

01LIP – Lineární programování

Stručný studijní text

Jan Bureš

Verze: srpen 2026

PŘEDMLUVA

Tento studijní text doprovází kurz lineárního programování 01LIP na FJFI ČVUT v Praze a přesně kopíruje jeho strukturu.

Obsah textu je do značné míry inspirován existujícími učebními texty a monografiemi, zejména pracemi *Lineární programování: Úvod pro informatiky* (Matoušek, 2006), *Lineární a nelineární programování* (Rohn, 1997), *Understanding and Using Linear Programming* (Matoušek and Gärtner, 2007), *Linear Programming: Foundations and Extensions* (Vanderbei, 2020), *Introduction to Linear Optimization* (Bertsimas and Tsitsiklis, 1997) a *Theory of Linear and Integer Programming* (Schrijver, 1986).

Text obsahuje také témata nad rámec zkoušky z tohoto předmětu.

Jan Bureš
Dallas, srpen 2026

OBSAH

Předmluva	iii
1 Formulace úlohy lineárního programování	1
1 Základy lineárního programování	2
1.1 Standardní a rovnicový tvar lineárního programu	2
1.2 Přípustná množina a optimální řešení	4
2 Převody mezi tvary	5
3 Celočíselné a smíšené celočíselné programování	7
3.1 Výpočetní složitost	7
3.2 LP-relaxace	8
4 Krátké srovnání s lineární algebrou	8
5 Jak vznikalo lineární programování	8
5.1 Leningrad, dnešní Petrohrad: od překližky k obecné metodě	9
5.2 Washington: když program znamenal plán	10
5.3 Setkání s von Neumannem: dualita vstupuje do amerického příběhu	10
5.4 První výpočty: 120 člověkodnů na jednu úlohu	10
5.5 Od praktického dřiče k teorii složitosti	11
6 Co všechno lze modelovat lineárním programem	12
6.1 Nejlevnější směs s předepsaným složením	12
6.2 Kolik dat projde sítí?	13
6.3 Vyrábět podle poptávky, nebo do zásoby?	14
6.4 Lze dvě skupiny bodů oddělit přímkou?	15
6.5 Příмка, která co nejlépe vystihuje měření	16
6.6 Největší kruh uvnitř mnohoúhelníku	17
2 Geometrie LP: konvexita a bážická řešení	19
1 Úvod do konvexity	20
1.1 Konvexní kombinace, konvexní a afinní obal	20
1.2 Nadroviny, poloprostory a mnohostěny	21
2 Bážická přípustná řešení	24
3 Základní věta LP a grafické řešení	29
1 Základní věta lineárního programování	30
2 Grafické řešení úloh	33
4 Simplexový algoritmus – základy metody	37
1 Úvodní příklad bez potíží	38
2 Neomezenost úlohy a více optimálních řešení	41
3 Simplexové tabulky	43
4 Simplexová metoda – obecné základy	44
4.1 Kritérium optimality	44

4.2	Pivot – volba vstupující a vystupující proměnné	44
5	Simplexový algoritmus II: dvoufázová metoda a M-úloha	47
1	Výchozí bázecké přípustné řešení	48
2	Metoda velkého M	48
2.1	Příklady	50
3	Dvoufázová metoda (metoda umělé báze)	52
3.1	První fáze	52
3.2	Druhá fáze	53
3.3	Příklady	53
6	Degenerace, pivotovací pravidla a efektivita	57
1	Degenerace v simplexové metodě	58
1.1	Příklad degenerovaného počátečního bodu	58
2	Pivotovací pravidla a cyklení	59
2.1	Pravidla pro volbu vstupující proměnné	60
2.2	Anticyklická pravidla	61
3	Efektivita simplexové metody	62
3.1	Kleevo–Mintyho příklad	62
3.2	Teoretické hranice	63
7	Dualita lineárního programování I	65
1	Motivace a konstrukce duální úlohy	66
1.1	Standardní primálně-duální dvojice	67
1.2	Oecná pravidla dualizace	67
1.3	Jiný pohled: Lagrangeovské odvození	68
2	Slabá a silná dualita	69
2.1	Podmínky komplementarity	70
8	Dualita lineárního programování II	73
1	Důkaz silné duality z optimální báze	74
2	Farkasovo lemma a druhý důkaz silné duality	75
2.1	Bod, kužel a oddělující nadrovina	75
2.2	Algebraická formulace a důkaz	75
2.3	Silná dualita z Farkasova lemmatu	77
3	Stínové ceny a citlivost na pravé strany	78
4	Duální simplexová metoda	79
4.1	Odvození pivotového pravidla	80
4.2	Úplný číselný příklad	81
9	Dopravní problém	83
1	Slovní zadání a značení	84
2	Formulace dopravního problému jako LP	84
2.1	Kanonická (vyrovnaná) formulace	84
2.2	Nevyrovnaná varianta dopravního problému	85
3	Struktura a vlastnosti dopravního problému	86
3.1	Počet nezávislých omezení a bázecká řešení	86
3.2	Degenerace	89
4	Metoda MODI (metoda potenciálů)	90
4.1	Počáteční bázecké přípustné řešení	90
4.2	Potenciály a redukované náklady	90
4.3	Zlepšovací krok a cyklus v tabulce	92

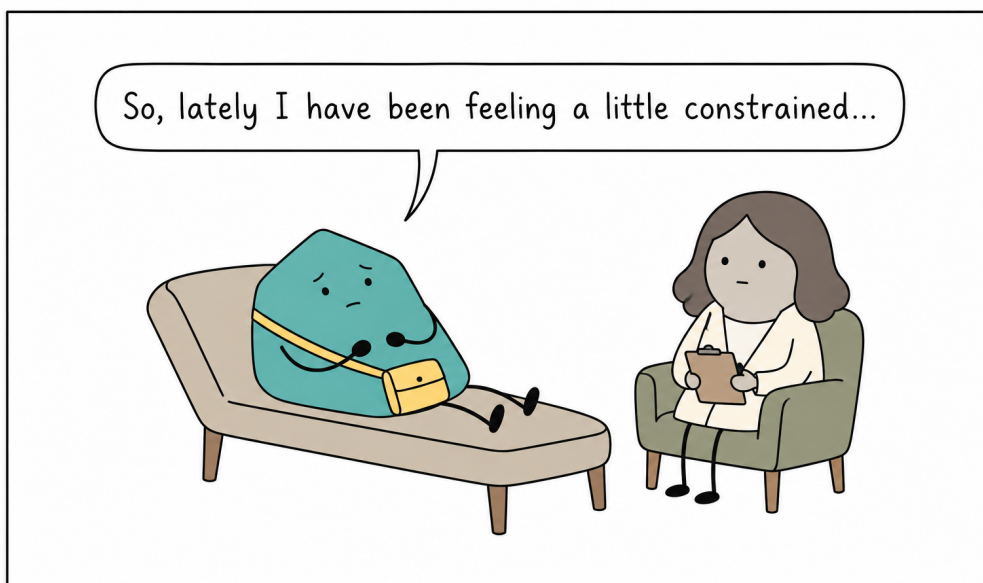
4.4	Shrnutí metody MODI	96
10	Maticové hry s nulovým součtem a věta o minimaxu	99
1	Maticová hra s nulovým součtem	100
2	Smíšené strategie a očekávaná výplata	103
2.1	Očekávaná výplata	104
3	Max–min, min–max a hodnota hry	104
3.1	Nejhorší případ pro danou strategii	105
3.2	Maximinové a minimaxové strategie, hodnota hry	105
4	Formulace maticové hry jako lineárního programu	107
4.1	LP pro sloupcového hráče (max–min)	107
4.2	LP pro řádkového hráče (min–max)	108
4.3	Dualita: každý hráč má svou úlohu	108
5	Krátký ilustrační příklad	110
11	Celočíselné programování: vybrané metody řešení	113
1	Krátké zopakování pojmů ILP/MILP	114
2	Dva základní modelovací příklady	115
3	LP-relaxace, větvení a řezy: obecné schéma	115
4	Metoda větví a mezí	116
4.1	Základní myšlenka	116
4.2	Schematický algoritmus	117
4.3	Jednoduchý ilustrativní příklad	118
5	Gomoryho řezy	119
5.1	Základní myšlenka	119
5.2	Gomoryho řez	120
5.3	Schematický algoritmus	122
5.4	Ilustrativní příklad Gomoryho řezů	123
	Literatura	127

KAPITOLA 1

FORMULACE ÚLOHY LINEÁRNÍHO PROGRAMOVÁNÍ

1	Základy lineárního programování	2
2	Převody mezi tvary	5
3	Celočíselné a smíšené celočíselné programování	7
4	Krátké srovnání s lineární algebrou	8
5	Jak vznikalo lineární programování	8
6	Co všechno lze modelovat lineárním programem	12

Zavedeme základní pojmy lineárního programování a ukážeme, jak převádět slovní zadání i různé zápisy úloh do jednotného tvaru. Vymezíme přípustná a optimální řešení, odlišíme celočíselné a smíšené celočíselné programování a přiblížíme vztah k lineární algebře i historii oboru. Na závěr sestavíme šest modelů, které ukážou šíři použití LP.



1 Základy lineárního programování

Na úvod připomeňme pro někoho možná překvapivý fakt: slovo *programování* zde neznamená psaní počítačového programu. V historickém významu označuje sestavení plánu či rozvrhu. Obor vznikl dříve, než byly počítače běžně dostupné. Dnes budeme *lineárním programem* (LP) rozumět optimalizační úlohu s lineární účelovou funkcí a konečně mnoha lineárními rovnicemi nebo nerovnicemi. Stejný obor se proto nazývá také *lineární optimalizace*. Základní teorii a české názvosloví lze porovnat například s texty Matoušek (2006) a Rohn (1997). Soustavný moderní výklad podávají Matoušek and Gärtner (2007).

Formální zápis LP je překvapivě jednoduchý, přesto jde o mimořádně užitečný aparát. Lineární model se hodí všude tam, kde lze podstatné vazby vyjádřit součty: kapacitami strojů a zdrojů, bilancemi materiálu či energie, rozpočtovými omezeními nebo minimálními požadavky na kvalitu. Rozhodovací proměnné popisují naše volby, omezení říkají, které volby jsou dovoleny, a účelová funkce rozhoduje, kterou z nich považujeme za nejlepší. Právě tento slovní popis nyní převedeme do maticového jazyka.

1.1 Standardní a rovnicový tvar lineárního programu

Nejprve ustálíme značení a dva tvary úlohy, mezi nimiž budeme v dalších kapitolách přecházet. Názvy těchto tvarů nejsou v literatuře jednotné, proto vždy uvedeme nejen název, ale i přesný zápis.

Značení 1.1: Notace a konvence

Nechť $m, n \in \mathbb{N}$ jsou počty omezení a proměnných. Vektory chápeme jako sloupcové. Matice $A \in \mathbb{R}^{m \times n}$ má i -tý řádek $\mathbf{a}_i^T$ a prvek a_{ij} . Dále $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{b} \in \mathbb{R}^m$ a $\mathbf{c} \in \mathbb{R}^n$. Vektorové nerovnosti, například $\mathbf{x} \geq \mathbf{0}$, vždy chápeme po složkách. Horní index T značí transpozici.

Standardní maximalizační tvar

V tomto textu budeme standardním maximalizačním tvarem rozumět úlohu

$$\begin{aligned} &\text{maximalizovat} && \mathbf{c}^T \mathbf{x}, \\ &\text{při} && A\mathbf{x} \leq \mathbf{b}, \\ &&& \mathbf{x} \geq \mathbf{0}. \end{aligned} \tag{1.1}$$

Definice 1.2: Účelová funkce

Lineární funkci

$$\mathbf{c}^T \mathbf{x} = \sum_{j=1}^n c_j x_j$$

nazýváme *účelovou funkcí*. Proměnné x_j jsou *rozhodovací proměnné* a složky vektoru $\mathbf{c}$ jsou jejich koeficienty v účelové funkci.

V obecné optimalizaci může být účelová funkce téměř libovolná. V tomto kurzu si však – docela věrni jeho názvu – vystačíme s funkcemi lineárními. Jejich jednoduchost nám později umožní spojit výpočet s velmi názornou geometrií.

Po složkách má (1.1) tvar

$$\max \sum_{j=1}^n c_j x_j \quad \text{při} \quad \sum_{j=1}^n a_{ij} x_j \leq b_i \quad (i = 1, \dots, m), \quad x_j \geq 0 \quad (j = 1, \dots, n).$$

Podmínku nezápornosti je nutné uvést: bez ní by šlo o jinou úlohu.

Analogicky budeme standardním minimalizačním tvarem rozumět

$$\begin{aligned} &\text{minimalizovat} \quad \mathbf{c}^T \mathbf{x}, \\ &\text{při} \quad \mathbf{A} \mathbf{x} \geq \mathbf{b}, \\ &\quad \mathbf{x} \geq \mathbf{0}. \end{aligned}$$

Volba opačného směru nerovností je užitečná při konstrukci duální úlohy. Každou minimalizaci lze zároveň převést na maximalizaci změnou znaménka účelové funkce.

Poznámka 1.3: Volná proměnná

Proměnnou bez znaménkového omezení, $x_j \in \mathbb{R}$, nazýváme *volnou*. Nahradíme ji rozdílem $x_j = x_j^+ - x_j^-$, kde $x_j^+, x_j^- \geq 0$. Rozšířený model má o jednu proměnnou více. Zobrazení $(x_j^+, x_j^-) \mapsto x_j$ není prosté, ale jeho obraz pokrývá všechny původní hodnoty. Po návratu k původním proměnným proto dostaneme právě původní přípustná řešení. V čistém ILP musí být také x_j^+ a x_j^- celočíselné. Posun celočíselné proměnné o neceločíselnou dolní mez naproti tomu celočíselnost obecně nezachovává.

Pro simplexovou metodu bude výhodný rovnicový tvar. U nerovnosti typu $\leq$ zavedeme *přičítanou doplňkovou proměnnou* (angl. *slack variable*) $s_i \geq 0$,

$$\mathbf{a}_i^T \mathbf{x} \leq b_i \quad \Longleftrightarrow \quad \mathbf{a}_i^T \mathbf{x} + s_i = b_i,$$

a u nerovnosti typu $\geq$ *odečítanou doplňkovou proměnnou* (angl. *surplus variable*) $s_i \geq 0$,

$$\mathbf{a}_i^T \mathbf{x} \geq b_i \quad \Longleftrightarrow \quad \mathbf{a}_i^T \mathbf{x} - s_i = b_i.$$

Rovnicový tvar LP

Po rozkladu volných proměnných a zavedení doplňkových proměnných lze lineární program zapsat ve tvaru

$$\max \mathbf{c}^T \mathbf{x} \quad \text{při} \quad \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}. \quad (1.2)$$

Značení $\mathbf{A}$, $\mathbf{x}$, $\mathbf{c}$ zde již zahrnuje i nově zavedené sloupce, proměnné a nulové koeficienty v účelové funkci.

Je-li původní soustava rozdělena na bloky $\leq$, $=$ a $\geq$, lze tentýž převod zapsat podrobněji jako

$$\begin{aligned} \mathbf{A}_{\leq} \mathbf{x} + \mathbf{s}_{\leq} &= \mathbf{b}_{\leq}, \\ \mathbf{A}_{=} \mathbf{x} &= \mathbf{b}_{=}, \\ \mathbf{A}_{\geq} \mathbf{x} - \mathbf{s}_{\geq} &= \mathbf{b}_{\geq}, \end{aligned} \quad \mathbf{x}, \mathbf{s}_{\leq}, \mathbf{s}_{\geq} \geq \mathbf{0}.$$

Původní a rozšířený model jsou ekvivalentní v následujícím přesném smyslu: každému původnímu přípustnému bodu odpovídají jednoznačně určené doplňkové proměnné. Po případném rozkladu volných proměnných se původní bod získá lineárním zobrazením rozšířeného bodu. Hodnota účelové funkce se nezmění.

Proč se namáhat s rovnicovým tvarem? Rovnosti spolu s nezáporností umožní vybrat několik sloupců matice, příslušné proměnné dopočítat a všechny ostatní položit na nulu. Tato zdánlivě čistě algebraická konstrukce bude přesně odpovídat “rohům” přípustné množiny. Odtud už vede přímá cesta k bázeickým přípustným řešením a simplexové metodě.

Poznámka 1.4: Různé názvy tvarů

Někteří autoři nazývají rovnicový tvar *standardním* nebo *kanonickým*. Jiní používají stejná slova pro nerovnicový tvar. V této knize znamená *standardní tvar* zápis (1.1) a *rovnicový tvar* zápis (1.2). Přesný vzorec má vždy přednost před názvem.

1.2 Přípustná množina a optimální řešení

Při každé optimalizaci musíme nejprve říci, mezi jakými body vůbec smíme vybírat. U standardního tvaru tuto množinu přečteme přímo z omezení.

Definice 1.5: Přípustná množina a optimální řešení

Pro úlohu (1.1) je

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$$

přípustná množina. Bod $\mathbf{x} \in P$ je *přípustné řešení*. Přípustný bod $\mathbf{x}^*$ je *optimální řešení* maximalizační úlohy, jestliže

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{c}^T \mathbf{x}^* \quad \text{pro všechna } \mathbf{x} \in P.$$

Číslo $z^* = \mathbf{c}^T \mathbf{x}^*$ je *optimální hodnota*. Pro minimalizaci se směr poslední nerovnosti obrátí.

Poznámka 1.6: Triáda osudů lineárního programu

Maximalizační lineární program je právě v jedné z těchto situací:

- je *nepřípustný*, tedy $P = \emptyset$,
- je *neomezený shora*, tedy $P \neq \emptyset$ a $\sup_{\mathbf{x} \in P} \mathbf{c}^T \mathbf{x} = +\infty$,
- má konečnou optimální hodnotu a alespoň jedno optimální řešení.

U minimalizace se ve druhém bodě mluví o neomezenosti zdola. Skutečnost, že konečné supremum lineární funkce na neprázdném mnohostěnu je skutečně dosaženo, zaručí základní věta lineárního programování, kterou dokážeme později. Při první analýze konkrétní úlohy se proto vždy ptáme, který z těchto tří osudů nastal.

2 Převody mezi tvary

Slovní modely přicházejí v nejrůznějších podobách. Následující praktické recepty nám dovolí převést je do jednotného tvaru, s nímž se dobře pracuje v geometrii i v algoritmech. Každá úprava zachovává přípustná řešení a hodnoty účelové funkce po uvedeném zpětném převodu.

1. Minimalizace a maximalizace.

$$\min \mathbf{c}^T \mathbf{x} \iff \max (-\mathbf{c})^T \mathbf{x}.$$

Optimální body se nemění a optimální hodnota změní znaménko.

2. Směr nerovnosti a rovnost.

$$\mathbf{a}_i^T \mathbf{x} \geq b_i \iff (-\mathbf{a}_i)^T \mathbf{x} \leq -b_i.$$

Rovnost lze nahradit dvojicí opačných nerovností. V modelu ji však obvykle ponecháme jako rovnost, protože zdvojení řádku je zbytečné.

3. Rovnicový tvar. K nerovnosti typu $\leq$ přičteme a od nerovnosti typu $\geq$ odečteme nezápornou doplňkovou proměnnou, jak bylo uvedeno výše. Samotné zavedení odečítané doplňkové proměnné obecně nevytvoří výchozí přípustnou bázi. Tuto otázku řeší kapitola 5.

Přičítaná doplňková proměnná (angl. *slack variable*) měří nevyužitou rezervu, odečítaná doplňková proměnná (angl. *surplus variable*) pak přebytek nad požadovanou mezí. Nejde tedy jen o algebraický trik: v mnoha modelech mají tyto proměnné přímý praktický význam.

4. Volné proměnné a meze. Volnou proměnnou nahradíme rozdílem dvou nezáporných proměnných. Má-li proměnná meze $L_j \leq x_j \leq U_j$, zavedeme

$$x_j = L_j + y_j, \quad 0 \leq y_j \leq U_j - L_j.$$

Dolní mez se tím odstraní a horní mez zůstane jediným lineárním omezením.

5. Absolutní hodnota a maximum lineárních funkcí. Pro lineární funkci ℓ platí

$$|\ell(\mathbf{x})| \leq t \iff -t \leq \ell(\mathbf{x}) \leq t,$$

přičemž $t \geq 0$ plyne již z těchto dvou nerovností, je-li soustava přípustná. Minimalizaci

$$\min \sum_i w_i |\ell_i(\mathbf{x})|, \quad w_i \geq 0,$$

lze linearizovat pomocí $t_i \geq 0$ a omezení $-t_i \leq \ell_i(\mathbf{x}) \leq t_i$, přičemž se minimalizuje $\sum_i w_i t_i$. Pro $w_i > 0$ musí v optimu platit $t_i = |\ell_i(\mathbf{x})|$. Je-li $w_i = 0$, rovnost nemusí být v každém optimálním řešení splněna, ale t_i lze na absolutní hodnotu snížit bez změny účelové hodnoty.

Podobně

$$\min_{\mathbf{x}} \max_{k=1,\dots,q} \{\mathbf{a}_k^T \mathbf{x} + b_k\}$$

převeďme na minimalizaci t při omezeních $\mathbf{a}_k^T \mathbf{x} + b_k \leq t$ pro všechna k .

Poznámka 1.7: Kontrolní seznam převodu

Neexistuje jediný povinný sled úprav. Vždy však zkontrolujte:

1. zda je správně změněno znaménko účelové funkce,
2. zda byly zachovány směry všech nerovností,
3. zda každá proměnná má výslovně uvedený obor,
4. zda doplňkové proměnné mají nulové koeficienty v účelové funkci,
5. jak se z řešení rozšířeného modelu obnoví původní proměnné.

Pro simplexovou metodu budeme zpravidla mířit k maximalizačnímu rovnicovému tvaru $\max\{\mathbf{c}^T \mathbf{x} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$.

Příklad 1.8: Převod úlohy s volnou proměnnou

Uvažujme

$$\min 2x_1 - x_2 \quad \text{při} \quad -x_1 + 2x_2 \geq 1, \quad x_1 + x_2 = 3,$$

kde $x_1 \geq 0$ a $x_2 \in \mathbb{R}$ je volná. Projdeme převod po jednotlivých krocích:

1. změnou znaménka účelové funkce dostaneme $\max(-2x_1 + x_2)$,
2. volnou proměnnou rozložíme jako $x_2 = x_2^+ - x_2^-$, kde $x_2^+, x_2^- \geq 0$,
3. nerovnost vynásobíme číslem -1 , takže $x_1 - 2x_2 \leq -1$,
4. doplníme rezervu $s \geq 0$ a nerovnost nahradíme rovností.

Po dosazení rozkladu x_2 je výsledný rovnicový tvar

$$\begin{aligned} \max \quad & -2x_1 + x_2^+ - x_2^-, \\ \text{při} \quad & x_1 - 2x_2^+ + 2x_2^- + s = -1, \\ & x_1 + x_2^+ - x_2^- = 3, \\ & x_1, x_2^+, x_2^-, s \geq 0. \end{aligned}$$

Záporná pravá strana první rovnice není chyba ekvivalence, ale naznačuje, že doplňková proměnná sama neposkytne výchozí přípustnou bázi.

Příklad 1.9: Převod intervalové meze

Intervalové meze potkáváme například tehdy, když proměnná představuje kapacitu s předepsaným minimem i maximem. Nemusíme kvůli nim zavádět dvě nové proměnné.

Pro $1 \leq x \leq 5$ položme $x = 1 + y$. Pak $0 \leq y \leq 4$. Ve všech omezeních i v účelové funkci nahradíme x výrazem $1 + y$. Pro rovnicový tvar lze horní mez zapsat jako $y + s = 4$, kde $s \geq 0$.

3 Celočíselné a smíšené celočíselné programování

V mnoha praktických úlohách nestačí dovolit proměnným libovolné reálné hodnoty. Počet vozidel nebo směn musí být celočíselný a rozhodnutí otevřít–neotevřít provoz lze přirozeně vyjádřit binární proměnnou. Takové podmínky zásadně mění geometrii i výpočetní obtížnost úlohy. Zatím se naučíme ILP a MILP pouze bezpečně rozpoznat a zapsat. K metodám jejich řešení se vrátíme v poslední kapitole.

Celočíselný lineární program (ILP)

V čistém celočíselném lineárním programu jsou všechny rozhodovací proměnné celočíselné, například

$$\max \mathbf{c}^T \mathbf{x} \quad \text{při} \quad A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \in \mathbb{Z}_+^n.$$

Binární proměnná navíc splňuje $x_j \in \{0, 1\}$.

Smíšený celočíselný lineární program (MILP)

Ve smíšeném modelu je celočíselná jen část proměnných:

$$\begin{aligned} \max \quad & \mathbf{c}_y^T \mathbf{y} + \mathbf{c}_z^T \mathbf{z}, \\ \text{při} \quad & A_y \mathbf{y} + A_z \mathbf{z} \leq \mathbf{b}, \\ & \mathbf{y} \in \mathbb{Z}_+^p, \quad \mathbf{z} \in \mathbb{R}_+^{n-p}. \end{aligned}$$

Přípustná množina čistého ILP je množina celočíselných bodů mnohostěnu, tedy diskrétní množina. Přípustná množina MILP může obsahovat celé spojité polyedrické části, ale obecně není konvexní. Nelze proto bez dalšího přenést geometrické závěry z LP na ILP či MILP.

3.1 Výpočetní složitost

Při tvrzeních o složitosti předpokládáme racionální vstupní data zapsaná v binárním kódování. Velikost vstupu zahrnuje i počet bitů koeficientů. Lineární programování je řešitelné v čase polynomiálním ve velikosti takového vstupu. Naproti tomu rozhodovací verze obecného celočíselného lineárního programování je NP-úplná a optimalizační verze je NP-těžká. MILP obsahuje ILP jako zvláštní případ (Schrijver, 1986, Papadimitriou, 1981). To nevylučuje rychlé řešení mnoha praktických instancí. Pokud však $P \neq NP$, neexistuje polynomiální algoritmus pro všechny instance obecného ILP.

Poznámka 1.10: Co zde znamenají pojmy NP-těžký a NP-úplný?

Rozhodovací problém se ptá pouze na odpověď ano–ne, například zda existuje přípustné celočíselné řešení s účelovou hodnotou alespoň K . Problém je *NP-těžký*, jestliže na něj lze v polynomiálním čase převést každý problém ze třídy NP. Je-li současně sám ve třídě NP, nazýváme jej *NP-úplným*. Neříkáme tím, že každá konkrétní úloha musí být v praxi pomalá. Jde o varování, že pro všechny instance obecně neočekáváme jeden vždy rychlý algoritmus.

3.2 LP-relaxace

LP-relaxaci získáme vypuštěním celočíselných podmínek při zachování všech ostatních omezení. Přípustná množina se tím zvětší. Proto pro maximalizaci platí $z_{LP}^* \geq z_{ILP}^*$ a pro minimalizaci opačná nerovnost, pokud jsou obě optimální hodnoty konečné.

K čemu je řešení uvolněné úlohy dobré, když jsme původně chtěli celá čísla? Právě proto, že optimalizujeme přes větší množinu, dostáváme optimistickou mez na výsledek původního modelu. U maximalizace je to horní mez, u minimalizace mez dolní. Později ji využijeme při rozhodování, které části stromu metody větvi a mezi už nemohou obsahovat lepší celočíselné řešení.

Nezáporná aditivní mez konkrétní instance je pro maximalizaci $z_{LP}^* - z_{ILP}^*$ a pro minimalizaci $z_{ILP}^* - z_{LP}^*$. Multiplikativní *poměr celočíselnosti* se obvykle definuje jen pro kladné optimální hodnoty: u minimalizace jako $z_{ILP}^*/z_{LP}^* \geq 1$ a u maximalizace jako $z_{LP}^*/z_{ILP}^* \geq 1$. Bez podmínky kladnosti může být poměr nedefinovaný nebo zavádějící. Termín *mez celočíselnosti* (angl. *integrality gap*) se navíc často používá pro supremum těchto poměrů nad celou určenou třídou instancí. Vždy proto uvedeme, zda mluvíme o jedné instanci, nebo o třídě úloh.

4 Krátké srovnání s lineární algebrou

První semestr lineární algebry jste pravděpodobně strávili u matic, soustav rovnic a Gaussovy eliminace. V lineárním programování se tytéž objekty vracejí, jen se na ně ptáme trochu ostřeji: nehledáme libovolné řešení, ale řešení nejlepší. Čtyři paralely stojí za to držet v paměti.

Ústřední úloha. Lineární algebra popisuje všechna řešení $A\mathbf{x} = \mathbf{b}$, tedy zpravidla afinní podprostor. Lineární programování přidá nerovnosti a účelovou funkci $\mathbf{c}^T\mathbf{x}$. Přípustná oblast je mnohostěn a nás zajímá jeho nejlepší bod.

Algoritmický “dělník”. V algebře dominuje Gaussova eliminace. V LP se stejné řádkové úpravy skrývají uvnitř simplexové metody, která přechází od jednoho vrcholu k druhému. Později zmíníme i metody vnitřního bodu, které naopak postupují vnitřkem mnohostěnu.

Certifikát neřešitelnosti. Neřešitelnost rovnic odhalí hodnotní kritérium. Pro soustavu nerovnic bude obdobnou roli hrát Farkasovo lemma.

Geometrie. Afinní podprostor je “plochý” a nemá výsadní body. V přípustném mnohostěnu našeho rovnicového tvaru budou naopak výsadními body vrcholy. Ukážeme, že je-li tento mnohostěn neprázdný a má-li úloha konečnou optimální hodnotu, alespoň jednoho optima lze dosáhnout v některém z nich.

Lineární algebra tedy není jen formálním předpokladem kurzu. Je to jazyk, v němž budeme geometrii i algoritmy lineárního programování skutečně počítat.

5 Jak vznikalo lineární programování

Lineární programování nevzniklo jediným objevem v jediném okamžiku. Jeho matematické součásti se objevovaly odděleně: teorie lineárních nerovnic vedla k Farkasovu lemmatu,

von Neumannova minimaxová věta spojila optimalizaci s maticovými hrami a Hitchcock formuloval dopravní úlohu ještě před vznikem obecného algoritmu (Farkas, 1902, von Neumann, 1928, Hitchcock, 1941). Moderní obor se z těchto proudů ustavil teprve tehdy, když se matematika setkala s naléhavými otázkami výroby, hospodářství a válečného plánování. Ani označení jednoho člověka za jediného „vynálezce“ proto nevystihuje celý příběh (Cottle et al., 2007).

5.1 Leningrad, dnešní Petrohrad: od překližky k obecné metodě

Roku 1938 oslovili v Leningradě mladého profesora Leonida V. Kantoroviče inženýři z laboratoře překližkářského trustu. Hledali rozvrh práce osmi loupacích strojů s různou produktivitou pro pět druhů dřeva. Podnik musel dodržet předepsaný sortiment a co nejlépe využít strojní čas. Kantorovič v úloze rozpoznal obecný problém optimálního rozdělení omezených zdrojů (Kantorovich, 1960, Boldyrev and Düppe, 2020).

Zapsal celou třídu úloh pomocí lineárních omezení a optimalizačního kritéria. Jeho multiplikátory omezení předznamenal duální proměnné a stínové ceny. Ty za vhodných podmínek popisují mezní změnu optimální hodnoty při malé změně dostupného množství zdroje. Přesné podmínky vyložíme v kapitolách o dualitě (Kantorovich, 1960, Boldyrev and Düppe, 2020).

Brožura *Matematické metody organizace a plánování výroby* vyšla v Leningradě roku 1939 v tisíci výtiscích. Anglický překlad následoval až roku 1960. Práce tak dlouho zůstala mimo běžný mezinárodní oběh a bezprostředně neovlivnila americký vývoj (Kantorovich, 1960, Boldyrev and Düppe, 2020).

Za obléhání Leningradu dostala Kantorovičova práce jinou naléhavost. Zásoby přivázela nákladní auta po zamrzlém Ladožském jezeře. Bezpečný průjezd po této „Cestě života“ závisel na únosnosti ledu. Kantorovič patřil k týmu, který ji počítal s ohledem na vlastnosti ledu, počasí a zatížení. Matematický výpočet zde pomáhal rozhodovat o bezpečném zásobování města (Nazarov and Sinkevich, 2019, s. 58).

Poznámka 1.11: Dva neobyčejné domácí úkoly

V roce 1939 nastoupil George B. Dantzig k doktorskému studiu u statistika Jerzyho Neymana v Berkeley. Podle své pozdější vzpomínky přišel jednou pozdě na přednášku, opsal si z tabule dvě úlohy a považoval je za domácí práci. Po několika dnech odevzdal řešení. Teprve potom se dozvěděl, že šlo o dva dosud nevyřešené problémy matematické statistiky. Výsledky se staly základem jeho disertace (Albers and Reid, 1986). Historka bývá převyprávěna jako legenda o úloze vyřešené přes noc. Dantzigovo vlastní podání je střízlivější: pracoval několik dnů a netušil, že řešení má být obtížné. Úlohy sice nebyly formulovány jako běžné konečné lineární programy, Dantzig však později připomínal, že geometrická práce se sloupci v disertaci mu pomohla při vzniku simplexové metody (Albers and Reid, 1986, Dantzig, 2002). Příběh tak dobře představuje člověka, který neváhal hledat obecnou metodu tam, kde ostatní viděli jen příliš mnoho možností.

5.2 Washington: když program znamenal plán

Druhá světová válka Dantzigovo doktorské studium přerušila. Od roku 1941 pracoval jako civilní statistik pro americké armádní letectvo a vedl oddělení, které vyhodnocovalo bojové operace. Po dokončení doktorátu v roce 1946 se vrátil do Pentagonu jako matematický poradce. Jeho kolegové mu položili otázku, která zněla spíše organizačně než matematicky: lze zrychlit plánování nasazení, výcviku a zásobování letectva v čase? Tehdejší „výpočetní technikou“ byly především děrné štítky, účetní stroje a lidé se stolními kalkulačkami (Albers and Reid, 1986, Cottle et al., 2007).

Slovo *program* zde označovalo plán činností. Výzkumná skupina, která později dostala název Project SCOOP (*Scientific Computation of Optimum Programs*), tedy nevznikla kvůli psaní počítačového kódu. Měla porovnávat možné plány, které musely respektovat dostupná letadla, materiál, personál i časovou návaznost činností. Dantzig vyšel z Leontiefových meziodvětvových modelů, připustil více alternativních činností a postupně nahradil množství dílčích plánovacích pravidel soustavou lineárních omezení. Zásadním krokem bylo přidání jediné účelové funkce: cílem už nebylo najít plán, který pouze splní pravidla, ale nejlepší ze všech přípustných plánů (Dantzig, 2002, Gass, 2002).

V létě 1947 Dantzig hledal algoritmus pro takto vzniklou obecnou úlohu. První nápad byl geometricky přirozený: přecházet po hranách mnohostěnu od jednoho vrcholu k sousednímu a pokaždé zlepšit účelovou funkci. Nápad nejprve zavrhl, protože se obával, že v mnoha rozměrech bude cesta beznadějně dlouhá. Když však úlohu začal chápat prostřednictvím sloupců matice a bází, stejný postup vypadal mnohem slibněji. Tak vznikla simplexová metoda. Dantzig později otevřeně připouštěl, že jeho geometrická intuice pro vysoké dimenze byla zpočátku příliš pesimistická (Albers and Reid, 1986, Dantzig, 2002).

5.3 Setkání s von Neumannem: dualita vstupuje do amerického příběhu

Dne 3. října 1947 navštívil Dantzig Johna von Neumanna v Institutu pro pokročilá studia (*Institute for Advanced Study*) v Princetonu. Simplexová metoda již existovala. Dantzig chtěl znát jiné možné způsoby řešení, s nimiž by ji mohl porovnat. Podle jeho vzpomínky von Neumann dlouhý úvod rychle přerušil a požádal o samotnou matematickou formulaci. Jakmile Dantzig napsal geometrickou a algebraickou podobu úlohy na tabuli, následoval přibližně devadesátiminutový výklad o Farkasově lemmatu a dualitě (Albers and Reid, 1986, Cottle et al., 2007).

Von Neumann v Dantzigově modelu rozpoznal obdobu teorie her dvou hráčů s nulovým součtem, na níž pracoval již dříve. Dantzig následně sepsal důkaz duality do interní zprávy z ledna 1948. Veřejnou podobu teorie dále rozvíjeli Albert Tucker, Harold Kuhn a David Gale (Albers and Reid, 1986, Cottle et al., 2007). Setkání tedy nebylo počátkem simplexové metody. Vysvětluje však, proč se dualita, Farkasovo lemma a maticové hry staly neoddělitelnou součástí lineárního programování. Ke všem třem tématům se v této knize později vrátíme.

5.4 První výpočty: 120 člověkodnů na jednu úlohu

Jedním z prvních velkých testů nové metody byla Stiglerova dietní úloha: z cenově dostupných potravin sestavit výživu, která splní stanovené minimální požadavky na živiny a přitom má nejnižší cenu (Stigler, 1945). Model měl 77 proměnných pro jednotlivé potraviny a devět výživových omezení. Na podzim 1947 jej tým vedený Jackem Ladermanem

v americkém Národním úřadu pro standardy (*National Bureau of Standards*) počítal na ručních stolních kalkulačkách. Výpočet spotřeboval přibližně 120 člověkodnů (Dantzig, 1963, Gass, 2002).

Výsledek byl poučný i z jiného důvodu. Stigler předtím sestavil promyšlený odhad s roční cenou 39,93 USD v cenách roku 1939. Prokázané optimum ve stejné cenové úrovni činilo 39,69 USD. Rozdíl pouhých 24 centů ukázal, že Stiglerova intuice byla výborná. Simplexová metoda k ní však přidala něco nového: záruku, že žádná levnější přípustná kombinace neexistuje (Dantzig, 1990).

