{x},\overline{x}) \equiv \{x \mid x \in M \ \& \ \underline{x} \le x \le \overline{x}\},$$

kde $M = \{x \mid a_i(x) = b_i(x) \ \forall \mathrm{i} \in \mathrm{S}\}$ pro

$$a_i(x) \equiv \max_{j \in N}(a_{ij} \wedge x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

$$b_i(x) \equiv \max_{j \in N}(b_{ij} \wedge x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

Podívejme se na algoritmy 3.2 a 3.3. Tyto algoritmy využívají algoritmus 3.1 pro nalezení největšího prvku na množině soustavy (max,+)-lineárních rovnic. Správnost a konečnost algoritmu 3.2 (resp. 3.3) ale není vůbec závislá na průběhu algoritmu 3.1, ani na vlastnostech množiny M na které se optimalizační úloha řeší. Pouze je potřeba, aby používaný algoritmus našel pro dané $\underline{x},\overline{x}$ největší prvek $M(\underline{x},\overline{x})$ nebo poznal, že $M(\underline{x},\overline{x}) = \emptyset$. Upravíme tedy algoritmus 4.2 tak, aby našel největší prvek množiny $M(\underline{x},\overline{x})$, nebo poznal, že $M(\underline{x},\overline{x}) = \emptyset$. K tomu použijeme následující lemma.

Lemma 4.4.1 Nechť $x^{\max}$ je prvek nalezený ALGORITMEM 4.2. Potom

(iii) Nechť $x^{\max} \ge \underline{x}$. Potom $x^{\max} \in M(\underline{x},\overline{x})$ a dokonce je největším prvkem.

(iv) Nechť $x^{\max} \not\ge \underline{x}$. Potom $M(\underline{x},\overline{x}) = \emptyset$.

Důkaz:

(ii)

$x^{\max} \ge \underline{x}$. Dle ALGORTIMU 4.2 je $x^{\max} \in M(\overline{x})$ a tedy $x^{\max} \le \overline{x}$. Z toho plyne, že $x^{\max} \in M(\underline{x},\overline{x})$.

To že je největší prvek dokážeme sporem. Nechť existuje $\tilde{x} \in M(\underline{x},\overline{x})$, $\tilde{x} > x^{\max}$. Tedy $\tilde{x} \in M(\overline{x})$ a $\tilde{x} > x^{\max}$, což je spor se správností ALGORITMU 4.2.

(ii)

Sporem. Nechť $x^{\max} < \underline{x}$ a $M(\underline{x},\overline{x}) \ne \emptyset$. Tedy nechť existuje $\tilde{x} \in M(\underline{x},\overline{x})$. Jelikož $M(\underline{x},\overline{x}) \subseteq M(\overline{x})$ máme $\tilde{x} \in M(\overline{x})$. Máme $x^{\max} < \underline{x} \le \tilde{x}$ a tedy $x^{\max}$ není největší v $M(\overline{x})$. Což je spor.

□

Můžeme tedy nechat proběhnout celý algoritmus a na konci zkontrolovat, jestli $x^{\max} \ge \underline{x}$. To by ale mohlo být velmi neefektivní pro případ kdy $x^{\max} < \underline{x}$. Je tedy lepší kontrolovat tuto podmínku při každé iteraci algoritmu. Upravený algoritmus bude tedy vypadat následovně:

ALGORITMUS 4.3

1. $\overline{y} := \overline{x}$

2. Jestliže $\overline{y} < \underline{x}$, potom $M(\overline{x}) = \emptyset$. KONEC

Jestliže $H(\overline{y}) = \emptyset$, potom $\overline{y}$ je největší prvek $M(\overline{x})$, KONEC

3. Najdi $P(\overline{y})$ pomocí ALGORITMU 4.1 (za $\overline{x}$ dosadíme $\overline{y}$).

4. Najdi $x(t)$ pomocí (5) a $t_1, t_2, t_{3,} t_4, t_5$ pomocí (6)-(10) (za $\overline{x}$ dosadíme $\overline{y}$)

5. $\tau := \min(t_1, t_2, t_3, t_4, t_5)$, $\overline{y} := x(\tau)$, přejdi na **2.**

Nechť tedy máme optimalizační úlohu:

$$\min_{x \in M(\underline{x}, \overline{x})} f(x)$$
$$M(\underline{x}, \overline{x}) \equiv \{x \mid x \in M \ \&\ \underline{x} \le x \le \overline{x}\}$$
$$M = \{x \mid a_i(x) = b_i(x) \ \forall \mathrm{i} \in \mathrm{S}\}$$

$$a_i(x) \equiv \max_{j \in N}(a_{ij} \wedge x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$b_i(x) \equiv \max_{j \in N}(b_{ij} \wedge x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

Jestliže $f(x) = \max_j f_j(x_j)$, kde $j \in J \subseteq \{1, ..., N\}$ a $f_j, j \in J$ jsou spojité, monotónní funkce a pro každou $f_j, j \in J$ známe také f_j^{-1}, Potom tuto úlohu můžeme řešit algoritmem 3.2, kde pro nalezení největšího prvku, místo algoritmu 3.1, budeme volat algoritmus 4.3.

Jestliže $f(x) = \max_j f_j(x_j)$, kde $j \in J \subseteq \{1, ..., N\}$ a $f_j, j \in J$ jsou spojité, unimodální funkce a pro každou $f_j, j \in J$ známe také $\dot{x}_j$ a inverzní funkce f_j' f_j'', kde

$$f_j' = f_j^{-1}(x) \text{ pro } x \in (-\infty, \dot{x}_j >$$
$$f_j'' = f_j^{-1}(x) \text{ pro } x \in < \dot{x}_j, \infty),$$

potom tuto úlohu můžeme řešit algoritmem 3.3, kde pro nalezení největšího prvku, místo algoritmu 3.1, budeme volat algoritmus 4.3.

5 Algoritmus řešící systém (max,+)-lineárních nerovnic

5.1 Formulace problému

Zaveďme následující značení:

$N = \{1,...,n\}, S = \{1,...,m\}, R = (-\infty, \infty),$
$R^n = R \times ... \times R$ (n - krát), $x = (x_1,...,x_n) \in R^n$,
$a_{ij}, b_{ij} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N}$ jsou dané,

$$
\begin{aligned}
&a_i(x) \equiv \max_{j \in N}(a_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S} \\
&b_i(x) \equiv \max_{j \in N}(b_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S} \\
&F_i(x) = \{j \in N \mid a_{ij} + x_j = a_i(x)\} \ \forall \mathrm{i} \in \mathrm{S}, \\
&G_i(x) = \{j \in N \mid b_{ij} + x_j = b_i(x)\} \ \forall \mathrm{i} \in \mathrm{S}
\end{aligned}
$$

Proměnné x_j, kde $j \in F_i(x)$ resp. $j \in G_i(x)$ budeme nazývat aktivní proměnné v bodě x pro $a_i(x)$, resp. $b_i(x)$.

Budeme uvažovat následující systém (max,+)-lineárních nerovnic:

$$
\begin{aligned}
a_i(x) &\le b_i(x) \quad \forall i \in S_1 \\
a_i(x) &= b_i(x) \quad \forall i \in S_2
\end{aligned}, \tag{1}
$$

kde platí $S = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$. Budeme předpokládat, že $S_1 \neq \emptyset$. Jinak bychom dostali již vyřešený případ.

Bez újmy na obecnosti můžeme předpokládat, že existuje $\bar{i} \in S$, tak že $S_1 = \{1,..,\bar{i}\}$, $S_2 = \{\bar{i}+1,..,m\}$. Pokud $\bar{i} = m$ potom $S_2 = \emptyset$. (Libovolný příklad můžeme převést do tohoto tvaru přečíslováním nerovností.)

Množinu všech řešení systému (1) budeme značit *M*. Dále si definujeme množinu $M(\bar{x})$ pro $\forall x \in M$ následovně:

$$
M(\bar{x}) \equiv \{x \mid x \in M \ \& \ x \le \bar{x}\} \tag{2}
$$

Funkce $a_i(x), b_i(x)$ jsou definovány jako v kapitole 1 a jsou tedy opět +-homogenní. Z toho plyne, že pokud $x \in M$, potom $x(\alpha) \in M$ pro všechny $\alpha \in R$.

Lemma 5.1.1 Máme dáno $\bar{x} \in R.$ Potom $M \neq \emptyset \Leftrightarrow M(\bar{x}) \neq \emptyset.$

Důkaz:

Z toho, že $M(\bar{x}) \subseteq M$ rovnou plyne $M(\bar{x}) \neq \emptyset \Rightarrow M \neq \emptyset$. Jestliže $M \neq \emptyset$ a $\tilde{x} \in M$, můžeme vždy nalézt takové $\alpha \in R$ aby $\tilde{x}(\alpha) \leq \bar{x}$. A jelikož $\tilde{x}(\alpha) \in M$ je také $\tilde{x}(\alpha) \in M(\bar{x})$. Máme tedy $M(\bar{x}) \neq \emptyset$.

□

5.2 Převedení na soustavu rovnic

V této kapitole se pokusíme převést tuto úlohu na soustavu rovnic. Podívejme se na libovolnou nerovnost. Vezměme první nerovnost.. Máme $a_1(x) \leq b_1(x)$. Tedy

$$\max_{j \in N}(a_{1j} + x_j) \leq \max_{j \in N}(b_{1j} + x_j)$$

Tuto nerovnost můžeme převést na rovnost zavedením nové neznámé x_{n+1} a definováním $a_{1n+1} = 0$. Musíme ale ještě definovat $a_{i,n+1}$ pro $i \in S \setminus \{1\}$ a $b_{i,n+1}$ pro $i \in S$. Tyto hodnoty musí být natolik malé, aby v konečném řešení nebyly aktivní. Definujme tedy $a_{i,n+1} = -K$ pro $i \in S \setminus \{1\}$ a $b_{i,n+1} - K$ pro $i \in S$, kde K je dostatečně velké. Později se podíváme podrobněji na velikost K.

Provedeme-li tuto úpravu pro všechna $i \in S_1$ dostáváme následující úlohu:

$$N = \{1,...,n\}, N' = \{n+1,...n+\bar{i}), S = \{1,...,m\},$$
$$x = (x_1,...,x_n, x_{n+1},...,x_{n+\bar{i}}) \in R^{n+\bar{i}},$$

$$a_{ij}, b_{ij} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N} \text{ jsou dané,`}$$

$$a_{ij} = \left\langle \begin{array}{l} 0 \text{ pro } i \in S, j \in N', j = n+i \\ -K \text{ pro } i \in S, j \in N', j \neq n+i \end{array} \right.$$

$$b_{ij} = -K, \text{pro } i \in S, j \in N'$$

, kde *K* je dostatečně velké.

$$a_i(x) \equiv \max_{j \in N \cup N'}(a_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

$$b_i(x) \equiv \max_{j \in N \cup N'}(b_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

Uvažujeme systém rovnic $a_i(x) = b_i(x), i \in S$. Množinu všech řešení této soustavy označíme M'.

Ještě je potřeba se podívat na velikost *K*. Potřebujeme, aby pro $x \in M'$ a pro všechna $i \in S, j \in N'$ platí $a_{ij} = -K \Rightarrow j \neq F_i(x)$ a pro všechna $i \in S, j \in N'$ platí $j \neq G_i(x)$. Jinak by mohlo vynecháním proměnných $x_{n+1},...,x_{n+\bar{i}}$ a návratem k původní soustavě nerovností mohlo

dojít k porušení podmínek (1).
Spočítejme

$$\overline{m} = \max_{i\in S, j\in N}(a_{ij}, b_{ij})$$
$$\underline{m} = \min_{i\in S, j\in N}(a_{ij}, b_{ij}) \tag{3}$$

a definujme K libovolně tak, aby platilo $K > \overline{m} - \underline{m}$.

Tvrzení: Nechť $K > \overline{m} - \underline{m}$ a $x \in M'$. Potom $i \in S, j \in N'$ platí $a_{ij} = -K \Rightarrow j \neq F_i(x)$ a pro všechna $i \in S, j \in N'$ platí $j \neq G_i(x)$.

Důkaz: Sporem. Nechť $x \in M'$ a existuje $i \in S, j \in N'$ tak, že $a_{ij} = -K \,\&\, j = F_i(x)$ nebo $j = G_i(x)$. Protože platí $b_{ij} = -K, \text{pro } i \in S, j \in N'$ můžeme druhou podmínku přepsat do tvaru $b_{ij} = -K \,\&\, j = G_i(x)$. Uvažujme nejprve první případ.

Platí tedy, že existuje $i \in S, \hat{j} \in N'$ tak, že $a_{i\hat{j}} = -K \,\&\, \hat{j} = F_i(x)$. Platí $a_{\hat{i}}(x) = -K + x_{\hat{i}\hat{j}}$. Pro všechna $j \in N'$ tedy platí pokud $a_{ij} = -K$, potom $-K + x_j \le a_i(x) = -K + x_{\hat{j}}$. Pokud $a_{ij} = 0$, potom $x_j \le a_i(x) = -K + x_{\hat{j}}$. Tedy:

$$\forall j \in N', x_j \le x_{\hat{j}} \tag{4}$$

Pro všechna $j \in N$ platí $a_{ij} + x_j \le a_i(x) = -K + x_{\hat{j}}$. Protože $\underline{a} \le a_{ij}$ máme

$$\underline{m} + x_j \le -K + x_{\hat{j}} \tag{5}$$

Pro $b_{ij} = -K \,\&\, j = G_i(x)$ dokážeme (4) a (5) analogicky.

Vezmeme $i \in S$ tak, aby $a_{i\hat{j}} = 0$. Potom $a_i(x) \ge 0 + x_{\hat{j}} = x_{\hat{j}}$. Budeme zkoumat $b_{ij} + x_j$.
Pokud $j \in N'$, potom díky (4) platí $b_{ij} + x_j = -K + x_j \le -K + x_{\hat{j}} < x_{\hat{j}}$.
Pokud $j \in N$, potom $b_{ij} + x_j \le \overline{m} + x_j$. Z předpokladu $K > \overline{m} - \underline{m}$ dostáváme $\overline{m} < K + \underline{m}$.
Díky tomu a (5) získáváme $b_{ij} + x_j \le \overline{m} + x_j < K + \underline{m} + x_j \le K - K + x_{\hat{j}} = x_{\hat{j}}$.
Tedy pro všechna $j \in N \cup N'$ platí $b_{ij} + x_j < x_{\hat{j}}$. Tedy $b_i(x) < x_{\hat{j}} \le a_i(x)$ což je spor s $x \in M'$.

□

Věta 5.2.1: Nechť $K > \overline{m} - \underline{m}$ a $x = (x_1, x_2, \ldots x_n)$. Potom platí $x \in M$ právě tehdy, když existuje $x_{n+1}, \ldots, x_{n+\bar{i}}$, tak že $x' = (x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+\bar{i}}) \in M'$.

Důkaz:
Nejprve nechť $x \in M$. V tom případě definujme $x_{n+k} = b_k(x)$ pro všechna $k \in 1, \ldots, \bar{i}$.

Předpokládejme, že existuje $i \in S, j \in N'$ takový, že $j \in G_i(x')$. Víme, že

$x_j = b_{j-\bar{i}}(x) = b_{j-\bar{i},j'} + x_{j'}$, kde $j' \in N$. Jestliže $j \in G_i(x')$, potom $-K + x_j \geq b_{ij'} + x_{j'} \geq \underline{m} + x_{j'} > \overline{m} - K + x_{j'} \geq b_{j-\bar{i},j'} - K + x_{j'} = -K + x_j$ což je spor.

Nyní předpokládejme, že existuje $i \in S, j \in N'$ takový, že $j \in F_i(x')$ a $a_{ij} = -K$. Stejným způsobem dojdeme opět ke sporu. Tedy pro všechna $i \in S$ platí

$a_i(x') = \max(a_i(x), \max_{j \in N'}(a_{ik} + x_k)) = \max(a_i(x), b_i(x)) = b_i(x) = b_i(x')$. Tedy $x' \in M'$.

Nechť $x' \in M'$. Dle předchozího tvrzení pro $i \in S, j \in N'$ platí $a_{ij} = -K \Rightarrow j \notin F_i(x)$ a pro všechna $i \in S, j \in N'$ platí $j \notin G_i(x)$.

Pro $i \in S_1$ platí $a_i(x) \leq a_i(x') = b_i(x') = b_i(x')$.

Pro $i \in S_2$ platí $a_i(x) = a_i(x') = b_i(x') = b_i(x')$.

Tedy $x \in M$.

□

Abychom mohli použít Algoritmus 1.2, je potřeba shora omezit proměnné $x_{n+1},...,x_{n+\bar{i}}$. Máme zadané $\bar{x}$. Je zřejmé, že pro $x \leq \bar{x}$ platí $b_i(x) \leq b_i(\bar{x})$ pro všechna $i \in S$. Tedy můžeme definovat $\bar{x}_{n+k} = b_k(\bar{x})$ pro $k = 1,...,\bar{i}$. Nyní již můžeme použít Algoritmus 1.2 na úlohu $M'(\bar{x}')$, kde $\bar{x}' = (\bar{x}_1,...,\bar{x}_n,\bar{x}_{n+1},...,\bar{x}_{n+\bar{i}})$. Algoritmus vrátí největší prvek množiny $M'(\bar{x}')$, což je (po vynechání $\bar{x}_{n+1},...,\bar{x}_{n+\bar{i}}$) největší prvek množiny $M(\bar{x})$.

Díky tomuto převodu jsme také ověřili existenci největšího prvku na množině $M(\bar{x})$. Neboť na $M'(\bar{x}')$ největší prvek existuje a ten je zároveň největším prvek (po vynechání $\bar{x}_{n+1},...,\bar{x}_{n+\bar{i}}$) množiny $M(\bar{x})$.

Abychom mohli využít optimalizační algoritmy, je potřeba určit také dolní mez. Obdobně jako pro horní mez stačí definovat $\underline{x}_{n+k} = b_k(\underline{x})$ pro $k = 1,...,\bar{i}$, neboť $b_k(\underline{x})$ je nejmenší hodnota, jaké může $b_k(x)$ pro $\underline{x} \leq x \leq \bar{x}$ dosáhnout.

5.3 Upravený algoritmus

Je zřejmé, že při převedení na soustavu rovnic zbytečně přidáváme nové proměnné a zvyšujeme složitost výpočtu. V této kapitole upravíme algoritmus 1.1 tak, aby řešil přímo soustavu nerovnic. Díky předchozí kapitole víme, že největší prvek existuje.

Pokud $\bar{x} \in M(\bar{x})$, potom je také řešením. Předpokládejme tedy, že $\bar{x} \notin M(\bar{x})$. Bez újmy na obecnosti můžeme předpokládat, že

$$a_i(\bar{x}) \geq b_i(\bar{x})\ i \in S_2 \tag{6}$$

Jestliže (6) neplatí, stačí přejmenovat některé koeficienty a_i, b_i. Definujme opět $H(\bar{x}) \equiv \{i \in S \mid a_i(\bar{x}) > b_i(\bar{x})\}$. Jestliže $\bar{x} \notin M(\bar{x})$, potom také $H(\bar{x})$ je neprázdná. Také platí, že $a_i(\bar{x}) \leq b_i(\bar{x})$ $i \in S_1 \backslash H(\bar{x})$ a $a_i(\bar{x}) = b_i(\bar{x})$ $i \in S_2 \backslash H(\bar{x})$.