Poznámka 1.12: Optimalizátor nemá zdravý rozum

Po přechodu do společnosti RAND roku 1952 (Cottle et al., 2007) se Dantzig pokusil využít počítač IBM 701 také pro vlastní redukční dietu. Model věrně optimalizoval zadaná čísla, ale nerozuměl tomu, co je přijatelné jídlo: první řešení požadovalo 500 galonů octa a po jeho vyřazení další doporučilo 200 bujónových kostek denně. Dantzig s manželkou postupně doplňovali horní meze a nevhodné potraviny vyřazovali, až Anne Dantzigová experiment ukončila a jídelníček stanovila sama (Dantzig, 1990). Anekdota vystihuje základní poučení modelování: algoritmus optimalizuje přesně to, co jsme zapsali, nikoli to, co jsme měli na mysli.

5.5 Od praktického dříče k teorii složitosti

Na konferenci uspořádané Koopmansem v Chicagu roku 1949, později nazývané „nulté sympozium“ matematického programování, se setkali matematici, ekonomové a statistici, kteří do té doby pracovali často nezávisle. Dantzig při vzniku simplexové metody Kantorovičovu práci neznal. Teprve postupně se ukázalo, že výrobní plánování v Leningradě, americká vojenská logistika, teorie her a dopravní modely mluví různými dialekty téhož matematického jazyka (Dantzig, 2002, Cottle et al., 2007).

Simplexová metoda se stala mimořádně úspěšným praktickým nástrojem. Roku 1972 však Klee a Minty sestrojili rodinu úloh, na niž simplexová metoda s Dantzigovým pravidlem volby dalšího kroku, zahájená v počátku souřadnic, navštíví všech 2^d vrcholů d -rozměrného mnohostěnu (Klee and Minty, 1972). Neznamená to, že každý simplexový výpočet nebo každé pravidlo volby kroku je pomalé. Ukazuje to, že výborné praktické chování samo o sobě není důkazem dobré složitosti v nejhorším případě.

Roku 1975 obdržel Kantorovič společně s Tjallingem Koopmansem Cenu Švédské národní banky za rozvoj ekonomické vědy na památku Alfreda Nobela za přínos k teorii optimální alokace zdrojů (Royal Swedish Academy of Sciences, 1975).

První algoritmus, který řeší LP v čase polynomiálním v délce binárního zápisu racionálních vstupních dat, publikoval Leonid Chačijan (Khachiyan, 1979). Použil elipsoidní metodu. Nový praktický směr metod vnitřního bodu otevřel Karmarkar (1984). Simplex tím nezmizel. Historie se spíše uzavřela dalším kontrastem: jeden algoritmus dlouho vítězil v praxi bez polynomiální záruky, jiný nejprve změnil teorii a další postupy pak změnil i výpočetní praxi.

6 Co všechno lze modelovat lineárním programem

Základní pojmy již máme. Zkusme nyní v několika zdánlivě nesouvisejících situacích rozpoznat tentýž matematický jazyk. Následujících šest úloh je inspirováno kapitolou 2 textu Matoušek (2006).

6.1 Nejlevnější směs s předepsaným složením

Výrobce připravuje jeden kilogram směsi ze tří sypkých surovin. Směs má obsahovat alespoň 120 gramů složky P a alespoň 100 gramů složky Q. Suroviny lze libovolně dělit a míchat, jejich obsahy se sčítají a při zpracování nevznikají ztráty. Jaká je nejnížší cena směsi? Jde o stejný typ modelu jako dietní úloha, v níž za co nejnížší cenu plníme požadavky na jednotlivé živiny (Matoušek, 2006, oddíl 2.1).

Údaj na 1 kg suroviny	A	B	C
Obsah složky P [g]	200	100	0
Obsah složky Q [g]	0	100	300
Cena [Kč]	8	6	5

Tabulka 1.1: Ilustrační údaje pro přípravu jednokilogramové směsi.

Označíme-li množství surovin v kilogramech jako x_A, x_B, x_C , dostaneme

$$\begin{aligned}
 \min \quad & 8x_A + 6x_B + 5x_C, \\
 \text{při} \quad & x_A + x_B + x_C = 1, \\
 & 200x_A + 100x_B \geq 120, \\
 & 100x_B + 300x_C \geq 100, \\
 & x_A, x_B, x_C \geq 0.
 \end{aligned}$$

První rovnice hlídá hmotnost, další dvě podmínky požadované složení. Všimněme si jednotek: součin obsahu v gramech na kilogram a množství v kilogramech dává gramy složky ve výsledné směsi.

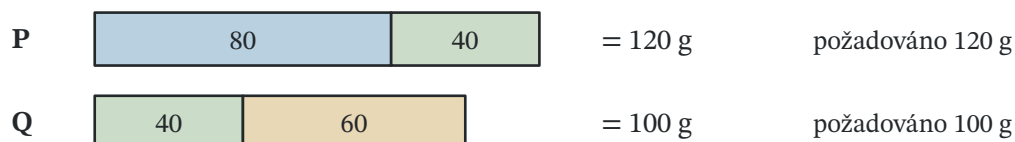
Směs s množstvími $x_A = 0,4$, $x_B = 0,4$ a $x_C = 0,2$ splňuje obě požadovaná minima přesně a stojí 6,60 Kč. Žádná levnější přípustná směs neexistuje, protože její cena splňuje

$$\begin{aligned}
 8x_A + 6x_B + 5x_C &= 2(x_A + x_B + x_C) \\
 &\quad + 0,03(200x_A + 100x_B) + 0,01(100x_B + 300x_C) \\
 &\geq 2 + 0,03 \cdot 120 + 0,01 \cdot 100 = 6,6.
 \end{aligned}$$

Složení tohoto optimálního řešení znázorňuje obrázek 1.1.

Složení jednoho kilogramu

Celkem 1 kg

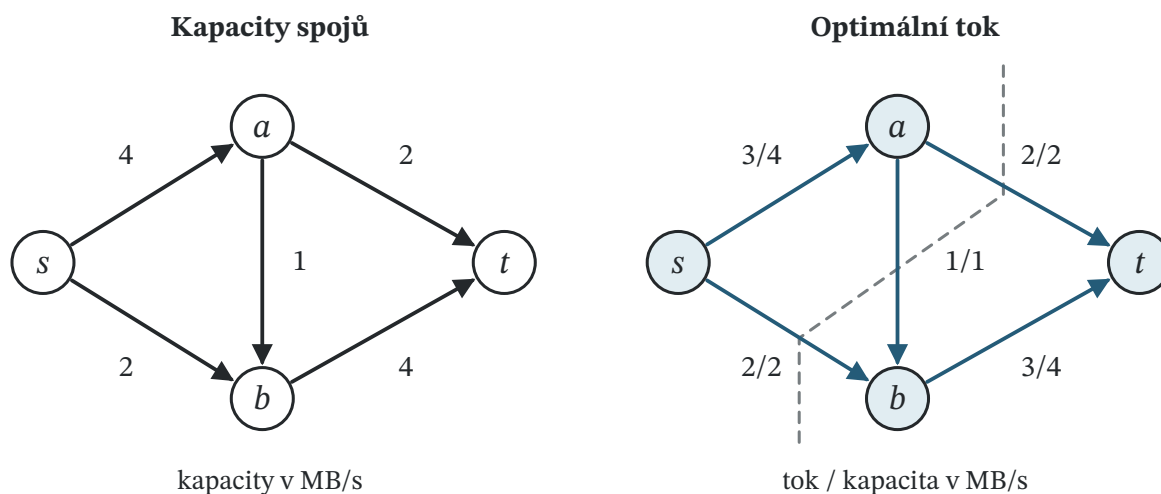
Cena 6,60 Kč

Obrázek 1.1: Optimální směs a příspěvky jednotlivých surovin k obsahu složek P a Q. Délky dílů horního pruhu odpovídají hmotnostem surovin, ve spodních pruzích množstvím složek v gramech.

Takový jednoduchý dolní odhad je předzvěstí duality. Pro první přednášku je však hlavní jiná myšlenka: model najde nejlevnější směs splňující *zapsané* požadavky. Pokud má mít také určitou chuť, barvu či maximální podíl suroviny C, musíme příslušné podmínky teprve doplnit.

6.2 Kolik dat projde sítí?

Data putují ze zdroje s přes dva mezilehlé uzly a, b do cíle t . Pět jednosměrných spojů má kapacity uvedené na obrázku 1.2 vlevo v MB/s. V mezilehlých uzlech se data neztrácejí ani nehromadí. Hledáme největší ustálenou přenosovou rychlost. Jde o variantu úlohy o toku v síti (Matoušek, 2006, oddíl 2.2).



Obrázek 1.2: Vlevo kapacity jednosměrných spojů, vpravo optimální tok o hodnotě 5 MB/s. Zápis $3/4$ znamená tok 3 MB/s při kapacitě 4 MB/s. Čárkovaná čára odděluje uzly s, a od uzlů b, t .

Proměnná x_{uv} označuje rychlost přenosu ze u do v . Model je

$$\begin{aligned} \max \quad & x_{sa} + x_{sb}, \\ \text{při} \quad & x_{sa} = x_{ab} + x_{at}, \\ & x_{sb} + x_{ab} = x_{bt}, \\ & 0 \leq x_{sa} \leq 4, \quad 0 \leq x_{sb} \leq 2, \quad 0 \leq x_{ab} \leq 1, \\ & 0 \leq x_{at} \leq 2, \quad 0 \leq x_{bt} \leq 4. \end{aligned}$$

Rovnosti vyjadřují bilanci v uzlech a a b . Jejich sečtením dostaneme $x_{sa} + x_{sb} = x_{at} + x_{bt}$, takže odeslaná a přijatá rychlost souhlasí.

Volba

$$(x_{sa}, x_{sb}, x_{ab}, x_{at}, x_{bt}) = (3, 2, 1, 2, 3)$$

přenáší 5 MB/s. Více nelze, neboť pro každý přípustný tok platí

$$x_{sa} + x_{sb} = x_{ab} + x_{at} + x_{sb} \leq 1 + 2 + 2 = 5.$$

Samotný součet kapacit spojů vycházejících ze zdroje, tedy 6 MB/s, by byl příliš optimistický. Omezení leží i uvnitř sítě. Bilance přitom nejsou zvláštností datových přenosů, stejným způsobem lze popisovat přepravu dělitelného materiálu.

6.3 Vyrábět podle poptávky, nebo do zásoby?

Podnik plánuje výrobu dělitelného výrobku na $T \geq 1$ měsíců. Poptávka $d_i \geq 0$ v měsíci i je předem daná a musí být včas uspokojena. Výroba může kolísat, její změny však něco stojí. Také skladování je placené. Chceme najít nejlevnější kompromis mezi plynulou výrobou a malými zásobami, podobně jako ve zmrzlinářském příkladu (Matoušek, 2006, oddíl 2.3).

Nechť x_i je výroba v měsíci i a s_i zásoba na jeho konci, obojí v tunách. Předchozí výroba $x_0 \geq 0$ je známá. Začínáme i končíme bez zásob, tedy $s_0 = s_T = 0$. Při skladování nejsou ztráty. Za jednu tunu ponechanou do dalšího měsíce zaplatíme $h > 0$ a za změnu měsíčního objemu výroby o jednu tunu zaplatíme $k > 0$, ať jde o zvýšení, nebo snížení. Samotná jednotková výrobní cena je ve všech měsících stejná. Protože celková výroba musí být $\sum_i d_i$, její náklady výběr plánu neovlivní a v účelové funkci je vynecháme.

S pomocnými proměnnými u_i pro velikost změny výroby sestavíme LP

$$\begin{aligned} \min \quad & h \sum_{i=1}^{T-1} s_i + k \sum_{i=1}^T u_i, \\ \text{při} \quad & s_{i-1} + x_i - s_i = d_i \quad (i = 1, \dots, T), \\ & -u_i \leq x_i - x_{i-1} \leq u_i \quad (i = 1, \dots, T), \\ & x_i, s_i, u_i \geq 0 \quad (i = 1, \dots, T), \\ & s_0 = s_T = 0. \end{aligned}$$

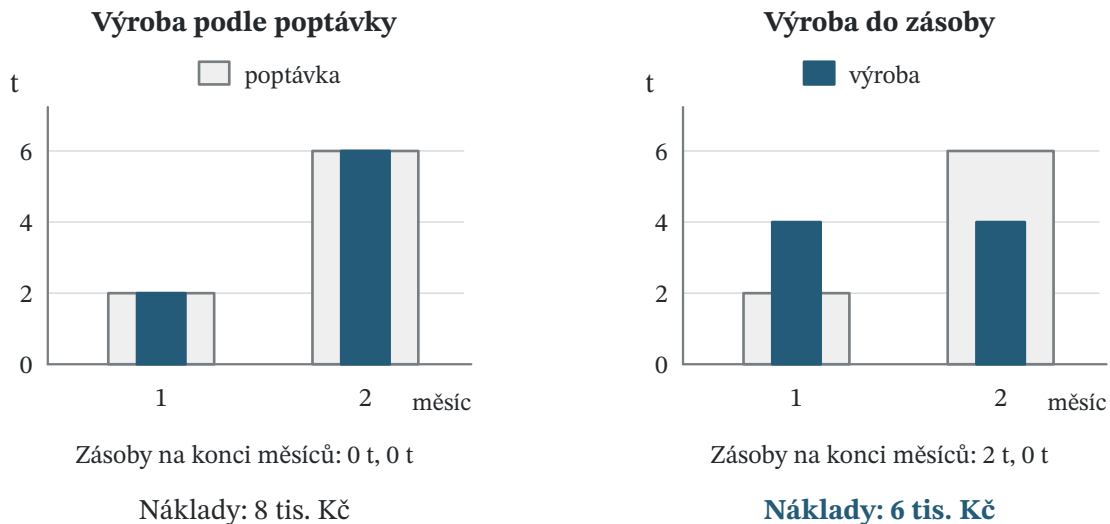
Nezápornost s_i brání odkládání dodávek do budoucna. Při kladném k musí v optimu platit $u_i = |x_i - x_{i-1}|$, jinak bychom mohli samotné u_i zmenšit a zlevnit plán. V libovolném přípustném řešení je u_i zatím jen horní mezí této absolutní hodnoty. Kapacitu výroby nebo skladu lze přidat jako další horní meze.

Pro malou ukázkou vezměme $T = 2$, $(d_1, d_2) = (2, 6)$, $x_0 = 2$, $h = 1$ a $k = 2$. Číselné náklady měřme v tisících Kč. Výroba přesně podle poptávky, $(x_1, x_2) = (2, 6)$, stojí na

změnách 8. Rovnoměrný plán $(4, 4)$ má zásobu $s_1 = 2$, změny $(u_1, u_2) = (2, 0)$ a cenu $2 + 2 \cdot 2 = 6$. Je i optimální. Položíme-li $x = x_1$, bilance dávají $s_1 = x - 2$, $x_2 = 8 - x$ a $2 \leq x \leq 8$. Nejnižší cena pro takové x je

$$(x - 2) + 2(|x - 2| + |8 - 2x|) = \begin{cases} 10 - x, & 2 \leq x \leq 4, \\ 7x - 22, & 4 \leq x \leq 8. \end{cases}$$

Obě větve mají společné minimum 6 při $x = 4$. Absolutní hodnota tedy nemusí znamenat konec lineárního programování. Oba výrobní plány porovnává obrázek 1.3.



Obrázek 1.3: Poptávka a výroba ve dvou měsících při předchozí výrobě $x_0 = 2$ t. Široké světlé sloupce ukazují poptávku, úzké tmavé výrobu. Rovnoměrný plán převádí dvě tuny z prvního měsíce do druhého.

6.4 Lze dvě skupiny bodů oddělit přímkou?

Máme dva neprázdné konečné soubory měření v rovině. Body $\mathbf{p}_1, \dots, \mathbf{p}_m$ představují výrobky jedné třídy a body $\mathbf{q}_1, \dots, \mathbf{q}_n$ výrobky druhé třídy. Souřadnice mohou například zachycovat dvě naměřené vlastnosti. Existuje přímka, která všechny body první třídy nechá na jedné straně a všechny body druhé třídy na druhé, aniž by některým z nich procházela? Tato úloha ukazuje souvislost LP s klasifikací dat (Matoušek, 2006, oddíl 2.4).

Přímku zapíšeme jako $\mathbf{w}^T \mathbf{z} + b = 0$, kde hledáme vektor $\mathbf{w} = (w_1, w_2)^T$ a číslo b . Místo ostrých nerovností budeme požadovat rezervu alespoň jedna:

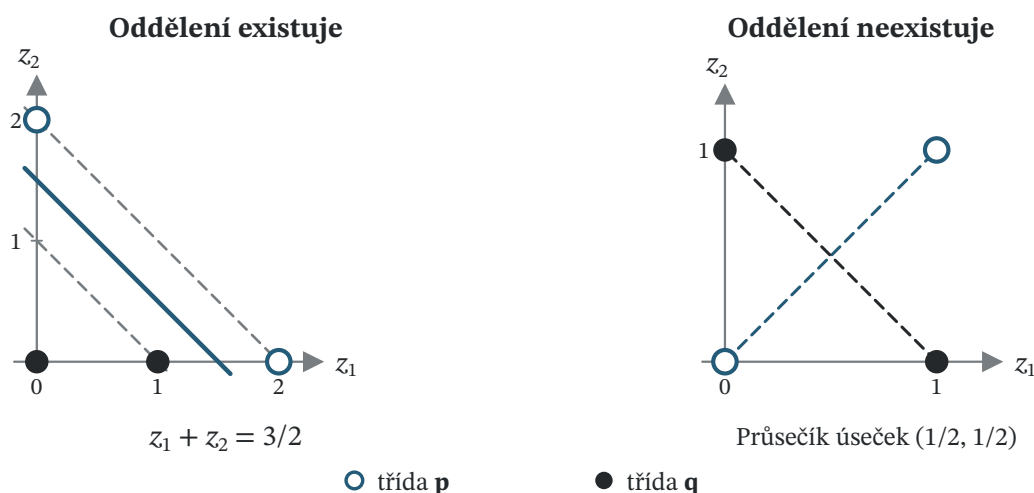
$$\begin{aligned} \min \quad & 0, \\ \text{při} \quad & \mathbf{w}^T \mathbf{p}_i + b \geq 1 \quad (i = 1, \dots, m), \\ & \mathbf{w}^T \mathbf{q}_j + b \leq -1 \quad (j = 1, \dots, n), \\ & \mathbf{w} \in \mathbb{R}^2, \quad b \in \mathbb{R}. \end{aligned}$$

Účelová funkce je konstantní, protože nás zajímá pouze přípustnost. Souřadnice měření jsou data, takže všechny podmínky jsou lineární v neznámých w_1, w_2, b . Zápis zahrnuje i svislé přímky.

Proč pevná rezerva neztratí žádné přísné oddělení? Pokud nějaké $(\mathbf{w}, b)$ všechny body přísně odděluje, minimum z konečně mnoha kladných čísel $\mathbf{w}^T \mathbf{p}_i + b$ a $-(\mathbf{w}^T \mathbf{q}_j + b)$ je číslo

$\delta > 0$. Vydělením všech koeficientů číslem δ dostaneme uvedené podmínky. Obrácená implikace je bezprostřední. Neprázdnot obou tříd navíc vylučuje $\mathbf{w} = \mathbf{0}$. Jednička zde vyjadřuje rezervu v hodnotě lineárního výrazu, nikoli pevnou geometrickou vzdálenost od přímky.

Například body $(2, 0)$ a $(0, 2)$ oddělíme od bodů $(0, 0)$ a $(1, 0)$ volbou $\mathbf{w} = (2, 2)^T$, $b = -3$. Dělicí přímka má rovnici $z_1 + z_2 = 3/2$. Naopak dvě třídy, z nichž jedna obsahuje $(0, 0)$, $(1, 1)$ a druhá $(1, 0)$, $(0, 1)$, takto oddělit nelze. Součet prvních dvou nerovností by vyžadoval $w_1 + w_2 + 2b \geq 2$, součet druhých dvou $w_1 + w_2 + 2b \leq -2$. Nepřípustnost LP zde má jasný význam: zvolený typ dělicího pravidla pro tato měření nestačí. Obě situace ukazuje obrázek 1.4.



Obrázek 1.4: Vlevo lze třídy přísně oddělit. Čárkované přímky $z_1 + z_2 = 1$ a $z_1 + z_2 = 2$ vymezují rezervu v hodnotě výrazu $2z_1 + 2z_2 - 3$. Vpravo se úsečky spojující body téže třídy protínají a přísné oddělení není možné.

6.5 Přímka, která co nejlépe vystihuje měření

Pro dané body (t_i, y_i) , $i = 1, \dots, N$, hledáme přímku $y = at + b$. Co znamená *co nejlépe*? Pokud minimalizujeme součet čtverců odchylek, dostaneme kvadratickou účelovou funkci. Součet absolutních odchylek se však dá vyjádřit pomocí LP (Matoušek, 2006, oddíl 2.5):

$$\begin{aligned}
 \min \quad & \sum_{i=1}^N e_i, \\
 \text{při} \quad & -e_i \leq at_i + b - y_i \leq e_i \quad (i = 1, \dots, N), \\
 & e_i \geq 0 \quad (i = 1, \dots, N), \\
 & a, b \in \mathbb{R}.
 \end{aligned}$$

Hodnoty t_i jsou známé koeficienty, nikoli proměnné. V optimu je $e_i = |at_i + b - y_i|$. Jde o odchylky ve směru osy y , nikoli o kolmé vzdálenosti bodů od přímky.

Jestliže má být co nejmenší *nejhorší* odchylka, použijeme jedinou pomocnou proměnnou e a jiný LP:

$$\min e \quad \text{při} \quad -e \leq at_i + b - y_i \leq e \quad (i = 1, \dots, N), \quad a, b \in \mathbb{R}, \quad e \geq 0.$$

Rozdíl mezi oběma cíli ukážou již tři měření $(0, 0)$, $(1, 1)$, $(2, 0)$. Pro součet absolutních odchylek je optimální přímka $y = 0$ s hodnotou 1. Pro nejhorší odchylku je optimální

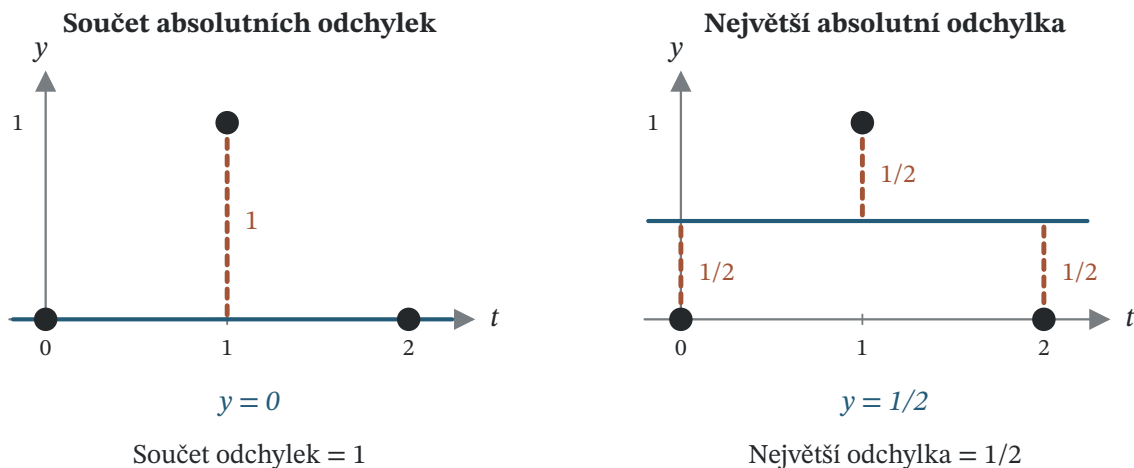
přímka $y = 1/2$ s hodnotou $1/2$. Správnost lze ověřit bez algoritmu. Pro podepsané odchylky

$$r_1 = b, \quad r_2 = a + b - 1, \quad r_3 = 2a + b$$

platí $r_1 - 2r_2 + r_3 = 2$, a tedy

$$2 \leq |r_1| + 2|r_2| + |r_3| \leq 2 \sum_{i=1}^3 |r_i|, \quad 2 \leq 4 \max_{i=1,2,3} |r_i|.$$

Uvedené přímky dosahují příslušných dolních mezí. Stejná měření tedy vedou k různým odpovědím podle toho, jakou chybu jsme se rozhodli měřit (obrázek 1.5).



Obrázek 1.5: Dvě optimální přímky pro stejná tři měření. Čárkované svislé úsečky znázorňují nenulové absolutní odchylky. Vlevo minimalizujeme jejich součet, vpravo největší z nich.

6.6 Největší kruh uvnitř mnohoúhelníku

Do konvexního mnohoúhelníku chceme umístit co největší kruh, například kruhovou plochu na pozemku s přímými hranicemi. Kruh není mnohoúhelník a v jeho rovnici jsou druhé mocniny. Přesto lze hledání jeho středu a poloměru vyřešit lineárním programem (Matoušek, 2006, oddíl 2.6).

Předpokládejme, že pozemek je neprázdný omezený mnohoúhelník s vnitřními body zadaný jako průnik polorovin

$$P = \{\mathbf{z} \in \mathbb{R}^2 : \mathbf{a}_i^\top \mathbf{z} \leq b_i, \quad i = 1, \dots, m\}, \quad \mathbf{a}_i \neq \mathbf{0}.$$

Hledáme střed $\mathbf{s} \in \mathbb{R}^2$ a poloměr $r \geq 0$. Kruh leží celý v i -té polorovině právě tehdy, když

$$\mathbf{a}_i^\top \mathbf{s} + r \|\mathbf{a}_i\|_2 \leq b_i, \quad \|\mathbf{a}_i\|_2 = \sqrt{a_{i1}^2 + a_{i2}^2}.$$

Skutečně, maximum výrazu $\mathbf{a}_i^\top \mathbf{z}$ na kruhu se středem $\mathbf{s}$ a poloměrem r se nabývá ve směru $\mathbf{a}_i$ a je rovno levé straně této nerovnosti. Pro $r = 0$ se tvrzení redukuje na podmínku pro samotný střed. Stačí tedy řešit

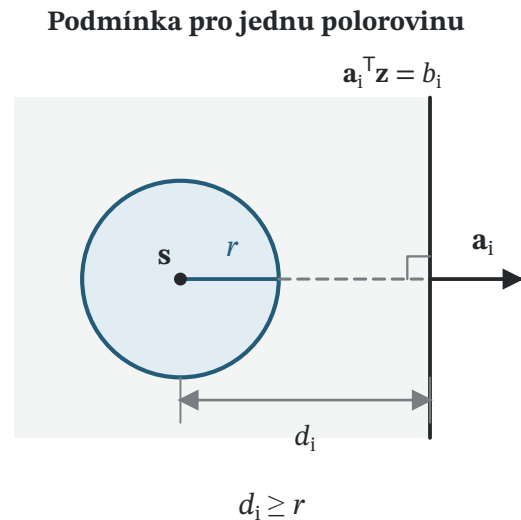
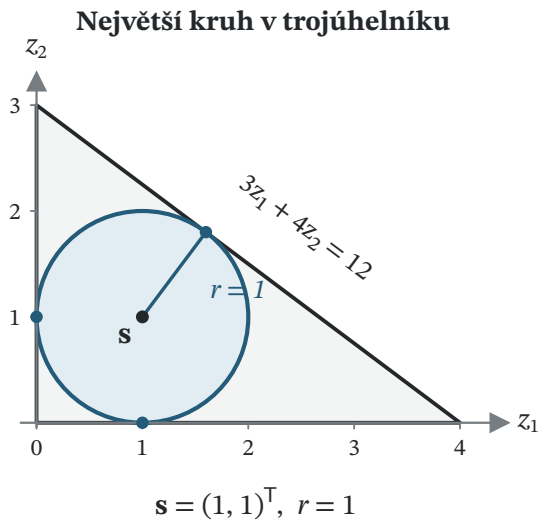
$$\max r \quad \text{při} \quad \mathbf{a}_i^\top \mathbf{s} + r \|\mathbf{a}_i\|_2 \leq b_i \quad (i = 1, \dots, m), \quad \mathbf{s} \in \mathbb{R}^2, \quad r \geq 0.$$

Odmocniny se počítají ze zadaných čísel. Vůči třem rozhodovacím proměnným s_1, s_2, r je model lineární.

V pravoúhlém trojúhelníku $z_1 \geq 0, z_2 \geq 0, 3z_1 + 4z_2 \leq 12$ má LP zvlášť jednoduchý tvar

$$\max r \quad \text{při} \quad s_1 \geq r, \quad s_2 \geq r, \quad 3s_1 + 4s_2 + 5r \leq 12, \quad r \geq 0.$$

První dvě podmínky spolu s třetí dávají $12r \leq 12$. Střed $\mathbf{s} = (1, 1)^\top$ a poloměr $r = 1$ jsou přípustné, takže jsme našli největší kruh (obrázek 1.6). Příklad ukazuje, že rozhodující není tvar geometrického objektu, ale závislost omezení na tom, co skutečně volíme.



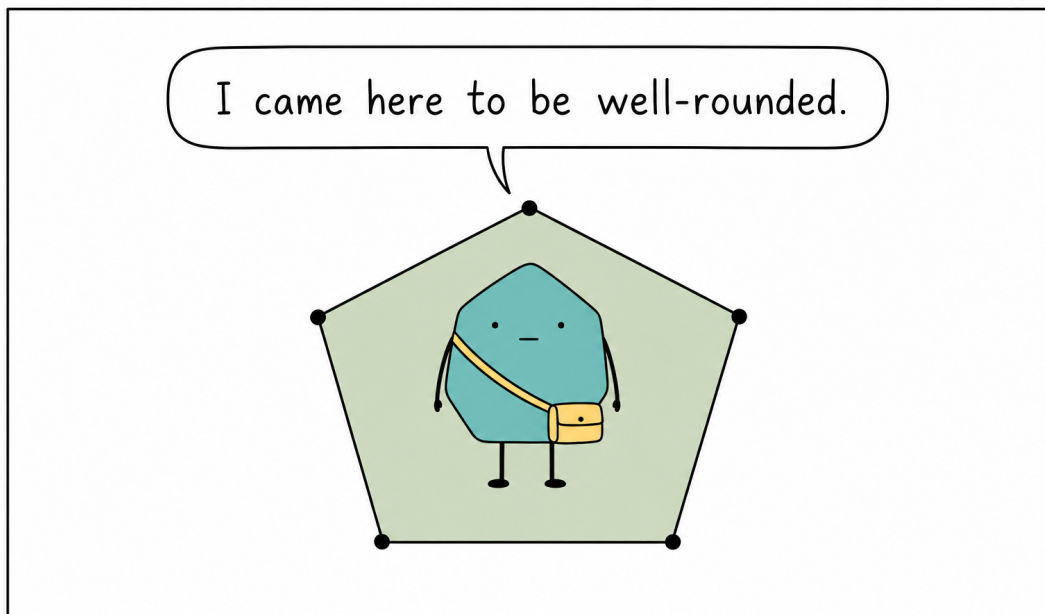
Obrázek 1.6: Vlevo největší kruh v zadaném trojúhelníku. Vpravo podmínka pro jednu polorovinu: kolmý odstup $d_i = (b_i - \mathbf{a}_i^\top \mathbf{s}) / \|\mathbf{a}_i\|_2$ musí být alespoň r . Normála $\mathbf{a}_i$ směřuje ven z poloroviny.

KAPITOLA 2

GEOMETRIE LP: KONVEXITA A BÁZICKÁ ŘEŠENÍ

1	Úvod do konvexity	20
2	Bázická přípustná řešení	24

Tato kapitola představuje geometrický základ lineárního programování. Zavedeme konvexní kombinace, afinní a konvexní obal, poloprostory, mnohohostěny a jejich vrcholy. Nejde jen o nový slovník: právě geometrie nám vysvětlí, proč má při hledání optima smysl soustředit se na “rohy” přípustné oblasti. Ve druhé části přejdeme k rovnicovému tvaru LP a přesně popíšeme bázická přípustná řešení, jejich nosič a degeneraci.



1 Úvod do konvexity

Konvexita je jedním ze základních kamenů geometrie lineárního programování. Připomeneme několik nejdůležitějších pojmů a výsledků, díky nimž si později budeme umět řešení LP nejen spočítat, ale také představit.

1.1 Konvexní kombinace, konvexní a afinní obal

Začneme dvěma body. Jestliže spolu s nimi množina obsahuje také celou úsečku, která je spojuje, chová se “bez prohlubní”. Přesně tuto vlastnost zachytí následující definice.

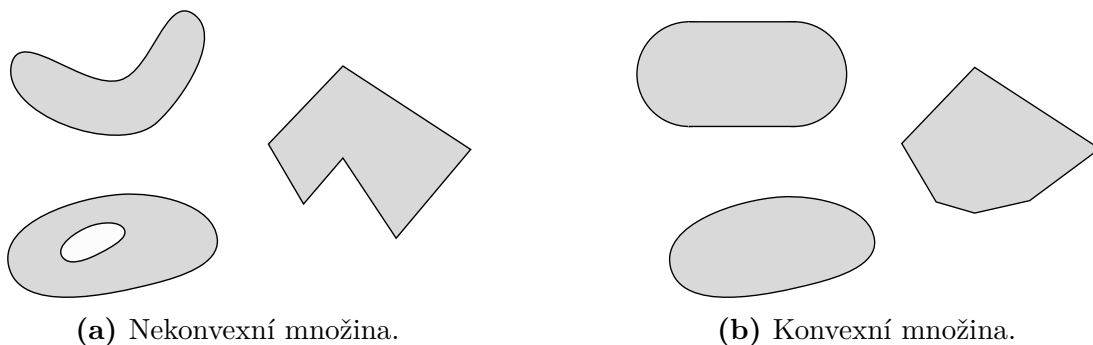
Definice 2.1: Konvexní kombinace a konvexní množina

Nechť $S \subseteq \mathbb{R}^n$. Bod

$$\mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{x}^{(i)}, \quad \lambda_i \geq 0, \quad \sum_{i=1}^k \lambda_i = 1,$$

je *konvexní kombinací* bodů $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)} \in S$. Množina $C \subseteq \mathbb{R}^n$ je *konvexní*, jestliže pro každé $\mathbf{x}, \mathbf{y} \in C$ a $\theta \in [0, 1]$ platí $\theta \mathbf{x} + (1 - \theta) \mathbf{y} \in C$.

Konvexní kombinace dvou bodů tedy probíhá po úsečce mezi nimi. U více bodů vyjadřují nezáporné koeficienty jejich “váhy” a podmínka, že se sečtou na jednu, brání tomu, abychom z jejich společného obalu utekli.



Obrázek 2.1: Úsečka mezi dvěma body zůstává v konvexní množině.

Definice 2.2: Afinní kombinace, afinní a konvexní obal

Afinní kombinace bodů $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k)}$ má tvar $\sum_i \alpha_i \mathbf{x}^{(i)}$ s $\sum_i \alpha_i = 1$. Koeficienty nemusí být nezáporné. *Afinní obal* $\text{aff}(S)$ je množina všech konečných afinních kombinací bodů z S .

Konvexní obal $\text{conv}(S)$ je průnik všech konvexních množin, které obsahují S . Je to tedy nejmenší konvexní množina obsahující S .

Slovo *obal* zde znamená nejmenší objekt daného typu, který původní množinu obsahuje. Konvexní obal mezi body doplňuje všechny potřebné úsečky. Afinní obal je naopak protahuje do celých přímek, rovin a jejich vícerozměrných obdob.

Příklad 2.3: Konvexní obal tří bodů

Jsou-li tři body $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^2$ afinně nezávislé, je jejich konvexním obalem vyplněný trojúhelník s těmito vrcholy. Jejich afinním obalem je celá rovina $\mathbb{R}^2$.

Trojúhelník je názorný, definice však platí pro libovolnou množinu a v libovolné dimenzi. Následující lemma říká, že k vytvoření konvexního obalu stačí všechny konečné konvexní kombinace.

Lemma 2.4: Charakterizace konvexního obalu

Pro každou množinu $X \subseteq \mathbb{R}^n$ platí

$$\text{conv}(X) = \left\{ \sum_{i=1}^k \lambda_i \mathbf{x}^{(i)} \mid k \geq 1, \mathbf{x}^{(i)} \in X, \lambda_i \geq 0, \sum_{i=1}^k \lambda_i = 1 \right\}.$$

Důkaz lemmatu.

Označme množinu na pravé straně D . Každá konvexní množina obsahující X obsahuje i každou konečnou konvexní kombinaci bodů z X . Ukažme to indukcí podle počtu bodů. Pro jeden bod je tvrzení zřejmé a pro dva je přímo definicí konvexity. Je-li $\mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{x}^{(i)}$ a $\lambda_k < 1$, položíme

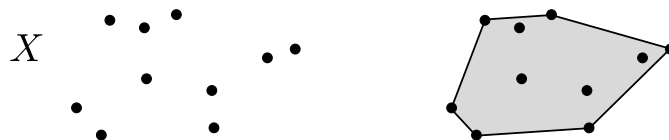
$$\mathbf{y} = \sum_{i=1}^{k-1} \frac{\lambda_i}{1 - \lambda_k} \mathbf{x}^{(i)}.$$

Koeficienty v $\mathbf{y}$ jsou opět nezáporné a jejich součet je jedna, takže podle indukčního předpokladu $\mathbf{y}$ v dané konvexní množině leží. Potom $\mathbf{x} = (1 - \lambda_k)\mathbf{y} + \lambda_k \mathbf{x}^{(k)}$ leží v téže množině. Příklad $\lambda_k = 1$ je okamžitý. Proto $D \subseteq \text{conv}(X)$.

Zbývá ověřit, že D je konvexní. Necht $\mathbf{x} = \sum_{i=1}^k \lambda_i \mathbf{x}^{(i)}$ a $\mathbf{y} = \sum_{j=1}^\ell \mu_j \mathbf{y}^{(j)}$ patří do D . Pro $\theta \in [0, 1]$ je

$$\theta \mathbf{x} + (1 - \theta) \mathbf{y} = \sum_{i=1}^k (\theta \lambda_i) \mathbf{x}^{(i)} + \sum_{j=1}^\ell ((1 - \theta) \mu_j) \mathbf{y}^{(j)}.$$

Všechny nové koeficienty jsou nezáporné a jejich součet je jedna, takže tento bod patří do D . Množina D je konvexní a obsahuje X . Z minimality konvexního obalu plyne $\text{conv}(X) \subseteq D$. ■



Obrázek 2.2: Konvexní obal konečné množiny bodů v $\mathbb{R}^2$.

1.2 Nadroviny, poloprostory a mnohostěny

Rovinné objekty ve vyšších dimenzích jsou základním stavebním materiálem LP. Jedna lineární nerovnost vymezí poloprostor. Všechny podmínky dohromady pak vyřiznou pří-

pustnou oblast.

Definice 2.5: Nadrovina a poloprostor

Nechť $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ a $\beta \in \mathbb{R}$. Množina

$$H(\mathbf{a}, \beta) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^\top \mathbf{x} = \beta\}$$

je *nadrovina*. Určuje dva uzavřené poloprostory, například

$$H^-(\mathbf{a}, \beta) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^\top \mathbf{x} \leq \beta\}.$$

Požadavek $\mathbf{a} \neq \mathbf{0}$ vylučuje prázdnou množinu a celý prostor, které nadrovinami nejsou.

Každá nadrovina tedy rozděluje prostor na dvě strany a je společnou hranicí obou odpovídajících uzavřených poloprostorů. Právě tak budeme číst jednotlivá omezení lineárního programu.

Definice 2.6: Afinní množina

Neprázdná množina $M \subseteq \mathbb{R}^n$ je *afinní*, jestliže je uzavřena na všechny afinní kombinace. Ekvivalentně má tvar $M = \mathbf{x}_0 + L$, kde L je lineární podprostor.

Konvexní kombinace a průniky spolu dobře fungují. Následující jednoduché tvrzení budeme v dalších kapitolách používat opakovaně.

Věta 2.7: Průnik konvexních množin je konvexní

Průnik libovolné rodiny konvexních množin v $\mathbb{R}^n$ je konvexní.

Důkaz věty.

Označme průnik $C = \bigcap_{i \in I} C_i$ a vezměme $\mathbf{x}, \mathbf{y} \in C$ a $\theta \in [0, 1]$. Pro každý index $i \in I$ platí $\mathbf{x}, \mathbf{y} \in C_i$. Z konvexity C_i tedy plyne $\theta \mathbf{x} + (1 - \theta) \mathbf{y} \in C_i$. Bod leží v každé množině rodiny, a proto i v jejich průniku C . ■

Tento jednoduchý fakt má dalekosáhlý důsledek: průnik konečně mnoha poloprostorů je vždy konvexní. Dostáváme tak geometrický objekt, s nímž se v lineárním programování budeme setkávat neustále.

Definice 2.8: Konvexní mnohostěn (polyedr)

Konvexní mnohostěn neboli *polyedr* je množina řešení konečného systému lineárních nerovnic,

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \leq \mathbf{b}\}.$$

Může být prázdný, omezený i neomezený. Omezený mnohostěn nazýváme *polytop*.

Rovnost lze zapsat jako dvě opačné nerovnosti, takže definice zahrnuje i množiny popsané rovnicemi a nerovnicemi současně. Každý mnohostěn je uzavřený a konvexní. Český termín *mnohostěn* zde tedy neznačí pouze omezené těleso. Stejnou konvenci používá také Matoušek (2006).

Všimněme si zejména, že mnohostěnem je i jediný poloprostor. Mnohostěn tedy nemusí připomínat uzavřenou krabici a může se táhnout do nekonečna. Je-li naopak omezený, používáme pro něj název *polytop*.

Definice 2.9: Dimenze mnohostěnu

Pro neprázdný mnohostěn P definujeme

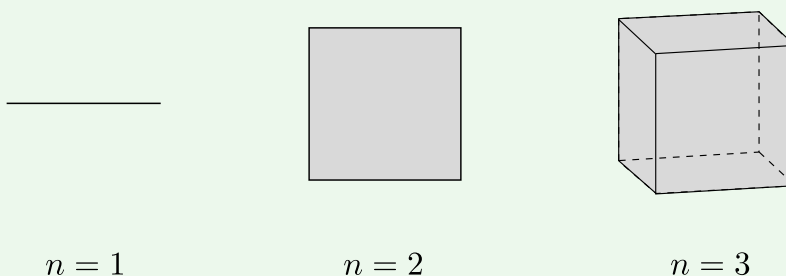
$$\dim P := \dim \text{aff}(P).$$

Ekvivalentně je $\dim P$ největší d , pro které P obsahuje $d + 1$ afinně nezávislých bodů. Dimenze se vztahuje k afinnímu obalu, nikoli nutně k okolnímu prostoru $\mathbb{R}^n$.

Následující známá tělesa ukazují, že jednou definicí zachytíme velmi různé tvary. Zároveň připomínají, jak lze geometrický objekt přeložit do konečné soustavy lineárních rovnic a nerovnic.

Příklad 2.10: Příklady konvexních mnohostěňů

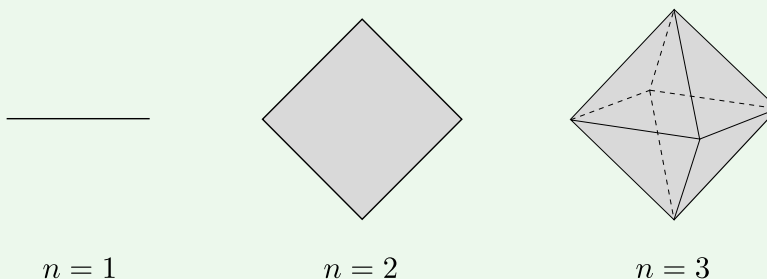
- Krychle $[-1, 1]^n$ je popsána $2n$ nerovnostmi $-1 \leq x_i \leq 1$.



- Křížový mnohostěn $\{\mathbf{x} \mid \sum_{i=1}^n |x_i| \leq 1\}$ je skutečně polyedr, neboť jej lze ekvivalentně popsat konečnou soustavou

$$\boldsymbol{\sigma}^T \mathbf{x} \leq 1 \quad \text{pro všechna } \boldsymbol{\sigma} \in \{-1, 1\}^n.$$

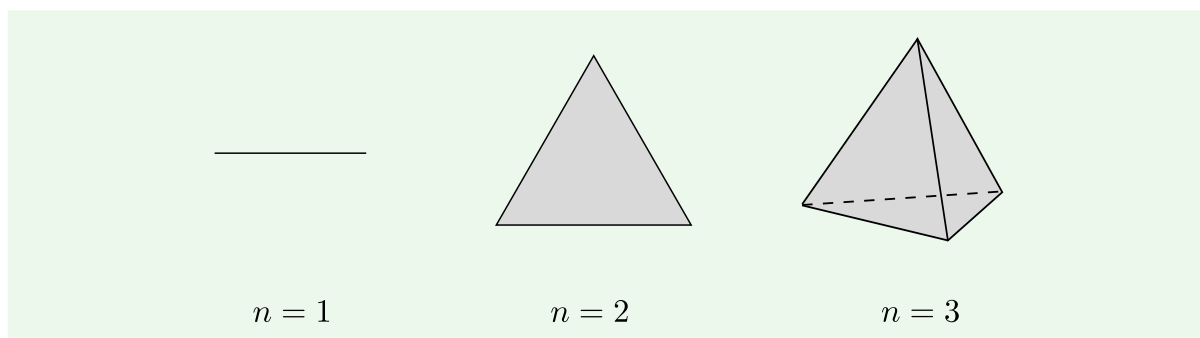
Pro $n = 3$ jde o pravidelný osmistěn.



- Pravděpodobnostní simplex

$$\Delta_n = \left\{ \mathbf{x} \in \mathbb{R}^{n+1} \mid \sum_{i=1}^{n+1} x_i = 1, x_i \geq 0 \right\}$$

má dimenzi n a vrcholy tvoří kanonické vektory $\mathbf{e}_1, \dots, \mathbf{e}_{n+1}$. Všimněme si, že jde přímo o přípustnou množinu lineárního programu s jedinou rovností a podmínkami nezápornosti.



Vrcholy mají mezi body mnohostěnu výsadní postavení: jsou to jeho “rohy” či “špičky”. Nechceme však spoléhat jen na obrázek, protože ve vyšší dimenzi už jej nakreslit nedokážeme. Potřebujeme definici, která pozná vrchol čistě z vlastností bodů množiny.

Definice 2.11: Vrchol konvexního mnohostěnu

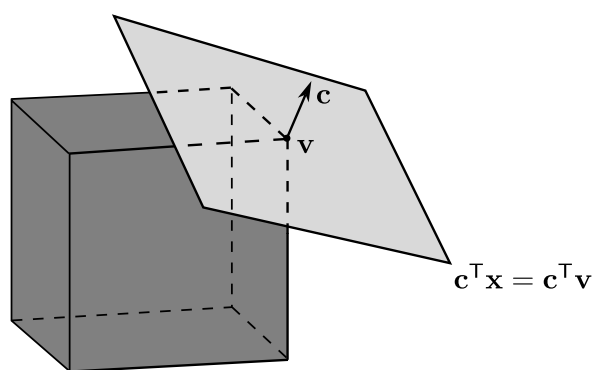
Bod $\mathbf{v} \in P$ je *vrcholem* (neboli extrémním bodem) konvexního mnohostěnu P , jestliže z rovnosti

$$\mathbf{v} = \theta \mathbf{x} + (1 - \theta) \mathbf{y}, \quad \mathbf{x}, \mathbf{y} \in P, \quad 0 < \theta < 1,$$

vždy plyne $\mathbf{x} = \mathbf{y} = \mathbf{v}$.

Vrchol tedy neleží uvnitř žádné netriviální úsečky obsažené v P . Pro mnohostěny platí také ekvivalentní charakterizace: $\mathbf{v}$ je jediným maximalizátorem nějaké lineární funkce na P . Tato ekvivalence je specifickou vlastností polyedrických množin. Pro obecnou uzavřenou konvexní množinu nemusí být každý extrémní bod takto exponovaný (Ziegler, 1995, Schrijver, 1986, Bertsimas and Tsitsiklis, 1997, Rockafellar, 1970).

Geometricky si druhou charakterizaci můžeme představit tak, že mnohostěnem posouváme podpůrnou nadrovinu kolmo ke zvolenému vektoru. Je-li jejím posledním dotykem jediný bod, exponovala právě vrchol. Tento obraz se v následující kapitole promění v grafickou metodu řešení LP.



Obrázek 2.3: Podpůrná nadrovina exponující vrchol mnohostěnu.

2 Bázecká přípustná řešení

Od této chvíle pracujeme s rovnicovým tvarem. Z lineární algebry víme, že je-li soustava $A\mathbf{x} = \mathbf{b}$ konzistentní, tvoří její řešení afinní množinu. Podmínka $\mathbf{x} \geq \mathbf{0}$ ji ořízne nezápor-

ným ortantem¹. Jejich průnikem je množina

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

kde $A \in \mathbb{R}^{m \times n}$, $m \leq n$, a

$$\text{rank } A = m. \quad (2.1)$$

Lineárně závislé řádky lze po kontrole konzistence odstranit. Právě tato kontrola je důležitá: závislý levý člen s neslučitelnou pravou stranou nepředstavuje nadbytečnou rovnici, ale důkaz nepřipustnosti. Podmínka plné řádkové hodnosti pak zaručuje, že v matici A lze vybrat m lineárně nezávislých sloupců. Množina P může být prázdná. V takovém případě nemá žádné bázecké přípustné řešení.

Přípustná množina LP je mnohostěn

Je-li soustava $A\mathbf{x} = \mathbf{b}$ konzistentní, vymezují její rovnosti afinní množinu a nezápornost $\mathbf{x} \geq \mathbf{0}$ nezáporný ortant. Jejich průnik je konvexní mnohostěn. Není-li soustava konzistentní, je přípustná množina prázdná, a tedy podle naší definice rovněž mnohostěn. Stejný závěr platí přímo i pro nerovnicový tvar LP.

Mezi přípustnými body budou mít výsadní postavení takzvaná bázecká přípustná řešení. Jejich konstrukce má jednoduchý recept: nebázecké složky položíme na nulu a bázecké dopočítáme ze čtvercového systému. Nejprve tento recept zapíšeme přesně.

Definice 2.12: Výběr sloupců a báze

Pro množinu indexů $B \subseteq \{1, \dots, n\}$ značí A_B matici tvořenou odpovídajícími sloupci A , zapsanými v rostoucím pořadí indexů, a $\mathbf{x}_B$ odpovídající podvektor. Množinu B nazveme *bází*, jestliže $|B| = m$ a čtvercová matice A_B je regulární. Doplněk $N = \{1, \dots, n\} \setminus B$ tvoří *nebázecké* indexy.

Slovo *báze* zde používáme jako pohodlnou zkratku. Přesněji říkáme, že sloupce matice A_B tvoří bázi sloupcového prostoru matice A .

Definice 2.13: Bázecké přípustné řešení (BFS)

Báze B určuje *bázecké řešení*

$$\mathbf{x}_N = \mathbf{0}, \quad \mathbf{x}_B = A_B^{-1}\mathbf{b}.$$

Je-li $\mathbf{x}_B \geq \mathbf{0}$, jde o *bázecké přípustné řešení* (BFS) a B je *přípustná báze*. Proměnné s indexy v B nazýváme *bázecké* a ostatní *nebázecké*.

V české literatuře se lze setkat i se zkratkou BPŘ. V této knize zachováme mezinárodně běžnou zkratku BFS, ale používáme český termín *bázecké přípustné řešení*.

¹Ortant je vícerozměrné zobecnění kvadrantu v $\mathbb{R}^2$ a oktantu v $\mathbb{R}^3$.

Příklad 2.14: Bázecké přípustné řešení

Pro

$$A = \begin{pmatrix} 1 & 5 & 3 & 4 & 6 \\ 0 & 1 & 3 & 5 & 6 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 14 \\ 7 \end{pmatrix}, \quad B = \{2, 4\}$$

je $A_B = \begin{pmatrix} 5 & 4 \\ 1 & 5 \end{pmatrix}$ regulární a $A_B^{-1}\mathbf{b} = (2, 1)^T$. Nebázecké indexy jsou $N = \{1, 3, 5\}$, a proto na ně doplníme nuly. Dostaneme

$$\mathbf{x} = (0, 2, 0, 1, 0)^T.$$

Všechny složky jsou nezáporné a přímým dosazením ověříme $A\mathbf{x} = A_B\mathbf{x}_B = \mathbf{b}$. Nezůstali jsme tedy jen u bázeckého kandidáta: skutečně jsme získali BFS.

Lemma 2.15: Kritérium pomocí nosiče

Pro přípustný bod $\mathbf{x} \in P$ položme $K = \{j \mid x_j > 0\}$. Bod $\mathbf{x}$ je bázecký právě tehdy, když jsou sloupce A_K lineárně nezávislé.

Důkaz lemmatu.

Je-li $\mathbf{x}$ určen přípustnou bází B , pak $K \subseteq B$, a tedy jsou sloupce A_K lineárně nezávislé.

Naopak předpokládejme, že A_K má lineárně nezávislé sloupce. Z toho plyne $|K| \leq m$. Je-li $|K| = m$, vezmeme rovnou $B = K$. Je-li $|K| < m$, z $\text{rank } A = m$ plyne, že lze tento nezávislý výběr doplnit dalšími sloupci matice A na m -prvkovou bázi B . Je to obvyklé doplnění lineárně nezávislé množiny na bázi známé z lineární algebry. Protože $K \subseteq B$, jsou všechny složky mimo B nulové. Rovnost $A\mathbf{x} = \mathbf{b}$ pak dává $A_B\mathbf{x}_B = \mathbf{b}$, takže $\mathbf{x}_B = A_B^{-1}\mathbf{b}$ a $\mathbf{x}$ je BFS. ■

Věta 2.16: Kandidát určený bází je jednoznačný

Každá báze B určuje právě jedno bázecké řešení. Toto řešení je BFS právě tehdy, když $A_B^{-1}\mathbf{b} \geq \mathbf{0}$.

Důkaz věty.

Rozepišme levou stranu jako $A\mathbf{x} = A_B\mathbf{x}_B + A_N\mathbf{x}_N$. Po položení nebázeckých proměnných $\mathbf{x}_N = \mathbf{0}$ se rovnice redukuje na $A_B\mathbf{x}_B = \mathbf{b}$. A právě zde se využije regulárnost A_B : soustava má jediné řešení $\mathbf{x}_B = A_B^{-1}\mathbf{b}$. Je-li tento vektor nezáporný, dostaneme po doplnění nulami právě jedno BFS. Obsahuje-li zápornou složku, báze neurčuje žádné přípustné bázecké řešení. ■

Poznámka 2.17: BFS a účelová funkce

Všimněme si, že v definici BFS se vůbec nevyskytuje vektor $\mathbf{c}$. Bázecká přípustná řešení jsou čistě geometrickou vlastností množiny P , nikoli vlastností konkrétní účelové funkce. Každá jednotlivá báze určuje nejvýše jeden přípustný bod. Platí to ale i obráceně – musí jednomu bodu odpovídat jen jedna báze? Nemusí. Právě degenerace dovolí, aby tentýž bod měl několik různých bázeckých popisů.

Definice 2.18: Degenerace bázeckého přípustného řešení

BFS $\mathbf{x}$ je *nedegenerované*, jestliže má právě m kladných složek, tedy $|\{j \mid x_j > 0\}| = m$. Je *degenerované*, jestliže má méně než m kladných složek. Ekvivalentně je při každé příslušné bázi alespoň jedna bázecká proměnná rovna nule. Úloha je degenerovaná, pokud má alespoň jedno degenerované BFS.

Příklad 2.19: Různé báze v degenerované úloze

Uvažujme

$$A = \begin{pmatrix} 1 & 2 & 1 & 2 & 1 \\ 3 & 0 & 2 & 1 & 1 \\ 2 & 2 & 1 & 1 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 5 \\ 8 \\ 7 \end{pmatrix}, \quad P = \{\mathbf{x} \geq \mathbf{0} \mid A\mathbf{x} = \mathbf{b}\}.$$

Projděme čtyři volby báze. Na jednom příkladu tak uvidíme rozdíl mezi nedegenerovaným BFS, nepřípustným bázeckým řešením a degenerovaným bodem s více bázemi.

Báze $B_1 = \{1, 2, 3\}$. Soustava $A_{B_1}\mathbf{x}_{B_1} = \mathbf{b}$ zní

$$a + 2b + c = 5, \quad 3a + 2c = 8, \quad 2a + 2b + c = 7.$$

Jejím řešením je $(a, b, c) = (2, 1, 1)$. Po doplnění nul na nebázecké pozice dostaneme $\mathbf{x}^{(1)} = (2, 1, 1, 0, 0)^\top$. Všechny tři bázecké složky jsou kladné, takže jde o nedegenerované BFS.

Báze $B_2 = \{2, 3, 4\}$. Nyní řešíme

$$2a + b + 2c = 5, \quad 2b + c = 8, \quad 2a + b + c = 7.$$

Vyjde $(a, b, c) = (2, 5, -2)$, tedy bázecké řešení $\mathbf{x}^{(2)} = (0, 2, 5, -2, 0)^\top$. Záporná čtvrtá složka znamená, že tento kandidát není přípustný.

Báze $B_3 = \{1, 3, 5\}$. Výsledkem je $\mathbf{x}^{(3)} = (0, 0, 3, 0, 2)^\top$. Bod je přípustný, ale jeho nosič je pouze $\{3, 5\}$ a má velikost $2 < m = 3$. Nulová první bázecká proměnná tedy prozrazuje degeneraci.

Báze $B_4 = \{2, 3, 5\}$. Výpočet vede znovu k témuž bodu:

$$\mathbf{x}^{(4)} = (0, 0, 3, 0, 2)^\top = \mathbf{x}^{(3)}.$$

Druhá bázecká proměnná je tentokrát nulová. Stejný degenerovaný bod jsme tedy popsali dvěma různými bázemi. Příklad zároveň ukazuje, že opakování bodu je typickým projevem degenerace, nikoli její definicí.

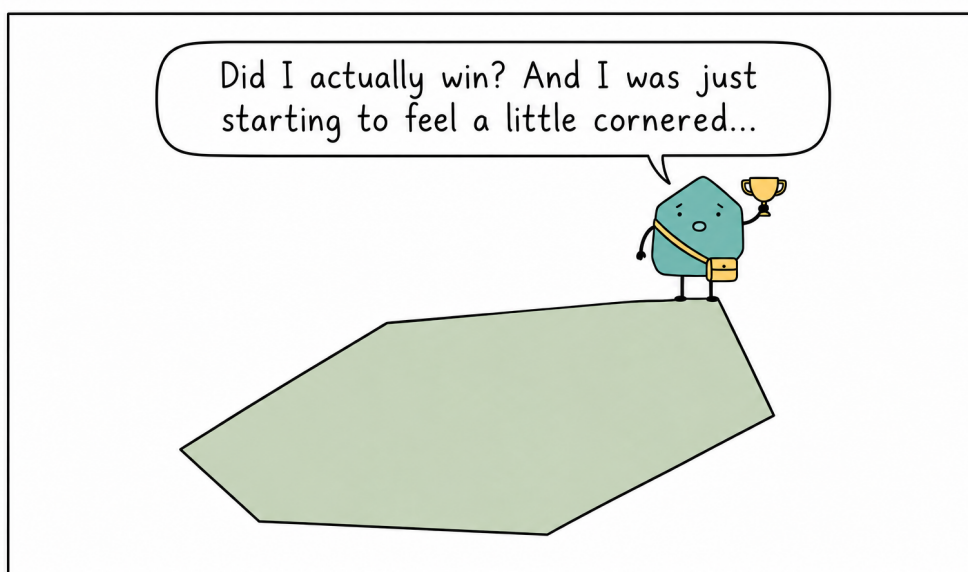
Podrobnou polyedrickou teorii lze nalézt u Schrijver (1986). Pro návaznost na simplexovou metodu viz Bertsimas and Tsitsiklis (1997).

KAPITOLA 3

ZÁKLADNÍ VĚTA LP A GRAFICKÉ ŘEŠENÍ

1	Základní věta lineárního programování	30
2	Grafické řešení úloh	33

V této kapitole propojíme algebru bázických přípustných řešení s geometrií vrcholů. Hledat maximum mezi nekonečně mnoha přípustnými body vypadá na první pohled beznadějně. Základní věta lineárního programování však ukáže, že při konečném optimu stačí obrátit pozornost k “rohům” přípustné oblasti. Dokážeme, že alespoň jedno bázické přípustné řešení je optimální, a poté tento princip uvidíme přímo na obrázcích. Přesně přitom odlišíme jediné optimum, optimální stěnu, nepřípustnost a neomezenost.



1 Základní věta lineárního programování

Z předchozích kapitol už máme všechny stavební kameny: rovnicový tvar, konvexní množiny a bázecká přípustná řešení. Teď je spojíme. Výsledkem bude věta, která z geometrického pozorování “optimum hledej v rohu” udělá přesný matematický nástroj.

Uvažujme přípustnou množinu

$$P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\},$$

kde $A \in \mathbb{R}^{m \times n}$ má plnou řádkovou hodnost $\text{rank } A = m$. Tento předpoklad je společný s definicí BFS v kapitole 2.

Budeme opakovaně používat kritérium pomocí nosiče z předchozí kapitoly: přípustný bod $\mathbf{x} \in P$ je BFS právě tehdy, když jsou sloupce A_K s indexy

$$K = \{j \mid x_j > 0\}$$

lineárně nezávislé.

Věta 3.1: Základní věta lineárního programování

Je-li $P \neq \emptyset$ a lineární funkce $\mathbf{c}^\top \mathbf{x}$ je na P shora omezená, pak

- (i) maximum je na P dosaženo,
- (ii) alespoň jedno bázecké přípustné řešení je optimální.

Důkaz věty.

V důkazu použijeme obrat, který se později v trochu praktičtější podobě objeví v simplexové metodě: budeme se pohybovat v množině přípustných řešení tak dlouho, až některá kladná složka klesne na nulu.

Nejprve ukážeme pomocné tvrzení: ke každému $\mathbf{x}^{(0)} \in P$ existuje BFS $\hat{\mathbf{x}} \in P$ splňující $\mathbf{c}^\top \hat{\mathbf{x}} \geq \mathbf{c}^\top \mathbf{x}^{(0)}$.

Proč by z něj měla plynout celá věta? Báze je jen konečně mnoho, a tedy je konečně mnoho BFS. Pokud můžeme každý přípustný bod nahradit alespoň stejně dobrým BFS, stačí mezi těmito konečně mnoha kandidáty vybrat nejlepší. Zbývá tedy pomocné tvrzení dokázat.

Mezi přípustnými body s účelovou hodnotou alespoň $\mathbf{c}^\top \mathbf{x}^{(0)}$ vyberme bod $\hat{\mathbf{x}}$ s nejmenším počtem kladných složek. Takový počet existuje, protože může nabývat jen hodnot $0, 1, \dots, n$. Označme nosič

$$K = \{j \mid \hat{x}_j > 0\}.$$

Jsou-li sloupce A_K lineárně nezávislé, je $\hat{\mathbf{x}}$ podle kritéria pomocí nosiče BFS.

Předpokládejme tedy, že sloupce A_K jsou závislé. To je okamžik, kdy se můžeme z bodu pohnout, aniž porušíme rovnice. Existuje totiž nenulový vektor $\mathbf{w}$ s nosičem obsaženým v K takový, že $A\mathbf{w} = \mathbf{0}$. Změnou znaménka můžeme zajistit $\mathbf{c}^\top \mathbf{w} \geq 0$.

Vektor $\mathbf{w}$ musí mít zápornou složku. Kdyby $\mathbf{w} \geq \mathbf{0}$ a $\mathbf{c}^\top \mathbf{w} > 0$, pak by body $\hat{\mathbf{x}} + t\mathbf{w}$, $t \geq 0$, byly přípustné a jejich účelová hodnota by rostla nade všechny meze, což odporuje omezenosti. Kdyby $\mathbf{w} \geq \mathbf{0}$ a $\mathbf{c}^\top \mathbf{w} = 0$, nahradíme $\mathbf{w}$ vektorem $-\mathbf{w}$. Účelová hodnota směru zůstane nulová a vznikne záporná složka.

Položme nyní

$$t^* = \min_{j: w_j < 0} \frac{\hat{x}_j}{-w_j} > 0.$$

Bod $\mathbf{x}' = \hat{\mathbf{x}} + t^* \mathbf{w}$ splňuje $A\mathbf{x}' = \mathbf{b}$ a $\mathbf{x}' \geq \mathbf{0}$. Alespoň jedna původně kladná složka se stala nulovou, žádná nová kladná složka mimo K nevznikla a

$$\mathbf{c}^\top \mathbf{x}' = \mathbf{c}^\top \hat{\mathbf{x}} + t^* \mathbf{c}^\top \mathbf{w} \geq \mathbf{c}^\top \hat{\mathbf{x}}.$$

Poměrový test tedy zastaví pohyb přesně ve chvíli, kdy první složka narazí na nulu. Dostali jsme stejně dobrý přípustný bod s menším nosičem, což odporuje volbě $\hat{\mathbf{x}}$. Sloupce A_K proto musejí být nezávislé a $\hat{\mathbf{x}}$ je BFS.

Bázi je nejvýše $\binom{n}{m}$, takže existuje jen konečně mnoho různých BFS. Vybereme-li z nich bod s největší účelovou hodnotou, předchozí odstavec zaručuje, že není horší než žádný přípustný bod. Tento bod je optimální, čímž jsou dokázány obě části věty. ■

Kompaktnost přípustné množiny jsme v důkazu nepotřebovali. Mnohostěn P může být neomezený. Stačí, aby na něm byla shora omezená konkrétní účelová funkce. Věta je standardním výsledkem polyedrické teorie (Matoušek and Gärtner, 2007, Bertsimas and Tsitsiklis, 1997).

Základní věta zatím mluví algebraickým jazykem bází. Geometrický obraz ale sliboval “rohy” mnohostěnu. Následující výsledek potvrzuje, že nejde o dvě různé představy: vrcholy a bázecká přípustná řešení jsou tytéž body, pouze popsány dvěma jazyky.

Věta 3.2: Vrcholy jsou právě bázecká přípustná řešení

Pro $\mathbf{v} \in P$ jsou ekvivalentní podmínky:

- (i) $\mathbf{v}$ je vrchol mnohostěnu P ,
- (ii) $\mathbf{v}$ je bázecké přípustné řešení.

Důkaz věty.

Nejprve: BFS je vrchol. Necht je $\mathbf{v}$ BFS určené bází B a označme $N = \{1, \dots, n\} \setminus B$. Předpokládejme, že

$$\mathbf{v} = \theta \mathbf{x} + (1 - \theta) \mathbf{y}, \quad \mathbf{x}, \mathbf{y} \in P, \quad 0 < \theta < 1.$$

Pro $j \notin B$ je $v_j = 0$. Z nezápornosti x_j, y_j plyne $x_j = y_j = 0$, a tedy $\mathbf{x}_N = \mathbf{y}_N = \mathbf{0}$. Rovnice přípustnosti dávají $A_B \mathbf{x}_B = \mathbf{b} = A_B \mathbf{y}_B$. Regulárnost A_B proto implikuje $\mathbf{x}_B = \mathbf{y}_B = \mathbf{v}_B$. Bod $\mathbf{v}$ je vrchol.

Nyní: nebázecký bod není vrchol. Naopak necht $\mathbf{v}$ není BFS a označme $K = \{j \mid v_j > 0\}$. Podle kritéria pomocí nosiče jsou sloupce A_K lineárně závislé. Existuje tedy nenulový $\mathbf{w}$ s nosičem v K a $A\mathbf{w} = \mathbf{0}$. Protože všechny v_j pro $j \in K$ jsou kladné, lze zvolit $\varepsilon > 0$ tak malé, že $\mathbf{v} \pm \varepsilon \mathbf{w} \geq \mathbf{0}$. Oba body jsou přípustné, různé a

$$\mathbf{v} = \frac{1}{2}(\mathbf{v} + \varepsilon \mathbf{w}) + \frac{1}{2}(\mathbf{v} - \varepsilon \mathbf{w}).$$

Bod $\mathbf{v}$ tedy není vrchol. ■

Důkaz zahrnuje i případ $n = m$. Tehdy je při plné hodnosti jedinou možnou bází celá množina sloupců a soustava $A\mathbf{x} = \mathbf{b}$ má nejvýše jedno řešení. Je-li nezáporné, je současně BFS i vrcholem.

Poznámka 3.3: Optimální stěna, více optim a degenerace

Je-li $\mathbf{c} \neq \mathbf{0}$ a má-li úloha konečné optimum, představme si podpůrnou nadrovinu, kterou posouváme ve směru $\mathbf{c}$, až se přípustné množiny dotkne naposledy. Posledním dotykem nemusí být jen jeden roh. Může jím být celá hrana, ploška, nebo dokonce neomezená stěna.

Má-li úloha konečnou optimální hodnotu z^* , množina

$$F^* = P \cap \{\mathbf{x} \mid \mathbf{c}^T \mathbf{x} = z^*\}$$

je *optimální stěna*. Je-li $\mathbf{c} \neq \mathbf{0}$, nadrovina v tomto zápisu je podpůrná: celý mnohostěn leží v poloprostoru $\mathbf{c}^T \mathbf{x} \leq z^*$. Je-li $\mathbf{c} = \mathbf{0}$, jsou optimální všechny přípustné body a $F^* = P$.

Dvě různá optimální BFS vždy určují celou úsečku optimálních řešení. Obrácená implikace však pro neomezený mnohostěn neplatí. Například množina $P = \{(x_1, x_2) \geq 0 \mid x_1 = 1\}$ má pro účelovou funkci x_1 nekonečně mnoho optimálních bodů, ale jen jeden vrchol $(1, 0)$. Je-li optimální stěna omezená, pak při kladné dimenzi obsahuje alespoň dva vrcholy.

Degenerace je jiný jev než nejednoznačnost optima: popisuje počet kladných složek BFS vzhledem k hodnotě omezení. Degenerované BFS může být jediným optimem a naopak optimální hrana může mít nedegenerované koncové body.

Enumerace bází. Základní věta okamžitě nabízí jednoduchý algoritmus. Není chytrý, ale dobře ukazuje, co jsme větou získali:

1. projdeme všechny m -prvkové množiny $B \subseteq \{1, \dots, n\}$,
2. u každé otestujeme regulárnost A_B , spočteme $\mathbf{x}_B = A_B^{-1} \mathbf{b}$ a mimo B doplníme nuly,
3. nezáporné kandidáty si ponecháme a porovnáme jejich účelové hodnoty.

Pokud nezískáme žádný kandidát, je úloha nepřípustná, protože neprázdné P by podle předchozího důkazu obsahovalo BFS. Háček je v počtu možností: $\binom{n}{m}$ může být obrovské číslo. Pro $n = 2m$ roste přibližně jako $4^m / \sqrt{\pi m}$. Konečný seznam kandidátů tedy ještě neznamená dobrý algoritmus – a právě zde později vstoupí na scénu simplexová metoda.

Samotná enumerace navíc nestačí k rozlišení konečného optima od neomezenosti. I neomezená úloha může mít mnoho BFS, zatímco účelová funkce uniká do nekonečna podél paprsku. Musíme proto umět odhalit také směr $\mathbf{d} \geq \mathbf{0}$ s $A\mathbf{d} = \mathbf{0}$ a $\mathbf{c}^T \mathbf{d} > 0$.

Věta 3.4: Základní věta LP – vše v jednom

Za předpokladu $\text{rank } A = m$ nastane právě jedna z možností:

- a) $P = \emptyset$,
- b) $P \neq \emptyset$ a existuje $\mathbf{d} \geq \mathbf{0}$ s $A\mathbf{d} = \mathbf{0}$ a $\mathbf{c}^T \mathbf{d} > 0$, takže úloha je neomezená,
- c) existuje konečné optimum a alespoň jedno optimální BFS, které je současně vrcholem P .

Důkaz věty.

Možnost a) se vylučuje s oběma zbývajících. Existuje-li při $P \neq \emptyset$ směr $\mathbf{d}$ z bodu b), potom pro libovolné $\mathbf{x}^0 \in P$ a každé $t \geq 0$ platí

$$A(\mathbf{x}^0 + t\mathbf{d}) = \mathbf{b}, \quad \mathbf{x}^0 + t\mathbf{d} \geq \mathbf{0},$$

zatímco $\mathbf{c}^T(\mathbf{x}^0 + t\mathbf{d}) \rightarrow +\infty$. Možnost b) tedy skutečně znamená neomezenost a nemůže nastat současně s c).

Zbývá ukázat, že při neprázdném P jiná situace nenastane. Použijeme redukci nosiče z důkazu základní věty. Začneme libovolným $\mathbf{x} \in P$. Jsou-li sloupce matice A s indexy v kladném nosiči $\mathbf{x}$ závislé, existuje nenulový vektor $\mathbf{w}$ s nosičem v něm obsaženým a $A\mathbf{w} = \mathbf{0}$. Jeho znaménko zvolíme tak, aby $\mathbf{c}^T\mathbf{w} \geq 0$. Má-li $\mathbf{w}$ zápornou složku, položíme

$$t = \min_{j:w_j < 0} \frac{x_j}{-w_j} > 0 \quad \text{a} \quad \mathbf{x} \leftarrow \mathbf{x} + t\mathbf{w}.$$

Účelová hodnota neklesne, přípustnost se zachová a nosič se zmenší. Nemá-li $\mathbf{w}$ zápornou složku a $\mathbf{c}^T\mathbf{w} > 0$, našli jsme směr z bodu b). Je-li tento skalární součin nulový, nahradíme $\mathbf{w}$ vektorem $-\mathbf{w}$ a opět zmenšíme nosič. Jsou-li naopak sloupce kladného nosiče nezávislé, je aktuální bod podle kritéria pomocí nosiče již BFS.

Nenastane-li b), konečně mnoha opakováními dojdeme z každého přípustného bodu k BFS s nejméně stejnou účelovou hodnotou. Bází je konečně mnoho, takže jejich největší účelová hodnota omezuje shora všechny body P . Základní věta pak dává konečné optimum i optimální BFS, tedy c). Alternativa je tím úplná (Bertsimas and Tsitsiklis, 1997, Schrijver, 1986). ■

2 Grafické řešení úloh

V rovině můžeme základní větu doslova vidět. Narýsujeme průnik poloprostorů a potom posouváme vrstevnici účelové funkce ve směru zlepšení tak dlouho, dokud se přípustné oblasti ještě dotýká. Grafická metoda samozřejmě není praktickým algoritmem pro vyšší dimenze, ale pro budování intuice je těžké najít lepší obrázek toho, co lineární program dělá.

Grafická metoda v $\mathbb{R}^2$

Je-li $\mathbf{c} = \mathbf{0}$, je každý přípustný bod optimální. V opačném případě postupujeme následovně.