Abychom získali prvek z množiny $M(\bar{x})$, je potřeba zmenšit $a_i(\bar{x})$ pro všechna $i \in H(\bar{x})$. Abychom toho dosáhli, je potřeba zmenšit všechny aktivní proměnné v $a_i(\bar{x}), i \in H(\bar{x})$. To znamená všechny proměnné $x_j, j \in A(\bar{x})$, kde

$$A(\bar{x}) \equiv \bigcup_{i \in H(\bar{x})} F_i(\bar{x}) \tag{7}$$

Během procesu zmenšování $x_j, j \in A(\bar{x})$ je třeba se opět postarat o udržení rovností pro $i \in S \setminus H(\bar{x})$. To bude probíhat stejně jako v Algoritmu 1.1, pouze pro $i \in S_1$ nám stačí hlídat, aby $a_i(\bar{x}) \leq b_i(\bar{x})$. Tedy algoritmus bude vypadat následovně:

ALGORITMUS 5.1

1. $P(\bar{x}) := A(\bar{x});$

2. $E_1 := \{i \in S_2 \setminus H(\bar{x}) \mid F_i(\bar{x}) \subseteq P(\bar{x})\ \&\ G_i(\bar{x}) \not\subset P(\bar{x})\},$

   $E_2 := \{i \in S_2 \setminus H(\bar{x}) \mid F_i(\bar{x}) \not\subset P(\bar{x})\ \&\ G_i(\bar{x}) \subseteq P(\bar{x})\},$

   $E_3 := \{i \in S_1 \setminus H(\bar{x}) \mid F_i(\bar{x}) \not\subset P(\bar{x})\ \&\ G_i(\bar{x}) \subseteq P(\bar{x})\ \&\ a_i(\bar{x}) = b_i(\bar{x})\};$

3. Jestliže $E_1 \cup E_2 \cup E_3 = 0$, potom $P(\bar{x})$ je množina proměnných, které je potřeba snížit, KONEC.

4. $P(\bar{x}) := P(\bar{x}) \cup \bigcup_{i \in E_1} (G_i(\bar{x}) \setminus P(\bar{x})) \cup \bigcup_{i \in E_2} (F_i(\bar{x}) \setminus P(\bar{x})) \cup \bigcup_{i \in E_3} (F_i(\bar{x}) \setminus P(\bar{x})),$

   přejdi na **2.**;

Protože $A(\bar{x})$ obsahuje alespoň jednu proměnnou a při každé iteraci se alespoň jedna proměnná přidá do $P(\bar{x})$, je ALGORITMUS 1.1 konečný a skončí po maximálně *n-1* iteracích.

Musíme ještě popsat proces snižování proměnných $, j \in P(\bar{x})$.

Opět definujme $x(t) = (x_1(t),...,x_n(t))$ pro $t \geq 0$ následovně:

$$x_j(t) \equiv \bar{x}_j - t \text{ pokud } j \in P(\bar{x}),$$
$$x_j(t) \equiv \bar{x}_j \text{ jinak} \tag{8}$$

I zde platí, že pokud snížíme t na hodnotu $\varepsilon > 0$, kde ε je dostatečně malé, potom pro všechna $i \in S$ zůstává

$$F_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow F_i(x(t)) = F_i(\bar{x}),$$
$$G_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow G_i(x(t)) = G_i(\bar{x}),$$

dále platí

$$F_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow a_i(x(t)) = a_i(\bar{x}) - t, \quad G_i(\bar{x}) \subseteq P(\bar{x}) \Rightarrow b_i(x(t)) = b_i(\bar{x}) - t$$

a nové $H(x(t)) = H(\bar{x})$.

Předpokládejme, že $P(\bar{x}) \neq N$ (později uvidíme, že pokud $P(\bar{x}) = N$ potom $M(\bar{x}) = \emptyset$).

Budeme zvyšovat *t*, dokud nenastane jedna z následujících podmínek.

(1) $F_i(x(t)) \neq F_i(\bar{x})$, tj. $a_i(x(t)) = \alpha_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})} (a_{ij} + \bar{x}_j)$ pro nějaké $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x})$;

(2) $G_i(x(t)) \neq G_i(\bar{x})$, tj. $b_i(x(t)) = \beta_i(\bar{x}) \equiv \max_{j \in N \setminus P(\bar{x})} (b_{ij} + \bar{x}_j)$ pro nějaké $i \in S$ takové, že $G_i(\bar{x}) \subseteq P(\bar{x})$;

(3) $H(x(t)) \neq H(\bar{x})$, tj. $a_i(x(t)) = \beta_i(\bar{x})$ pro nějaké $i \in H(\bar{x})$.

(4) $a_i(x(t)) = b_i(x(t))$ pro $i \in S_1, a_i(\bar{x}) < b_i(\bar{x})$. Tj. $\alpha_i(\bar{x}) = b_i(x(t))$ pro nějaké $i \in S_1$ takové, že $F_i(\bar{x}) \not\subseteq P(\bar{x})$ a $G_i(\bar{x}) \subseteq P(\bar{x})$.

Budeme definovat proměnné *t₁, t₂, t₃, t₄* jako hodnoty, ve kterých nastaly předchozí podmínky.

Pokud $P(\bar{x}) \neq N$ (tj. $P(\bar{x}) \subset N$) a a_{ij}, b_{ij} jsou konečné, potom $\alpha_i(\bar{x}), \beta_i(\bar{x})$ jsou vždy konečné. Z předchozích podmínek dostaneme následující definice:

$$t_1 = \min_{i \in L_1}(a_i(\bar{x}) - \alpha_i(\bar{x})), \tag{9}$$

$$\text{kde } L_1 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x})\};$$

$$t_2 = \min_{i \in L_2}(b_i(\bar{x}) - \beta_i(\bar{x})), \tag{10}$$

$$\text{kde } L_2 \equiv \{i \in S \mid G_i(\bar{x}) \subseteq P(\bar{x})\};$$

$$t_3 = \min_{i \in L_3}(a_i(\bar{x}) - \beta_i(\bar{x})), \tag{11}$$

kde $L_3 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x}) \,\&\, i \in H(\bar{x})\}$.

$$t_4 = \min_{i \in L_4}(b_i(\bar{x}) - \alpha_i(\bar{x})), \tag{12}$$

kde $L_4 \equiv \{i \in S_1 \mid F_i(\bar{x}) \not\subset P(\bar{x}) \,\&\, G_i(\bar{x}) \subseteq P(\bar{x})\}$.

Nyní definujme $\tau \equiv \min(t_1, t_2, t_3, t_4)$, a $x(\tau)$ určíme jako novou horní mez. Jestliže $H(x(\tau)) \neq \emptyset$, začneme další iteraci s touto horní mezí, v opačném případě jsme nalezli řešení. Nyní již můžeme napsat ALGORITMUS 1.2 řešící celou naší úlohu.

ALGORITMUS 5.2

1. $\bar{y} := \bar{x}$

2. Jestliže $H(\bar{y}) = \emptyset$, potom $\bar{y}$ je největší prvek $M(\bar{x})$, KONEC

3. Najdi $P(\bar{y})$ pomocí ALGORITMU 5.1 (za $\bar{x}$ dosadíme $\bar{y}$). Jestliže $P(\bar{y}) = N$, potom $M(\bar{x}) = \emptyset$. KONEC

4. Najdi $x(t)$ pomocí (5) a t_1, t_2, t_3, t_4 pomocí (9)-(12) (za $\bar{x}$ dosadíme $\bar{y}$)

5. τ:=min(t_1, t_2, t_3, t_4), $\bar{y} := x(\tau)$, přejdi na **2.**

5.4 Důkaz Algoritmu 5.2

Věta 5.4.1 Nechť $\bar{y}$ je současná horní mez v Algoritmu 5.2, $\bar{y} \notin M(\bar{y}), t \in (0, \tau), \tilde{x} \leq \bar{y}$ a $\tilde{x} \not\leq x(t)$. Potom $\tilde{x} \notin M(\bar{y})$.

Důkaz:

Nejprve přepokládejme, že $\tilde{x}_p > x_p(t)$ pro nějaké $p \in A(\bar{y}) = \bigcup_{i \in H(\bar{y})} F_i(\bar{y})$. Potom $p \in F_k(\bar{y})$ pro nějaké $k \in H(\bar{y})$ a tedy $\bar{y}_p \geq \tilde{x}_p > x_p(t) \geq x_p(\tau)$. Potom musí existovat $\tilde{t} \in (0, t)$ takové, že $\tilde{x}_p = \bar{y}_p - \tilde{t} = x_p(\tilde{t})$ a $a_k(x(\tilde{t})) = a_k(\bar{y}) - \tilde{t}$. Předpokládejme, že $G_k(\bar{y}) \not\subset P(\bar{y})$ a tedy $b_k(x(\tilde{t})) = b_k(\bar{y}) \geq b_k(\tilde{x})$ a jelikož $F_k(\bar{y}) = F_k(x(\tilde{t})) = F_k(x(t))$, $a_k(x(\tilde{t})) = a_{kp} + \bar{y}_p - \tilde{t}$ a tím máme $a_k(\bar{y}) \geq a_k(\tilde{x}) \geq a_{kp} + \bar{y} - \tilde{t} = a_k(x(\tilde{t})) > b_k(\bar{y}) \geq b_k(\tilde{x})$ a tedy $\tilde{x} \in M(\bar{y})$.

Jestliže $G_k(\bar{y}) \subseteq P(\bar{y})$, potom $b_k(x(t)) = b_k(\bar{y}) - t$ a existuje $t^* \in (0,\tau)$ takový, že $b_k(\tilde{x}) = b_k(\bar{y}) - t^*$. A protože $p \in F_k(\bar{y}) = F_k(x(t^*))$ máme $a_k(\tilde{x}) \geq a_{kp} + x_p(t^*) = a_{kp} + \bar{y}_p - t^* = a_k(\bar{y}) - t^* > b_k(\bar{y}) - t^* = b_k(\tilde{x})$ a tedy opět $\tilde{x} \in M(\bar{y})$.

Nyní předpokládejme, že $p \in P(\bar{y}) \setminus A(\bar{y})$. Bez újmy na obecnosti předpokládejme, že $\tilde{x}_j \leq x_j(t)$ pro všechna $j \in A(\bar{y})$. Jinak, jak jsme již dokázali, platí $\tilde{x} \notin M(\bar{y})$. Jelikož platí $\tilde{x} \leq \bar{y}$, máme

$\tilde{x}_j \leq x_j(t)\ \forall j \in A(\bar{y})$

$\tilde{x}_j \leq x_j(t) = \bar{y}_j\ \forall j \in N \setminus P(\bar{y})$

$x_j(t) < \tilde{x}_j \leq \bar{y}_j$ pro $j \in N_1 \subseteq P(\bar{y}) \setminus A(\bar{y})$ a $\tilde{x}_j \leq x_j(t)$ pro zbývající $j \in (P(\bar{y}) \setminus A(\bar{y}) \setminus N_1$.