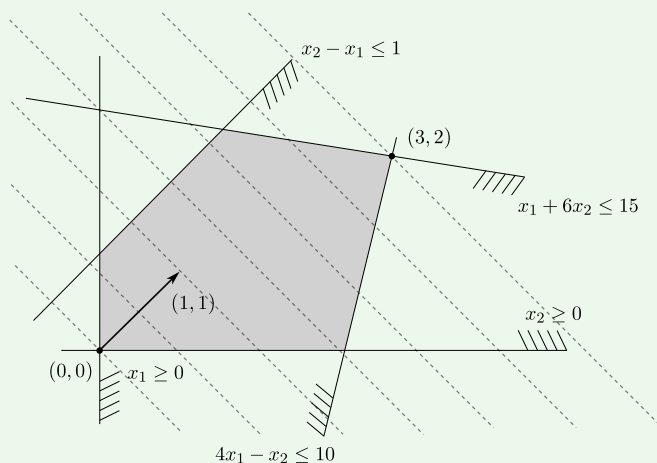
1. Pro každé omezení sestrojíme hraniční přímku a určíme správný poloprostor. Jejich průnik je přípustná oblast P .
2. Sestrojíme jednu vrstevnici $\mathbf{c}^T\mathbf{x} = \alpha$. Pro maximalizaci roste hodnota ve směru normálového vektoru $\mathbf{c}$.
3. Přímku posouváme rovnoběžně ve směru růstu. Při konečném optimu je poslední dotyk optimální stěnou. Může jít o jediný vrchol nebo o celou hranu.
4. Souřadnice kandidátů ověříme řešením příslušných rovnic a dosazením do všech omezení i do účelové funkce.

Příklad 3.5: Čtyři možné výsledky grafického řešení

Začneme velmi jednoduchou úlohou, na níž postupně vyvoláme všechny čtyři základní situace. Maximalizujme $x_1 + x_2$ při omezeních

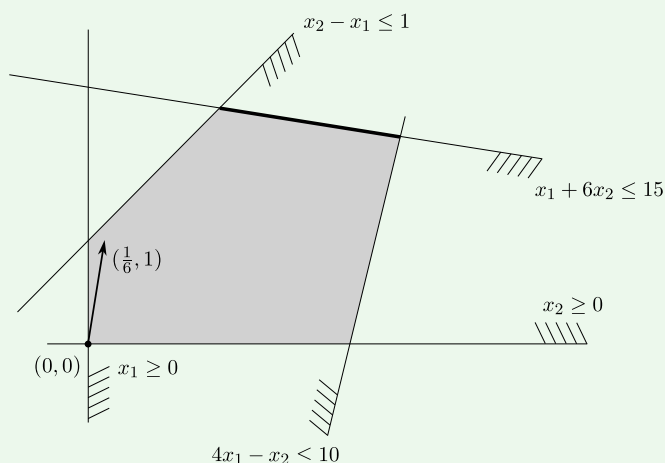
$$x_1, x_2 \geq 0, \quad x_2 - x_1 \leq 1, \quad x_1 + 6x_2 \leq 15, \quad 4x_1 - x_2 \leq 10.$$

Průnik pěti poloprostorů tvoří konvexní mnohoúhelník. Představme si přímku $x_1 + x_2 = \alpha$, kolmou na vektor $\mathbf{c} = (1, 1)^T$. Když ji posouváme ve směru tohoto vektoru, hodnota α roste. Poslední dotyk s přípustnou oblastí nastane v průsečíku posledních dvou hraničních přímek, tedy v bodě $(3, 2)$. Bod splňuje všechna omezení a dosahuje hodnoty 5.



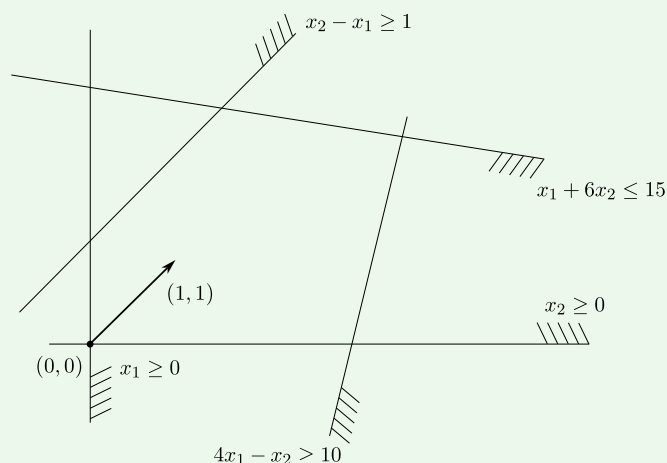
Obrázek 3.1: Jediné optimum v bodě $(3, 2)$.

Změníme-li účelovou funkci na $\frac{1}{6}x_1 + x_2$, jsou její vrstevnice rovnoběžné s hranou $x_1 + 6x_2 = 15$. Optimální je celá příslušná hrana, nikoli důsledkem degenerace, ale proto, že účelová funkce je na této stěně konstantní.



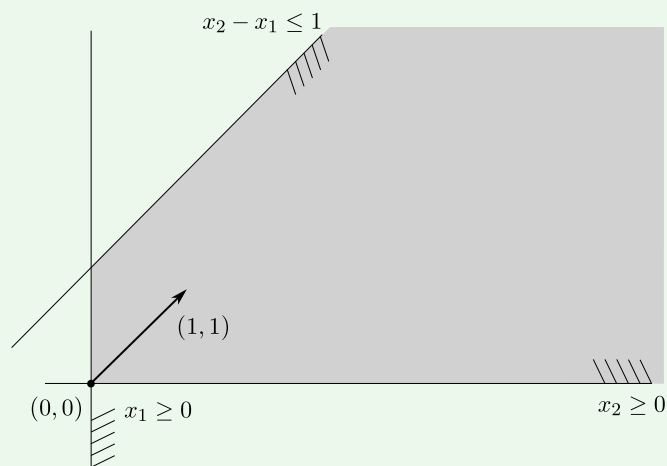
Obrázek 3.2: Optimální hrana a nekonečně mnoho optimálních bodů.

Nahradíme-li dvě šikmá omezení podmínkami $x_2 - x_1 \geq 1$ a $4x_1 - x_2 \geq 10$ a ostatní omezení zachováme, vznikne prázdný průnik. Úloha je nepřípustná.



Obrázek 3.3: Neslučitelná omezení a prázdná přípustná množina.

Odstraníme-li naopak omezení $x_1 + 6x_2 \leq 15$ a $4x_1 - x_2 \leq 10$, zůstává přípustný paprsek $(x_1, x_2) = (t, t)$ pro $t \geq 0$. Hodnota $x_1 + x_2 = 2t$ roste bez omezení.



Obrázek 3.4: Neomezená přípustná oblast i účelová funkce.

Poznámka 3.6: Co z obrázku skutečně plyne

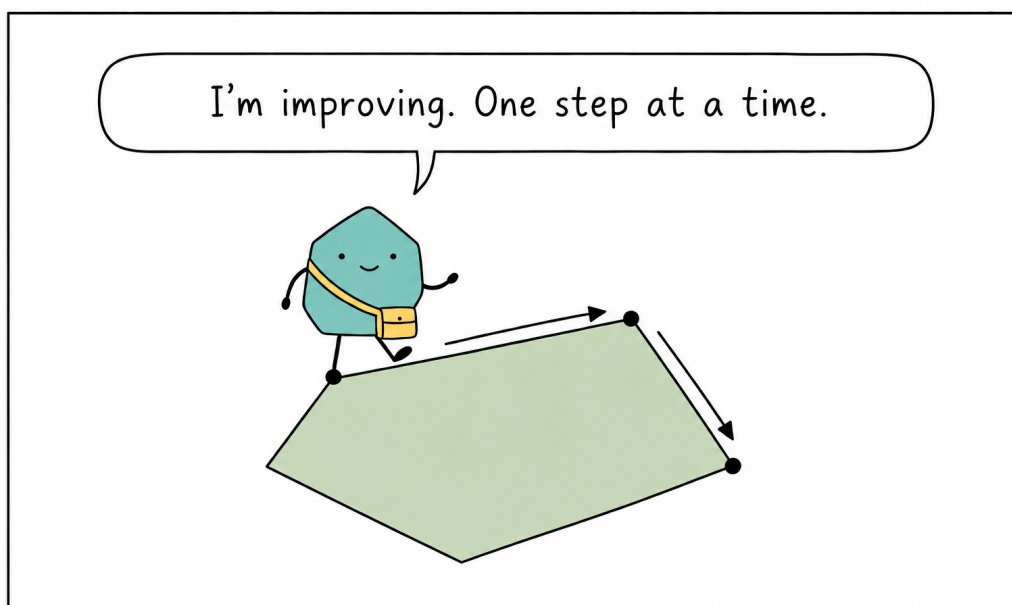
Omezenost přípustné oblasti zaručuje omezenost každé lineární účelové funkce. Obráceně to neplatí: neomezený mnohostěn může mít pro konkrétní účelovou funkci konečné optimum. Při grafickém řešení proto sledujeme nejen tvar P , ale i směr $\mathbf{c}$.

KAPITOLA 4

SIMPLEXOVÝ ALGORITMUS – ZÁKLADY METODY

1	Úvodní příklad bez potíží	38
2	Neomezenost úlohy a více optimálních řešení	41
3	Simplexové tabulky	43
4	Simplexová metoda – obecné základy	44

Tato kapitola představuje výpočetní jádro lineárního programování: simplexovou metodu pro maximalizační úlohu v rovnice-
vém tvaru. Začneme konkrétním výpočtem a teprve potom z něj vytáhneme obecný postup. Odvodíme simplexovou tabulku, kritérium optimality, poměrový test a jeden korektní pivot. Na příkladech také ukážeme, jak tabulka sama prozradí neomezenost nebo celou úsečku optimálních řešení. Degeneraci, cyklení a pravidla zaručující konečnost probereme v kapitole 6.



1 Úvodní příklad bez potíží

Simplexovou metodu zavedl George B. Dantzig roku 1947 (Dantzig, 1963, 2002). Geometricky si ji můžeme představit jako cestu po hranách přípustného mnohostěnu: v každém kroku se rozhlédneme po sousedních vrcholech, vybereme směr zlepšení a po jedné hraně se vydáme dál. Algebraicky přitom měníme jediný sloupec báze a upravujeme ekvivalentní soustavu rovnic podobně jako při Gaussově eliminaci.

Tento obrázek je velmi užitečný, potřebuje však jednu důležitou poznámku. V nede-generovaném kroku skutečně přejdeme do sousedního vrcholu a účelovou hodnotu přísně zlepšíme. Při degeneraci se může změnit báze, aniž se změní bod nebo účelová hodnota. Simplex se tedy někdy “otočí v jednom vrcholu”, aniž by se z něj zatím pohnul.

Budeme používat úlohu v *maximalizačním rovnicovém tvaru*

$$\max \mathbf{c}^T \mathbf{x} \quad \text{při} \quad A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0},$$

kde $A \in \mathbb{R}^{m \times n}$ má řádky lineárně nezávislé.

Pro zahájení metody předpokládáme, že máme k dispozici výchozí přípustnou bázi a jí určené BFS $\mathbf{x}^{[0]}$. Jak takovou bázi nalézt, ukáže kapitola 5. Jednotlivé pivoty generují posloupnost přípustnýchází a jim odpovídajících bodů $\mathbf{x}^{[0]}, \mathbf{x}^{[1]}, \dots$, kde po sobě jdoucí báze sdílejí $m - 1$ indexů. Odpovídající vrcholy jsou při kladné délce kroku sousední. Při nulovém kroku mohou oba zápisy určovat tentýž degenerovaný vrchol. V jednom kroku nastane jedna z možností:

1. $\mathbf{x}^{[p]}$ je optimální.
2. Úloha je neomezená.
3. Lze provést pivot do nové přípustné báze s nezmenšenou účelovou hodnotou. Hodnota vzroste přísně právě tehdy, když je délka kroku kladná.

V následujícím příkladu nejprve projdeme celý výpočet bez obecných vzorců. Až uvidíme, co jednotlivé kroky dělají, popíšeme simplexovou metodu obecně.

Příklad 4.1: Úvodní příklad na simplexovou metodu

Začneme řešením elementární úlohy v nerovnicovém tvaru

$$\begin{array}{llll} \text{maximalizovat} & x_1 + x_2 & \text{za podmíněk} & \begin{array}{l} -x_1 + x_2 \leq 1, \\ x_1 \leq 3, \\ x_2 \leq 2, \\ x_1, x_2 \geq 0. \end{array} \end{array}$$

Po zavedení doplňkových proměnných x_3, x_4, x_5 dostaneme rovnicový tvar

$$-x_1 + x_2 + x_3 = 1, \quad x_1 + x_4 = 3, \quad x_2 + x_5 = 2, \quad x_j \geq 0 \quad (j = 1, \dots, 5),$$

a rovnici účelové funkce $z = x_1 + x_2$. Této úloze přiřadíme tzv. počáteční *simplexovou*

tabulku:

$$\begin{aligned}x_3 &= 1 + x_1 - x_2, \\x_4 &= 3 - x_1, \\x_5 &= 2 - x_2, \\z &= x_1 + x_2.\end{aligned}$$

Pro $x_1 = x_2 = 0$ je báze přípustné řešení $\mathbf{x} = (0, 0, 1, 3, 2)^\top$ a $z = 0$.

Krok 1. Vstupuje x_2 , vystupuje x_3 . Zvyšujeme z volbou nebázické proměnné s kladným koeficientem v posledním řádku. Vhodné jsou x_1 i x_2 . Zvolíme x_2 . Nezápornost bázeických proměnných omezuje růst x_2 podmínkami

$$x_3 = 1 - x_2 \geq 0, \quad x_4 = 3 \geq 0, \quad x_5 = 2 - x_2 \geq 0,$$

tedy nejpřísněji $x_2 \leq 1$. Položíme $x_2 = 1$ a $x_3 = 0$. Z první rovnice vyjádříme

$$x_2 = 1 + x_1 - x_3$$

a dosazením získáme novou tabulku

$$\begin{aligned}x_2 &= 1 + x_1 - x_3, \\x_4 &= 3 - x_1, \\x_5 &= 1 - x_1 + x_3, \\z &= 1 + 2x_1 - x_3,\end{aligned}$$

s bází $\{2, 4, 5\}$ a hodnotou $z = 1$ (pro $x_1 = x_3 = 0$).

Krok 2. Vstupuje x_1 , vystupuje x_5 . z -řádek má kladný koeficient u x_1 , takže můžeme růst z zajišťovat zvyšováním x_1 . Omezení:

$$x_2 = 1 + x_1 \text{ (neomezuje)}, \quad x_4 = 3 - x_1 \geq 0 \Rightarrow x_1 \leq 3, \quad x_5 = 1 - x_1 \geq 0 \Rightarrow x_1 \leq 1.$$

Nejpřísnější je $x_1 \leq 1$, proto nastavíme $x_1 = 1$ a $x_5 = 0$. Z třetí rovnice vyjádříme

$$x_1 = 1 + x_3 - x_5$$

a po dosazení:

$$\begin{aligned}x_1 &= 1 + x_3 - x_5, \\x_2 &= 2 - x_5, \\x_4 &= 2 - x_3 + x_5, \\z &= 3 + x_3 - 2x_5.\end{aligned}$$

Báze je nyní $\{1, 2, 4\}$ a pro $x_3 = x_5 = 0$ dostáváme $z = 3$.

Krok 3. Vstupuje x_3 , vystupuje x_4 . z -řádek dovoluje zvyšovat x_3 . Nezápornost omezuje jen rovnice

$$x_4 = 2 - x_3 + x_5 \geq 0 \Rightarrow x_3 \leq 2 + x_5,$$

ostatní bázecké proměnné nezápornost neporuší. Položíme $x_5 = 0$ a $x_3 = 2$, čímž $x_4 = 0$. Z třetí rovnice dostaneme

$$x_3 = 2 + x_5 - x_4,$$

a finální tabulka má tvar

$$x_1 = 3 - x_4,$$

$$x_2 = 2 - x_5,$$

$$x_3 = 2 - x_4 + x_5,$$

$$z = 5 - x_4 - x_5.$$

V posledním řádku jsou koeficienty nebázeckých proměnných nekladné, takže nelze zvýšit z bez porušení přípustnosti. Optimální řešení je

$$\mathbf{x} = (3, 2, 2, 0, 0)^T, \quad z^* = 5.$$

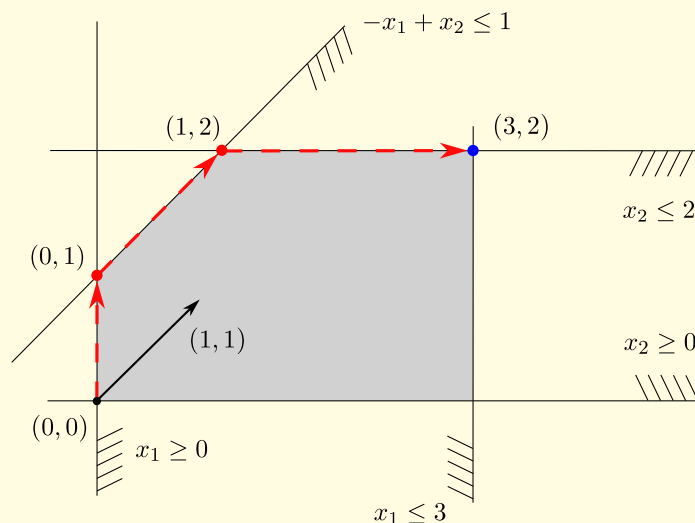
Poslední tabulka současně poskytuje certifikát optimality. Pro libovolné přípustné řešení $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_5)^T$ s hodnotou $\tilde{z}$ platí všechny její rovnice, protože vznikly z původní úlohy ekvivalentními úpravami. Tedy

$$\tilde{z} = 5 - \tilde{x}_4 - \tilde{x}_5.$$

Z nezápornosti $\tilde{x}_4, \tilde{x}_5$ plyne $\tilde{z} \leq 5$. Tabulka navíc dokazuje jednoznačnost optima: rovnost $z = 5$ vyžaduje $x_4 = x_5 = 0$, což jednoznačně určí i ostatní proměnné. Bod $\mathbf{x} = (3, 2, 2, 0, 0)^T$ je proto jediným optimálním řešením.

Poznámka 4.2: Geometrická ilustrace

Postup metody odpovídá posunu z vrcholu do vrcholu po hranách přípustného mnohoúhelníku ve směru růstu $x_1 + x_2$. Trajektorii ukazuje obrázek:



Obrázek 4.1: Tři nedegenerované kroky simplexové metody.

2 Neomezenost úlohy a více optimálních řešení

V předchozím jednoduchém příkladu proběhl algoritmus bez zádrhelů. V praxi se však mohou objevit komplikace, které podrobně popíšeme později. Nejdříve prozkoumáme, jak se metoda chová při neomezenosti úlohy a jak odhalí více optimálních řešení úlohy.

Příklad 4.3: Neomezenost úlohy

Ukažme na příkladu, jak simplex rozpozná neomezenou úlohu

$$\begin{array}{llll} \text{maximalizovat} & x_1 & \text{za podmíněk} & \begin{array}{l} x_1 - x_2 \leq 1, \\ -x_1 + x_2 \leq 2, \\ x_1, x_2 \geq 0. \end{array} \end{array}$$

Po zavedení x_3, x_4 máme spolu s rovnicí $z = x_1$ soustavu

$$\begin{aligned} x_3 &= 1 - x_1 + x_2, \\ x_4 &= 2 + x_1 - x_2, \\ z &= x_1. \end{aligned}$$

V z -řádku má x_1 kladný koeficient, proto jej zavedeme do báze. Z první rovnice

$$x_1 = 1 + x_2 - x_3,$$

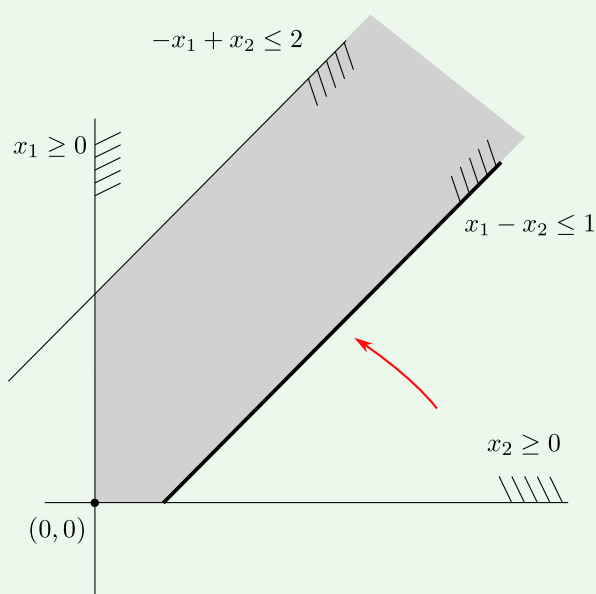
a po dosazení

$$x_4 = 3 - x_3, \quad z = 1 + x_2 - x_3.$$

Pokud nyní chceme zavést do báze x_2 , všimneme si, že žádná rovnice nerestriktuje jeho růst: v pravých stranách se x_2 neobjevuje se záporným koeficientem. Můžeme tedy volit $x_2 = t$, $x_3 = 0$ libovolně velké a dostáváme

$$(x_1, x_2, x_3, x_4, z) = (1 + t, t, 0, 3, 1 + t), \quad t \geq 0,$$

což dokládá neomezenost. Geometricky jde o polopřímku na hraně přípustné oblasti:



Obrázek 4.2: Přípustný paprsek dokazující neomezenost.

Příklad 4.4: Simplexová metoda – úsečka optimálních řešení

Pomocí simplexové tabulky vyřešme následující maximalizační úlohu:

$$\begin{array}{llll} & & & x_1 + x_2 \leq 5, \\ & & & x_1 \leq 4, \\ \text{maximalizovat} & x_1 + x_2 & \text{za podmínek} & x_2 \leq 3, \\ & & & x_1, x_2 \geq 0. \end{array}$$

Po přidání doplňkových proměnných x_3, x_4, x_5 získáme soustavu

$$x_1 + x_2 + x_3 = 5, \quad x_1 + x_4 = 4, \quad x_2 + x_5 = 3, \quad x_j \geq 0 \quad (j = 1, \dots, 5),$$

a účelovou funkci

$$z = x_1 + x_2.$$

Počáteční simplexová tabulka:

$$\begin{aligned} x_3 &= 5 - x_1 - x_2, \\ x_4 &= 4 - x_1, \\ x_5 &= 3 - x_2, \\ z &= x_1 + x_2. \end{aligned}$$

Krok 1. Vybereme x_2 jako vstupující. Omezení:

$$x_3 = 5 - x_1 - x_2, \quad x_4 = 4 - x_1, \quad x_5 = 3 - x_2.$$

Nejpřísněji omezuje $x_2 \leq 3$ (třetí rovnice), proto x_5 vystupuje a

$$x_2 = 3 - x_5.$$

Dosadíme:

$$\begin{aligned} x_3 &= 2 - x_1 + x_5, \\ x_4 &= 4 - x_1, \\ x_2 &= 3 - x_5, \\ z &= 3 + x_1 - x_5. \end{aligned}$$

Báze $\{2, 3, 4\}$, pro $x_1 = x_5 = 0$ je $z = 3$.

Krok 2. Zvyšujeme x_1 . Omezení:

$$x_3 = 2 - x_1 + x_5 \geq 0, \quad x_4 = 4 - x_1 \geq 0.$$

Nejpřísněji omezuje $x_1 \leq 2$ (první rovnice). Vstoupí x_1 , vystoupí x_3 :

$$x_1 = 2 + x_5 - x_3.$$

Po dosazení:

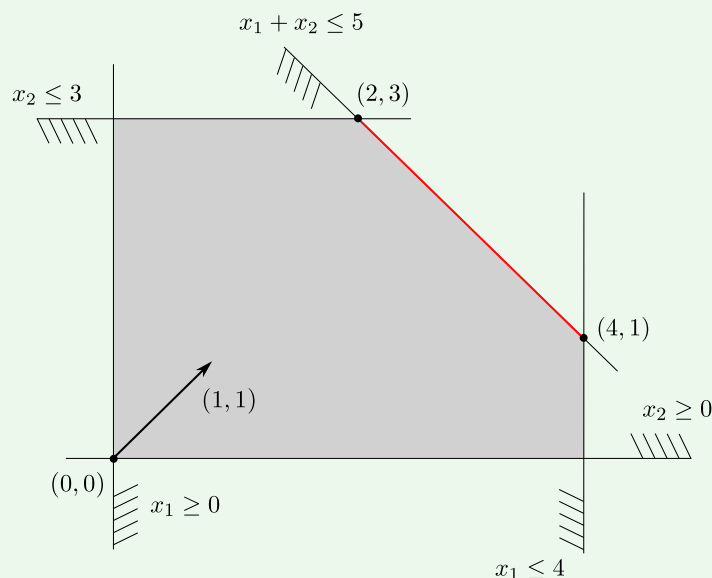
$$\begin{aligned} x_1 &= 2 + x_5 - x_3, \\ x_2 &= 3 - x_5, \\ x_4 &= 2 - x_5 + x_3, \\ z &= 5 - x_3. \end{aligned}$$

Báze $\{1, 2, 4\}$. Pro $x_3 = x_5 = 0$ je $z = 5$.

Optimální řešení. V z -řádku nejsou kladné koeficienty a u x_5 je nula, jsme v optimu. Avšak rozebereme-li si řešení, vidíme, že při $x_3 = 0$ a $x_5 \in [0, 2]$:

$$(x_1, x_2) = (2 + x_5, 3 - x_5), \quad z = 5,$$

tj. úsečka mezi $(2, 3)$ a $(4, 1)$ určuje body, kde v každém z nich je nabýváno optimální hodnoty.



Obrázek 4.3: Úsečka optimálních řešení.

3 Simplexové tabulky

To, co jsme zatím sledovali na příkladech, nyní zformulujeme obecně a se stejnou znaménkovou konvencí, kterou budeme používat i v dalších kapitolách.

Uvažujme obecnou maximalizační úlohu v rovnicovém tvaru

$$\text{maximalizovat } \mathbf{c}^T \mathbf{x} \quad \text{za podmíněk } A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0},$$

kde $A \in \mathbb{R}^{m \times n}$ má řádky lineárně nezávislé a přípustná množina není prázdná. Pro přípustnou bázi $B \subseteq \{1, \dots, n\}$, $|B| = m$, píšme $B = \{j_1 < \dots < j_m\}$ a označme $N = \{1, \dots, n\} \setminus B$.

Simplexová tabulka

Simplexová tabulka přiřazená bázi B je soustava

$$\begin{aligned} \mathbf{x}_B &= \mathbf{p} + Q \mathbf{x}_N, \\ z &= z_0 + \mathbf{r}^T \mathbf{x}_N, \end{aligned} \tag{4.1}$$

jejíž množina řešení v proměnných $(\mathbf{x}, z)$ je shodná s množinou řešení $A\mathbf{x} = \mathbf{b}$ a $z = \mathbf{c}^T \mathbf{x}$. Vektor $\mathbf{x}_B$ obsahuje bázecké proměnné. Pro $N = \{1, \dots, n\} \setminus B$ obsahuje $\mathbf{x}_N$ proměnné nebázecké. Rozměry koeficientů jsou $\mathbf{p} \in \mathbb{R}^m$, $Q \in \mathbb{R}^{m \times (n-m)}$, $\mathbf{r} \in \mathbb{R}^{n-m}$

a $z_0 \in \mathbb{R}$. Souřadnice $\mathbf{x}_N$ a $\mathbf{r}$ i sloupce Q budeme pro pohodlí indexovat přímo původními indexy $j \in N$. Symboly r_j , q_{ij} a $Q_{\cdot j}$ tedy označují souřadnici či sloupec příslušející proměnné x_j , i když indexy množiny N nemusejí tvořit posloupnost $1, \dots, n - m$.

Bázické přípustné řešení k simplexové tabulce okamžitě vyčteme: dosadíme $\mathbf{x}_N = \mathbf{0}$, tedy $\mathbf{x}_B = \mathbf{p}$ a hodnota účelové funkce je $z = z_0$. Protože B je přípustná báze, vždy platí $\mathbf{p} \geq \mathbf{0}$. Složky vektoru $\mathbf{r}$ nazýváme v této znaménkové konvenci *redukované náklady*.

Tvary $\mathbf{p}$, Q , $\mathbf{r}$ a z_0 lze snadno vyjádřit pomocí B a A , $\mathbf{b}$, $\mathbf{c}$. Podotkneme, že následující vzorečky je ale zbytečné si pamatovat, v případě potřeby se lehce odvodí.

Lemma 4.5: Jednoznačnost a tvar koeficientů simplexové tabulky

Pro každou přípustnou bázi B existuje právě jedna tabulka (4.1) a její koeficienty jsou dány

$$Q = -A_B^{-1}A_N, \quad \mathbf{p} = A_B^{-1}\mathbf{b}, \quad z_0 = \mathbf{c}_B^T A_B^{-1}\mathbf{b}, \quad \mathbf{r} = \mathbf{c}_N - (\mathbf{c}_B^T A_B^{-1}A_N)^T.$$

Důkaz lemmatu.

Rozklad $A\mathbf{x} = A_B\mathbf{x}_B + A_N\mathbf{x}_N = \mathbf{b}$ a regulárnost A_B dávají

$$\mathbf{x}_B = A_B^{-1}\mathbf{b} - A_B^{-1}A_N\mathbf{x}_N.$$

Tím jsou jednoznačně určeny $\mathbf{p}$ a Q . Po dosazení do $z = \mathbf{c}_B^T\mathbf{x}_B + \mathbf{c}_N^T\mathbf{x}_N$ dostaneme

$$z = \mathbf{c}_B^T A_B^{-1}\mathbf{b} + (\mathbf{c}_N^T - \mathbf{c}_B^T A_B^{-1}A_N)\mathbf{x}_N,$$

což jednoznačně určuje z_0 i $\mathbf{r}$. Všechny úpravy jsou ekvivalentní, takže tabulka popisuje právě původní rovnice. ■

4 Simplexová metoda – obecné základy

4.1 Kritérium optimality

Jsou-li v tabulce (4.1) všechny redukované náklady nekladné, tj.

$$\mathbf{r} \leq \mathbf{0},$$

pak příslušné bázické přípustné řešení je optimální. Skutečně, bázické přípustné řešení příslušné takové tabulce má účelovou funkci rovnou z_0 , a pro libovolné jiné přípustné řešení $\tilde{\mathbf{x}}$ máme $\tilde{\mathbf{x}}_N \geq \mathbf{0}$ a

$$\mathbf{c}^T \tilde{\mathbf{x}} = z_0 + \mathbf{r}^T \tilde{\mathbf{x}}_N \leq z_0.$$

Jde o postačující certifikát optimality. V degenerované tabulce nemusí být podmínka $\mathbf{r} \leq \mathbf{0}$ nutná pro optimalitu samotného bodu, protože kladnému redukovanému nákladu může odpovídat pouze nulový přípustný krok.

4.2 Pivot – volba vstupující a vystupující proměnné

Jeden pivot odpovídá dvěma rozhodnutím. Nejprve vybereme, kterou nebázickou proměnnou začneme zvětšovat – tedy kudy se chceme vydat. Potom zjistíme, která z dosavadních

bázických proměnných narazí jako první na nulu – tedy kde nás zastaví hranice přípustné oblasti.

Vstupující proměnná: kudy se vydat. Zvolme nebázickou proměnnou x_v s $r_v > 0$. Její kladný redukováný náklad říká, že každé přípustné kladné zvětšení x_v zlepší účelovou hodnotu. Pokud ostatní nebázické proměnné zůstanou nulové a položíme $x_v = t \geq 0$, tabulka má tvar

$$x_{j_i} = p_i + q_{iv}t \quad (i = 1, \dots, m), \quad z = z_0 + r_v t.$$

Vystupující proměnná: kde narazíme na hranici. Protože $\mathbf{p} \geq \mathbf{0}$, řádek s $q_{iv} \geq 0$ růst t neomezuje. Řádek s $q_{iv} < 0$ vyžaduje $t \leq -p_i/q_{iv}$. Je-li alespoň jeden takový řádek, maximální přípustná délka kroku je

$$\theta = \min_{i: q_{iv} < 0} \left(-\frac{p_i}{q_{iv}} \right) \geq 0. \quad (4.2)$$

Jako vystupující zvolíme některou bázickou proměnnou x_{j_s} , pro jejíž řádek se minima dosáhne. Při shodě více podílů dosáhne nuly několik bázických proměnných současně. Volba jedné z nich je součástí pivotovacího pravidla. Jinými slovy: sledujeme pouze záporné koeficienty ve sloupci vstupující proměnné, spočteme příslušné podíly a vezmeme nejmenší. Právě tato hodnota je nejdelší krok, který ještě zachová nezápornost všech bázických proměnných. Nová báze je

$$B' = (B \setminus \{j_s\}) \cup \{v\}.$$

Z rovnice pro x_{j_s} vyjádříme x_v a dosadíme jej do všech ostatních řádků. To je vlastní pivotová úprava.

A co když ve sloupci žádný záporný koeficient není? Potom nás při zvyšování x_v ne zastaví žádná bázická proměnná. Směr zlepšení pokračuje libovolně daleko a tabulka tím přímo vydává certifikát neomezenosti.

Lemma 4.6: Korektnost pivotu

Nechť B je přípustná báze, $r_v > 0$ a existuje řádek s $q_{iv} < 0$. Zvolíme-li s poměrovým testem (4.2), pak B' je přípustná báze. Nové BFS má účelovou hodnotu $z_0 + \theta r_v$. Ta je větší než z_0 pro $\theta > 0$ a stejná pro $\theta = 0$. Jestliže pro zvolené v platí $q_{iv} \geq 0$ ve všech řádcích, je úloha neomezená.

Důkaz lemmatu.

Pro $0 \leq t \leq \theta$ jsou všechny hodnoty $p_i + q_{iv}t$ nezáporné. V bodě $t = \theta$ je vystupující proměnná x_{j_s} rovna nule, takže tento bod je přípustný.

Zbývá ověřit, že B' je skutečně báze. Z lemmatu 4.5 máme $Q = -A_B^{-1}A_N$, a tedy $q_{sv} \neq 0$. Kdyby sloupce s indexy v B' byly lineárně závislé, existovala by netriviální rovnost

$$\alpha_v A_v + \sum_{i \neq s} \alpha_i A_{j_i} = \mathbf{0}.$$

Po vynásobení A_B^{-1} má její s -tá složka tvar $-\alpha_v q_{sv} = 0$, takže $\alpha_v = 0$. Zbývající kanonické vektory pak dávají $\alpha_i = 0$ pro všechna $i \neq s$, což je spor. Matice $A_{B'}$ je regulární. Pivotová soustava je s původní soustavou ekvivalentní a při položení nových nebázických proměnných na nulu dává právě bod pro $t = \theta$. Báze B' je proto přípustná. Z účelového řádku plyne hodnota $z_0 + \theta r_v$.

Neexistuje-li záporné q_{iv} , pak pro každé $t \geq 0$ platí $\mathbf{x}_B = \mathbf{p} + tQ_{\cdot v} \geq \mathbf{0}$. Tyto body jsou přípustné a $z = z_0 + r_v t \rightarrow +\infty$, protože $r_v > 0$. ■

Poznámka 4.7: Nula v účelovém řádku není celý příběh

V optimální tabulce může některá nebázická proměnná splňovat $r_v = 0$. Její zvyšování nemění účelovou hodnotu, ale další závěr závisí na přípustné délce kroku. Je-li $\theta > 0$, dostaneme odlišné optimální body a při $t = \theta$ jiné optimální BFS. Je-li $\theta = 0$, pivot může pouze změnit bázi téhož degenerovaného bodu. Není-li v daném sloupci žádné záporné q_{iv} , vzniká optimální paprsek. Samotná rovnost $r_v = 0$ tedy bez kontroly sloupce $Q_{\cdot v}$ neprokazuje existenci jiného optimálního vrcholu.

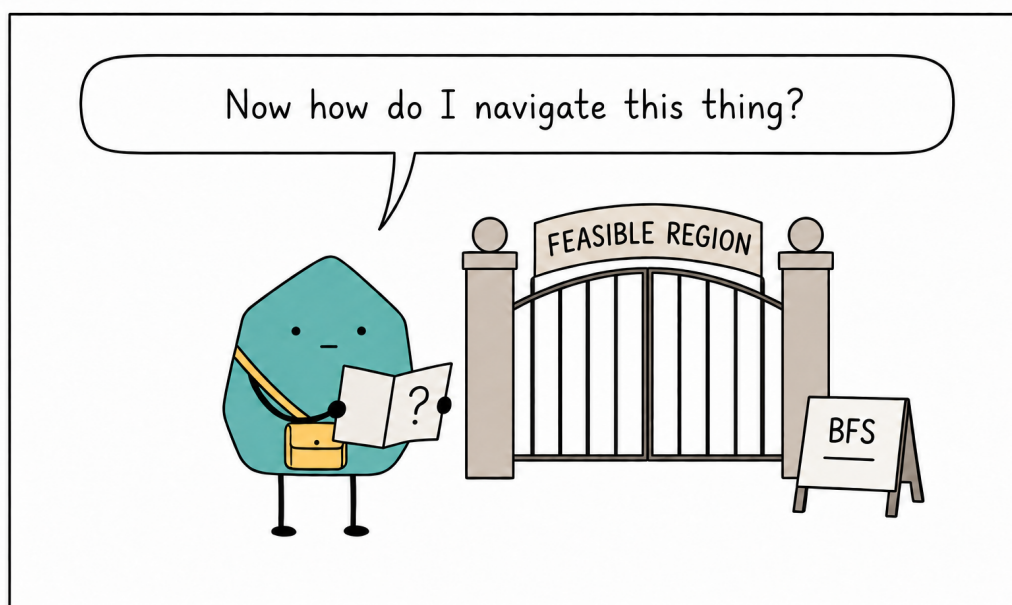
Jednotlivý pivot je tím odůvodněn. Z tabulky už umíme vyčíst optimum, směr neomezenosti i další přípustnou bázi. Zbývá jediná nepříjemnost: při nulových krocích by se metoda mohla mezi bázemi točit a nikam se neposunout. Aby byl celý algoritmus konečný i při degeneraci, potřebujeme anticyklické pravidlo. Tomu je věnována kapitola 6. Soustavné odvození simplexové metody lze porovnat s Vanderbei (2020) a Bertsimas and Tsitsiklis (1997).