Nechť p je ten index z množiny N_1, který byl do $P(\bar{y})$ vložen jako první. Tedy existuje index $i_0 \in S_2 \setminus H(\bar{y})$ takový, že $p \notin F_{i_0}(\bar{y})$ a $p \in G_{i_0}(\bar{y})$ nebo $p \in F_{i_0}(\bar{y})$ a $p \notin G_{i_0}(\bar{y})$. Nebo existuje $i_0 \in S_1 \setminus H(\bar{y})$ takový, že $p \in F_{i_0}(\bar{y})$, $p \notin G_{i_0}(\bar{y})$ a $a_i(\bar{x}) = b_i(\bar{x})$.

Nejprve předpokládejme, že $p \in F_{i_0}(\bar{y})$ a $p \notin G_{i_0}(\bar{y})$ pro $i_0 \in S \setminus H(\bar{y})$. Platí $F_{i_0}(x(t)) = F_{i_0}(\bar{y})$ a $G_{i_0}(x(t)) = G_{i_0}(\bar{y})$. Tedy také platí, že $p \in F_{i_0}(x(t))$ a $p \notin G_{i_0}(x(t))$. Protože p je první index z N_1 vložený do $P(\bar{y})$ je zřejmé, že $j \notin G_{i_o}(x(t))$ pro všechny ostatní $j \in N_1$ a tedy máme

$$b_{i_0 j} + \tilde{x}_j < b_{i_0}(x(t)) \text{ pro všechny } j \in N_1$$

Protože pro všechna ostatní $\tilde{x}_j, j \in N \setminus N_1$ platí, že $\tilde{x}_j \leq x_j(t)$, dostáváme , že $b_{i_0}(\tilde{x}) \leq b_{i_0}(x(t)) = a_{i_0}(x(t)) = a_{i_0 p} + x_p(t)$, díky tomu že $p \in F_{i_0}(x(t))$.

A tedy nyní máme $b_{i_0}(\tilde{x}) \leq a_{i_0}(x(t)) = a_{i_0 p} + x_p(t) < a_{i_0 p} + \tilde{x}_p \leq a_{i_0}(\tilde{x})$ a tedy $\tilde{x} \notin M(\bar{y})$.

Budeme-li předpokládat, že $p \notin F_{i_0}(\bar{y})$ a $p \in G_{i_0}(\bar{y})$ a použijeme stejný postup, dostaneme $b_{i_0}(\tilde{x}) > a_{i_0}(\tilde{x})$. Vzhledem k tomu, že nyní $i_0 \in S_2 \setminus H$ dostáváme opět $\tilde{x} \notin M(\bar{y})$.

□

Věta 5.4.2 Pokud $P(\bar{y}) = N$, potom $M(\bar{y}) = \emptyset$.

Důkaz:

Předpokládejme, že $P(\bar{y}) = N$ a existuje prvek $\tilde{x} \in M(\bar{y})$. Potom můžeme najít t>0 takový, že $\tilde{x} \not\leq x(t)$. Pokud $P(\bar{y}) = N$, potom žádná z podmínek (1)-(4) nemůže být splněna pro žádné t, tedy $\tau = \infty$. Nyní použitím věty 5.4.1. dostáváme $\tilde{x} \notin M(\bar{y})$.

□

Pozn.: *Jako důsledek věty 5.4.1 dostaneme, že pokud* $M(\bar{x}) \neq \emptyset$ *a* $x(\tau)$ *je řešení obdržené* *v posledním kroku Algoritmu 5.2., potom alespoň jedna z proměnných* $x_j, j \in N$ *nebyla snížena. Skutečně, jestliže* $x_j(\tau) < \bar{x}_j$ *pro všechna* $j \in N$*, potom existuje* $\varepsilon > 0$ *takový, že* $x_j(\tau) + \varepsilon < \bar{x}_j$ *pro všechna* $j \in N$ *a* $(x_1(\tau) + \varepsilon, ..., x_n(\tau) + \varepsilon) \in M(\bar{x})$ *což je ve sporu s Větou 4.1.*

V každé iteraci Algoritmu 5.2 je nejméně jedna z proměnných $x_j, j \in N$ snížena a alespoň jedna z funkcí $a_i(x), b_i(x), i \in S$ a nikdy není znovu zvýšena. Nejméně jedna z proměnných nikdy není snížena, tedy pouze (*n*-1) proměnných v každé rovnici(resp. nerovnici) může být sníženo. Existuje maximálně *n*-1 různých hodnot $a_{ij} + \bar{y}_j$ a stejně maximálně *n*-1 různých hodnot $b_{ij} + \bar{y}_j$. Nazveme tyto hodnoty jako mezní hodnoty $a_i(\bar{y})$, resp. $b_i(\bar{y})$. V každé iteraci se nejméně jedna mezní hodnota sníží a dosáhne jiné mezní hodnoty. Předpokládejme, že v nějaké iteraci máme
$a_h(\bar{y}) > b_h(\bar{y}), a_i(\bar{y}) \le b_i(\bar{y})\ \forall i \in K \cap S_1$ a $a_i(\bar{y}) = b_i(\bar{y})\ \forall i \in K \cap S_2$, kde $K \subset S$ má *k* prvků. Každá z funkcí $a_i(\bar{y}), b_i(\bar{y}),\ i \in K \cup \{h\}$ má maximálně (*n*-1) různých mezních hodnot v bodě $\bar{y}$, které mohou být sníženy v další iteraci, tedy potřebujeme maximálně $2(n-1)(k+1)$ iterací, než bude muset být rovnost $a_h(x(\tau)) = b_h(x(\tau))$ dosažena v dalším kroku. Jestliže se *k* mění od 0 do *m*-1, je celkový počet mezních hodnot, které můžou být dosaženy v Algoritmu 5.2 nežli se zastaví

$$\sum_{k=0}^{m-1} (2(k+1)(n-1)+1) = (n-1)m(m-1)+m$$, tedy algoritmus je konečný.

6 Algoritmus řešící systém (max,+)-lineárních rovnic s různými proměnnými na obou stranách.

6.1 Formulace problému

Zaveďme následující značení:

$$N_1 = \{1,...,n_1\}, N_2 = \{1,...,n_2\}, S = \{1,...,m\}, R = (-\infty, \infty),$$
$$R^n = R \times ... \times R \text{ (n - krát)}, x = (x_1,...,x_{n_1}) \in R^{n_1}, y = (y_1,...,y_{n_2}) \in R^{n_2}$$
$$a_{ij}, b_{ik} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N}_1, \mathrm{k} \in \mathrm{N}_2 \text{ jsou dané,}$$

$$a_i(x) \equiv \max_{j \in N_1}(a_{ij} + x_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$
$$b_i(y) \equiv \max_{j \in N_2}(b_{ij} + y_j) \text{ pro } \forall \mathrm{i} \in \mathrm{S}$$

Budeme uvažovat následující systém (max,+)-lineárních rovnic:

$$a_i(x) = b_i(y) \quad \forall i \in S_2 , \tag{1}$$

Množinu všech dvojic (x,y) která řeší (1) označíme M. Obdobně jako v předchozích kapitolách zavedeme značení

$$M((\bar{x}, \bar{y})) = \{(x, y) \mid (x, y) \in M, x \leq \bar{x}, y \leq \bar{y}\} \tag{2}$$

Můžeme si všimnout, že jestliže $(x, y) \in M$, potom $(x(\alpha), y(\alpha)) \in M$ pro všechny $\alpha \in R$. Z toho plyne následující Lemma

Lemma 6.1.1 Máme dáno $\bar{x} \in R, \bar{y} \in R$. Potom $M \neq \emptyset \Leftrightarrow M((\bar{x}, \bar{y})) \neq \emptyset$.

Důkaz:

Z toho, že $M((\bar{x}, \bar{y})) \subseteq M$ rovnou plyne $M((\bar{x}, \bar{y})) \neq \emptyset \Rightarrow M \neq \emptyset$. Jestliže $M \neq \emptyset$ a $\tilde{x}, \tilde{y} \in M$, můžeme vždy nalézt takové $\alpha \in R$ aby $\tilde{x}(\alpha) \leq \bar{x}$ a zároveň $\tilde{y}(\alpha) \leq \bar{y}$. A jelikož $(\tilde{x}(\alpha), \tilde{y}(\alpha)) \in M$ je také $(\tilde{x}(\alpha), \tilde{y}(\alpha)) \in M((\bar{x}, \bar{y}))$. Máme tedy $M((\bar{x}, \bar{y})) \neq \emptyset$.

□

Opět se budeme snažit najít největší prvek množiny $M((\bar{x}, \bar{y}))$.

6.2 Převedení na soustavu rovnic

V této kapitole se pokusíme převést tuto úlohu na úlohu z kapitoly 1. Označme:

$$N = \{1,\dots,n_1+n_2\}$$
$$z \equiv (x,y) = (x_1,\dots,x_{n_1},y_1,\dots,y_{n_2}) \in R^{n_1+n_2}.$$
$$a'_{ij}, b'_{ij} \in R, \forall \mathrm{i} \in \mathrm{S}, \mathrm{j} \in \mathrm{N}, \text{kde}$$
$$a'_{ij} = a_{ij} \text{ pro } 1 \le j \le n_1 \quad b'_{ij} = b_{i,j-n_1} \text{ pro } n_1+1 \le j \le n_1+n_2$$
$$a'_{ij} = -K \text{ jinak} \qquad b'_{ij} = -K \text{ jinak}$$

, kde $K \in R$ je dané. Toto je původní úloha s maticemi A', B' a neznámou $y' \in R^{n_1+n_2}$. Množinu všech řešení soustavy odpovídající tomuto zadání označme M', resp. $M'(\bar{z})$. Je třeba pouze dokázat, že existuje takové K, že $z \in M \Leftrightarrow z \in M'$, resp $z \in M(\bar{z}) \Leftrightarrow z \in M'(\bar{z})$.

Lemma 6.2.1: Nechť $K < \min(\min_{i\in S, j\in N_1}(a_{ij}), \min_{i\in S, k\in N_2}(b_{ik}))$, potom

(i) Neexistuje $i \in S, n_1 < j \le n_1+n_2, z \in M'$ takové, že $j \in F'_i(z)$.

(ii) Neexistuje $i \in S, 1 \le j \le n_1, z \in M'$ takové, že $j \in G'_i(z)$.

Důkaz:

(i) Nechť existuje $i \in S, n_1 < j \le n_1+n_2, z \in M'$ takové, že $j \in F'_i(z)$. Potom $a'_i(z) = K + z_j < b'_{ij} + z_j \le b'_i(z)$ což je spor s $z \in M'$.