KAPITOLA 5

SIMPLEXOVÝ ALGORITMUS II: DVOUFÁZOVÁ METODA A M-ÚLOHA

1	Výchozí bázecké přípustné řešení	48
2	Metoda velkého M	48
3	Dvofázová metoda (metoda umělé báze)	52

Aby se simplexový algoritmus mohl vůbec rozběhnout, musíme mu na začátku dodat bázecké přípustné řešení. V dosavadních příkladech jsme je dostávali téměř zadarmo, ale z obecného rovnicového zadání nemusí být patrné. V této kapitole si ukážeme dvě standardní techniky: M -úlohu neboli metodu velkého M a dvofázovou metodu neboli metodu umělé báze. Obě pracují s pomocnou rozšířenou úlohou, která buď odkryje přípustnou bázi původní úlohy, nebo prokáže, že ji hledáme marně.



1 Výchozí bázecké přípustné řešení

Již jsme viděli, jak simplexová metoda pracuje, jakmile má k dispozici přípustnou bázi. Pečlivému čtenáři však mohla zůstat jedna nepříjemná otázka: odkud vezmeme úplně první bázi? Ta totiž není součástí obecného zadání a musíme ji teprve nalézt.

V příkladech, které jsme zatím uváděli, jsme přípustnou bázi dostali víceméně zadarmo. Tak tomu je u všech úloh v nerovnicovém tvaru

$$\text{maximalizovat } \mathbf{c}^T \mathbf{x} \text{ za podmínek } A\mathbf{x} \leq \mathbf{b} \text{ a } \mathbf{x} \geq \mathbf{0}$$

s $\mathbf{b} \geq \mathbf{0}$. Doplnkové proměnné zavedené při převodu na rovnicový tvar pak tvoří přípustnou bázi.

Obecně však rozhodnutí o přípustnosti úlohy ani nalezení výchozího bázeckého přípustného řešení není samo o sobě jednoduché. Nestačí tedy jen umět z jedné tabulky přejít do druhé. Musíme nejprve najít místo, odkud vyrazit. Ukážeme si, jak to udělat systematicky, nebo jak cestou prokázat, že původní úloha přípustné řešení vůbec nemá.

Použijeme dvě standardní metody (Dantzig, 1963): metodu velkého M , zkráceně M -úlohu, a dvoufázovou metodu neboli metodu umělé báze. Obě přidávají *umělé proměnné*, které vytvoří snadno dostupnou počáteční bázi. Tyto proměnné nejsou součástí původního modelu a před zahájením vlastního řešení původní úlohy je musíme odstranit.

Nejprve dokažme tvrzení, na němž jsou oba postupy založeny.

Tvrzení 5.1: Pomocná úloha a přípustnost původního systému

Nechť $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{b} \geq \mathbf{0}$. Pro systém $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, uvažujme pomocnou úlohu

$$A\mathbf{x} + I\mathbf{w} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0},$$

se záměrem minimalizovat $\mathbf{1}^T \mathbf{w}$. Pomocná úloha má přípustné řešení $(\mathbf{0}, \mathbf{b})$, její optimum je dosaženo a platí

$$\min\{\mathbf{1}^T \mathbf{w} : A\mathbf{x} + \mathbf{w} = \mathbf{b}, \mathbf{x}, \mathbf{w} \geq \mathbf{0}\} = 0 \iff \exists \mathbf{x} \geq \mathbf{0} : A\mathbf{x} = \mathbf{b}.$$

Důkaz tvrzení.

Účelová funkce je zdola omezená nulou. Existuje-li $\mathbf{x} \geq \mathbf{0}$ s $A\mathbf{x} = \mathbf{b}$, potom je $(\mathbf{x}, \mathbf{0})$ přípustné pro pomocnou úlohu a dosahuje hodnoty nula. Naopak, dosáhne-li pomocná úloha hodnoty nula v bodě $(\mathbf{x}^*, \mathbf{w}^*)$, z $\mathbf{w}^* \geq \mathbf{0}$ a $\mathbf{1}^T \mathbf{w}^* = 0$ plyne $\mathbf{w}^* = \mathbf{0}$. Rovnosti pomocné úlohy pak dávají $A\mathbf{x}^* = \mathbf{b}$. Existence optima plyne ze základní věty lineárního programování, protože pomocná úloha je přípustná a její účelová funkce je zdola omezená. ■

2 Metoda velkého M

Začneme *metodou velkého M*. Její myšlenka je přímočará: kde původní sloupce nedávají vhodnou počáteční bázi, na chvíli si vypomůžeme umělými proměnnými. Současně je však v účelové funkci potrestáme tak silně, aby je algoritmus odstranil, jakmile je to možné. Pro původní maximalizační úlohu v rovnicovém tvaru řešíme rozšířenou úlohu

$$\max \mathbf{c}^T \mathbf{x} - \mathbf{m}^T \mathbf{w}$$

za podmínek

$$A\mathbf{x} + \mathbf{w} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \mathbf{w} \geq \mathbf{0},$$

kde $\mathbf{m} = M\mathbf{1}$ a $\mathbf{w} \in \mathbb{R}^m$ je vektor umělých proměnných. Zápis svádí k představě, že prostě vezmeme nějaké obrovské číslo. Matematicky bezpečný význam symbolu M je ale *lexikografický*: nejprve minimalizujeme $\mathbf{1}^T \mathbf{w}$ a teprve mezi řešeními s jeho nejmenší hodnotou maximalizujeme $\mathbf{c}^T \mathbf{x}$. Symbol M tedy není předem zvolené velké reálné číslo, nýbrž stručný zápis naprosté priority prvního cíle před druhým.

Poznámka 5.2: Proč nevolit obrovské konečné M

Rčení „ M je větší než všechna čísla v úloze“ je užitečná pomůcka, nikoli návod k volbě konkrétního čísla. Pro obecnou úlohu neexistuje pravidlo typu „několik řádů nad největším koeficientem“, které by bez další analýzy zaručilo správnou prioritu a současně dobrou numerickou stabilitu. Příliš malé M může připustit kladnou umělou proměnnou výměnou za zisk v původní účelové funkci. Příliš velké M zase zvětšuje dynamický rozsah koeficientů účelového řádku a může vést ke ztrátě přesnosti při odečítání i při výpočtu redukováných nákladů. Samotnou podmíněnost bázecké matice A_B však hodnota M nemění. Proto:

- v ručním a teoretickém výpočtu zacházíme s M symbolicky a porovnáváme zvlášť koeficienty u M a konstantní části,
- v numerické implementaci dáváme přednost dvoufázové metodě, která žádnou velkou konstantu nepotřebuje.

Není nutné přidávat umělou proměnnou do každého řádku. Jestliže už některý původní nebo doplňkový sloupec tvoří příslušný jednotkový sloupec počáteční báze, použijeme jej. Umělou proměnnou w_i přidáme pouze k tomu i -tému řádku, pro který vhodný bázecký sloupec chybí. Index umělé proměnné tedy označuje *řádek omezení*, nikoli sloupec původní matice.

Jakmile mají všechny umělé proměnné hodnotu nula a jsou mimo bázi, máme bázecké přípustné řešení původního systému. Jejich sloupce odstraníme, znovu sestavíme řádek původní účelové funkce $z = \mathbf{c}^T \mathbf{x}$ pro aktuální bázi a pokračujeme běžnou simplexovou metodou. Znaménka redukováných nákladů u již odstraněných umělých proměnných nejsou pro tento přechod další podmínkou.

Jestliže lexikografická optimalizace skončí a některá umělá proměnná má kladnou hodnotu, je původní úloha nepřípustná. Zůstane-li umělá proměnná v bázi s nulovou hodnotou, postupujeme stejně jako na konci první fáze dvoufázové metody: buď ji vypivotujeme pomocí původního sloupce s nenulovým koeficientem v jejím řádku, nebo odstraníme nadbytečný řádek.

Poznámka 5.3: Proč funguje metoda velkého M

Uvažujme rozšířenou úlohu s účelovou funkcí

$$\tilde{z} = \mathbf{c}^T \mathbf{x} - \mathbf{m}^T \mathbf{w} \quad \text{při} \quad A\mathbf{x} + \mathbf{w} = \mathbf{b}, \quad \mathbf{x}, \mathbf{w} \geq \mathbf{0},$$

kde význam $\mathbf{m}$ je popsán výše. Část účelové funkce obsahující M je dominantní. Maximalizace $\tilde{z}$ je tedy lexikograficky^a ekvivalentní dvoukroku:

1. minimalizuj $\sum_j w_j$,
2. mezi řešeními s minimálním $\sum_j w_j$ maximalizuj $\mathbf{c}^\top \mathbf{x}$.

Z tvrzení 5.1 víme, že $\min \sum_j w_j = 0$ právě tehdy, když existuje $\mathbf{x} \geq \mathbf{0}$ s $A\mathbf{x} = \mathbf{b}$. Pokud v optimu lexikografické úlohy zůstane některé $w_j > 0$, původní úloha je nepřípustná.

^a*Lexikografické uspořádání:* pro vektory $\mathbf{u}, \mathbf{v} \in \mathbb{R}^k$ platí $\mathbf{u} <_{\text{lex}} \mathbf{v}$, jestliže existuje nejmenší index i s $u_i \neq v_i$ a $u_i < v_i$. Maximalizace vektorové funkce $(f_1, \dots, f_k)$ v lexikografickém smyslu znamená: nejprve maximalizovat f_1 , pokud je více řešení se stejným f_1 , maximalizovat mezi nimi f_2 , atd.

2.1 Příklady

Na prvním příkladu projdeme standardní chod metody velkého M a umělé proměnné skutečně z báze vytlačíme. Druhý příklad je záměrně prostý: chceme na něm jasně vidět okamžik, kdy kladná umělá proměnná prozradí nepřípustnost původní úlohy.

Příklad 5.4: Metoda velkého M

V úvodním příkladu budeme chtít

$$\begin{aligned} & \text{maximalizovat} \quad z = x_1 + 4x_2 + 2x_3 \quad \text{za podmíněk} \quad \begin{aligned} x_1 - x_2 + x_3 &= 1, \\ x_1 + x_2 + 2x_3 &= 4, \\ x_1, x_2, x_3 &\geq 0. \end{aligned} \end{aligned}$$

Po rozšíření o umělé proměnné w_1, w_2 dostaneme rovnicový tvar

$$x_1 - x_2 + x_3 + w_1 = 1, \quad x_1 + x_2 + 2x_3 + w_2 = 4, \quad x_1, x_2, x_3, w_1, w_2 \geq 0,$$

a rovnicí účelové funkce

$$\tilde{z} = x_1 + 4x_2 + 2x_3 - M(w_1 + w_2),$$

kde M chápeme jako lexikograficky dominantní symbol.

Počáteční simplexová tabulka. Umělé proměnné zvolíme jako počáteční báze-zické proměnné:

$$\begin{aligned} w_1 &= 1 - x_1 + x_2 - x_3, \\ w_2 &= 4 - x_1 - x_2 - 2x_3, \\ \tilde{z} &= -5M + (2M + 1)x_1 + 4x_2 + (3M + 2)x_3. \end{aligned}$$

Pivotováním se nyní snažíme umělé proměnné přesunout mezi nebázické proměnné a snížit jejich součet na nulu.

Krok 1. Vstupuje x_3 , vystupuje w_1 .

$$\begin{aligned} x_3 &= 1 - x_1 + x_2 - w_1, \\ w_2 &= 2 + x_1 - 3x_2 + 2w_1, \\ \tilde{z} &= 2 - 2M + (-M - 1)x_1 + (3M + 6)x_2 - (3M + 2)w_1. \end{aligned}$$

Krok 2. Vstupuje x_2 , vystupuje w_2 .

$$\begin{aligned}x_2 &= \frac{2}{3} + \frac{1}{3}x_1 + \frac{2}{3}w_1 - \frac{1}{3}w_2, \\x_3 &= \frac{5}{3} - \frac{2}{3}x_1 - \frac{1}{3}w_1 - \frac{1}{3}w_2, \\z &= 6 + x_1 + (2 - M)w_1 - (M + 2)w_2.\end{aligned}$$

Obě umělé proměnné jsou nyní nebázické a mají hodnotu nula. Jejich sloupce proto odstraníme a pro aktuální bázi znovu sestavíme řádek původní účelové funkce.

Bázické přípustné řešení původní úlohy.

$$\mathbf{x} = (0, \frac{2}{3}, \frac{5}{3})^\top, \quad z = 6.$$

Pokračování simplexovým algoritmem. Po odstranění $\mathbf{w}$ máme úvodní simplexovou tabulku ve tvaru:

$$x_2 = \frac{2}{3} + \frac{1}{3}x_1, \quad x_3 = \frac{5}{3} - \frac{2}{3}x_1, \quad z = 6 + x_1.$$

Vstupuje x_1 , vystupuje x_3 . Po pivotu

$$x_1 = \frac{5}{2} - \frac{3}{2}x_3, \quad x_2 = \frac{3}{2} - \frac{1}{2}x_3, \quad z = \frac{17}{2} - \frac{3}{2}x_3.$$

Z tvaru účelové funkce jsme našli optimum, pokládáme tedy $x_3 = 0$ a dostaneme:

$$\mathbf{x}^* = (\frac{5}{2}, \frac{3}{2}, 0)^\top, \quad z^* = \frac{17}{2}.$$

Příklad 5.5: Metoda velkého M – nepřípustná úloha

V tomto příkladu budeme chtít

$$\begin{aligned}\text{maximalizovat} \quad z &= x_1 + x_2 & \text{za podmíněk} \quad x_1 + x_2 &= 1, \\ & & & x_1 + x_2 &= 2, \\ & & & x_1, x_2 &\geq 0.\end{aligned}$$

Po rozšíření o w_1, w_2 :

$$x_1 + x_2 + w_1 = 1, \quad x_1 + x_2 + w_2 = 2, \quad x_1, x_2, w_1, w_2 \geq 0,$$

a účelová funkce

$$\tilde{z} = x_1 + x_2 - M(w_1 + w_2),$$

kde je M opět lexikograficky dominantní.

Počáteční simplexová tabulka.

$$w_1 = 1 - x_1 - x_2, \quad w_2 = 2 - x_1 - x_2, \quad \tilde{z} = -3M + (2M + 1)(x_1 + x_2).$$

Krok 1. Vstupuje x_1 , vystupuje w_1 .

$$x_1 = 1 - x_2 - w_1, \quad w_2 = 1 + w_1, \quad \tilde{z} = -M + 1 - (2M + 1)w_1 + 0 \cdot x_2.$$

V řádku $\tilde{z}$ nejsou kladné koeficienty, tedy jsme našli optimum pomocné úlohy při $w_1 = 0, x_2 = 0$:

$$x_1 = 1, \quad x_2 = 0, \quad w_1 = 0, \quad w_2 = 1, \quad \tilde{z} = -M + 1.$$

Protože v optimu platí $w_2 > 0$, původní úloha je **nepřípustná**.

Rychlá diagnostika.

$$(x_1 + x_2 + w_2) - (x_1 + x_2 + w_1) = 1 \Rightarrow w_2 - w_1 = 1,$$

takže $\min(w_1 + w_2) = 1$ a stavu $w = 0$ nelze dosáhnout.

3 Dvoufázová metoda (metoda umělé báze)

Druhou metodou pro nalezení výchozí báze je *dvoufázová metoda*, někdy také nazývaná metoda umělé báze. Sleduje stejnou myšlenku jako metoda velkého M , jen obě priority nepíše do jediné účelové funkce. Původní úlohu v rovnicovém tvaru rozšíříme tak, abychom snadno získali výchozí BFS pomocné úlohy. Z jejího optima rozhodneme o přípustnosti původního systému a případně připravíme bázi pro vlastní optimalizaci.

Jak název napovídá, postup má dvě fáze. V první se nestaráme o původní zisk: jediným cílem je minimalizovat součet umělých proměnných a dostat jej pokud možno na nulu. Teprve ve druhé fázi umělé proměnné odstraníme, vrátíme původní účelovou funkci a z nalezené přípustné báze pokračujeme simplexovou metodou.

Poznamenejme, že první fáze dvoufázové metody je postavena na přímém využití tvrzení 5.1.

3.1 První fáze

Nejprve převedeme úlohu do rovnicového tvaru. Řádek se zápornou pravou stranou vynásobíme číslem -1 , takže $\mathbf{b} \geq \mathbf{0}$. Do řádků, v nichž chybí vhodný bázecký sloupec, přidáme umělé proměnné. Pro jednoduchost zápisu nyní předpokládejme, že jsou potřeba ve všech řádcích:

$$A\mathbf{x} + I\mathbf{w} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{w} \geq \mathbf{0}.$$

Pak simplexovou metodou řešíme *pomocnou maximalizační úlohu*

$$\max (-w_1 - \dots - w_m) \quad \text{při} \quad A\mathbf{x} + I\mathbf{w} = \mathbf{b}, \quad \mathbf{x}, \mathbf{w} \geq \mathbf{0}.$$

Výchozí BFS v první fázi získáme snadno – je jím bod $\mathbf{x} = \mathbf{0}, \mathbf{w} = \mathbf{b}$.

V závislosti na optimu první fáze mohou nastat tři situace:

1. $\sum_{i=1}^m w_i^* > 0$, ekvivalentně je optimální hodnota maximalizační pomocné úlohy záporná. Podle tvrzení 5.1 je původní úloha **nepřípustná**.
2. $\sum_{i=1}^m w_i^* = 0$ a všechny umělé proměnné jsou mimo bázi. Potom $\mathbf{w}^* = \mathbf{0}$ a nalezený bázecký bod v původních proměnných je výchozím BFS pro druhou fázi.

3. $\sum_{i=1}^m w_i^* = 0$, ale některá umělá proměnná w_i zůstala v bázi s nulovou hodnotou. V jejím řádku vyhledáme nebázickou neumělou proměnnou s nenulovým koeficientem a provedeme nulový pivot, jímž w_i z báze odstraníme. Pokud takový koeficient neexistuje, odpovídající rovnice je lineárně závislá na ostatních. V takovém případě odstraníme tento řádek i sloupec w_i . Postup opakujeme, dokud v bázi nezůstane žádná umělá proměnná.

3.2 Druhá fáze

Je-li optimum první fáze rovno nule, po uvedené úpravě báze odstraníme všechny sloupce umělých proměnných, vrátíme původní účelovou funkci $\mathbf{c}^T \mathbf{x}$ a její řádek přepočítáme pro aktuální bázi. Pak pokračujeme obvyklou simplexovou metodou. První fáze zaručuje pouze přípustnost: ve druhé fázi může původní úloha mít konečné optimum, ale také se může ukázat jako neomezená.

3.3 Příklady

Na následujících třech příkladech ilustrujeme všechny tři scénáře, které mohou na konci první fáze nastat.

Příklad 5.6: Dvoufázová metoda

Ukažme si postup metody umělé báze na jednoduchém příkladu, kdy chceme

$$\begin{aligned} \text{maximalizovat} \quad z = x_1 + 4x_2 + 2x_3 \quad \text{za podmínek} \quad & x_1 - x_2 + x_3 = 1, \\ & x_1 + x_2 + 2x_3 = 4, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

1. fáze. Přidáme umělé proměnné w_1, w_2 a řešíme

$$\max(-w_1 - w_2) \quad \text{při} \quad x_1 - x_2 + x_3 + w_1 = 1, \quad x_1 + x_2 + 2x_3 + w_2 = 4, \quad \mathbf{x}, \mathbf{w} \geq \mathbf{0}.$$

Označme $\tilde{z} = -w_1 - w_2$.

Počáteční tabulka.

$$\begin{aligned} w_1 &= 1 - x_1 + x_2 - x_3, \\ w_2 &= 4 - x_1 - x_2 - 2x_3, \\ \tilde{z} &= -5 + 2x_1 + 3x_3. \end{aligned}$$

Krok 1. Vstupuje x_3 , vystupuje w_1 .

$$\begin{aligned} x_3 &= 1 - x_1 + x_2 - w_1, \\ w_2 &= 2 + x_1 - 3x_2 + 2w_1, \\ \tilde{z} &= -2 - x_1 + 3x_2 - 3w_1. \end{aligned}$$

Krok 2. Vstupuje x_2 , vystupuje w_2 .

$$\begin{aligned} x_2 &= \frac{2}{3} + \frac{1}{3}x_1 + \frac{2}{3}w_1 - \frac{1}{3}w_2, \\ x_3 &= \frac{5}{3} - \frac{2}{3}x_1 - \frac{1}{3}w_1 - \frac{1}{3}w_2, \\ \tilde{z} &= -w_1 - w_2. \end{aligned}$$

Optimum 1. fáze je $\tilde{z}^* = 0$ při $w_1 = w_2 = 0$. Získáme BFS

$$\mathbf{x}^{(0)} = (0, \frac{2}{3}, \frac{5}{3})^\top.$$

2. fáze. Odstraníme sloupce $\mathbf{w}$ a řešíme původní úlohu

$$z = x_1 + 4x_2 + 2x_3 = 6 + x_1 \quad \text{při} \quad \begin{aligned} x_2 &= \frac{2}{3} + \frac{1}{3}x_1, \\ x_3 &= \frac{5}{3} - \frac{2}{3}x_1. \end{aligned}$$

Vstupuje x_1 , vystupuje x_3 . Po pivotu a dosazení $x_3 = 0$:

$$\mathbf{x}^* = (\frac{5}{2}, \frac{3}{2}, 0)^\top, \quad z^* = \frac{17}{2}.$$

Příklad 5.7: Dvoufázová metoda a odhalení nepřipustnosti

Abychom si připomněli obecný postup, uvažujme úlohu nejprve v nerovnicovém tvaru, v níž je cílem

$$\begin{aligned} \text{maximalizovat} \quad z = 3x_1 + 5x_2 \quad \text{za podmíněk} \quad & \begin{aligned} x_1 - 2x_2 &\leq 6, \\ 2x_1 + \frac{1}{2}x_2 &\leq 7, \\ x_1 + x_2 &\geq 16, \\ x_1, x_2 &\geq 0. \end{aligned} \end{aligned}$$

Převod na rovnosti a 1. fáze.

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 6, \\ 2x_1 + \frac{1}{2}x_2 + x_4 &= 7, \\ x_1 + x_2 - x_5 + w &= 16, \\ x_1, x_2, x_3, x_4, x_5, w &\geq 0. \end{aligned} \quad \max \tilde{z} = \max(-w).$$

Počáteční bázecké proměnné jsou x_3, x_4, w :

$$x_3 = 6 - x_1 + 2x_2, \quad x_4 = 7 - 2x_1 - \frac{1}{2}x_2, \quad w = 16 - x_1 - x_2 + x_5.$$

Nejmenší možné w dostaneme volbou $x_5 = 0$. Z druhé rovnice platí

$$2x_1 + \frac{1}{2}x_2 \leq 7 \Rightarrow \frac{1}{2}(x_1 + x_2) \leq 2x_1 + \frac{1}{2}x_2 \leq 7 \Rightarrow x_1 + x_2 \leq 14.$$

Proto

$$w \geq 16 - (x_1 + x_2) \geq 2, \quad \Rightarrow \quad \tilde{z} \leq -2.$$

Meze dosáhneme například volbou $x_1 = 0, x_2 = 14, x_5 = 0$, pro niž jsou zbývající bázecké proměnné nezáporné a $w = 2$. Proto je $\tilde{z}^* = -2 < 0$.

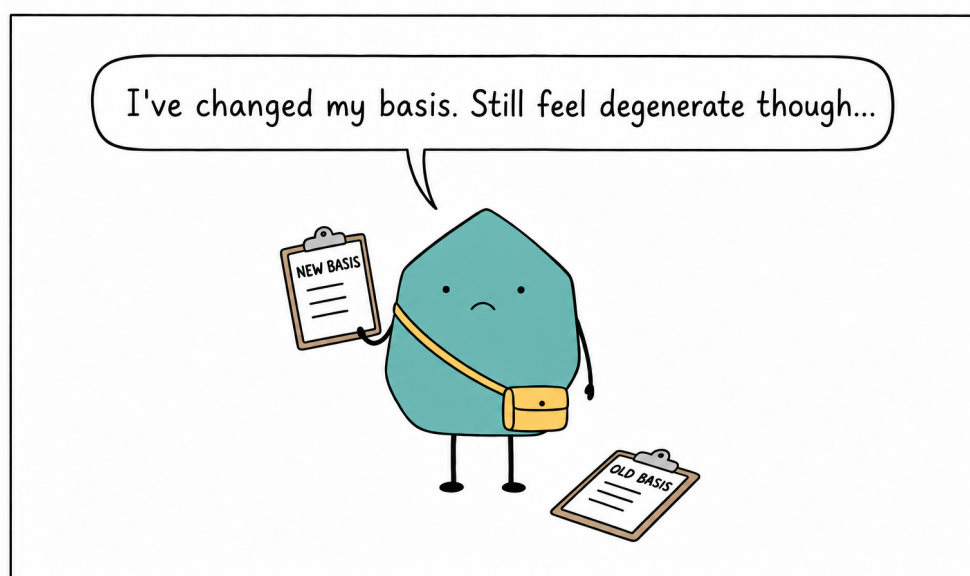
Optimální hodnota 1. fáze je záporná, tedy $w^* > 0$. Původní úloha je **nepřipustná**.

KAPITOLA 6

DEGENERACE, PIVOTOVACÍ PRAVIDLA A EFEKTIVITA

- | | | |
|---|--|----|
| 1 | Degenerace v simplexové metodě | 58 |
| 2 | Pivotovací pravidla a cyklení | 59 |
| 3 | Efektivita simplexové metody | 62 |

V předchozích kapitolách se simplexová metoda pohybovala od jednoho bázeického přípustného řešení ke druhému a hodnota účelové funkce přitom rostla. Nyní narazíme na zvláštní situaci: báze se změní, ale bod ani jeho hodnota se nepohnou. Vysvětlíme degeneraci, ukážeme, jak se simplexový výpočet může kolem jediného vrcholu doslova „otáčet“, a představíme pravidla, která zabrání cyklení. Nakonec porovnáme výborné praktické chování simplexové metody s jejími překvapivě špatnými teoretickými případy.



1 Degenerace v simplexové metodě

Degenerace se nejlépe pozná ve slovníku, ale nejprve ji pojmenujme přesně. Uvažujme rovnicový tvar

$$A\mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \quad \text{rank } A = m,$$

a přípustnou bázi B . Bázické přípustné řešení je *degenerované*, jestliže alespoň jedna jeho bázická složka je nulová. Ekvivalentně má takový bod méně než m kladných složek. Degenerace je tedy vlastností bázického bodu. Nulový krok v poměrovém testu není její definicí, ale jedním z projevů, který při výpočtu skutečně uvidíme.

Ve slovníku

$$\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \quad z = z_0 + \mathbf{r}^\top \mathbf{x}_N$$

zvolme vstupující proměnnou x_j s $r_j > 0$. Při jejím zvýšení o $\theta \geq 0$ se bázické proměnné mění podle

$$x_{B_i}(\theta) = p_i + q_{ij}\theta.$$

Přípustnost proto vyžaduje

$$0 \leq \theta \leq \min_{i: q_{ij} < 0} \frac{p_i}{-q_{ij}}.$$

Je-li minimum rovno nule, proběhne *nulový krok*: báze se změní, ale reprezentovaný bod i hodnota účelové funkce zůstanou stejné.

Geometricky jsme tedy neopustili vrchol, pouze jsme změnili způsob, jakým jej popisujeme pomocí báze. Pokud se takové nulové kroky opakují, simplexový výpočet se může kolem jediného vrcholu „otáčet“. Teprve když se některá báze zopakuje a spolu s ní se vrátí celý slovník, mluvíme o *cyklení*.

Poznámka 6.1: Nulový krok a cyklení

Jeden nebo několik nulových kroků není chybou výpočtu. Počet bází je ovšem konečný, nejvýše $\binom{n}{m}$. Kdyby tedy simplexový výpočet pokračoval nekonečně, musel by některou bázi dříve či později navštívit znovu. Cyklení může nastat pouze za degenerace. Vhodné anticyklické pravidlo je však zcela vyloučí.

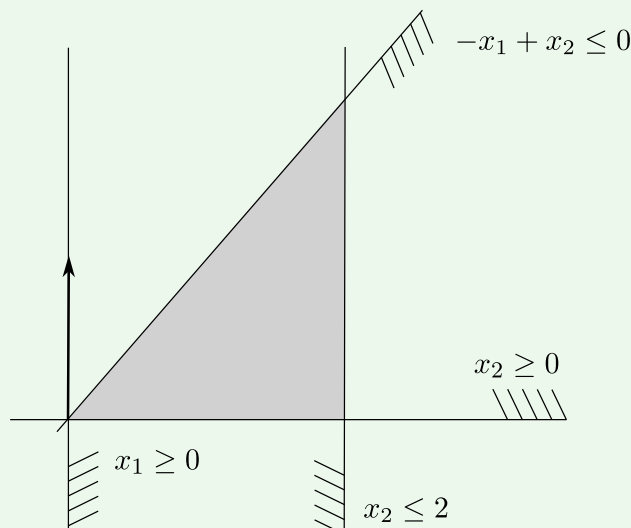
1.1 Příklad degenerovaného počátečního bodu

Příklad 6.2: Degenerovaný počáteční bod a nulový krok

Uvažujme malou úlohu, na níž lze celý jev sledovat bez dlouhých výpočtů:

$$\begin{aligned} \text{maximalizovat} \quad & x_2, \\ \text{za podmíněk} \quad & -x_1 + x_2 \leq 0, \\ & x_1 \leq 2, \\ & x_1, x_2 \geq 0. \end{aligned}$$

Přípustná oblast je znázorněna na obrázku. V počátku jsou aktivní nerovnosti $-x_1 + x_2 \leq 0$, $x_1 \geq 0$ a $x_2 \geq 0$: v rovině se zde setkává více aktivních omezení, než je pro určení vrcholu potřeba. Degenerovaný je proto počáteční bod $(0, 0)$, nikoli optimální bod $(2, 2)$.



Obrázek 6.1: Přípustná oblast. V bodě $(0,0)$ jsou současně aktivní tři nerovnosti.

Po přidání doplňkových proměnných dostaneme

$$-x_1 + x_2 + x_3 = 0, \quad x_1 + x_4 = 2, \quad x_1, x_2, x_3, x_4 \geq 0.$$

Pro bázi $B = \{3, 4\}$ má počáteční slovník tvar

$$x_3 = x_1 - x_2, \quad x_4 = 2 - x_1, \quad z = x_2.$$

Odpovídá bodu $(0, 0, 0, 2)$. Bázická proměnná x_3 je nulová, takže jde o degenerované BFS. Proměnná x_2 má kladný redukovaný náklad, a chtěli bychom ji proto zvětšit. První řádek nás ale okamžitě zastaví: při $x_1 = 0$ vyžaduje $x_3 = x_1 - x_2 \geq 0$, tedy $x_2 \leq 0$. Poměrový test dovolí jen krok $\theta = 0$. Přesto pivot provedeme: x_2 vstoupí a x_3 vystoupí,

$$x_2 = x_1 - x_3, \quad x_4 = 2 - x_1, \quad z = x_1 - x_3.$$

Báze se změnila, avšak geometricky stále stojíme v počátku a hodnota $z = 0$ se nezměnila. Nový slovník už ale dovolí skutečný pohyb: do báze může vstoupit x_1 , poměrový test dává krok 2 a z báze vystoupí x_4 . Dostaneme

$$x_1 = 2 - x_4, \quad x_2 = 2 - x_3 - x_4, \quad z = 2 - x_3 - x_4.$$

Optimální BFS je tedy $(x_1, x_2, x_3, x_4) = (2, 2, 0, 0)$. V bázi $\{x_1, x_2\}$ jsou obě bázické proměnné kladné, takže toto optimum není degenerované.

2 Pivotovací pravidla a cyklení

Viděli jsme, že degenerace může vést k nulovým krokům a v krajním případě i k cyklení. Pro samotný algoritmus je proto věta „vybereme zlepšující proměnnou“ příliš neurčitá. Je-li více nebázických proměnných s kladným redukovaným nákladem, musí algoritmus určit, která z nich vstoupí do báze. Při shodě v poměrovém testu musí rovněž určit vystupující proměnnou. Souhrn těchto voleb nazýváme *pivotovacím pravidlem*. Volba pravidla

nemění, co považujeme za optimum, může však dramaticky změnit cestu, kterou k němu simplexová metoda projde.

2.1 Pravidla pro volbu vstupující proměnné

Uvažujme značení zavedené pro obecný simplexový slovník. Máme-li několik kandidátů x_j s $r_j > 0$, nabízí se několik přirozených strategií. Každá odpovídá na trochu jinou otázku: která proměnná vypadá nejlépe na jednotku své změny, která přinese největší zlepšení v celém kroku, nebo která vede po geometricky nejstrmější hraně?

Dantzigovo pravidlo. Vstupuje index

$$j \in \arg \max_{k \in N: r_k > 0} r_k.$$

Jde o nejprůchoďejší volbu: vezmeme proměnnou, která v řádku účelové funkce slibuje největší okamžitý růst na jednotku svého zvýšení. Toto pravidlo však ještě neví, jak daleko smíme po příslušné hraně dojít, a samo nezabraňuje cyklení (Dantzig, 1963).

Pravidlo největšího zlepšení. Pro každého kandidáta j nejprve spočteme přípustný krok

$$\theta_j = \min_{i: q_{ij} < 0} \frac{p_i}{-q_{ij}}.$$

Je-li množina omezujících řádků prázdná, směr dokládá neomezenost úlohy. Jinak vybereme kandidáta s největším součinem $r_j \theta_j$. Tentokrát se tedy neptáme jen, jak prudce hodnota roste, ale i jak dlouhý krok je přípustný. Cenou za tuto informaci je větší množství poměrových testů.

Pravidlo nejstrmější hrany. Při jednotkovém zvýšení nebázické proměnné x_j má příslušný směr v prostoru všech proměnných složky

$$\mathbf{d}_N = \mathbf{e}_j, \quad \mathbf{d}_B = -A_B^{-1} \mathbf{a}_j.$$

Jeho eukleidovská délka je

$$\|\mathbf{d}\|_2 = \sqrt{1 + \|A_B^{-1} \mathbf{a}_j\|_2^2}.$$

Základní varianta pravidla proto vybírá

$$j \in \arg \max_{k \in N: r_k > 0} \frac{r_k}{\sqrt{1 + \|A_B^{-1} \mathbf{a}_k\|_2^2}}.$$

Z geometrického pohledu tedy vybíráme hranu s největším „stoupáním“ vzhledem k její délce. Jmenovatel obsahuje i jednotkovou složku vstupující proměnné. Právě odtud pochází jednička pod odmocninou. Samotná norma $\|A_B^{-1} \mathbf{a}_j\|$ by nebyla délkou úplného hranového směru. Praktické implementace používají účinné aktualizace těchto vah (Goldfarb and Reid, 1977).

Nejjednodušší je samozřejmě vybrat kandidáta náhodně. Taková volba může být užitečnou heuristikou, ale bez dalšího opatření nezaručuje konečnost konkrétního běhu.

Poznámka 6.3: Proč se někdy mluví o největším poklesu

V literatuře se pravidlo největšího zlepšení často popisuje pro minimalizační úlohu a nazývá se *maximum decrement rule*: hledá proměnnou způsobující největší pokles hodnoty účelové funkce. My pracujeme s maximalizačním slovníkem, a proto je touž myšlenkou největší zvýšení.

2.2 Anticyklická pravidla

Degenerace sama o sobě simplexovou metodu nekazí. Potíž vzniká teprve tehdy, když ne-jednoznačné volby dovolí návrat k již navštívené bázi. Následující pravidla dávají shodám pevný řád a tím takový návrat vyloučí.

Blandovo pravidlo. Pro maximalizační slovník v naší znaménkové konvenci postupujeme takto:

1. Ze všech nebázických indexů s $r_j > 0$ zvolíme nejmenší index j .
2. Pro tento sloupec provedeme poměrový test. Mezi všemi řádky, které dosahují minima

$$\frac{p_i}{-q_{ij}}, \quad q_{ij} < 0,$$

zvolíme řádek, jehož *bázická proměnná* má nejmenší index.

Bland dokázal, že toto pravidlo necyklí (Bland, 1977). Důkaz nestojí na tvrzení, že by navštívené báze tvořily obvykle lexikograficky rostoucí posloupnost. Uvažuje proměnnou s největším indexem mezi těmi, které by v předpokládaném cyklu měnily svůj bázický či nebázický stav, a odvodí spor s volbou nejmenších indexů podle pravidla.

Pravidlo je až odzbrojující prosté: při každé nejednoznačnosti dá přednost nejmenšímu indexu. Právě tato důslednost je podstatná. Nestačí ji použít jen pro vstupující proměnnou a při shodě v poměrovém testu rozhodnout libovolně.

Lexikografické pivotování. Místo samotných indexů můžeme při shodě porovnat celé vhodně normalizované řádky. První rozdílná složka pak rozhodne stejně, jako rozhoduje první rozdílné písmeno při řazení slov ve slovníku. Přesněji postupujeme následovně. Pevně uspořádejme složky a označme B_0 zvolenou výchozí přípustnou bázi. Pro již zvolený vstupující sloupec j porovnáme mezi řádky s $q_{ij} < 0$ vektory

$$\ell_i = \frac{1}{-q_{ij}} (p_i, (A_B^{-1} A_{B_0})_{i1}, \dots, (A_B^{-1} A_{B_0})_{im})$$

v uvedeném lexikografickém pořadí a vybereme řádek s nejmenším ℓ_i . První složka nejprve provede obvyklý poměrový test. Další složky jednoznačně rozhodnou případnou shodu. Pro zvolený vstupující sloupec je tak vystupující řádek jednoznačný. Má-li být jednoznačný celý pivotovací krok, musí být předem určeno také pravidlo pro volbu vstupující proměnné. Lexikografická přípustnost přitom zaručuje, že se žádná báze nezopakuje.