(ii) Stejně jako (i).

□

Věta 6.2.1: Nechť $K < \min(\min_{i\in S, j\in N_1}(a_{ij}), \min_{i\in S, k\in N_2}(b_{ik}))$, potom pro všechna $z \in R^{n_1+n_2}$ platí $z \in M \Leftrightarrow z \in M'$.

Důkaz:

Nechť $z \in M'$. Dle předchozího tvrzení neexistuje $i \in S, n_1 < j \le n_1+n_2$ takové, že $j \in F'_i(z)$ a neexistuje $i \in S, 1 \le j \le n_1$ takové, že $j \in G'_i(z)$. Tedy pro všechny $i \in S$ existuje $1 \le j_1 \le n_1$ a $n_1 < j_2 \le n_1+n_2$ takové, že $j_1 \in F'_i(z)$ a $j_2 \in G'_i(z)$. Nyní máme

$$a_i(x) = a_{ij_1} + x_{j_1} = a'_{ij_1} + z_{j_1} = a'_i(z) = b'_i(z) = b'_{ij_2} + z_{j_2} = b_{i,j_2-n_1} + y_{j_2-n_1} = b_i(y)$$

a z toho plyne, že $z \in M$.

Nechť tedy $z \in M$. Pro všechna $i \in S$ a

1, pro $1 \le j' \le n_1$ platí $a_i(x) \ge a_{ij'} + x_{j'} = a'_{ij'} + z_{j'} > K + z_{j'} = b'_{ij'} + z_{j'}$

2, pro $n_1 < j' \le n_1+n_2$ platí $a_i(x) = b_i(y) \ge b_{i,j-n_1} + y_{j-n_1} = b'_{ij} + z_j$.

Tedy pro všechna $j \in N$ platí $a_i(x) \ge b'_{ij} + z_j$. A tedy $a_i(x) \ge b'_i(z)$. Protože $z \in M$ platí, že pro všechna $i \in S$ existuje $j \in N_2$, takové, že $a_i(x) = b_i(y) = b_{ij} + y_j = b'_{i,n_1+j} + z_{n_1+j}$. A tedy $a_i(x) = b_i'(z)$. Obdobně můžeme dokázat $b_i(y) = a_i'(z)$.

Tím jsme získali $a_i'(z) = b_i(y) = a_i(x) = b_i'(z)$. A tedy $z \in M'$.

□

Pozn.: *Tvrzení* $z \in M(\bar{z}) \Leftrightarrow z \in M'(\bar{z})$ *je zřejmé z* $z \in M \Leftrightarrow z \in M'$.

Pozn.: *Převedením na základní úlohu jsme zároveň dokázali existenci největšího prvku množiny* $M((\bar{x}, \bar{y}))$.

7 Algoritmus řešící systém (max,+)-lineárních rovnic s koeficienty -∞.

7.1 Formulace problému

V této kapitole budeme uvažovat variaci původní úlohy. Umožníme, aby koeficienty a_{ij}, b_{ij} byly z intervalu $< -\infty, \infty)$ a budeme zkoumat, jak bude třeba změnit algoritmus a jak se změní množina všech řešení.

V předchozích dvou kapitolách jsme používali konstantu *K*, která vlastně nahrazovala nekonečno. Byly to speciální případy, kdy místo mínus nekonečna bylo možno uvažovat dostatečně malou konstantu a přitom získat množinu všech řešení. Nyní budeme zkoumat obecný případ.

Pozn.: *Hodnota -∞ v* a_{ij} *vlastně znamená, že proměnná j nemá žádný vliv na rovnost i.*

Pozn.: *Pro* a_{ij}, b_{ij} *neuvažujeme hodnotu +∞, protože v tom případě by rovnost byla splněna triviálně, pokud by +∞ bylo na obou stranách rovnosti, nebo by rovnost naopak byla nesplnitelná pro libovolné* $x \in R^N$ *, pokud by +∞ bylo pouze na jedné straně rovnosti.*

Pozn.: *Při programování musíme buď implementovat počítání s -∞ (-∞+c=-∞ pro* $c \in R$*), nebo tyto prvky označit a při počítání* $a_i(x), \text{resp}\, b_i(x)$ *je vůbec neuvažovat.*

Pro algoritmus 1.1 musíme být v prve řadě schopni spočítat $a_i(x), \text{resp}\, b_i(x)$. Zde je zřejmé, že pokud by $\forall j \in N$ bylo $a_{ij} = -\infty$, potom $a_i(x) = -\infty$ pro libovolné *x* a rovnost *i* by tedy platila pouze tehdy, pokud by pro $\forall j \in N$ platilo $b_{ij} = -\infty$. V tom případě by rovnost platila triviálně. Budeme tedy uvažovat pouze takové úlohy, kde pro $\forall i \in S$ existuje $j_1, j_2 \in N$ takové, že $a_{ij_1} \in R$ a $b_{ij_2} \in R$. V tom případě tedy platí, že pro všechna $i \in S$ a $x \in R^n$ je $a_i(x) \in R$ a $b_i(x) \in R$.

Dále si uvědomme, že v tomto případě také pokud $j \in F_i(x), \text{resp.}\, j \in G_i(x)$ potom $a_{ij} \in R, \text{resp.}\, b_{ij} \in R$.

Funkce $a_i(x), b_i(x)$ zůstávají +homogenní. Díky tomu také platí tvrzení, že pro všechna $\bar{x} \in R^n$ platí $M \neq \emptyset \Leftrightarrow M(\bar{x})$.

Věta 7.1.1: Nechť $\bar{x} \in R^n$, a $M(\bar{x})$ je neprázdná. Potom existuje největší prvek množiny $M(\bar{x})$.

Důkaz: V tomto důkazu budeme vycházet z toho, že pokud $a_{ij}, b_{ij} \in R$ potom $M(\bar{x})$ má největší prvek.

Idea důkazu je následující:

Uvažujme libovolné $x^1, x^2 \in M(\bar{x})$, $x^1 \neq x^2$. Vytvoříme vhodnou úlohu s konečnými koeficienty, takovou aby $x^1, x^2 \in M'(\bar{x})$. $M'(\bar{x})$ není prázdná a tedy má největší prvek. Tato množina má největší prvek. Úlohu vytvoříme tak, aby tento prvek byl také prvkem $M(\bar{x})$. Tedy pro dva libovolné různé $x^1, x^2 \in M(\bar{x})$ existuje $x^{\max} \geq x^1$ a zároveň $x^{\max} \geq x^2$. Protože $M(\bar{x})$ je shora omezena prvkem $\bar{x}$ existuje největší prvek množiny $M(\bar{x})$.

Nejprve nalezneme takové L aby pro něj platilo:

pro všechna $i \in S, j \in N$ takové, že $a_{ij} = -\infty$ platí $L < a_i(x^1) - x_j^1$ a $L < a_i(x^2) - x_j^2$.

pro všechna $i \in S, j \in N$ takové, že $b_{ij} = -\infty$ platí $L < b_i(x^1) - x_j^1$ a $L < b_i(x^2) - x_j^2$.

Díky tomu, že všechny tyto hodnoty jsou konečné, L existuje a je konečné.
Dále definujme $K \equiv L - \max_{j \in N}(\bar{x}_j - x_j^1)$. Tedy $K<L$.

Nyní položme $\begin{array}{l} a'_{ij} = a_{ij} \text{ pokud } a_{ij} \in R \\ a'_{ij} = K \text{ jinak} \end{array}$ a $\begin{array}{l} b'_{ij} = b_{ij} \text{ pokud } b_{ij} \in R \\ b'_{ij} = K \text{ jinak} \end{array}$.

Tím jsme vytvořili úlohu s konečnými koeficienty. Množinu všech jejích řešení menších nežli $\bar{x}$ označme $M'(\bar{x})$. Můžeme snadno ověřit, že $x^1, x^2 \in M'(\bar{x})$.

Vezměme libovolné $i \in S$. Pro všechna $i \in N$ platí

pokud $a_{ij} = -\infty$, potom $a'_{ij} + x_j^1 = K + x_j^1 < L + x_j^1 < a_i(x^1) - x_j^1 + x_j^1 = a_i(x^1)$ a

pokud $a_{ij} \in R$, potom $a'_{ij} + x_j^1 = a_{ij} + x_j^1 \leq a_i(x^1)$.

Tedy $a'_i(x^1) \leq a_i(x^1)$. Také existuje $j \in N, j \in F_i(x^1), a_{ij} \in R$. Potom $a_i(x^1) = a_{ij} + x_j^1 = a'_{ij} + x_j^1 \leq a'_i(x^1)$. Tedy $a'_i(x^1) = a_i(x^1)$.

To samé dokážeme pro $b_i(x^1)$. Máme $a'_i(x^1) = a_i(x^1) = b_i(x^1) = b'_i(x^1)$ a tedy $x^1 \in M'(\bar{x})$.
Stejně dokážeme $x^2 \in M'(\bar{x})$.

Množina $M'(\bar{x})$ má největší prvek. Označme ho $x^{\max}$. Musíme dokázat, že $x^{\max} \in M(\bar{x})$.
Pro všechna $i \in S$ platí

pokud $a_{ij} \in R$ potom $a'_{ij} + x_j^{\max} = a_{ij} + x_j^{\max} \leq a_i(x^{\max})$

pokud $a_{ij} = -\infty$ potom $a'_{ij} + x_j^{\max} = K + x_j^{\max} \leq K + \bar{x}_j = L - \max_{k \in N}(\bar{x}_k - x_k^1) + \bar{x}_j \leq$

$$\leq L - \bar{x}_j + x_j^1 + \bar{x}_j = L + x_j^1 < a_i(x^1) - x_j^1 + x_j^1 = a_i(x^1) \leq a_i(x^{\max}).$$

A tedy $a_i'(x^{\max}) \le a_i(x^{\max})$. Také existuje $j \in N, j \in F_i(x^{\max}), a_{ij} \in R$. Potom $a_i(x^{\max}) = a_{ij} + x_j^{\max} = a_{ij}' + x_j^{\max} \le a_i'(x^{\max})$. Tedy $a_i'(x^{\max}) = a_i(x^{\max})$. Stejně dokážeme, že $b_i'(x^{\max}) = b_i(x^{\max})$.

Tím jsme dokázali, že pro všechna $i \in S$ platí $a_i(x^{\max}) = a_i'(x^{\max}) = b_i'(x^{\max}) = b_i(x^{\max})$ a tedy $x^{\max} \in M(\bar{x})$.

□

7.1 Upravení původního algoritmu

Díky tomu, že pokud $j \in F_i(x), \text{resp.}\, j \in G_i(x)$ potom $a_{ij} \in R, \text{resp.}\, b_{ij} \in R$, je zřejmé, že algoritmus 1.1 bude fungovat správně i na tuto úlohu. Problém nastane teprve v okamžiku, kdy budeme chtít nalézt τ o které máme snížit proměnné $x_j, j \in P(\bar{x})$. Připomeňme si jak se τ spočítá.

Budeme zvyšovat t, dokud nenastane jedna z následujících podmínek.

(1) $F_i(x(t)) \ne F_i(\bar{x})$, tj. $a_i(x(t)) = \alpha_i(\bar{x}) \equiv \max\limits_{j \in N \setminus P(\bar{x})} (a_{ij} + \bar{x})$ pro nějaké $i \in S$ takové, že $F_i(\bar{x}) \subseteq P(\bar{x})$;

(2) $G_i(x(t)) \ne G_i(\bar{x})$, tj. $b_i(x(t)) = \beta_i(\bar{x}) \equiv \max\limits_{j \in N \setminus P(\bar{x})} (b_{ij} + \bar{x})$ pro nějaké $i \in S$ takové, že $G_i(\bar{x}) \subseteq P(\bar{x})$;

(3) $H(x(t)) \ne H(\bar{x})$, tj. $a_i(x(t)) = \beta_i(\bar{x})$ pro nějaké $i \in H(\bar{x})$.

Budeme definovat proměnné t_1, t_2, t_3 jako hodnoty, ve kterých nastaly předchozí podmínky. Protože $a_{ij}, b_{ij} \in <-\infty, \infty)$ může některé z t_1, t_2, t_3 dosáhnout hodnoty ∞. Tato situace nastane pokud:

(1) pro všechna $i \in L_1, j \in N$ platí $a_{ij} \in R \Rightarrow j \in P(\bar{x})$, kde $L_1 \equiv \{i \in S \mid F_i(\bar{x}) \subseteq P(\bar{x})\}$;

(2) pro všechna $i \in L_2, j \in N$ platí $b_{ij} \in R \Rightarrow j \in P(\bar{x})$, kde $L_2 \equiv \{i \in S \mid G_i(\bar{x}) \subseteq P(\bar{x})\}$;

(3) pro všechna $i \in L_3, j \in N$ platí $b_{ij} \in R \Rightarrow j \in P(\bar{x})$, kde $L_3 \equiv \{i \in H(\bar{x}) \mid F_i(\bar{x}) \subseteq P(\bar{x})\}$.

V okamžiku, kdy $\tau \equiv \min(t_1, t_2, t_3) = \infty$, je zřejmé, že můžeme proměnné $x_j, j \in P(\bar{x})$ libovolně snižovat, aniž bychom měli možnost se někde zastavit. V tomto případě je $M(\bar{x}) = \emptyset$.

Můžeme tedy použít Algoritmus 1.2 i na řešení naší úlohy, pouze s tím rozdílem, že kontrolujeme, zda-li $\tau \equiv \min(t_1, t_2, t_3) = \infty$. V tom případě úloha nemá řešení.

Pro důkaz správnosti kroku algoritmu předpokládejme tedy následující:

(1) Pro $\forall i \in S$ existuje $j_1, j_2 \in N$ takové, že $a_{ij_1} \in R$ a $b_{ij_2} \in R$.

Díky tomu pro všechna $i \in S, x \in R^N$ platí

(i) $a_i(x) \in R, b_i(x) \in R$.

(ii) pokud $j \in F_i(x)$, resp. $j \in G_i(x)$ potom $a_{ij} \in R$, resp. $b_{ij} \in R$.

(2) $\tau \equiv \min(t_1, t_2, t_3) \neq \infty$

V tomto případě je důkaz správnosti kroku i konečnosti algoritmu 1.2, uvedený v [2], platný i pro důkaz algoritmu naší úlohy.

Pozn.: *Úlohy řešené v předchozích dvou kapitolách je tady možné řešit také pomocí tohoto algoritmu, kdy konstantu K nahradíme v případě kapitoly 5 hodnotou ∞, v kapitole 6 hodnotou -∞.*

8 Uživatelská dokumentace Optim_MP

8.1 Program Optim_MP.exe

Optim_MP je program pro počítání algoritmů uvedených v této diplomové práci. Název je odvozen od Optimalizace Max-Plus úloh. Program Optim_MP je implementován s důrazem na názornost uvedených algoritmů. Výsledkem je přesný postup programu, odpovídající uvedeným algoritmům.

Optim_MP se ovládá z příkazové řádky. Zadání dostává ze standardního vstupu nebo častěji ze souboru.

Syntaxe spuštění programu je následující:

```
Optim_MP.exe [-log] [-f filename] [-out filename]
```

`-log` Parametrem `log` se zapíná logování průběhu výpočtu. Není-li zadáno, je výstupem pouze výsledný vektor nebo popis případné chyby výpočtu.

`-f filename` Určení vstupního souboru se zadáním úlohy. Není-li zadán, očekává zadání ze standartního vstupu.

`-out filename` Určení souboru pro výpis průběhu řešení a výsledku. Není-li zadáno provádí se výpis na obrazovku.

Soubor pro zadání úlohy je rozdělen na několik samostatných častí. Každá část začíná hlavičkou – řádka obsahující pouze název části v hranatých závorkách a končí začátkem další části nebo koncem souboru. Na pořadí jednotlivých části nezáleží. Všechna desetinná čísla jsou zadávána s tečkou jako oddělovačem desetinných míst. Pro zadávání vektorů a matic se jako oddělovač jednotlivých hodnot používá čárka.

8.1.1 Část `[MAIN]`

Hlavní část zadání úlohy. Obsahuje parametry určující velikost úlohy (`n`,`m`), algoritmus (`alg`), který se má použít a případně přesnost úlohy ε (`eps`). Parametry jsou zapsány každý na samostatné řádce ve tvaru `název_parametru=xxx`. Na pořadí jednotlivých parametrů nezáleží.

Algoritmy jsou určovány číslem použitým v této diplomové práci. Tedy například ro algoritmus 1.2 použijeme parametr `alg=1_2`.

Hlavní část úlohy tedy může vypadat následovně:

```
[MAIN]
m=2
n=5
alg=3_5
eps=0.001
```

8.1.2 Část `[BOUND]`

Tato část obsahuje pouze dva parametry určující mezení množiny M - $\overline{x}, \underline{x}$. Značí se `maxx` a `minx`. Pro algoritmus 1.2 je parametr `minx` nepovinný.

```
[BOUND]
maxx=10,10,10,10,10
minx=-10,-10,-10,-10,-10
```

8.1.3 Část `[MATRIX A]`, `[MATRIX B]`

Tyto části určují matici A, resp. matici B.

V každé části musí být `m x n` hodnot oddělených čárkou nebo novou řádkou. Hodnoty jsou zapsány v pořadí $a_{11}, a_{12}, a_{13}, ..., a_{1n}, a_{21}, ..., a_{mn}$. Pro přehlednost doporučují zapisovat jako `m` řádků, na každé `n` hodnot.

```
[MATRIX A]
1, 5, 7, 3, 4
3, 5, 7, 4, 2
[MATRIX B]
5, 3, 2, 6, 7
2, 3, 4, 6, 3
```

8.1.4 Část `[FUNC]`

Zde jsou nadefinovány jednotlivé funkce. Tato část není povinná pro algoritmy 1.2, 3.1 protože ty pouze hledají největší prvek na množině.

Pro každé $j \in J$ je zde jeden řádek ve tvaru `Fj=definice`. Definice funkce se skládá z několika parametrů oddělených středníkem.

První parametr je typ funkce. Zde jsou vypsány typy, které je možné použít.

0	`FCE_MONO_ROST`	-obecná rostoucí funkce na intervalu $<\underline{x}_j, \overline{x}_j>$
1	`FCE_MONO_KLES`	-obecná klesající funkce na intervalu $<\underline{x}_j, \overline{x}_j>$
2	`FCE_UNI_MAX`	-obecná rostoucí funkce na intervalu $<\underline{x}_j, sx>$ a klesající na $<sx, \overline{x}_j>$
3	`FCE_UNI_MIN`	-obecná rostoucí funkce na intervalu $<\underline{x}_j, sx>$ a klesající na $<sx, \overline{x}_j>$
4	`FCE_LIN`	-lineární funkce
5	`FCE_LINLOM`	-lineární lomená funkce

Další parametry se liší v závislosti na typu funkce

FCE_MONO_ROST, FCE_MONO_KLES

Druhý parametr je zápis funkce, třetí parametr je funkce k ní inverní na intervalu $<\underline{x}_j, \overline{x}_j>$. Například `0;x^2;sqrt(x);` pokud platí $\underline{x}_j, \overline{x}_j \subseteq R^+$.

FCE_UNI_MAX, FCE_UNI_MIN

Zde jsou následující parametry:

- zápis funkce
- funkce k ní inverzní na intervalu $<\underline{x}_j, sx>$
- funkce k ní inverzní na intervalu $<sx, \overline{x}_j>$
- sx

Například 0;x^2;-sqrt(-x); sqrt(x);0;

FCE_LIN

Zde jsou pouze dva parametry `k` a `d`. Tedy funkce `3x-7` se zapíše jako 4;3;-7;

FCE_LINLOM

Zde jsou parametry `k1, d1, k2, d2`. Tedy funkce `5x-2` na pro `x<1 a -3x+6` pro `x>1` se zapíše jako `5;5;-2;-3;6;`

Nyní můžeme zapsat libovolnou úlohu pro Optim_MP. Jako příklad použijme příklad z kapitoly 3.6

Příklad:

$A = (0,-10,-10) \quad B = (-10,-10,0)$

$\overline{x} = (10,10,10) \qquad f_1(x_1) = x_1$

$\underline{x} = (0,0,0) \qquad f_2(x_2) = 2x_2 - 10$

$\qquad f_3(x_3) = -x_3 + 10$

.

```
[MAIN]
n=3
m=1
alg=3_5
[BOUND]
maxx=10,10,10
minx=0,0,0
[MATRIX A]
0,-10,-10
[MATRIX B]
-10,-10,0
```

```
[FUNC]
F1=4;1;0;
F2=4;2;-10;
F3=4;-1;10;
```

Pokud tuto úlohu spustíme v programu Optim_MP dostaneme následující postup řešení.