Symbolická perturbace pravé strany. Lexikografické pravidlo lze chápat i jinak: rovné hodnoty od sebe symbolicky nepatrně odsuneme. Pracujeme s hierarchicky malou perturbací

$$\mathbf{b}(\varepsilon) = \mathbf{b} + A_{B_0}(\varepsilon, \varepsilon^2, \dots, \varepsilon^m)^\top, \quad 0 < \varepsilon \ll 1,$$

takže počáteční bázecký vektor je $A_{B_0}^{-1}\mathbf{b} + (\varepsilon, \dots, \varepsilon^m)^\top$ a jeho původně nulové složky jsou symbolicky kladné. Výrazy se nevyhodnocují v plovoucí řádové čárce, ale porovnávají se podle první nenulové mocniny ε . Taková generická hierarchická perturbace odstraní rovnosti odpovědné za degeneraci navštívených bází. Po výpočtu necháme $\varepsilon \rightarrow 0^+$. Pro původní úlohu s konečným optimelem tak získáme optimální limitní bod. Libovolné malé navzájem různé desetinné přídatky tuto vlastnost obecně nezaručují a mohou změnit numerický problém.

Poznámka 6.4: Perturbace je myšlenková pomůcka

Čísla $\varepsilon, \varepsilon^2, \dots$ nemáme bez rozmyslu dosazovat jako velmi malá desetinná čísla. Zacházíme s nimi symbolicky a po skončení necháme $\varepsilon \rightarrow 0^+$. Původní úlohu tedy nechceme potají změnit. Chceme pouze jednoznačně rozhodnout situace, v nichž by několik kroků bylo z pohledu neperturované úlohy nerozlišitelných.

3 Efektivita simplexové metody

V praxi je simplexová metoda pozoruhodně úspěšná i u úloh s velkým počtem proměnných a omezení. Právě proto se stala jedním z pracovních nástrojů optimalizace, nikoli jen hezkým algoritmem do učebnice. Teorie nám však vzápětí připomene, že dobré praktické chování není totéž jako záruka pro každý možný vstup. Počet pivotů závisí na geometrii úlohy, počáteční bázi i pivotovacím pravidle. Bez přesného vymezení třídy dat proto nelze slíbit pevný násobek počtu omezení.

3.1 Klee–Mintyho příklad

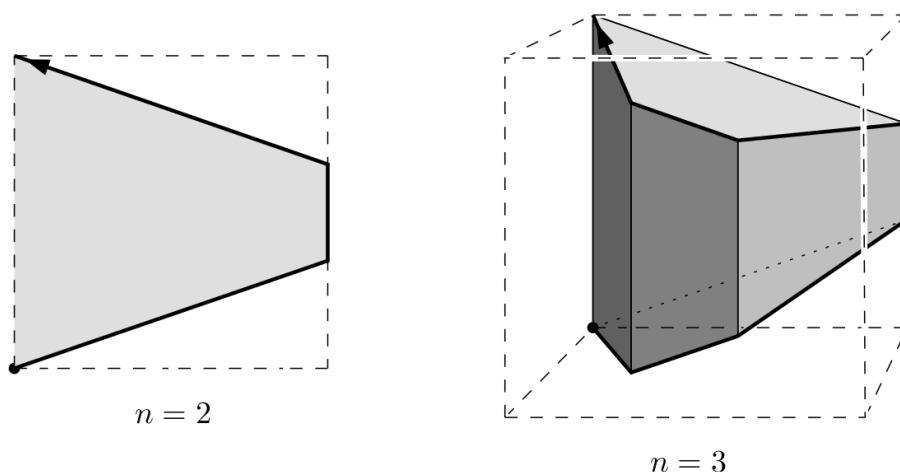
Z teoretického hlediska přišla v roce 1972 studená sprcha. Victor Klee a George Minty sestrojili rodinu lineárních programů, na níž Dantzigovo pravidlo od počáteční báze doplňkových proměnných navštíví všech 2^n vrcholů deformované n -rozměrné krychle (Klee and Minty, 1972). Jedna běžná podoba této rodiny je

$$\begin{aligned} &\text{maximalizovat} && \sum_{j=1}^n 2^{n-j} x_j, \\ &\text{za podmíněk} && \sum_{j=1}^{i-1} 2^{i-j+1} x_j + x_i \leq 5^i, \quad i = 1, \dots, n, \\ &&& x_1, \dots, x_n \geq 0. \end{aligned}$$

Pro $n = 3$ tedy dostáváme konkrétní úlohu

$$\begin{aligned} \text{maximalizovat} \quad & 4x_1 + 2x_2 + x_3, \\ & x_1 \leq 5, \\ & 4x_1 + x_2 \leq 25, \\ & 8x_1 + 4x_2 + x_3 \leq 125, \\ & x_1, x_2, x_3 \geq 0. \end{aligned}$$

Krychle je „zešikmená“ právě tak, aby lokálně nejlákavější volba vedla algoritmus po velmi dlouhé trase. Od počátku a při Dantzigově pravidle potřebuje simplexová metoda $2^n - 1$ pivotů. Jde o exponenciální dolní mez pro tuto konkrétní kombinaci rodiny úloh, počáteční báze a pivotovacího pravidla, nikoli o tvrzení, že každé pravidlo prochází touto úlohou stejně.



Schematická představa deformované krychle. Přesný průběh simplexové trajektorie závisí na koeficientech a pivotovacím pravidle.

Obrázek zachycuje pouze myšlenku deformované krychle. Skutečná soustava koeficientů, která donutí Dantzigovo pravidlo projít všechny vrcholy ve správném pořadí, je podstatně bizarnější. Autor si zde proto dovoluje ušetřit čtenáře i vlastní grafické schopnosti a ponechává na obrázku jen geometrickou představu.

3.2 Teoretické hranice

Kleeho–Mintyho příklad neříká, že je lineární programování samo o sobě exponenciálně těžké. Lineární programování lze řešit v čase polynomiálním vzhledem k bitové délce racionálně zadaných dat, například elipsoidovou metodou nebo vhodnou metodou vnitřního bodu (Khachiyan, 1979, Karmarkar, 1984). To však samo o sobě neomezuje počet simplexových pivotů polynomem v počtu proměnných a omezení.

Zde je snadné smísit dvě různé otevřené otázky. Pro simplexovou metodu není známo efektivně realizovatelné pivotovací pravidlo, které by na každé úloze provedlo jen polynomiálně mnoho pivotů. Nové dolní meze vylučují další široké třídy přirozených pravidel, nikoli však všechna možná pravidla (Disser et al., 2026). Zvláště zůstává otevřena existence libovolného silně polynomiálního algoritmu pro obecné lineární programování (Allamigeon et al., 2021). Efektivní pravidlo s polynomiálním počtem pivotů by poskytlo

silně polynomiální simplexový algoritmus. Případný jiný silně polynomiální algoritmus by však nemusel určovat žádnou simplexovou trajektorii.

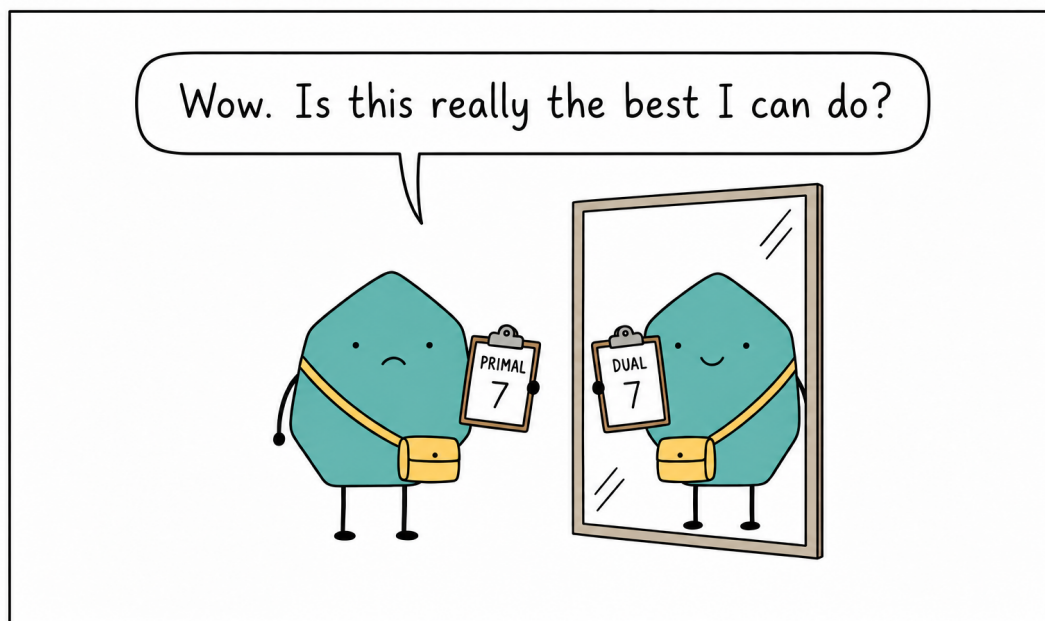
Podobně nesmíme zaměnit krátkou cestu, která v grafu existuje, za cestu, kterou skutečně vybere algoritmus. Průměr grafu mnohostěnu a počet pivotů určité simplexové trajektorie nejsou totožné veličiny. Průměr je nejkratší vzdálenost mezi dvojicemi vrcholů, zatímco pivotovací pravidlo může zvolit mnohem delší cestu. Původní Hirschova domněnka o lineární mezi pro průměr obecných mnohostěnů je navíc vyvrácena (Santos, 2012). Sama by proto ani v kladném případě neposkytovala polynomiální záruku pro libovolné pivotovací pravidlo.

KAPITOLA 7

DUALITA LINEÁRNÍHO PROGRAMOVÁNÍ I

- | | | |
|---|---|----|
| 1 | Motivace a konstrukce du-
ální úlohy | 66 |
| 2 | Slabá a silná dualita | 69 |

Dualita nám dovolí podívat se na jediný lineární program ze dvou stran. Z jedné strany hledáme nejlepší výrobní plán, tok nebo rozvrh. Z druhé oceňujeme zdroje tak, aby jejich celková cena uhlídala hodnotu každého přípustného plánu. Nejprve tuto možná překvapivou myšlenku odvodíme na příkladu, potom uvedeme obecná pravidla dualizace, slabou a silnou větu o dualitě a podmínky komplementarity.



1 Motivace a konstrukce duální úlohy

Lineární programy mají jednu důležitou a možná trochu překvapivou vlastnost: každou úlohu můžeme číst dvěma způsoby. První čtení už známe. Hledáme například nejlepší výrobní plán tak, aby byl zisk co největší a současně jsme nepřekročili dostupné zdroje. Druhé čtení obrací pozornost od výrobků ke zdrojům: jaké ceny bychom jim museli přiřadit, aby žádný přípustný plán nemohl mít větší hodnotu, než je cena všech použitých zdrojů?

První formulaci budeme nazývat *primární úlohou*, druhou *duální úlohou*. Nejde o dvě nesouvisející úlohy, které se náhodou podobají. Jsou to dva matematické popisy téže situace: jeden hledá nejlepší rozhodnutí a druhý nejlepší důkaz, že už žádné rozhodnutí nemůže být lepší.

Algebraická cesta k tomuto druhému pohledu je překvapivě jednoduchá. Uvažujme maximalizační úlohu. Nezápornou lineární kombinací jejích omezení můžeme někdy vytvořit nerovnost, jejíž levá strana dominuje účelové funkci. Pravá strana této kombinace je pak horní mezí hodnoty každého přípustného řešení. Duální úloha hledá nejlepší takovou mez.

Příklad 7.1: Horní meze z kombinací omezení

Uvažujme úlohu

$$\begin{array}{ll} \text{maximalizovat} & 2x_1 + 3x_2, \\ \text{za podmínek} & 4x_1 + 8x_2 \leq 12, \\ & 2x_1 + x_2 \leq 3, \\ & 3x_1 + 2x_2 \leq 4, \\ & x_1, x_2 \geq 0. \end{array}$$

Aniž bychom hledali optimum, okamžitě můžeme napsat například

$$2x_1 + 3x_2 \leq 4x_1 + 8x_2 \leq 12.$$

Hodnota 12 je tedy bezpečný, ale zjevně nepřiliš chytrý horní odhad. Omezení nemusíme používat pouze po jednom. Součet jedné třetiny prvního a jedné třetiny druhého omezení dává

$$2x_1 + 3x_2 \leq \frac{1}{3}(4x_1 + 8x_2) + \frac{1}{3}(2x_1 + x_2) \leq \frac{1}{3}12 + \frac{1}{3}3 = 5.$$

Hodnota 5 je tedy lepší horní mezí optima. Nabízí se pokračovat: která nezáporná kombinace omezení vytvoří vůbec nejnižší platnou horní mez?

Označme nezáporné váhy tří omezení y_1, y_2, y_3 . Jejich kombinace má levou stranu

$$(4y_1 + 2y_2 + 3y_3)x_1 + (8y_1 + y_2 + 2y_3)x_2.$$

Aby dominovala účelové funkci pro každé $\mathbf{x} \geq \mathbf{0}$, musí platit

$$4y_1 + 2y_2 + 3y_3 \geq 2, \quad 8y_1 + y_2 + 2y_3 \geq 3.$$

Koeficienty y_i zde hrají dvojí roli. Algebraicky jsou vahami, jimiž skládáme původní nerovnosti. V příběhu o výrobě je ale můžeme číst jako ceny jednotek tří zdrojů. Duální omezení pak říkají, že cena zdrojů spotřebovaných jedním kusem výrobku

nesmí být nižší než jeho zisk. Nejlepší horní mez proto hledá duální úloha

$$\begin{array}{ll} \text{minimalizovat} & 12y_1 + 3y_2 + 4y_3, \\ \text{za podmíněk} & 4y_1 + 2y_2 + 3y_3 \geq 2, \\ & 8y_1 + y_2 + 2y_3 \geq 3, \\ & y_1, y_2, y_3 \geq 0. \end{array}$$

Primární řešení $\mathbf{x}^* = (\frac{1}{2}, \frac{5}{4})^T$ i duální řešení $\mathbf{y}^* = (\frac{5}{16}, 0, \frac{1}{4})^T$ jsou přípustná a obě mají hodnotu 19/4. První nám říká „takto dobře umím vyrábět“, druhé „lépe vyrábět nelze“. Slabá dualita níže ukáže, že tato shoda současně dokazuje optimalitu obou řešení.

1.1 Standardní primárně-duální dvojice

Pro matici $A \in \mathbb{R}^{m \times n}$, vektory $\mathbf{b} \in \mathbb{R}^m$ a $\mathbf{c} \in \mathbb{R}^n$ tvoří standardní dvojici

$$\max \mathbf{c}^T \mathbf{x} \quad \text{při} \quad A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \quad (\text{P})$$

a

$$\min \mathbf{b}^T \mathbf{y} \quad \text{při} \quad A^T \mathbf{y} \geq \mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}. \quad (\text{D})$$

Úlohu (P) nazýváme *primární* a (D) *duální*. Jejich přípustné množiny jsou zpravidla odlišné. Slovem „duální“ neříkáme, že jde o tutéž optimalizační úlohu. Jejich těsný vztah se týká zejména mezi a optimálních hodnot.

Ve výrobní interpretaci má primární proměnná x_j význam množství j -tého výrobku, zatímco duální proměnná y_i oceňuje i -tý zdroj. Každý sloupec matice A tak lze číst jako recept na jeden výrobek a každý řádek jako rozpočet jednoho zdroje. Transpozice v duálu pouze prohodí tyto dvě perspektivy. Matice se nezmění, jen ji jednou čteme po řádcích a podruhé po sloupcích.

Každé *duálně přípustné* $\mathbf{y}$, tedy vektor splňující současně $\mathbf{y} \geq \mathbf{0}$ a $A^T \mathbf{y} \geq \mathbf{c}$, poskytuje pro všechna primárně přípustná $\mathbf{x}$ horní mez

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}.$$

Samotná podmínka $\mathbf{y} \geq \mathbf{0}$ k tomuto závěru nestačí.

1.2 Obecná pravidla dualizace

Standardní dvojice se pamatuje snadno, skutečná zadání ale obsahují i rovnosti, obrácené nerovnosti a volné proměnné. Následující recept není nová teorie, pouze pečlivé hlídání znamének. Vyjděme z maximalizační primární úlohy, jejíž jednotlivá omezení mohou mít smysl $\leq$, $\geq$ nebo $=$, a jejíž proměnné mohou být nezáporné, nekladné nebo volné. Každému primárnímu omezení odpovídá jedna duální proměnná a každé primární proměnné jedno duální omezení. Platí:

Prvek maximalizační primární úlohy	Odpovídající prvek duální úlohy
omezení i ve smyslu $\leq$	$y_i \geq 0$
omezení i ve smyslu $\geq$	$y_i \leq 0$
omezení i ve smyslu $=$	$y_i \in \mathbb{R}$
proměnná $x_j \geq 0$	$(A^T \mathbf{y})_j \geq c_j$
proměnná $x_j \leq 0$	$(A^T \mathbf{y})_j \leq c_j$
volná proměnná $x_j \in \mathbb{R}$	$(A^T \mathbf{y})_j = c_j$

Duální účelová funkce je $\min \mathbf{b}^T \mathbf{y}$. Všimněme si jednoduché symetrie: primární omezení vytvářejí duální proměnné a primární proměnné vytvářejí duální omezení. Je-li výchozí primární úloha minimalizační, můžeme ji nejprve převést na maximalizační změnou znaménka účelové funkce, nebo použít symetrická pravidla. Dualizujeme-li ještě jednou, dostaneme po obvyklém ztotožnění zápisů zpět původní úlohu.

Příklad 7.2: Dualizace smíšených směrů a znamének

Pro úlohu

$$\begin{aligned}
 &\text{maximalizovat} && 3x_1 + x_2 - 2x_3, \\
 &&& x_1 + 2x_2 - x_3 \leq 4, \\
 &&& 2x_1 - x_2 + x_3 \geq 1, \\
 &&& x_1 + x_2 + x_3 = 2, \\
 &&& x_1 \geq 0, \quad x_2 \leq 0, \quad x_3 \in \mathbb{R}
 \end{aligned}$$

zavedeme $y_1 \geq 0$, $y_2 \leq 0$ a volnou proměnnou y_3 . Duální úloha je

$$\begin{aligned}
 &\text{minimalizovat} && 4y_1 + y_2 + 2y_3, \\
 &&& y_1 + 2y_2 + y_3 \geq 3, \\
 &&& 2y_1 - y_2 + y_3 \leq 1, \\
 &&& -y_1 + y_2 + y_3 = -2, \\
 &&& y_1 \geq 0, \quad y_2 \leq 0, \quad y_3 \in \mathbb{R}.
 \end{aligned}$$

Směr každého duálního omezení určuje znaménko odpovídající primární proměnné, nikoli směr některého jednotlivého primárního omezení.

1.3 Jiný pohled: Lagrangeovské odvození

Předchozí konstrukce skládala nerovnosti a je pro první setkání s dualitou nejpřirozenější. Tatáž myšlenka se dá zapsat jazykem Lagrangeových multiplikátorů. Tento pohled není pro praktickou dualizaci nutný, ale ukazuje, proč se dualita lineárního programování později bez násilí zobecní do konvexní optimalizace.

Pro standardní úlohu (P) a multiplikátor $\mathbf{y} \geq \mathbf{0}$ definujeme Lagrangian

$$L(\mathbf{x}, \mathbf{y}) = \mathbf{c}^T \mathbf{x} + \mathbf{y}^T (\mathbf{b} - A\mathbf{x}) = \mathbf{b}^T \mathbf{y} + \mathbf{x}^T (\mathbf{c} - A^T \mathbf{y}).$$

Je-li $\mathbf{x}$ primárně přípustné, pak $\mathbf{b} - A\mathbf{x} \geq \mathbf{0}$, a tedy $L(\mathbf{x}, \mathbf{y}) \geq \mathbf{b}^T \mathbf{y}$. Proto je

$$g(\mathbf{y}) = \sup_{\mathbf{x} \geq \mathbf{0}} L(\mathbf{x}, \mathbf{y})$$

horní mezí primární optimální hodnoty. Pokud má $\mathbf{c} - A^T \mathbf{y}$ kladnou složku, je $g(\mathbf{y}) = +\infty$. Je-li naopak $A^T \mathbf{y} \geq \mathbf{c}$, supremum je dosaženo v $\mathbf{x} = \mathbf{0}$ a

$$g(\mathbf{y}) = \mathbf{b}^T \mathbf{y}.$$

Minimalizací této konečné horní meze dostáváme právě úlohu (D). Multiplikátor y_i zde není mírou „aktivity“ omezení. Jeho vztah k aktivitě přesně popíší až podmínky komplementarity.

2 Slabá a silná dualita

Duální úloha má primární optimum „hlídat shora“. Slabá dualita potvrzuje, že tuto strážní roli plní každé duálně přípustné řešení. Silná dualita pak říká něco mnohem překvapivějšího: je-li konečného optima dosaženo, nejlepší horní mez se s ním přesně setká.

Věta 7.3: Slabá věta o dualitě

Je-li $\mathbf{x}$ přípustné pro (P) a $\mathbf{y}$ přípustné pro (D), potom

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}.$$

Je-li tedy (P) neomezená shora, (D) je nepřipustná. Je-li (D) neomezená zdola, (P) je nepřipustná.

Důkaz věty.

Stejnou myšlenku dokážeme dvěma způsoby. První je krátký řetězec nerovností, druhý ukáže, z čeho se skládá rozdíl mezi duální a primární hodnotou.

První pohled: řetězec nerovností. Z $\mathbf{x} \geq \mathbf{0}$ a $A^T \mathbf{y} \geq \mathbf{c}$ dostáváme první nerovnost a z $\mathbf{y} \geq \mathbf{0}$ a $A\mathbf{x} \leq \mathbf{b}$ druhou:

$$\mathbf{c}^T \mathbf{x} \leq (A^T \mathbf{y})^T \mathbf{x} = \mathbf{y}^T A\mathbf{x} \leq \mathbf{y}^T \mathbf{b} = \mathbf{b}^T \mathbf{y}.$$

Druhý pohled: mezeru duality. Rozdíl obou hodnot lze rozepsat jako

$$\begin{aligned} \mathbf{b}^T \mathbf{y} - \mathbf{c}^T \mathbf{x} &= \mathbf{y}^T (\mathbf{b} - A\mathbf{x}) + \mathbf{x}^T (A^T \mathbf{y} - \mathbf{c}) \\ &\geq 0. \end{aligned}$$

Každý z obou skalárních součinů je nezáporný, neboť v něm násobíme dva nezáporné vektory. Rozdíl nazýváme *mezerou duality*. Tento druhý důkaz nejen znovu potvrzuje její nezápornost, ale později nám také umožní přesně určit, kdy je mezeru nulová.

Kdyby byla primární úloha neomezená shora a existovalo duálně přípustné $\mathbf{y}$, konečné číslo $\mathbf{b}^T \mathbf{y}$ by podle právě dokázané nerovnosti omezovalo všechny primární hodnoty shora, což je spor. Druhý důsledek je symetrický. ■

Věta 7.4: Silná věta o dualitě

Pro standardní dvojici (P), (D) platí: je-li jedna z úloh přípustná a její účelová funkce je v příslušném směru omezená, mají obě úlohy optimální řešení a jejich optimální hodnoty jsou stejné. Ekvivalentně nastane právě jedna z možností:

1. obě úlohy jsou nepřípustné,
2. (P) je neomezená shora a (D) je nepřípustná,
3. (P) je nepřípustná a (D) je neomezená zdola,
4. obě úlohy mají optimální řešení $\mathbf{x}^*, \mathbf{y}^*$ a

$$\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*.$$

Slabá dualita nechávala mezi nejlepším plánem a nejlepším horním odhadem možnou mezeru. Silná dualita ji uzavírá: v případě konečného optima existuje ocenění zdrojů, které hodnotu optimálního plánu vystihne přesně. Důkaz uvedeme v následující kapitole nejprve ze simplexové tabulky a poté pomocí Farkasova lemmatu. Jde o jeden ze základních výsledků lineárního programování (Dantzig, 1963, Matoušek and Gärtner, 2007).

2.1 Podmínky komplementarity

Rovnost optimálních hodnot má ještě jemnější důsledek. V rozkladu mezery duality je každý sčítanec nezáporný, a proto se jejich součet může rovnat nule jen tehdy, když zmizí každý z nich zvlášť. Právě to vyjadřují podmínky komplementarity.

Věta 7.5: Podmínky komplementarity

Nechť $\mathbf{x}$ je přípustné pro (P) a $\mathbf{y}$ přípustné pro (D). Potom jsou $\mathbf{x}$ a $\mathbf{y}$ optimální právě tehdy, když

$$y_i(b_i - \mathbf{a}_i^T \mathbf{x}) = 0 \quad (i = 1, \dots, m)$$

a současně

$$x_j((A^T \mathbf{y})_j - c_j) = 0 \quad (j = 1, \dots, n).$$

Důkaz věty.

Pro přípustný pár jsou všechny členy v rozkladu mezery duality

$$\sum_{i=1}^m y_i(b_i - \mathbf{a}_i^T \mathbf{x}) + \sum_{j=1}^n x_j((A^T \mathbf{y})_j - c_j)$$

nezáporné. Mezera je proto nulová právě tehdy, když je nulový každý uvedený součin. Nulová mezera podle slabé duality znamená, že žádné primárně přípustné řešení nemůže mít vyšší hodnotu než $\mathbf{x}$ a žádné duálně přípustné řešení nižší hodnotu než $\mathbf{y}$. Obě řešení jsou optimální. Naopak pro optimální pár je mezera podle silné duality nulová. ■

Podmínky mají dva důležité, ale pouze jednosměrné důsledky:

$$b_i - \mathbf{a}_i^T \mathbf{x}^* > 0 \implies y_i^* = 0, \quad y_i^* > 0 \implies \mathbf{a}_i^T \mathbf{x}^* = b_i,$$

a obdobně

$$(A^T \mathbf{y}^*)_j > c_j \implies x_j^* = 0, \quad x_j^* > 0 \implies (A^T \mathbf{y}^*)_j = c_j.$$

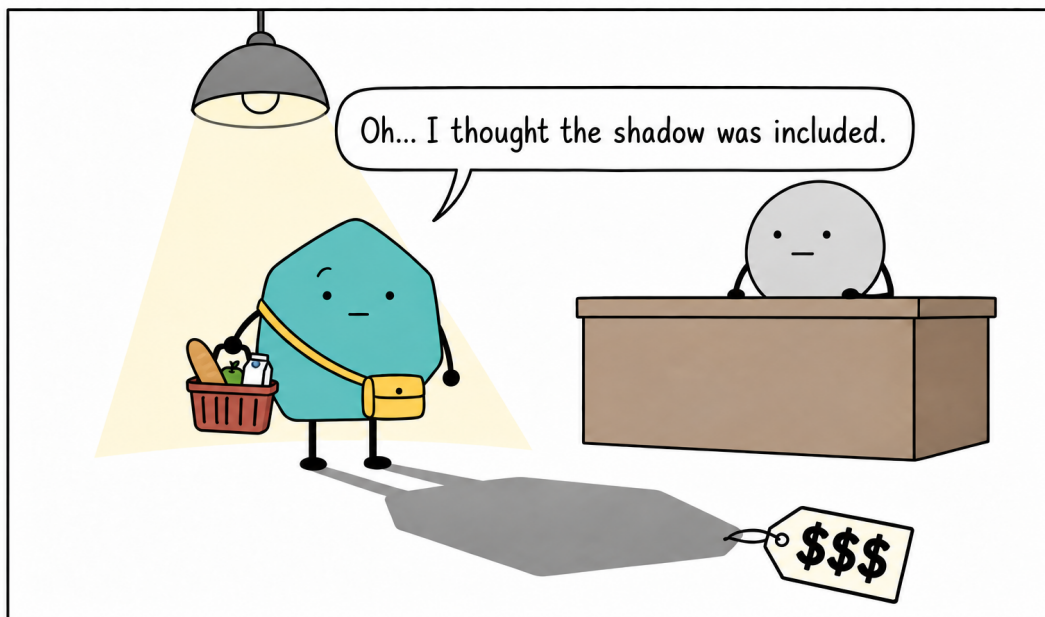
Jinými slovy: má-li zdroj v optimu rezervu, jeho cena musí být nulová. Má-li kladnou cenu, musí být zcela vyčerpán. Pozor však na obrácené implikace. Aktivní primární omezení může mít nulovou duální proměnnou a těsné duální omezení může odpovídat nulové primární proměnné, zejména při degeneraci nebo při více optimálních řešeních.

KAPITOLA 8

DUALITA LINEÁRNÍHO PROGRAMOVÁNÍ II

1	Důkaz silné duality z optimální báze	74
2	Farkasovo lemma a druhý důkaz silné duality	75
3	Stínové ceny a citlivost na pravé strany	78
4	Duální simplexová metoda	79

V minulé kapitole jsme silné dualitě zatím museli uvěřit. Nyní dluh splatíme, a dokonce dvakrát: poprvé přečteme duální řešení z optimální simplexové tabulky, podruhé je získáme pomocí Farkasova lemmatu a geometrie konvexního kužele. Potom se vrátíme od důkazů k interpretaci duálních proměnných jako stínových cen. Na závěr obrátíme logiku simplexové metody a představíme její duální variantu.



1 Důkaz silné duality z optimální báze

První důkaz se dívá přímo do poslední simplexové tabulky. Spoléhá proto na správnost a konečnost simplexové metody s anticyklickým pravidlem. Čtenář, který tomuto algoritmu východisku zatím nedůvěřuje, nemusí zoufat: druhý důkaz v této kapitole povede jinou cestou.

Připomeňme standardní primárně-duální dvojici

$$\max \mathbf{c}^T \mathbf{x} \quad \text{při} \quad A\mathbf{x} \leq \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0} \quad (\text{P})$$

a

$$\min \mathbf{b}^T \mathbf{y} \quad \text{při} \quad A^T \mathbf{y} \geq \mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}. \quad (\text{D})$$

Předpokládejme, že (P) je přípustná a její účelová funkce je shora omezená. Po přidání doplňkových proměnných $\mathbf{s} \geq \mathbf{0}$ dostaneme

$$\tilde{A}\tilde{\mathbf{x}} = \mathbf{b}, \quad \tilde{A} = (A \ I_m), \quad \tilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ \mathbf{s} \end{pmatrix}, \quad \tilde{\mathbf{c}} = \begin{pmatrix} \mathbf{c} \\ \mathbf{0} \end{pmatrix}.$$

Matice $\tilde{A}$ má plnou řádkovou hodnost. Základní věta lineárního programování a konečný simplexový postup s anticyklickým pravidlem proto poskytují závěrečnou bázi B splňující kritérium optimality: $\mathbf{p} \geq \mathbf{0}$ a $\mathbf{r} \leq \mathbf{0}$. Těmto dvěma podmínkám zde společně říkáme optimalita báze. Samotná optimalita příslušného BFS při degeneraci nestačí. Právě nekladné redukováné náklady se ukážou být skrytým duálním certifikátem. Označme

$$\mathbf{p} = \tilde{A}_B^{-1} \mathbf{b} \geq \mathbf{0}, \quad \mathbf{y}^* = \tilde{A}_B^{-T} \tilde{\mathbf{c}}_B.$$

Lemma 8.1: Duální řešení z optimální báze

Vektor $\mathbf{y}^*$ je přípustný pro (D) a hodnota příslušného optimálního BFS $\mathbf{x}^*$ splňuje

$$\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*.$$

Důkaz lemmatu.

Kandidáta na duální řešení jsme ze závěrečné báze již vyrobili. Musíme ověřit dvě věci: že je skutečně duálně přípustný a že jeho hodnota souhlasí s hodnotou nalezeného primárního řešení.

Pro každý sloupec j rozšířené matice definujeme redukováný náklad

$$\tilde{r}_j = \tilde{c}_j - \tilde{\mathbf{a}}_j^T \mathbf{y}^*.$$

U bázeických sloupců je $\tilde{r}_j = 0$, protože

$$\tilde{A}_B^T \mathbf{y}^* = \tilde{\mathbf{c}}_B.$$

U nebázeických sloupců jsou redukováné náklady v optimální tabulce nekladné. Pro všechny sloupce tedy platí

$$\tilde{A}^T \mathbf{y}^* \geq \tilde{\mathbf{c}}.$$

Prvních n složek této nerovnosti dává $A^T \mathbf{y}^* \geq \mathbf{c}$. Posledních m složek odpovídá sloupcům jednotkové matice a dává $\mathbf{y}^* \geq \mathbf{0}$. Tady je důležitý krok, který nelze vynechat: duální nerovnosti samy nestačí, musí platit i znaménková podmínka. Vektor $\mathbf{y}^*$ je tedy skutečně duálně přípustný.

Hodnota bázeického řešení je

$$\begin{aligned}\mathbf{c}^\top \mathbf{x}^* &= \tilde{\mathbf{c}}_B^\top \tilde{A}_B^{-1} \mathbf{b} = (\tilde{A}_B^{-\top} \tilde{\mathbf{c}}_B)^\top \mathbf{b} \\ &= (\mathbf{y}^*)^\top \mathbf{b} = \mathbf{b}^\top \mathbf{y}^*.\end{aligned}$$

Našli jsme tedy primárně přípustné $\mathbf{x}^*$ a duálně přípustné $\mathbf{y}^*$ se stejnou hodnotou. Slabá dualita mezi ně už nepustí žádné lepší řešení, a proto jsou obě optimální. Tím je dokázána hlavní část silné duality. Je-li namísto primární úlohy přípustná a omezená duální úloha, použijeme stejný argument na duál a skutečnost, že duál duálu je původní úloha. Případy neomezenosti a nepřípustnosti pak doplňuje slabá dualita. Důkaz současně vysvětluje, proč simplexové multiplikátory $\mathbf{y}^* = \tilde{A}_B^{-\top} \tilde{\mathbf{c}}_B$ tvoří duální řešení (Dantzig, 1963).

2 Farkasovo lemma a druhý důkaz silné duality

Farkasovo lemma je *věta alternativy*: právě jedna ze dvou soustav má řešení. Jedna soustava popisuje hledaný nezáporný rozklad, druhá poskytuje certifikát, že takový rozklad neexistuje (Farkas, 1902, Matoušek and Gärtner, 2007). Než si lemma zapíšeme algebraicky, vytvořme si jeho geometrický obraz.

2.1 Bod, kužel a oddělovací nadrovina

Definice 8.2: Konvexní kužel

Nechť $\mathbf{a}_1, \dots, \mathbf{a}_n \in \mathbb{R}^m$. *Konvexní kužel* generovaný těmito vektory je množina všech jejich nezáporných lineárních kombinací,

$$\text{cone}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} = \left\{ \sum_{j=1}^n x_j \mathbf{a}_j : x_j \geq 0 \right\}.$$

Představme si, že ze sloupců matice $A = (\mathbf{a}_1 \cdots \mathbf{a}_n)$ stavíme pomocí nezáporných koeficientů vektor $\mathbf{b}$. Všechny vektory, které takto dokážeme postavit, vyplní kužel

$$C = \text{cone}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} = \{A\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}.$$

Soustava $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, má řešení právě tehdy, když bod $\mathbf{b}$ leží v tomto kuželi. Jestliže v něm neleží, chceme tuto skutečnost potvrdit jedinou nadrovinou: celý kužel zůstane na její nezáporné straně, zatímco $\mathbf{b}$ bude striktně na straně záporné. Normálový vektor této nadroviny bude právě hledaným certifikátem.

2.2 Algebraická formulace a důkaz

Lemma 8.3: Farkasovo lemma

Nechť $A \in \mathbb{R}^{m \times n}$ a $\mathbf{b} \in \mathbb{R}^m$. Právě jedna z následujících soustav má řešení:

$$(F1) \quad A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0},$$

(F2) existuje $\mathbf{y} \in \mathbb{R}^m$ takové, že $A^T \mathbf{y} \geq \mathbf{0}$ a $\mathbf{b}^T \mathbf{y} < 0$.

Důkaz lemmatu.

Nemohou nastat obě možnosti současně. Kdyby $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$, a zároveň $A^T \mathbf{y} \geq \mathbf{0}$, potom

$$\mathbf{b}^T \mathbf{y} = \mathbf{x}^T A^T \mathbf{y} \geq 0,$$

což odporuje podmínce ve (F2).

Zbývá dokázat existenci certifikátu, jestliže (F1) nemá řešení. Označme sloupce matice A jako $\mathbf{a}_1, \dots, \mathbf{a}_n$ a položme

$$C = \text{cone}\{\mathbf{a}_1, \dots, \mathbf{a}_n\} = \{A\mathbf{x} : \mathbf{x} \geq \mathbf{0}\}.$$

Konvexita C plyne přímo z definice. Uzavřenost si krátce zdůvodněme. Každý bod C lze vyjádřit nezápornou kombinací lineárně nezávislé podmnožiny generátorů. Jsou-li totiž generátory s kladnými koeficienty závislé, existuje na jejich indexech nenulový vektor $\mathbf{d}$ s $A\mathbf{d} = \mathbf{0}$. Po případné změně znaménka má některou složku $d_j > 0$. Pro

$$\lambda = \min_{j: d_j > 0} \frac{x_j}{d_j}$$

je $\mathbf{x} - \lambda \mathbf{d} \geq \mathbf{0}$, popisuje tentýž bod kužele a alespoň jedna jeho kladná složka zmizí. Po konečně mnoha opakováních zůstanou jen nezávislé generátory.