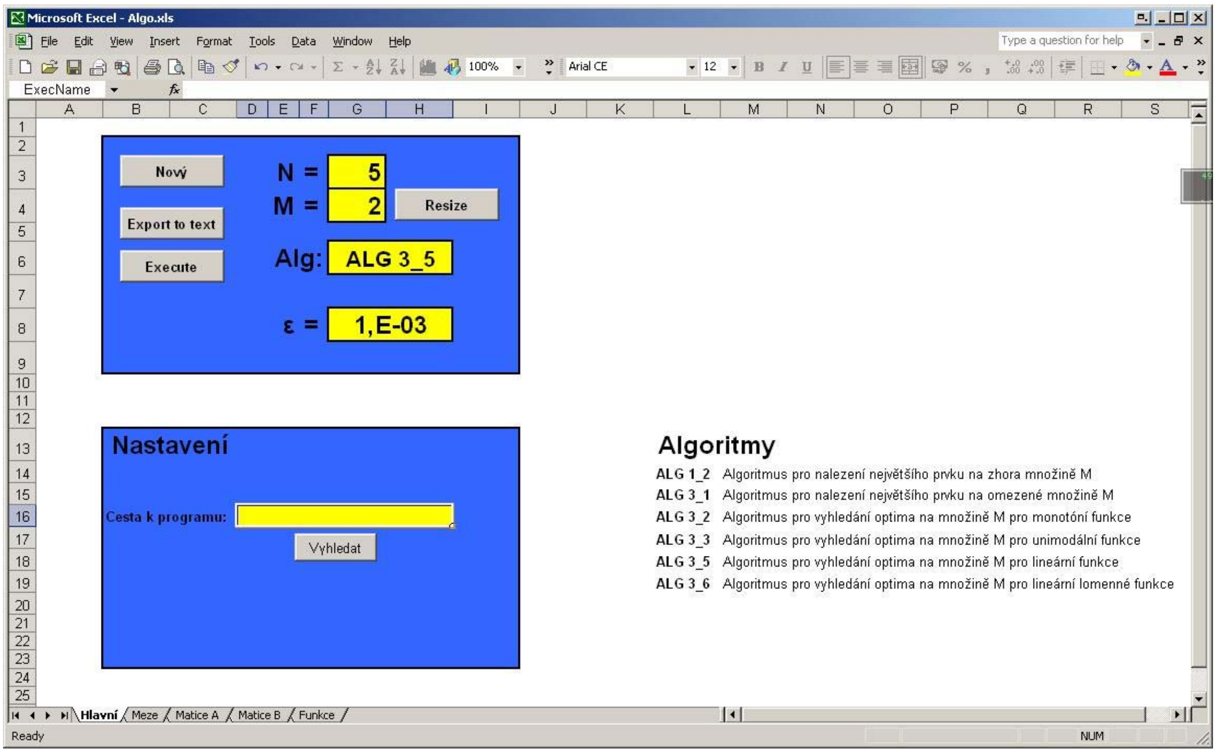
```
******************
  ALGORITMUS 3.5
******************

N = 3
M = 1
Matice A:
       0  -10  -10

Matice B:
     -10  -10    0

Maxx= (10,10,10)
Minx= (0,0,0)
J= { 1,2,3,}
    F(x1)=1x+0
    F(x2)=2x-10
    F(x3)=-1x+10

KROK 1
Spouštím Algoritmus 3.1
  Maxx = (10,10,10)

KROK 2
  J(y) = { 1,2,}

KROK 3-4
  Spouštím algoritmus 3.4 pro výpočet P
  P = { 1,2,3,}
    Aktivní t4
  J' = { 1,}
  Spouštím algoritmus 3.4 pro výpočet P'
  P' = { 1,3,}
    Aktivní t4

KROK 5
  Tau = 5
  Maxx = (5,10,5)
  H(y) = { }
  Skok na krok 2

KROK 2
  J(y) = { 2,}

KROK 3-4
  Spouštím algoritmus 3.4 pro výpočet P
  P = { 2,}
    Aktivní t4
  J' = { 2,}
  Spouštím algoritmus 3.4 pro výpočet P'
  P' = { 2,}
    Aktivní t4

KROK 5
  Tau = 2.5
  Maxx = (5,7.5,5)
  H(y) = { }
  Skok na krok 2

KROK 2
  J(y) = { 1,2,3,}
  Některá funkce z J(y) je klesající
Nalezeno řešení, konec algoritmu
Iterací:  3
Řešení: (5,7.5,5)
F(x)= 5
************************
```

8.2 Zjednodušené ovládání

Protože vytváření definičního souboru není přiliž intuitivní ani komfortní, vytvořil jsem sešit (*workbook*) v aplikaci Microsoft Excel sloužící k zadávání úlohy. Aplikaci Microsoft Excel jsem vybral s ohledem na komfort a snadnou práci s daty uspořádanými do tabulky.

K ovládání jsou využity makra v programovacím jazyku Visual Basic. Z toho důvodu je uživatel po otevření sešitu dotázán na povolení maker. Tato je třeba povolit, jinak by sešit správně nepracoval.

Sešit obsahuje několik listů. Základní list je nadepsán *Hlavní*. Slouží k ovládání aplikace a k nastavení základních parametrů úlohy



Nový – Zeptá se na velikost úlohy a vytvoří nové čisté formuláře pro zadání úlohy. Pokud není původní úloha uložena je uživatel dotázán, zda ji chce uložit. Po vytvoření nových formulářů je uživatel dotázán na jméno nového souboru.

Export to text – Uloží úlohu do textového souboru pro aplikaci Optim_MP.

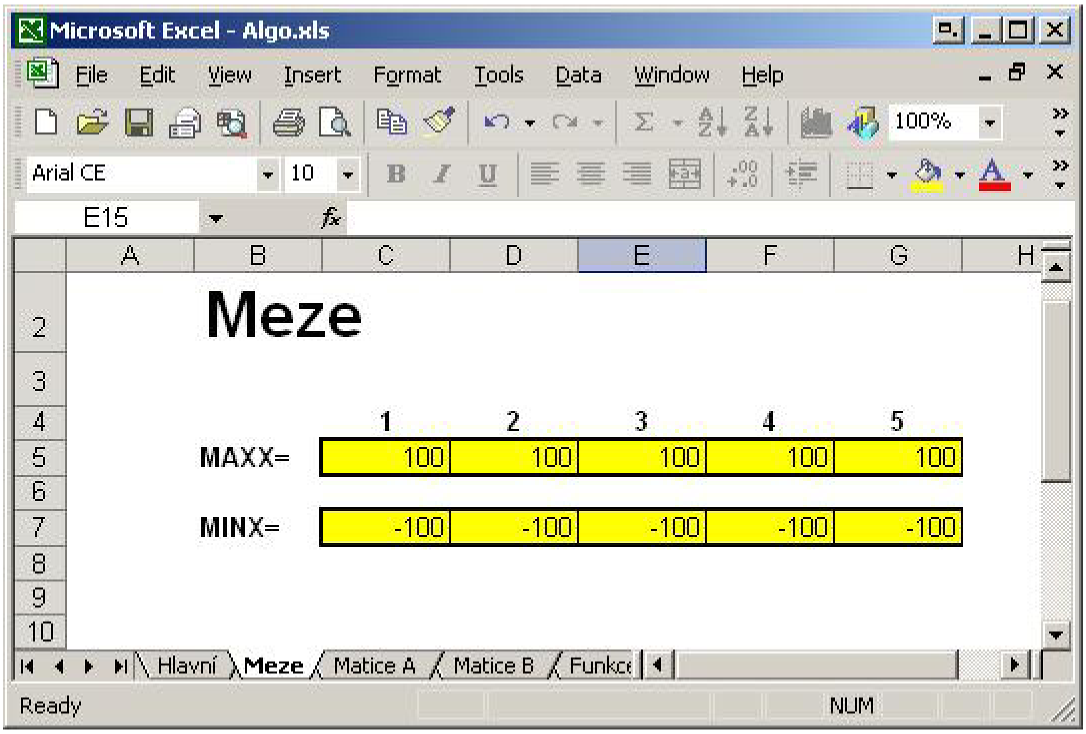
Execute – Spustí úlohu v aplikaci Optim_MP. Výsledek otevře v novém okně.

Resize – Změní velikost úlohy. Pokud je nová velikost větší než původní jsou všechna data zachována. Pokud se úloha zmenšuje jsou ořezána a smazána nadbytečná data.