Kužel C je tedy sjednocením konečně mnoha kuželů $\{A_I \mathbf{x}_I : \mathbf{x}_I \geq \mathbf{0}\}$, kde sloupce A_I jsou lineárně nezávislé. Prázdná podmnožina dává pouze bod $\mathbf{0}$. Každý ostatní z těchto kuželů je uzavřený: jestliže $A_I \mathbf{x}_I^{(k)} \rightarrow \mathbf{v}$, pak

$$\mathbf{x}_I^{(k)} = (A_I^T A_I)^{-1} A_I^T (A_I \mathbf{x}_I^{(k)})$$

konverguje k nezápornému vektoru $\mathbf{x}_I$ a $\mathbf{v} = A_I \mathbf{x}_I$. Konečné sjednocení těchto uzavřených kuželů je uzavřené, a tedy je uzavřený i C .

Protože $\mathbf{b} \notin C$, existuje bod $\mathbf{z} \in C$ nejbližší bodu $\mathbf{b}$. Pro každé $\mathbf{w} \in C$ a $t \in [0, 1]$ leží $\mathbf{z} + t(\mathbf{w} - \mathbf{z})$ v C . Minimum vzdálenosti v $t = 0$ dává po derivaci zprava

$$(\mathbf{b} - \mathbf{z})^T (\mathbf{w} - \mathbf{z}) \leq 0.$$

Geometrický význam této nerovnosti je prostý: kdyby některý směr z $\mathbf{z}$ dovnitř kužele svíral s vektorem k bodu $\mathbf{b}$ ostrý úhel, mohli bychom se po něm k $\mathbf{b}$ ještě přiblížit. To by odporovalo volbě nejbližšího bodu. Protože C je kužel, můžeme dosadit $\mathbf{w} = \mathbf{0}$ a $\mathbf{w} = 2\mathbf{z}$. Dostaneme

$$(\mathbf{b} - \mathbf{z})^T \mathbf{z} = 0.$$

Položme $\mathbf{y} = \mathbf{z} - \mathbf{b}$. Pro všechna $\mathbf{w} \in C$ pak platí $\mathbf{y}^T \mathbf{w} \geq 0$, a tedy zejména

$$A^T \mathbf{y} \geq \mathbf{0}.$$

Navíc

$$\mathbf{b}^T \mathbf{y} = \mathbf{b}^T (\mathbf{z} - \mathbf{b}) = -\|\mathbf{b} - \mathbf{z}\|_2^2 < 0,$$

kde jsme použili $(\mathbf{b} - \mathbf{z})^T \mathbf{z} = 0$ a $\mathbf{b} \neq \mathbf{z}$. Vektor $\mathbf{y}$ tedy splňuje (F2). ■

Geometricky tvrzení říká: buď bod $\mathbf{b}$ leží v kuželu generovaném sloupci matice A , nebo

jej od tohoto kuželu odděluje nadrovina s normálou $\mathbf{y}$. Nerovnosti $\mathbf{y}^\top \mathbf{a}_j \geq 0$ umísťují celý kužel do jednoho uzavřeného poloprostoru, zatímco $\mathbf{y}^\top \mathbf{b} < 0$ umísťuje bod $\mathbf{b}$ do opačného otevřeného poloprostoru.

Pro dualitu potřebujeme následující důsledek.

Lemma 8.4: Farkasovo lemma pro nerovnosti

Soustava

$$H\mathbf{x} \leq \mathbf{h}, \quad \mathbf{x} \geq \mathbf{0}$$

je nepřipustná právě tehdy, když existuje $\boldsymbol{\lambda} \geq \mathbf{0}$ takové, že

$$H^\top \boldsymbol{\lambda} \geq \mathbf{0}, \quad \mathbf{h}^\top \boldsymbol{\lambda} < 0.$$

Důkaz lemmatu.

Po přidání doplňkových proměnných je systém ekvivalentní rovnostem

$$(H \ I) \begin{pmatrix} \mathbf{x} \\ \mathbf{s} \end{pmatrix} = \mathbf{h}, \quad \mathbf{x}, \mathbf{s} \geq \mathbf{0}.$$

Použití Farkasova lemmatu na matici $(H \ I)$ dává $(H \ I)^\top \boldsymbol{\lambda} \geq \mathbf{0}$, což je právě dvojice podmínek $H^\top \boldsymbol{\lambda} \geq \mathbf{0}$ a $\boldsymbol{\lambda} \geq \mathbf{0}$. ■

2.3 Silná dualita z Farkasova lemmatu

Myšlenka důkazu je následující. Nad optimální hodnotu γ položíme latku o libovolně malé $\varepsilon > 0$. Žádné primární přípustné řešení přes ni nemůže přeskočit, a Farkasovo lemma proto vydá certifikát neřešitelnosti. Po vhodné normalizaci se z tohoto certifikátu stane téměř optimální duální řešení. Když necháme ε klesat k nule, mezera se uzavře.

Latka nad primárním optimem. Předpokládejme, že (P) má optimální řešení $\mathbf{x}^*$, a označme $\gamma = \mathbf{c}^\top \mathbf{x}^*$. Pro každé $\varepsilon > 0$ je systém

$$A\mathbf{x} \leq \mathbf{b}, \quad -\mathbf{c}^\top \mathbf{x} \leq -(\gamma + \varepsilon), \quad \mathbf{x} \geq \mathbf{0}$$

nepřipustný. Použijeme předchozí lemma s

$$H = \begin{pmatrix} A \\ -\mathbf{c}^\top \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} \mathbf{b} \\ -(\gamma + \varepsilon) \end{pmatrix}.$$

Farkasův certifikát. Existují tedy $\mathbf{u} \geq \mathbf{0}$ a $t \geq 0$ takové, že

$$A^\top \mathbf{u} - t\mathbf{c} \geq \mathbf{0}, \quad \mathbf{b}^\top \mathbf{u} - t(\gamma + \varepsilon) < 0.$$

Vynásobením první nerovnosti vektorem $\mathbf{x}^* \geq \mathbf{0}$ a využitím $A\mathbf{x}^* \leq \mathbf{b}$ dostaneme

$$\mathbf{b}^\top \mathbf{u} \geq \mathbf{u}^\top A\mathbf{x}^* \geq t\mathbf{c}^\top \mathbf{x}^* = t\gamma.$$

Proto musí být $t > 0$. Pro $t = 0$ by poslední dvě nerovnosti dávaly současně $\mathbf{b}^\top \mathbf{u} \geq 0$ a $\mathbf{b}^\top \mathbf{u} < 0$.

Normalizace na duální řešení. Položme $\mathbf{y}_\varepsilon = \mathbf{u}/t$. Pak

$$\mathbf{y}_\varepsilon \geq \mathbf{0}, \quad A^T \mathbf{y}_\varepsilon \geq \mathbf{c}, \quad \gamma \leq \mathbf{b}^T \mathbf{y}_\varepsilon < \gamma + \varepsilon.$$

Duální úloha je tedy přípustná a její infimum je podle slabé duality a posledního odhadu rovno γ . Její účelová funkce je na přípustném mnohostěnu zdola omezená, takže základní věta lineárního programování zaručuje dosažení minima. Existuje proto duální optimální řešení $\mathbf{y}^*$ a

$$\mathbf{b}^T \mathbf{y}^* = \gamma = \mathbf{c}^T \mathbf{x}^*.$$

Druhý směr a zbývající alternativy. Tím jsme dokázali směr od optimálního řešení (P) k optimálnímu řešení (D) se stejnou hodnotou. Je-li naopak (D) přípustná a její účelová funkce zdola omezená, má optimální řešení. Zapišeme ji jako standardní maximalizační úlohu

$$\max (-\mathbf{b})^T \mathbf{y} \quad \text{při} \quad (-A^T) \mathbf{y} \leq -\mathbf{c}, \quad \mathbf{y} \geq \mathbf{0}.$$

Právě dokázaný směr aplikovaný na tuto úlohu dává optimum jejího duálu, který je po změně znaménka účelové funkce totožný s (P). Jde o skutečnost, že duál duálu je původní úloha. Slabá dualita dále vylučuje duálně přípustné řešení, je-li (P) neomezená shora, a symetricky vylučuje primárně přípustné řešení, je-li (D) neomezená zdola. Je-li jedna úloha nepřípustná a druhá přípustná, nemůže být druhá omezená v příslušném směru, jinak by právě dokázaný výsledek zaručil přípustnost první. Tím dostáváme všechny čtyři alternativy silné duality bez použití simplexové tabulky.

3 Stínové ceny a citlivost na pravé strany

Duální proměnné jsou důležitým a zároveň velmi intuitivním výstupem lineárního programování. Zatímco primární proměnné popisují konkrétní rozhodnutí – kolik vyrábět, kudy vést materiál nebo které aktivity spustit – duální proměnné oceňují omezení, tedy zdroje, kapacity a limity použité v modelu.

U kapacitního omezení se optimální duální proměnná y_i^* často nazývá *stínová cena*. Představme si například, že b_i udává počet hodin na stroji. Stínová cena odpovídá na otázku: kolik má v současném optimu hodnotu jedna další hodina? Slovo „současném“ je podstatné. Odpověď je lokální a platí jen tak dlouho, dokud se nezmění optimální báze.

Toto tvrzení nyní zapišeme přesně. Po přechodu do rovnicového tvaru označme jeho matici $\tilde{A}$, vektor koeficientů účelové funkce $\tilde{\mathbf{c}}$ a optimální bázi B splňující $\mathbf{p} \geq \mathbf{0}$, $\mathbf{r} \leq \mathbf{0}$. Položme

$$\mathbf{y}^* = \tilde{A}_B^{-T} \tilde{\mathbf{c}}_B.$$

Změní-li se pouze pravá strana z $\mathbf{b}$ na $\mathbf{b} + \Delta \mathbf{b}$, redukované náklady se nemění. Pokud

$$\tilde{A}_B^{-1}(\mathbf{b} + \Delta \mathbf{b}) \geq \mathbf{0},$$

zůstává stejná báze přípustná, a tedy i optimální. V této oblasti platí *přesně*

$$\begin{aligned} v(\mathbf{b} + \Delta \mathbf{b}) &= \tilde{\mathbf{c}}_B^T \tilde{A}_B^{-1}(\mathbf{b} + \Delta \mathbf{b}) \\ &= v(\mathbf{b}) + (\mathbf{y}^*)^T \Delta \mathbf{b}. \end{aligned}$$

Nejde pouze o neurčitý odhad. Linearita je přesná, dokud zůstává tatáž báze přípustná. Jakmile narazíme na hranici této oblasti, může se optimální báze změnit a s ní i stínová cena.

Podmínky komplementarity z předchozí kapitoly dávají

$$b_i - \mathbf{a}_i^T \mathbf{x}^* > 0 \implies y_i^* = 0, \quad y_i^* > 0 \implies \mathbf{a}_i^T \mathbf{x}^* = b_i.$$

To dává praktické pravidlo: má-li zdroj rezervu, jeho stínová cena je nulová. Má-li kladnou stínovou cenu, omezení je aktivní a zdroj je zcela využit. Obrácené implikace ale obecně neplatí. Aktivní omezení může mít nulovou stínovou cenu, například při redundanci nebo degeneraci. Rovněž $y_i^* = 0$ samo o sobě neznamená, že má omezení rezervu nebo že je možné je odstranit.

V příkladu z předchozí kapitoly bylo

$$\mathbf{y}^* = \left(\frac{5}{16}, 0, \frac{1}{4}\right)^T.$$

Dokud zůstává stejná báze přípustná, změna pravých stran splňuje

$$\Delta v = \frac{5}{16} \Delta b_1 + \frac{1}{4} \Delta b_3.$$

Druhé omezení má pro tuto bázi nulovou lokální cenu. Tento krátký vztah je přímočarým modelem typu „co kdyby“: dokud báze zůstává přípustná, nemusíme celou úlohu po každé malé změně řešit znovu.

Poznámka 8.5: Jak číst stínové ceny

Kladná stínová cena ukazuje na zdroj, jehož malé rozšíření může zlepšit optimum. Není to ještě automatický investiční pokyn. Hodnoty y_i^* lze mezi zdroji porovnávat až po zohlednění jejich fyzikálních a ekonomických jednotek, ceny získání další jednotky a intervalu, v němž zůstává současná báze přípustná. Největší číselná stínová cena proto sama o sobě neurčuje nejlepší investici.

Poznámka 8.6: Když je optimálních cen více

Duální optimální řešení nemusí být jediné. Různé optimální vektory $\mathbf{y}^*$ mají pro současnou pravou stranu stejnou celkovou hodnotu, mohou však jednotlivé zdroje oceňovat odlišně. Proto při nejednoznačném duálním optimu nelze bez další analýzy mluvit o jediné „té správné“ stínové ceně ani mechanicky předpovědět změnu pro libovolný směr. Pro náš výklad postačí vzorec spojený s konkrétní optimální bází a intervalem, v němž tato báze zůstává přípustná. Obecnější popis patří do analýzy citlivosti a konvexní optimalizace (Boyd and Vandenberghe, 2004, Rockafellar, 1970).

Z tabulky získáme duální vektor bezpečně vztahem

$$(\mathbf{y}^*)^T = \tilde{\mathbf{c}}_B^T \tilde{A}_B^{-1}.$$

Ve standardním tvaru s doplňkovými sloupci I_m je redukovaný náklad i -té doplňkové proměnné roven $-y_i^*$. Umístění v konkrétním grafickém řádku tabulky závisí na použitém formátu. Nelze je obecně popsat jako „předposlední řádek“ bez stanovení konvence.

4 Duální simplexová metoda

Klasická primární simplexová metoda začíná na „správném místě“, tedy v primárně přípustné bázi. Tuto přípustnost po celou dobu chrání a postupně odstraňuje kladné redukované náklady. Duální simplexová metoda obrátí role: začne sice v bodě, který může

porušovat některé primární podmínky, ale její redukované náklady jsou už na správné straně. Při každém pivotu zachovává duální přípustnost a napravljuje záporné bázičké proměnné, dokud se obě podmínky nepotkají v optimu (Lemke, 1954, Dantzig, 1963).

Pracujeme důsledně se slovníkem

$$\boxed{\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \quad z = z_0 + \mathbf{r}^T \mathbf{x}_N,} \quad Q = -A_B^{-1} A_N.$$

Pro maximalizační úlohu znamená

$$\mathbf{p} \geq \mathbf{0} \quad \text{primární přípustnost,} \quad \mathbf{r} \leq \mathbf{0} \quad \text{duální přípustnost.}$$

Platí-li obě podmínky, bázičké řešení je optimální: pro každé přípustné $\mathbf{x}_N \geq \mathbf{0}$ totiž $z = z_0 + \mathbf{r}^T \mathbf{x}_N \leq z_0$.

4.1 Odvození pivotového pravidla

V primárním simplexu jsme nejprve volili zlepšující sloupec a poměrovým testem hledali řádek, který nás zastaví. Zde postup obrátíme. Nejprve vybereme porušený řádek, tedy zápornou bázičkou proměnnou, a teprve potom hledáme sloupec, který ji může napravit, aniž by pokazil duální přípustnost.

Nechť $p_s < 0$. Řádek vystupující bázičké proměnné má tvar

$$x_{B_s} = p_s + \sum_{j \in N} q_{sj} x_j.$$

Aby bylo možné zápornou pravou stranu zvýšením některé nebázičké proměnné napravit, musí pro vstupující index platit $q_{sj} > 0$. Jestliže $q_{sj} \leq 0$ pro všechna $j \in N$, pak pro každé $\mathbf{x}_N \geq \mathbf{0}$

$$x_{B_s} \leq p_s < 0,$$

a původní úloha je nepřípustná.

Pro kandidáta v s $q_{sv} > 0$ vyřešíme klíčový řádek pro x_v :

$$x_v = -\frac{p_s}{q_{sv}} + \frac{1}{q_{sv}} x_{B_s} - \sum_{j \in N \setminus \{v\}} \frac{q_{sj}}{q_{sv}} x_j.$$

Po dosazení do účelové funkce jsou nové redukované náklady

$$r'_{B_s} = \frac{r_v}{q_{sv}}, \quad r'_j = r_j - \frac{r_v}{q_{sv}} q_{sj} \quad (j \in N \setminus \{v\}).$$

Protože původně $\mathbf{r} \leq \mathbf{0}$, zůstávají složky s $q_{sj} \leq 0$ automaticky nekladné. Pro všechny sloupce s $q_{sj} > 0$ potřebujeme

$$\frac{r_v}{q_{sv}} \geq \frac{r_j}{q_{sj}}.$$

Duální poměrový test tedy volí *největší*, nikoli *nejmenší* poměr r_j/q_{sj} . Právě zde bývají chyby ve znaménkách nejčastější: pravidlo vždy čteme z konkrétního slovníku $\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N$, nikoli z tabulky převzaté z jiné znaménkové konvence.

Jeden krok duální simplexové metody

Předpokládejme $\mathbf{r} \leq \mathbf{0}$.

1. Je-li $\mathbf{p} \geq \mathbf{0}$, aktuální bázecké řešení je optimální.
2. Jinak zvolíme řádek s s $p_s < 0$, například nejzápornější pravou stranu.
3. Je-li $q_{sj} \leq 0$ pro všechna $j \in N$, úloha je nepřipustná.
4. Jinak zvolíme

$$v \in \arg \max_{j \in N: q_{sj} > 0} \frac{r_j}{q_{sj}}$$

a pivotujeme: x_v vstoupí do báze a x_{B_s} z ní vystoupí.

5. Opakujeme od prvního kroku.

Při shodách a degeneraci je třeba stanovit anticyklické pravidlo, například vhodnou duální variantu Blandova pravidla nebo lexikografické pivotování. Samotná volba nejzápornější pravé strany konečnost nezaručuje.

4.2 Úplný číselný příklad

Krátký příklad ověří současně obě části pravidla: kandidátní sloupec musí mít v porušeném řádku kladný koeficient a mezi kandidáty vybíráme největší podíl r_j/q_{sj} .

Příklad 8.7: Jeden krok duální simplexové metody

Uvažujme rovnicovou úlohu

$$\text{maximalizovat } z = -x_3 - 3x_4,$$

$$x_1 - x_3 - 2x_4 = -1,$$

$$x_2 + x_3 - x_4 = 3,$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

Pro bázi $B = \{1, 2\}$ dostaneme slovník

$$x_1 = -1 + x_3 + 2x_4, \quad x_2 = 3 - x_3 + x_4, \quad z = -x_3 - 3x_4.$$

Vektor redukovaných nákladů $(-1, -3)$ je nekladný, takže báze je duálně přípustná. Není však primárně přípustná, protože $p_1 = -1$.

V prvním řádku jsou oba kandidátní koeficienty kladné. Poměry jsou

$$\frac{r_3}{q_{13}} = -1, \quad \frac{r_4}{q_{14}} = -\frac{3}{2}.$$

Volíme větší poměr, tedy vstupuje x_3 a vystupuje x_1 . Z prvního řádku

$$x_3 = 1 + x_1 - 2x_4.$$

Po dosazení do ostatních řádků dostaneme

$$x_3 = 1 + x_1 - 2x_4, \quad x_2 = 2 - x_1 + 3x_4, \quad z = -1 - x_1 - x_4.$$

Nová báze $\{x_3, x_2\}$ je primárně přípustná, protože její pravé strany jsou $(1, 2)$, a stále duálně přípustná, protože nové redukované náklady jsou $(-1, -1)$. Slovník je tedy optimální:

$$(x_1, x_2, x_3, x_4) = (0, 2, 1, 0), \quad z^* = -1.$$

Příklad zároveň ukazuje význam znamének: kandidáty jsou sloupce s $q_{1j} > 0$. Opačná podmínka by zde chybně prohlásila přípustnou úlohu za nepřípustnou.

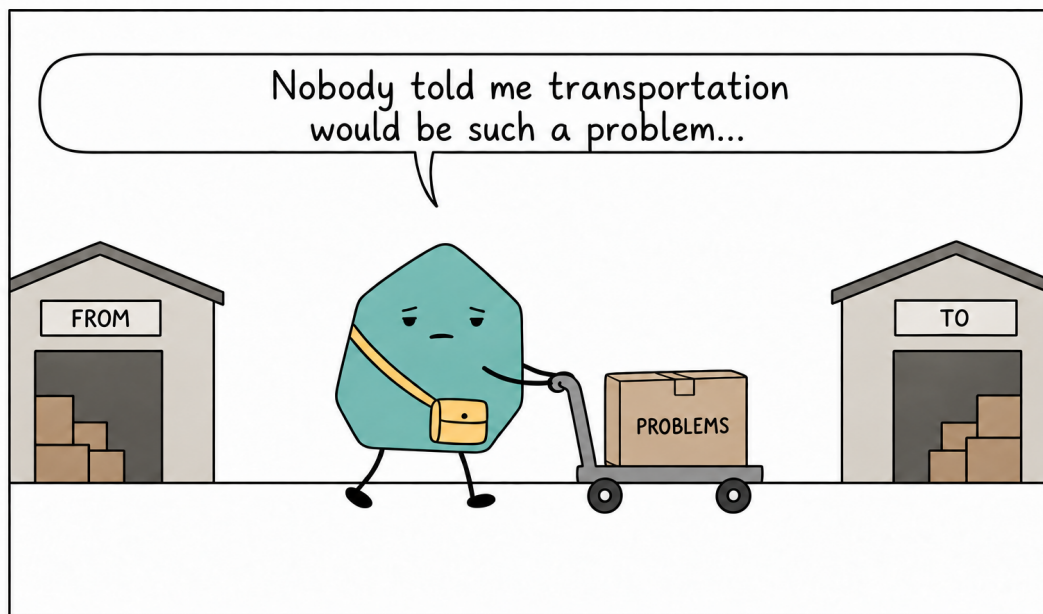
Duálně přípustná báze bývá přirozeně k dispozici při reoptimalizaci. Představme si, že už máme optimálně vyřešený výrobní model a dodatečně přidáme nové kapacitní omezení. Redukované náklady mohou zůstat nekladné, ale nová pravá strana poruší primární přípustnost. Duální simplex pak nemusí zahazovat všechnu předchozí práci: z původní optimální báze znovu „opraví“ přípustnost. Pokud výchozí báze duálně přípustná není, je třeba ji získat jiným korektním postupem. Samotný duální simplex tuto inicializační úlohu neřeší.

KAPITOLA 9

DOPRAVNÍ PROBLÉM

1	Slovní zadání a značení . . .	84
2	Formulace dopravního problému jako LP	84
3	Struktura a vlastnosti dopravního problému	86
4	Metoda MODI (metoda potenciálů)	90

Dopravní problém hledá nejlevnější způsob přepravy od dodavatelů k odběratelům při daných zásobách a požadavcích. Ukážeme jeho formulaci jako lineárního programu, popíšeme strukturu omezení, bážická přípustná řešení a degeneraci. Poté představíme *metodu potenciálů* (MODI) pro nalezení optimálního řešení.



1 Slovní zadání a značení

Dopravní problém patří mezi klasické a zároveň nejpraktičtější modely lineárního programování. Přirozeně se objevuje všude tam, kde je potřeba efektivně přiřadit zdroje k odběratelům a minimalizovat související náklady – od logistiky a distribuce zboží, přes plánování výroby, až po alokaci energie a podobně. Nejdříve danou úlohu formulujeme.

Uvažujme m dodavatelů (zdrojů) $S_1, \dots, S_m$ a n odběratelů $D_1, \dots, D_n$, kde $m, n \geq 1$. Dodavatel S_i má k dispozici množství $s_i \geq 0$ (kapacita, zásoba), odběratel D_j požaduje množství $d_j \geq 0$ (poptávka). Pro každou dvojici (i, j) je dána jednotková cena (náklad) přepravy $c_{ij} \geq 0$.

Rozhodovací proměnné dopravního problému budeme značit

$$x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n,$$

a budeme si je interpretovat jako množství přepravené ze zdroje S_i k odběrateli D_j .

Poznámka 9.1: Slovní zadání dopravního problému

Najít takové přiděly $\{x_{ij}\}$, aby

- celková zásilka od zdroje S_i nepřekročila dostupnou zásobu s_i ,
- celkový odběr u odběratele D_j pokryl požadovanou poptávku d_j ,
- celkové přepravní náklady byly minimální.

V typické „čisté“ variantě dopravního problému budeme předpokládat, že *celá zásoba má být odvezena a celá poptávka má být uspokojena*. Formálně to znamená

$$\sum_{j=1}^n x_{ij} = s_i, \quad i = 1, \dots, m,$$

$$\sum_{i=1}^m x_{ij} = d_j, \quad j = 1, \dots, n.$$

2 Formulace dopravního problému jako LP

2.1 Kanonická (vyrovnaná) formulace

Začněme standardním případem, kdy je součet všech zásob stejný jako součet všech požadavků:

$$\sum_{i=1}^m s_i = \sum_{j=1}^n d_j.$$

Takovému zadání se často říká *vyrovnaný* (balanced) dopravní problém.

LP formulace dopravního problému

Pro vyrovnaný dopravní problém hledáme řešení minimalizační úlohy

$$\begin{aligned}
 &\text{minimalizovat} && \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\
 &\text{při} && \sum_{j=1}^n x_{ij} = s_i, \quad i = 1, \dots, m, \\
 &&& \sum_{i=1}^m x_{ij} = d_j, \quad j = 1, \dots, n, \\
 &&& x_{ij} \geq 0 \quad \text{pro všechna } i, j.
 \end{aligned} \tag{9.1}$$

Vidíme, že jde o úlohu lineárního programování s mn proměnnými a $m + n$ rovnicemi (které však nejsou všechny nezávislé, viz níže).

2.2 Nevyrovaná varianta dopravního problému

Vyrovaná varianta dopravního problému má mnoho výhodných vlastností, v praxi často ale přirozeně nastává, že

$$\sum_{i=1}^m s_i \neq \sum_{j=1}^n d_j.$$

Tedy například součet zásob převyšuje součet poptávky (část kapacity zůstane nevyužita), nebo naopak zásoby na všechny požadavky nestačí. Taková varianta se nazývá *nevyrovnaný* dopravní problém. Dobrou zprávou je, že ji lze po doplnění fiktivního řádku či sloupce výpočetně převést na vyrovnanou tabulku. Musíme si však předem přesně říci, co fiktivní přiděly znamenají – zejména při nedostatku nabídky tím původní model nejen přepisujeme, ale také měníme.

Poznámka 9.2: Vyrovnání dopravního problému

- Pokud $\sum_i s_i > \sum_j d_j$, přidáme do tabulky „fiktivního odběratele“ D_{n+1} s poptávkou

$$d_{n+1} := \sum_{i=1}^m s_i - \sum_{j=1}^n d_j$$

a jednotkovými náklady $c_{i,n+1} = 0$. Proměnná $x_{i,n+1}$ pak reprezentuje nevyužitou zásobu dodavatele i . Tato úprava je ekvivalentní modelu, v němž jsou nabídky horními mezemi a všechny skutečné poptávky musí být uspokojeny. Má-li likvidace nebo skladování přebytku skutečný náklad, dosadíme místo nuly právě tento náklad.

- Pokud $\sum_i s_i < \sum_j d_j$, přidáme „fiktivního dodavatele“ S_{m+1} se zásobou

$$s_{m+1} := \sum_{j=1}^n d_j - \sum_{i=1}^m s_i$$

a náklady $c_{m+1,j}$. Příděl $x_{m+1,j}$ vyjadřuje neuspokojenou část poptávky odběratele j , proto se za $c_{m+1,j}$ obvykle volí odpovídající jednotková penalizace nedostatku.

Ve druhém případě nejde o ekvivalentní převod původního modelu s povinným uspokojením všech poptávek: ten je při nedostatku nabídky nepřipustný. Fiktivní dodavatel model mění tím, že výslovně dovoluje nedostatek a oceňuje jej zvolenou penalizací. Nulové náklady by znamenaly, že neuspokojená poptávka není nijak postihována.

V dalším textu analyzujeme vyrovnanou úlohu (9.1). Nevyrovnaný model lze těmito metodami řešit až po výše popsaném doplnění a po jasném stanovení významu i nákladů fiktivních přidělů.

3 Struktura a vlastnosti dopravního problému

Matice omezení v (9.1) má velmi speciální strukturu. Prvních m řádků matice popisuje součty v řádcích dopravní tabulky a dalších n řádků součty v jejích sloupcích. To má několik důsledků, které krátce rozebereme v této sekci.

3.1 Počet nezávislých omezení a bázecká řešení

Nejprve ukažme, že některé z omezení musí být nutně lineárně závislé. Součet všech řádkových omezení (nabídkových) dává

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = \sum_{i=1}^m s_i,$$

zatímco součet všech sloupcových omezení (poptávkových) dává

$$\sum_{j=1}^n \sum_{i=1}^m x_{ij} = \sum_{j=1}^n d_j.$$

Levá strana obou rovností je totožná, jde pouze o jinak zapsaný dvojný součet. Ve vyrovnaném dopravním problému jsou stejné také pravé strany, tedy $\sum_i s_i = \sum_j d_j$.

Pro důkaz hodnosti označme symbolem $\mathbf{r}_i$ koeficientový řádek i -tého nabídkového omezení a symbolem $\mathbf{c}_j$ koeficientový řádek j -tého poptávkového omezení. Oba typy řádků patří do $\mathbb{R}^{mn}$. Pravé strany s_i, d_j nejsou součástí matice A . Z předešlého plyne, že

$$\sum_{i=1}^m \mathbf{r}_i - \sum_{j=1}^n \mathbf{c}_j = \mathbf{0}.$$

Jde o netriviální lineární kombinaci řádků matice A , která je rovna nule, a proto

$$\text{rank}(A) \leq m + n - 1.$$

Nyní ověříme, že žádná další lineární závislost mezi omezeními neexistuje. Uvažujme lineární kombinaci

$$\sum_{i=1}^m \alpha_i \mathbf{r}_i + \sum_{j=1}^{n-1} \beta_j \mathbf{c}_j = \mathbf{0}.$$

Každá proměnná x_{ij} má nenulový koeficient právě v řádku $\mathbf{r}_i$ a (pokud $j < n$) v řádku $\mathbf{c}_j$. Proto koeficient u x_{ij} v této lineární kombinaci je

$$\alpha_i + \beta_j \quad (\text{pro } j < n),$$

a u proměnných x_{in} , pro něž jsme řádek $\mathbf{c}_n$ vynechali, je koeficient roven pouze α_i . Aby celá lineární kombinace byla nulová, musí být všechny tyto koeficienty rovny nule. Z proměnných x_{in} tedy dostáváme

$$\alpha_1 = \alpha_2 = \dots = \alpha_m = 0,$$

a dosazením do ostatních koeficientů dostáváme

$$\beta_1 = \beta_2 = \dots = \beta_{n-1} = 0.$$

Jediná lineární kombinace daných $m + n - 1$ rovnic, která dává nulu, je tedy triviální, což znamená, že uvedených $m + n - 1$ rovnic je lineárně nezávislých.

Teď už víme, že $\text{rank}(A) = m + n - 1$. Aby pojem báze přesně odpovídal definici z předchozích kapitol, odstraníme například poslední poptávkovou rovnici a označíme vzniklou matici $\bar{A} \in \mathbb{R}^{(m+n-1) \times mn}$ a příslušnou pravou stranu $\bar{\mathbf{b}}$. Ve vyrovnané úloze je odstraněná rovnice důsledkem všech nabídkových rovnic a prvních $n - 1$ poptávkových rovnic, takže

$$A\mathbf{x} = \mathbf{b} \iff \bar{A}\mathbf{x} = \bar{\mathbf{b}}.$$

Matice $\bar{A}$ má plnou řádkovou hodnost $m + n - 1$. Bází dopravní úlohy budeme proto rozumět výběr $m + n - 1$ lineárně nezávislých sloupců právě této redukované matice. Přesné kritérium přípustnosti je nyní velmi jednoduché.

Věta 9.3: Charakterizace přípustnosti dopravního problému

Za předpokladu $s_i, d_j \geq 0$ jsou pro dopravní problém s rovnostními omezeními (9.1) následující podmínky ekvivalentní:

1. Úloha má báziké přípustné řešení.
2. Úloha má optimální řešení.
3. Jedná se o vyrovnaný dopravní problém, tj. platí rovnost

$$\sum_{i=1}^m s_i = \sum_{j=1}^n d_j.$$

Důkaz věty.

Každé přípustné řešení po sečtení řádkových a sloupcových rovnic nutně splňuje

$$\sum_i s_i = \sum_j d_j.$$

Je-li naopak úloha vyrovnaná, metoda severozápadního rohu popsaná níže sestrojí báziké přípustné řešení ekvivalentního redukovaného systému. Přípustná množina je uzavřená a omezená, neboť $0 \leq x_{ij} \leq \min\{s_i, d_j\}$. Lineární účelová funkce na ní proto nabývá minima. A konečně, existuje-li optimální řešení, základní věta lineárního programování zaručuje existenci optimálního bázikého přípustného řešení. ■

Podrobnější výklad síťové struktury dopravní úlohy a jejích bází uvádějí (Bazaraa et al., 2010, Ahuja et al., 1993).

Rozebereme dále strukturu matice omezení. Ta má rozměr $(m + n) \times mn$ a přirozené blokové členění

$$A = \begin{pmatrix} A^{(S)} \\ A^{(D)} \end{pmatrix}, \quad A^{(S)} \in \mathbb{R}^{m \times mn}, \quad A^{(D)} \in \mathbb{R}^{n \times mn},$$

kde blok $A^{(S)}$ obsahuje (nabídková) omezení

$$\sum_{j=1}^n x_{ij} = s_i, \quad i = 1, \dots, m,$$

a blok $A^{(D)}$ (poptávková) omezení

$$\sum_{i=1}^m x_{ij} = d_j, \quad j = 1, \dots, n.$$

Každému políčku tabulky, tj. proměnné x_{ij} , odpovídá jeden sloupec matice A . Tento sloupec má právě dvě jedničky: jednu v řádku příslušného dodavatele S_i a jednu v řádku příslušného odběratele D_j , ostatní prvky jsou nulové.

Pro konkrétní případ se třemi dodavateli a třemi odběrateli dostaneme například

$$A = \begin{array}{c|cccccccccc} & x_{11} & x_{12} & x_{13} & x_{21} & x_{22} & x_{23} & x_{31} & x_{32} & x_{33} \\ \hline S_1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ S_2 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ S_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ D_1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ D_2 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ D_3 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

První tři řádky zde vyjadřují, že součet přes řádky tabulky dává nabídky s_1, s_2, s_3 , další tři řádky vyjadřují, že součet přes sloupce dává poptávky d_1, d_2, d_3 .

Poznamenejme, že je užitečné dívat se na dopravní problém jako na síťový model na bipartitním grafu. Vrcholy tvoří zdroje S_i a odběratelé D_j , hrana mezi S_i a D_j reprezentuje proměnnou x_{ij} .

Poznámka 9.4: Bipartitní graf

Bipartitní graf je graf, jehož množinu vrcholů lze rozdělit na dva disjunktní podmnožiny U a V tak, že každá hrana spojuje vrchol z U s vrcholem z V . Mezi vrcholy uvnitř téhož dílu tedy neexistují žádné hrany.

V dopravním problému tvoří přirozené dvě části množina zdrojů $\{S_1, \dots, S_m\}$ a množina odběratelů $\{D_1, \dots, D_n\}$. Struktura dopravní tabulky je tedy přesně strukturou bipartitního grafu.

Poznámka 9.5: Stromová struktura báze

Dobrou zprávou je, že zde abstraktní pojem báze dostává velmi názornou podobu. Necht F je množina buněk dopravní tabulky, chápaná jako množina hran bipartitního grafu na vrcholech $\{S_1, \dots, S_m\} \cup \{D_1, \dots, D_n\}$. Sloupce redukované matice $\bar{A}$ s indexy v F jsou lineárně nezávislé právě tehdy, když F neobsahuje cyklus. Skutečně, po vynásobení všech poptávkových řádků číslem -1 dostaneme redukovanou vrcholově-hranovou incidenční matici. Sloupce cyklu mají střídavou netriviální lineární závislost. Nezávislost lesa se naopak dokáže postupným odstraněním listu. Má-li F právě $m + n - 1$ hran, je takový les souvislý, a tedy je kostrou bipartitního grafu. Báze dopravní úlohy proto odpovídají právě kostrám (Bazaraa et al., 2010, Ahuja et al., 1993).

Řešení dopravního problému i hledání výchozího báze příпустného řešení je proto přirozené organizovat v tabulce. Sloupce znázorňují odběratele a řádky dodavatele. V nedegenerovaném řešení lze kladné báze přiděly zapsat do příslušných buněk a nebáze buňky ponechat prázdné. V degenerovaném řešení však může být báze i buňka s nulovým přidělem. Takovou buňku je nutné v tabulce zvlášť označit, aby byla báze určena jednoznačně.

	D_1	D_2	D_3	nabídka
S_1	c_{11} x_{11}	c_{12} x_{12}	c_{13} x_{13}	s_1
S_2	c_{21} x_{21}	c_{22} x_{22}	c_{23} x_{23}	s_2
S_3	c_{31} x_{31}	c_{32} x_{32}	c_{33} x_{33}	s_3
poptávka	d_1	d_2	d_3	$d_1 + d_2 + d_3$

3.2 Degenerace

Stejně jako u obecného LP může být báze příпустné řešení dopravního problému *degenerované*.

Definice 9.6: Degenerace v dopravním problému

Báze příпустné řešení dopravního problému nazveme *degenerované*, pokud má

$$\#\{(i, j) \mid x_{ij} > 0\} < m + n - 1.$$

V opačném případě je *nedegenerované*.