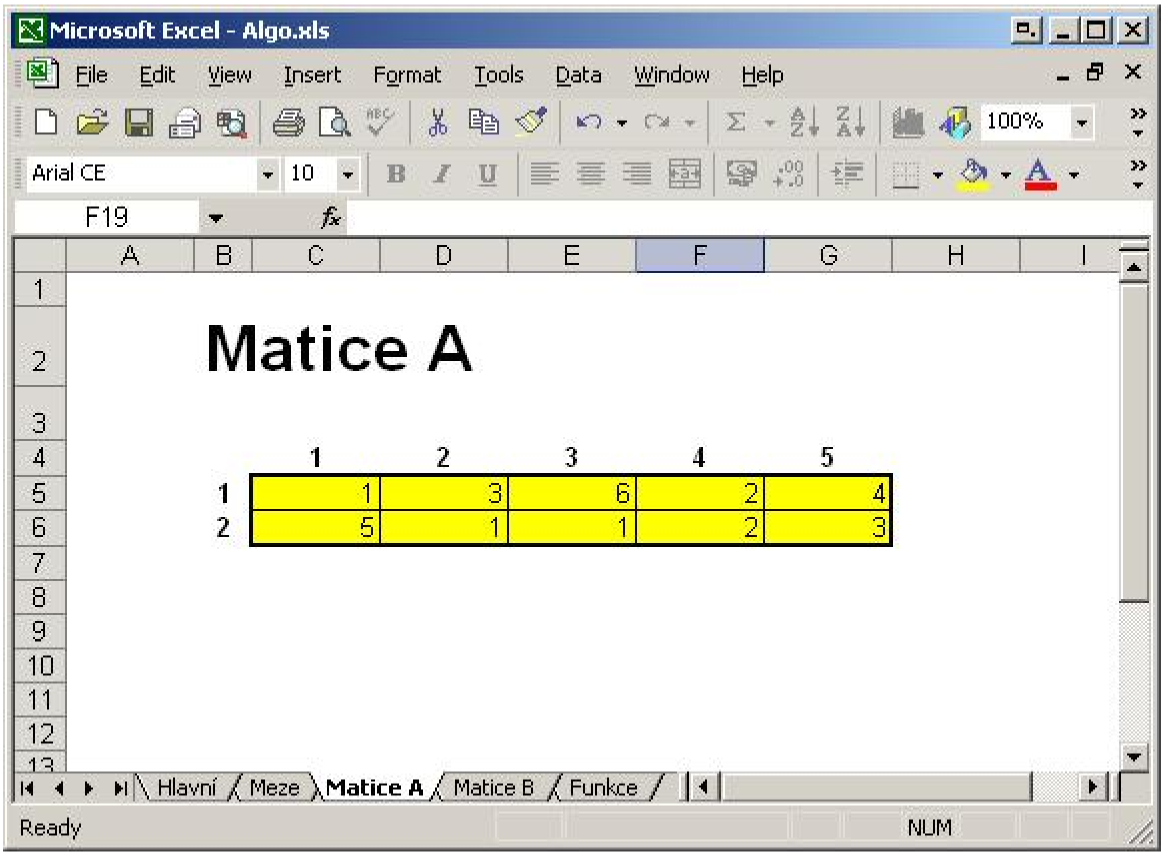
Alg – Výběr algoritmu, který bude použit pro výpočet

ε – Zadání přesnosti výpočtu přibližných algoritmů.

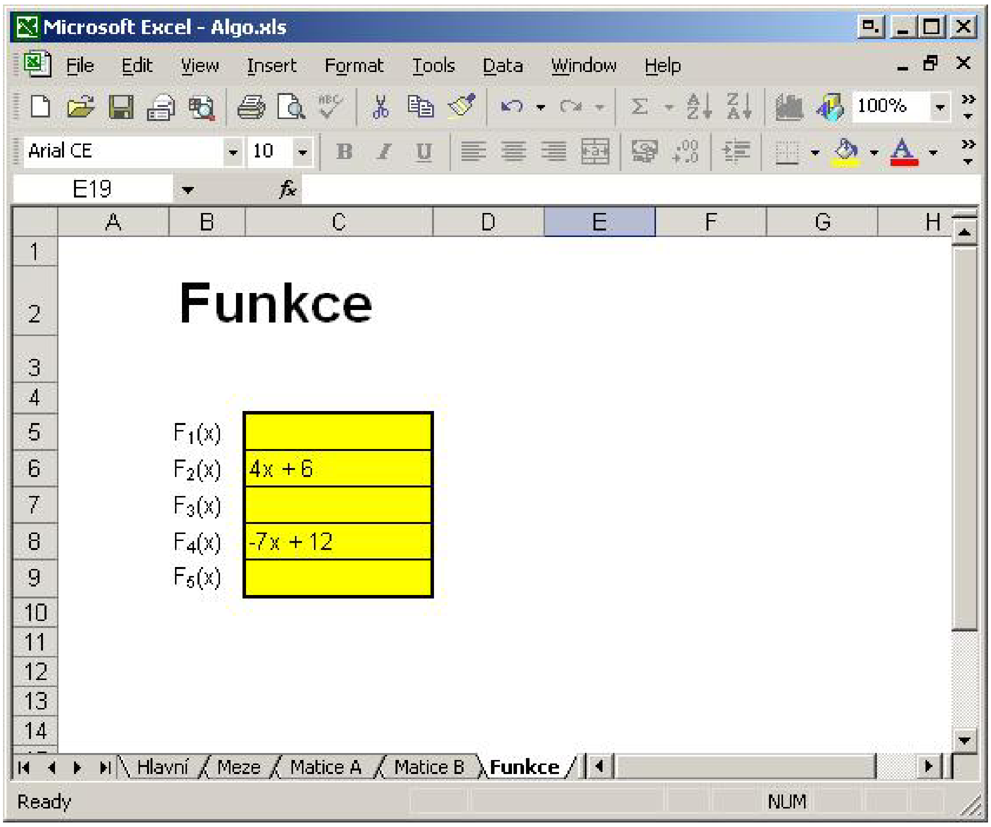
Cesta programu – Pro automatické spouštění úlohy tlačítkem Execute je zde třeba uvést cestu k programu Optim_MP. Tuto cest je možné určit pomocí tlačítka Vyhledat.



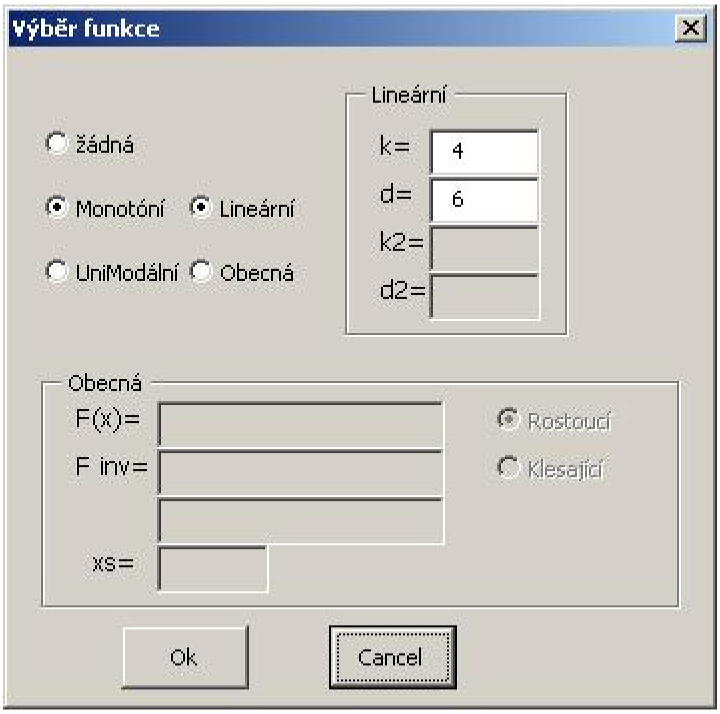
V listu *Meze* je možné zadat horní a dolní mez množiny M.



listy *Matice A* a *Matice B* slouží k zadávání hodnot matic A a B.



Pro zadáváni funkcí slouží poslední list *Funkce*. V tabulce jsou zobrazeny zadané funkce. Pro kliknutí na vybrané políčko se zobrazí dialog s možností výběru funkce.



Pro uložení a načtení úlohy je možné použít standardní funkce aplikace Excel *Otevřít*, *Uložit*, *Uložit jako*.

9 Programátorská dokumentace Optim_MP

Aplikace Optim_MP je naprogramována ve Visual C++. Je naprogramována ta, aby co nejpřesněji dodržovala algoritmy popsané v této diplomové práci.

Kód je psaný pokud možno názorně a srozumitelně. Detailnější informace jsou uvedeny v komentářích ve zdrojových kódech

Samotný program se skládá z následujících částí:

9.1 Struct.cpp, Struct.h

Modul obsahuje definice používaných datových typů a operace nad nimi.

`Vektor` **-** Třída implementující vektor a operace nad ním.

`Matice` **-** Třída implementující matici a operace nad ní.

`IndexSet` **-** Třída implementující množinu indexů.

`Uloha` – Třída implementující kompletní zadání úlohy. Dále obsahuje dočasné proměnné a mezivýpočty.

`PodUloha` – Třída implementující podúlohu – používaná algoritmem 3.3.

`SetPU` – Množina podúloh

`Fronta` – Fronta vektorů používaná v algoritmu 3.3 při hledání všech možných intervalů, na kterých můžeme pokračovat

9.2 Výpis.cpp, Výpis.h

Tento modul obsahuje jednoduché funkce pro výpis postupu a výsledků.

`void setoutput(FILE * output)` – nastaví soubor, do kterého je výpis prováděn. Standardně je to stdout.

`void print(const char *str)` – vypíše do logu zadaný řetězec.

`void printnl(const char *str)` – vypíše do logu prázdnou řádku a poté zadaný řetězec.

Následující funkce vypíšou do logu zadaný řetězec a formátovaný obsah dané struktury.

```
void print(const char *str,const double d)
void print(const char *str,vektor &v)
void print(const char *str,matice &m)
void print(const char *str,IndexSet &is)
```

9.3 cFunkce.cpp, cFunkce.h

Modul pro práci s funkcemi.

Pro vyčíslování obecných výrazů je použita knihovna *eval*. Tuto knihovnu jsem vytvořil před několika lety pro jiný projekt. Zpracovává a vyhodnocuje textově zadané funkce. Tato knihovna není při výpočtech efektivní a umí pouze omezené množství funkcí, ale pro účely předvedení těchto algoritmů dostačuje. Pro zefektivnění výpočtů stačí upravit tento modul, aby pracoval s jinou knihovnou.

9.4 algoritmy.cpp, algoritmy.h

Hlavní modul implementující jednotlivé algoritmy. Funkce jsou pojmenovány podle algoritmů které reprezentují.

9.5 Optim_MP.cpp

Modul implementující rozhraní aplikace. Pouze zpracuje textový vstup a spustí odpovídající algoritmus.

Odkazy:

[1] Cuninghame-Green, R.A., Zimmermaenn, K.: *Equation with Residual Functions*, Comment. Math. Univ. Carolinae 42(2001), 4, pp. 729-740

[2] Butkovič, P., Zimmermann, K. - *A Strongly Polynomial Algorithm for Solving Two-sided Linear Systems in Max-algebra*, Discrete Applied Math

[3] Cuninghame-Green, R. A.: *Minmax Algebra*, Lecture Notes in Economics and Math. Sys., 166, 1979, 258 stran, Springer Verlog

[4] Vorolejov N., N.: *Extremální algebra kladných matic*, Elektronische Informationsverarbeitung und Kybernetik, 1967, vol. 3, No 1, 1967, str. 39 - 71