Degenerace tedy znamená, že alespoň jedna z $m + n - 1$ báze buněk má nulový

příděl. Na obrázku tabulky taková buňka snadno „zmizí“, bázi však musíme i nadále uchovávat jako strom s přesně $m + n - 1$ hranami. V ručním výpočtu lze nulovou bázeckou buňku označit symbolem 0_B . Symbolické ε je pouze účetní pomůcka a nesmí být zaměněno za skutečně přepravované kladné množství. Při degenerovaném pivotu může vyjít délka kroku $\theta = 0$: změni se báze, nikoli přepravní plán ani hodnota účelové funkce. Pro zajištění konečnosti algoritmu je pak třeba použít anticyklické pravidlo pro volbu vstupující a vystupující buňky, například lexikografické pravidlo.

4 Metoda MODI (metoda potenciálů)

Dopravní problém bychom samozřejmě mohli řešit obecnou simplexovou metodou. Byla by však škoda nevyužít jeho pravidelnou „tabulkovou“ strukturu: specializované algoritmy s ní pracují přímo a jsou výrazně úspornější. Jedním z nich je *metoda MODI* (*Modified Distribution Method*), často nazývaná také *metoda potenciálů* nebo *u-v metoda*. Vychází ze specializovaného síťového simplexu (Bazaraa et al., 2010, Ahuja et al., 1993).

4.1 Počáteční bázecké přípustné řešení

Metoda MODI předpokládá, že na vstupu máme BFS dopravního problému. V praxi se počáteční BFS získá například jednou z následujících jednoduchých konstrukcí:

- *metoda severozápadního rohu* (North–West Corner),
- *metoda nejmenších nákladů* (Least Cost),
- *Vogelova aproximační metoda*.

Nejjednodušší je metoda severozápadního rohu. Její název říká téměř celý postup: začneme v levém horním rohu tabulky a posouváme se doprava a dolů. Nulové nabídky a poptávky můžeme rovnou označit za vyčerpané. V první dosud nevyčerpané buňce v pořadí zleva doprava a shora dolů přiřadíme

$$x_{ij} = \min\{\text{zbývající nabídka řádku } i, \text{zbývající poptávka sloupce } j\}.$$

Je-li tím vyčerpána nabídka řádku, přejdeme o řádek níže. Je-li uspokojena poptávka sloupce, přejdeme o sloupec doprava. Nastane-li obojí současně, posuneme se v obou směrech a později doplníme jednu nulovou bázeckou buňku. V nové buňce opět přiřadíme minimum ze zbývajících nabídky a poptávky a pokračujeme, dokud nejsou všechny sumy splněny. Buňky s kladným přídělem vytvoří les. Pokud jich je méně než $m + n - 1$ – například při současném vyčerpání nabídky a poptávky nebo při nulových nabídkách či poptávkách – doplníme je nulovými bázeckými buňkami tak, aby nevznikl cyklus, dokud nedostaneme kostru s $m + n - 1$ hranami. V úplném bipartitním grafu lze jednotlivé komponenty lesa vždy takto spojit. Výsledkem je báze a BFS vyrovnané úlohy. Jedinou slabinou této velmi snadné konstrukce je, že se při volbě buněk na náklady vůbec nedívá, a proto nemusí dát dobrý počáteční plán.

4.2 Potenciály a redukované náklady

V dopravním problému je vhodné zapisovat bázi jako množinu dvojic indexů, tedy jako množinu buněk tabulky.

Nechť máme bázecké přípustné řešení dopravního problému, tj. množinu bázeckých buněk $B \subseteq \{1, \dots, m\} \times \{1, \dots, n\}$ o velikosti $|B| = m + n - 1$. V degenerovaném případě mohou mít některé bázecké přiděly x_{ij} hodnotu 0.

Metoda MODI zavádí dvě sady pomocných veličin:

$$u_1, \dots, u_m \quad (\text{potenciály řádků}), \quad v_1, \dots, v_n \quad (\text{potenciály sloupců}).$$

Pro každou buňku $(i, j) \in B$ zavedeme potenciály u_i a v_j tak, aby platilo

$$u_i + v_j = c_{ij}. \quad (9.2)$$

Bázecké buňky tvoří strom v bipartitním grafu se $m + n$ vrcholy, a proto má tento strom právě $m + n - 1$ hran. Stejný závěr plyne i z výše rozebrané lineární nezávislosti. Soustava rovnic (9.2) tedy obsahuje $m + n - 1$ rovností pro $m + n$ neznámých $u_1, \dots, u_m, v_1, \dots, v_n$. Potenciály jsou určeny až na transformaci

$$u_i \mapsto u_i + \lambda, \quad v_j \mapsto v_j - \lambda, \quad \lambda \in \mathbb{R},$$

kteřá zachovává všechny součty $u_i + v_j$. Obvykle si zvolíme jeden potenciál, například $u_1 := 0$, a všechny ostatní potenciály poté jednoznačně dopočítáme z rovnic (9.2) podél hran stromu.

Poznámka 9.7: Proč právě rozklad $c_{ij} = u_i + v_j$?

V duální úloze k dopravnímu problému vystupují u_i a v_j jako duální proměnné k rovnicím pro nabídky S_i a poptávky D_j . Každá duální nerovnost má tvar $u_i + v_j \leq c_{ij}$, protože ve sloupci matice omezení pro proměnnou x_{ij} jsou právě dvě jedničky: jedna u řádku S_i a jedna u řádku D_j . Rovnosti $u_i + v_j = c_{ij}$ na zvolené bázi definují simplexové multiplikátory. Pokud vypočtené potenciály splní všechny duální nerovnosti, jsou duálně přípustné a z duality plyne optimalita aktuálního plánu. V optimu jsou rovnosti na bázeckých buňkách také v souladu s podmínkami komplementarity.

Jakmile máme potenciály u_i, v_j , definujeme pro každou (i nebázeckou) buňku *redukováné náklady* neboli *odchylky*

$$\Delta_{ij} := c_{ij} - (u_i + v_j). \quad (9.3)$$

Optimalitní kritérium metody MODI

Uvažujme minimalizační dopravní problém (9.1) a bázecké přípustné řešení s potenciály u_i, v_j a redukovanými náklady Δ_{ij} definovanými v (9.2)–(9.3). Pak platí:

- je-li $\Delta_{ij} \geq 0$ pro všechny (zejména nebázecké) buňky, je dané bázecké řešení *optimální*,
- existuje-li nebázecká buňka s $\Delta_{ij} < 0$, odpovídající směr nezvyšuje náklady. Je-li délka přípustného kroku $\theta > 0$, celkové náklady se přísně sníží. Při degenerovaném kroku může být $\theta = 0$ a změna se pouze báze.

Intuitivně lze Δ_{ij} chápat jako „jednotkový přírůstek nákladů“, kdybychom chtěli do buňky (i, j) přidat malé množství přepravy a ostatní přiděly upravit tak, aby zůstala zachována rovnost řádků a sloupců. Nezáporné odchylky tedy říkají, že žádnou lokální úpravou nelze náklady snížit, což je přesně charakteristika optima.

Poznámka 9.8: Souvislost se simplexovou metodou

V obecném LP vyjadřují redukované náklady, jak se změní hodnota účelové funkce při zavedení nebázické proměnné do báze (viz příslušná kapitola o simplexové metodě). Metoda MODI je v podstatě „simplex na míru“ pro speciální strukturu dopravních tabulek, kdy se potenciály u_i, v_j chovají jako duální proměnné a Δ_{ij} jako redukované náklady.

4.3 Zlepšovací krok a cyklus v tabulce

Předpokládejme, že pro dané báze řešení existuje alespoň jedna buňka (p, q) s $\Delta_{pq} < 0$. Chceme zavést buňku (p, q) do báze, což odpovídá tomu, že x_{pq} zvětšíme z nuly na nějakou kladnou hodnotu. Aby se přitom zachovaly všechny řádkové a sloupcové sumy, musíme ostatní přiděly přenastavit.

Geometricky se v bipartitním grafu po přidání hrany (p, q) vytvoří *jediný cyklus*. V tabulce se projeví jako uzavřená lomená čára se sudým počtem buněk, v níž se střídají vodorovné a svislé kroky. Všechny buňky cyklu kromě vstupující buňky jsou báze. Při degeneraci některé z nich mohou mít nulový přiděl.

Postup je následující:

1. Vybereme nebázickou buňku (p, q) se zápornou odchylkou $\Delta_{pq} < 0$ (například s nejmenší hodnotou) a dočasně ji přidáme ke stromu báze buněk.
2. V tabulce najdeme uzavřený cyklus procházející buňkou (p, q) tak, že se střídají horizontální a vertikální kroky a v každém „zlomu“ leží báze buňka.
3. Buňky v cyklu střídavě označíme znaménky $+$ a $-$, počínaje $+$ v buňce (p, q) .
4. Spočítáme

$$\theta := \min\{x_{ij} \mid (i, j) \text{ leží na cyklu se znaménkem } -\},$$

tj. nejmenší přiděl v báze buňkách označených $-$. Může vyjít $\theta = 0$.

5. Všem buňkám se znaménkem $+$ zvětšíme přiděl o θ , všem buňkám se znaménkem $-$ přiděl o θ zmenšíme. Alespoň jedna z „minusových“ buněk má po změně nulový přiděl a opustí bázi. Při shodě minim vybereme právě jednu, ostatní nulové buňky zůstanou báze. Buňka (p, q) se stane báze s přidělem θ .

Podél tohoto cyklu se hodnota účelové funkce změní přesně o $\theta\Delta_{pq}$. Pro $\Delta_{pq} < 0$ a $\theta > 0$ tedy skutečně klesne. Při $\theta = 0$ zůstane stejná.

Příklad 9.9: Metoda MODI na konkrétním dopravním problému

Uvažujme dopravní problém se třemi zdroji S_1, S_2, S_3 a třemi odběrateli D_1, D_2, D_3 . Nabídky, poptávky i jednotkové náklady c_{ij} jsou dány

$$(c_{ij}) = \begin{pmatrix} 6 & 8 & 10 \\ 7 & 11 & 11 \\ 4 & 5 & 12 \end{pmatrix}, \quad (s_1, s_2, s_3) = (15, 17, 18), \quad (d_1, d_2, d_3) = (20, 10, 20).$$

Hledáme plán přeprav, který minimalizuje celkové náklady.

Předpokládejme, že některou z konstrukčních heuristik (např. metodou nejmenších nákladů) jsme již získali počáteční bázecké přípustné řešení. Tomu odpovídají příděly

$$x_{11} = 2, \quad x_{12} = 10, \quad x_{13} = 3, \quad x_{23} = 17, \quad x_{31} = 18,$$

ostatní x_{ij} jsou nulové. Takové řešení je zachyceno v následující tabulce. V horní části buněk jsou uvedeny náklady c_{ij} , vlevo dole přidělená množství x_{ij} .

	D_1	D_2	D_3	nabídka	u_i
S_1	6 2	8 10	10 3	15	$u_1 = 0$
S_2	7	11	11 17	17	$u_2 = 1$
S_3	4 18	5	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 8$	$v_3 = 10$		$z = 381$

Bázecké buňky jsou

$$B = \{(1, 1), (1, 2), (1, 3), (2, 3), (3, 1)\}.$$

Pro každou bázeckou buňku platí vztah pro potenciály

$$u_i + v_j = c_{ij}, \quad (i, j) \in B.$$

Dostáváme soustavu pěti rovnic

$$\begin{aligned} u_1 + v_1 &= 6, \\ u_1 + v_2 &= 8, \\ u_1 + v_3 &= 10, \\ u_2 + v_3 &= 11, \\ u_3 + v_1 &= 4. \end{aligned}$$

Protože potenciály jsou určeny pouze až na transformaci $u_i \mapsto u_i + \lambda$, $v_j \mapsto v_j - \lambda$, můžeme si zafixovat $u_1 := 0$. Potom ze soustavy postupně dopočítáme

$$v_1 = 6, \quad v_2 = 8, \quad v_3 = 10, \quad u_2 = 1, \quad u_3 = -2,$$

což je právě zaneseno v tabulce výše.

Pro nebázické buňky spočítáme redukované náklady $\Delta_{ij} = c_{ij} - (u_i + v_j)$:

$$\begin{aligned}\Delta_{21} &= c_{21} - (u_2 + v_1) = 7 - (1 + 6) = 0, \\ \Delta_{22} &= c_{22} - (u_2 + v_2) = 11 - (1 + 8) = 2, \\ \Delta_{32} &= c_{32} - (u_3 + v_2) = 5 - ((-2) + 8) = -1, \\ \Delta_{33} &= c_{33} - (u_3 + v_3) = 12 - ((-2) + 10) = 4.\end{aligned}$$

Jediná záporná odchylka je $\Delta_{32} = -1$, takže do báze se pokusíme zavést buňku $(3, 2)$.

Zlepšovací cyklus. V tabulce vyhledáme cyklus procházející buňkou $(3, 2)$, ve kterém se střídají bázecké buňky v řádcích a sloupcích. Tento cyklus je

$$(3, 2) (+) \rightarrow (3, 1) (-) \rightarrow (1, 1) (+) \rightarrow (1, 2) (-) \rightarrow (3, 2),$$

kde v závorce je vždy znaménko, kterým je buňka označena. Označme $\theta > 0$ množství, o které zvýšíme příděl v buňce $(3, 2)$. Potom musíme o θ snížit příděl v buňkách se znaménkem $-$ a o θ zvýšit v buňkách se znaménkem $+$. Rozpis přídělů v závislosti na θ je v následující tabulce.

	D_1	D_2	D_3	nabídka	u_i
S_1	6 $2 + \theta$	8 $10 - \theta$	10 3	15	$u_1 = 0$
S_2	7	11	11 17	17	$u_2 = 1$
S_3	4 $18 - \theta$	5 θ	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 8$	$v_3 = 10$		$z = 381 - \theta$

Aby byly všechny příděly nezáporné, musí platit

$$10 - \theta \geq 0, \quad 18 - \theta \geq 0,$$

tj. $\theta \leq 10$. Protože $\Delta_{32} = -1 < 0$, zvolíme největší možnou hodnotu, tedy $\theta^* = 10$, čímž dosáhneme největšího poklesu účelové funkce. Po dosazení $\theta = 10$ dostáváme nový bázecký plán

	D_1	D_2	D_3	nabídka	u_i
S_1	6 12	8	10 3	15	$u_1 = 0$
S_2	7	11	11 17	17	$u_2 = 1$
S_3	4 8	5 10	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 7$	$v_3 = 10$		$z = 371$

Z nových bázeckých buněk

$$B' = \{(1, 1), (1, 3), (2, 3), (3, 1), (3, 2)\}$$

opět dopočítáme potenciály. Soustava rovnic pro u_i, v_j je nyní

$$\begin{aligned} u_1 + v_1 &= 6, \\ u_1 + v_3 &= 10, \\ u_2 + v_3 &= 11, \\ u_3 + v_1 &= 4, \\ u_3 + v_2 &= 5. \end{aligned}$$

Po položení $u_1 := 0$ dostaneme

$$v_1 = 6, \quad v_3 = 10, \quad u_2 = 1, \quad u_3 = -2, \quad v_2 = 7,$$

což je opět vyznačeno v tabulce. Redukované náklady pro nebázecké buňky jsou

$$\begin{aligned} \Delta_{12} &= c_{12} - (u_1 + v_2) = 8 - (0 + 7) = 1, \\ \Delta_{21} &= c_{21} - (u_2 + v_1) = 7 - (1 + 6) = 0, \\ \Delta_{22} &= c_{22} - (u_2 + v_2) = 11 - (1 + 7) = 3, \\ \Delta_{33} &= c_{33} - (u_3 + v_3) = 12 - ((-2) + 10) = 4. \end{aligned}$$

Všechny odchylky jsou nezáporné, takže nalezený bázecký plán je optimální. Zároveň však $\Delta_{21} = 0$, tedy existuje alespoň jedno další optimální bázecké řešení.

Alternativní optimální řešení. Zkusme proto zavést do báze buňku $(2, 1)$ a zkonstruovat nový cyklus

$$(2, 1) (+) \rightarrow (2, 3) (-) \rightarrow (1, 3) (+) \rightarrow (1, 1) (-) \rightarrow (2, 1).$$

Označíme-li opět změnu po cyklu symbolem θ , pak příděly mají tvar

$$x_{11} = 12 - \theta, \quad x_{13} = 3 + \theta, \quad x_{21} = \theta, \quad x_{23} = 17 - \theta,$$

ostatní bázecké buňky $(3, 1)$ a $(3, 2)$ zůstávají beze změny. Tomu odpovídá tabulka

	D_1	D_2	D_3	nabídka	u_i
S_1	6 $12 - \theta$	8	10 $3 + \theta$	15	$u_1 = 0$
S_2	7 θ	11	11 $17 - \theta$	17	$u_2 = 1$
S_3	4 8	5 10	12	18	$u_3 = -2$
poptávka	20	10	20	50	
v_j	$v_1 = 6$	$v_2 = 7$	$v_3 = 10$		$z = 371$

Z podmínek nezápornosti přídělů plyne

$$12 - \theta \geq 0, \quad 17 - \theta \geq 0,$$

tedy $0 \leq \theta \leq 12$. Protože $\Delta_{21} = 0$, je hodnota účelové funkce $z = 371$ pro všechna θ v tomto intervalu stejná. Například pro $\theta = 12$ dostaneme alternativní optimální plán

$$x_{11} = 0, \quad x_{13} = 15, \quad x_{21} = 12, \quad x_{23} = 5, \quad x_{31} = 8, \quad x_{32} = 10,$$

který splňuje všechna omezení a má stejné minimální náklady jako předchozí řešení.

4.4 Shrnutí metody MODI

Metodu MODI můžeme shrnout v následujícím přehledu:

Metoda MODI – přehled kroků
1. Najdi nějaké bázecké přípustné řešení (např. metodou severozápadního rohu, metodou nejmenších nákladů apod.). 2. Spočítej potenciály u_i, v_j tak, aby pro všechny bázecké buňky (i, j) platilo $u_i + v_j = c_{ij}$ (po zafixování jednoho potenciálu, např. $u_1 = 0$). 3. Pro každou buňku spočítej redukované náklady $\Delta_{ij} = c_{ij} - (u_i + v_j)$.

4. Pokud $\Delta_{ij} \geq 0$ pro všechna (i, j) , je aktuální bázecké řešení optimální a algoritmus končí.
5. Jinak vyber nebázeckou buňku (p, q) s $\Delta_{pq} < 0$ (typicky s nejmenší hodnotou), sestroj příslušný cyklus v tabulce a proved' zlepšovací krok popsany výše. Při shodě minim zvol jednu vystupující buňku podle anticyklického pravidla, například lexikografického. Tím získáš novou bázi a bázecké přípustné řešení. Při $\theta = 0$ může přepravní plán zůstat stejný.
6. Vrať se na krok 2.

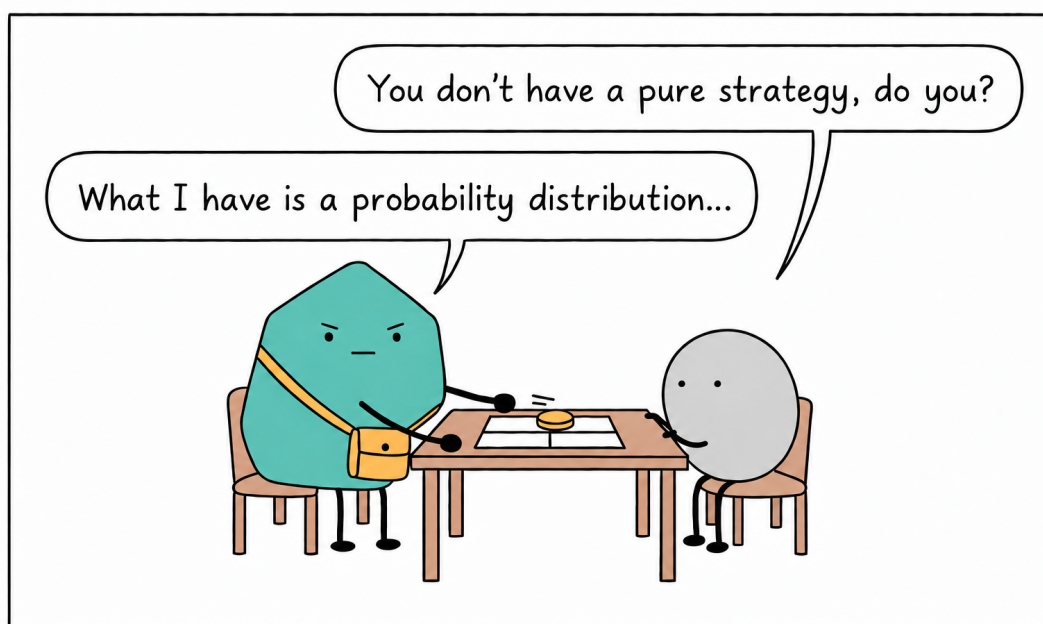
Metoda MODI je v praxi velmi oblíbená: využívá speciální strukturu dopravního problému, je relativně snadno implementovatelná a pro úlohy s několika desítkami až stovkami řádků a sloupců bývá velmi efektivní. Z teoretického hlediska však nejde o zásadně nový algoritmus – lze ji chápat jako specializovanou variantu simplexové metody, která pracuje přímo s tabulkou přidělů a s potenciály odpovídajícími duálními proměnným.

KAPITOLA 10

MATICOVÉ HRY S NULOVÝM SOUČTEM A VĚTA O MINIMAXU

1	Maticová hra s nulovým součtem	100
2	Smíšené strategie a očekávaná výplata	103
3	Max–min, min–max a hodnota hry	104
4	Formulace maticové hry jako lineárního programu . .	107
5	Krátký ilustrační příklad . .	110

V této kapitole se omezíme na speciální, ale velmi důležitou třídu tzv. dvouhráčových her: *hry s nulovým součtem*, které lze zapsat pomocí jediné reálné matice. Ukážeme, že hledání jejich optimálních smíšených strategií lze převést na řešení dvojice navzájem duálních lineárních programů. To poskytuje jak efektivní algoritmus, tak i velmi čisté propojení teorie her a teorie lineárního programování.



1 Maticová hra s nulovým součtem

V této kapitole chceme ukázat, jak se lineární programování přirozeně objevuje v jednoduchém modelu strategické interakce dvou hráčů. Představme si situaci typu „já proti tobě“: dva hráči si současně zvolí nějakou akci a podle kombinace těchto akcí jeden získá určitý počet bodů (nebo např. peněz, interpretace je na nás) a druhý přesně o stejný počet přijde.

- Pokud známe všechny možné akce obou hráčů a pro každou dvojici akcí umíme říct, kdo kolik „vyhraje“ nebo „prohraje“, mluvíme o *hře v normální (strategické) formě*.
- Pokud navíc platí, že zisk jednoho je přesně ztráta druhého, jde o *hru s nulovým součtem*.
- Když jsou obě množiny akcí konečné, můžeme celou hru zapsat jednou *maticí*, proto často říkáme *maticová hra*.

Na následujícím příkladu naznačíme, co si pod maticovou hrou s nulovým součtem lze představit.

Příklad 10.1: Pašerák a celní kontrola

Pašerák se snaží převézt nelegální zboží z Rakouska do ČR. Má na výběr dvě trasy:

- **Trasa 1** – vedlejší silnice přes málo frekventovaný přechod,
- **Trasa 2** – dálnice, kde by v případě úspěchu mohl převézt mnohem více zboží.

Celní správa má jen jednu mobilní hlídku, kterou každý den pošle buď na vedlejší silnici, nebo na dálnici. Hlídka neví, kterou trasu si pašerák ten den zvolí, a pašerák neví, kde přesně bude hlídka. Rozhodují se tedy *souběžně*, bez znalosti tahu toho druhého.

Označme:

- řádkový hráč = celní správa,
- sloupcový hráč = pašerák,
- řádky: R_1 = hlídka na vedlejší silnici, R_2 = hlídka na dálnici,
- sloupce: C_1 = pašerák jede vedlejší silnicí, C_2 = pašerák jede dálnicí.

Následující tabulka popisuje *zisk pašeráka* (například v desítkách tisíc korun):

	C_1 (vedlejší)	C_2 (dálnice)
R_1 (hlídka na vedlejší)	0	6
R_2 (hlídka na dálnici)	2	0

Číslo a_{ij} v tabulce je zisk pašeráka, pokud celní správa zvolí řádek R_i a pašerák sloupec C_j . Hra je *s nulovým součtem*: zisk pašeráka je zároveň ztráta celní správy, takže užitek celní správy je $-a_{ij}$.

Jak by se měli oba hráči chovat?

- Kdyby celní správa *vždy* hlídala jen jednu trasu, pašerák by se okamžitě přizpůsobil a jezdil by tou druhou.
- Kdyby pašerák *vždy* jezdil jen po dálnici, celní správa by měla jasno a posílala by hlídku na dálnici.

Je tedy rozumné, aby oba hráči nějakým způsobem *náhodně střídali* své možnosti (volili takzvané *smíšené strategie*), aby protistraně „znepříjemnili“ život. V této konkrétní hře se ukáže, že existuje rovnováha, ve které

- pašerák volí vedlejší silnici C_1 s pravděpodobností $\frac{3}{4}$ a dálnici C_2 s pravděpodobností $\frac{1}{4}$,
- celní správa posílá hlídku na vedlejší silnici R_1 s pravděpodobností $\frac{1}{4}$ a na dálnici R_2 s pravděpodobností $\frac{3}{4}$.

Pro uvedenou smíšenou strategii celní správy má každá čistá trasa pašeráka očekávanou výplatu $3/2$. Obdobně proti uvedené strategii pašeráka vede každé čisté rozmístění hlídky ke stejné očekávané platbě $3/2$. Ani jeden z hráčů si proto nemůže jednostrannou změnou své strategie polepšit. Takto vyvážené řešení plyne z věty o minimaxu. V další části kapitoly uvidíme, že právě hledání takových optimálních pravděpodobností lze elegantně přepsat jako úlohu lineárního programování.

Z pohledu lineárního programování je na těchto hrách zajímavé, že:

1. nejlepší (optimální) chování hráčů lze popsat jako řešení určitých optimalizačních úloh,
2. tyto úlohy lze přepsat jako lineární programy a jejich duály, což odpovídá větě o minimaxu.

Než ukážeme, že „hledání rozumné strategie ve hře“ je v tomto speciálním případě totéž jako řešení LP, tak formálně popíšeme, co přesně maticová hra s nulovým součtem je.

Definice 10.2: Hra v normální formě

(Konečná) hra v normální formě je trojice

$$G = (P, \mathcal{A}, \mathbf{u}),$$

kde $P = \{1, \dots, N\}$ pro nějaké $N \geq 1$ a platí:

- P je konečná množina hráčů,
- $\mathcal{A} = \mathcal{A}_1 \times \dots \times \mathcal{A}_N$ je množina všech *profilů akcí*, kde $\mathcal{A}_i$ je neprázdná konečná množina akcí hráče i ,
- $\mathbf{u} = (u_1, \dots, u_N)$ je N -tice *užitkových (výplatních) funkcí*, $u_i : \mathcal{A} \rightarrow \mathbb{R}$, kde $u_i(\mathbf{a})$ je užitek hráče i při profilu akcí $\mathbf{a} \in \mathcal{A}$.

V dalším se omezíme na hry se dvěma hráči. Pro přehlednost je budeme nazývat *řádkový hráč* a *sloupcový hráč* a jejich akce si očíslováme:

$$R = \{1, \dots, m\}, \quad C = \{1, \dots, n\}.$$

Profil akcí je tedy dvojice (i, j) , kde i je zvolený řádek a j zvolený sloupec.

Definice 10.3: Hra s nulovým součtem v maticové formě

Dvouhráčová hra v normální formě

$$G = (\{\text{řádkový hráč, sloupcový hráč}\}, R \times C, \mathbf{u})$$

je hra s nulovým součtem, pokud pro každý profil akcí $(i, j) \in R \times C$ platí

$$u_{\text{řádkový}}(i, j) + u_{\text{sloupcový}}(i, j) = 0.$$

Protože nás zajímá jen relativní zisk mezi hráči, můžeme celou hru popsat jedinou *výplatní maticí* $A \in \mathbb{R}^{m \times n}$. Každý prvek

$$a_{ij} = u_{\text{sloupcový}}(i, j)$$

chápeme jako *platbu od řádkového hráče sloupcovému hráči*. Kladná hodnota a_{ij} znamená, že sloupcový hráč vyhrává (řádkový platí jemu), záporná hodnota naopak.

Z podmínky nulového součtu okamžitě plyne, že užitek řádkového hráče je vždy $-a_{ij}$. Celou hru tedy reprezentuje jediná matice A , která popisuje „kolik zaplatí řádkový hráč sloupcovému“ v každé kombinaci voleb.

Poznámka 10.4: Maticová hra a značení

V dalším budeme používat označení:

- *řádky* $1, \dots, m$ jsou akce řádkového hráče,
- *sloupce* $1, \dots, n$ jsou akce sloupcového hráče,
- matice A je *výplatní matice* sloupcového hráče.

Záměrně tedy používáme A jako výplatní matici sloupcového hráče, který svou výplatu maximalizuje. V části literatury značí A naopak výplatní matici řádkového hráče. Při zachování stejných řádkových a sloupcových akcí se mezi oběma konvencemi přechází náhradou $A_{\text{zde}} = -A_{\text{řádkový}}$. Současně se proto obrátí směr příslušných optimalizačních nerovností. Tomuto nastavení se v literatuře říká *maticová hra* (*matrix game*) a je to standardní model dvouhráčové hry s nulovým součtem.

V tuto chvíli umíme popsat, co se stane, pokud si každý hráč vybere konkrétní řádek/sloupec (tzv. *čistá strategie*, tedy volba jedné akce z konečné množiny). Z pohledu strategie ale často dává smysl volit akce náhodně, například ve hře kámen–nůžky–papír: kdyby jeden hráč opakoval stále stejný tah, druhý ho snadno „přečte“. K tomu potřebujeme pojem *smíšené strategie*.

2 Smíšené strategie a očekávaná výplata

Intuitivní představa smíšené strategie je jednoduchá: hráč si nepředepisuje „vždy hraj řádek 3“, ale „řádek 1 hraj s pravděpodobností 40 %, řádek 2 s pravděpodobností 30 %, řádek 3 s pravděpodobností 30 %“. Tím vnáší do hry náhodu a brání protistraně v předvídání.

Formálně smíšená strategie je právě *pravděpodobnostní rozdělení* na množině čistých strategií.

Definice 10.5: Smíšená strategie a pravděpodobnostní simplex

Vektor $\mathbf{x} \in \mathbb{R}^n$ nazveme *stochastický* (též *pravděpodobnostní vektor*), pokud

$$x_j \geq 0 \quad \text{pro všechna } j \quad \text{a} \quad \sum_{j=1}^n x_j = 1.$$

Množinu všech takových vektorů délky n označíme

$$S_n = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} \geq \mathbf{0}, \sum_{j=1}^n x_j = 1\}.$$

- *Smíšená strategie sloupcového hráče* je vektor $\mathbf{x} \in S_n$, kde x_j je pravděpodobnost, že zvolí sloupec j .
- *Smíšená strategie řádkového hráče* je vektor $\mathbf{y} \in S_m$, kde y_i je pravděpodobnost, že zvolí řádek i .

Poznámka 10.6: Pravděpodobnostní simplex je skutečně simplex

Množinu S_n lze chápat jako konvexní obal vektorů kanonické báze $\mathbf{e}_1, \dots, \mathbf{e}_n \in \mathbb{R}^n$. Každý stochastický vektor $\mathbf{x} \in S_n$ totiž splňuje

$$\mathbf{x} = x_1\mathbf{e}_1 + x_2\mathbf{e}_2 + \dots + x_n\mathbf{e}_n,$$

kde součinitelé x_j jsou nezáporné a sečtou se na 1. To znamená, že S_n je přesně množina všech konvexních kombinací vrcholů $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$, tedy $(n - 1)$ -rozměrný *simplex*.

Čistá strategie je zvláštní případ smíšené strategie. Jde o vektor, který má v jedné souřadnici hodnotu 1 a ve všech ostatních 0. Smíšené strategie nám tedy jen „rozšíří“ prostor možných strategií o konvexní kombinace čistých.

2.1 Očekávaná výplata

Značení 10.7: Pravděpodobnost a střední hodnota

Budeme pracovat s prvky elementární pravděpodobnosti.

- Symbolem $\mathbb{P}[\cdot]$ budeme označovat *pravděpodobnost* daného jevu.
- Symbolem $\mathbb{E}[\cdot]$ budeme označovat *střední hodnotu* (očekávanou hodnotu) náhodné veličiny.

Předpokládáme, že oba hráči používají vlastní nezávislé zdroje náhody a před volbou akce nepozorují výsledek náhodné volby soupeře. Pokud řádkový hráč hraje smíšenou strategii $\mathbf{y} \in S_m$ a sloupcový hráč $\mathbf{x} \in S_n$, pak pravděpodobnost, že nastane konkrétní profil akcí (i, j) , je

$$\mathbb{P}[(i, j)] = \mathbb{P}[\text{řádek } i] \cdot \mathbb{P}[\text{sloupec } j] = y_i x_j.$$

Výplata sloupcového hráče při profilu (i, j) je a_{ij} . Očekávanou výplatu při opakovaném použití stejných strategií $\mathbf{x}, \mathbf{y}$ získáme jako součet výplat všech možných profilů vážených jejich pravděpodobnostmi:

$$\mathbb{E}[\text{výplata sloupcového hráče}] = \sum_{i=1}^m \sum_{j=1}^n \underbrace{\mathbb{P}[(i, j)]}_{y_i x_j} a_{ij} = \sum_{i=1}^m \sum_{j=1}^n y_i a_{ij} x_j.$$

Tento dvojitý součet lze zapsat elegantně maticově. Vektor $\mathbf{y}$ chápeme jako sloupcový vektor délky m , $\mathbf{x}$ jako sloupcový vektor délky n , potom

$$\mathbb{E}[\text{výplata sloupcového hráče}] = \mathbf{y}^\top \mathbf{A} \mathbf{x}.$$

Protože hra má nulový součet, očekávaná výplata řádkového hráče je

$$\mathbb{E}[\text{výplata řádkového hráče}] = -\mathbf{y}^\top \mathbf{A} \mathbf{x}.$$

Poznámka 10.8: Zisk vs. ztráta

Z pohledu sloupcového hráče je číslo $\mathbf{y}^\top \mathbf{A} \mathbf{x}$ jeho *očekávaný zisk*, zatímco z pohledu řádkového hráče je to *očekávaná ztráta*. Je tedy přirozené, že:

- sloupcový hráč se snaží $\mathbf{y}^\top \mathbf{A} \mathbf{x}$ *maximalizovat*,
- řádkový hráč se snaží $\mathbf{y}^\top \mathbf{A} \mathbf{x}$ *minimalizovat*.

Podstata maticové hry s nulovým součtem je právě v tomto konfliktu cílů obou hráčů. Jedno a totéž číslo je kritériem pro oba hráče, ale jeden ho chce co největší a druhý co nejmenší.

3 Max–min, min–max a hodnota hry

Pro danou strategii jednoho hráče se druhý hráč bude chovat tak, aby si maximalizoval (nebo minimalizoval) svůj očekávaný užitek. To vede na pojmy *max–min*, *min–max* a *hodnota hry*.

3.1 Nejhorší případ pro danou strategii

Fixujme smíšenou strategii sloupcového hráče $\mathbf{x} \in S_n$. Pro každou možnou strategii řádkového hráče $\mathbf{y} \in S_m$ je očekávaná výplata sloupcového hráče rovna $\mathbf{y}^\top \mathbf{A} \mathbf{x}$.

Dívejme se na situaci z pohledu sloupcového hráče. Chceme-li pro strategii $\mathbf{x}$ posoudit *nejhorší možný scénář*, kdy se řádkový hráč zachová co nejvíce proti sloupcovému hráči, díváme se na minimum této hodnoty přes všechny $\mathbf{y} \in S_m$:

$$\min_{\mathbf{y} \in S_m} \mathbf{y}^\top \mathbf{A} \mathbf{x}.$$

To motivuje následující definici.

Definice 10.9: Funkce β a α

Pro strategii sloupcového hráče $\mathbf{x} \in S_n$ definujeme

$$\beta(\mathbf{x}) := \min_{\mathbf{y} \in S_m} \mathbf{y}^\top \mathbf{A} \mathbf{x},$$

tj. nejhorší možná (nejmenší) očekávaná výplata sloupcového hráče, pokud hraje $\mathbf{x}$ a řádkový hráč reaguje optimálně.

Analogicky pro strategii řádkového hráče $\mathbf{y} \in S_m$ definujeme

$$\alpha(\mathbf{y}) := \max_{\mathbf{x} \in S_n} \mathbf{y}^\top \mathbf{A} \mathbf{x},$$

tj. nejhorší možnou (největší) očekávanou ztrátu řádkového hráče, pokud hraje $\mathbf{y}$ a sloupcový hráč reaguje optimálně.

Minimum ani maximum nám zde nemůže „utéci“: množiny S_m a S_n jsou neprázdné kompaktní simplexy a očekávaná výplata je spojitá. Minima a maxima v těchto definicích jsou proto vždy dosažena.

Z definice okamžitě plyne

$$\beta(\mathbf{x}) \leq \mathbf{y}^\top \mathbf{A} \mathbf{x} \leq \alpha(\mathbf{y}) \quad \text{pro všechna } \mathbf{x} \in S_n, \mathbf{y} \in S_m,$$

protože pro daný pár $(\mathbf{x}, \mathbf{y})$ nemůže hodnota $\mathbf{y}^\top \mathbf{A} \mathbf{x}$ být menší než minimum přes všechna $\mathbf{y}$ ani větší než maximum přes všechna $\mathbf{x}$.

3.2 Maximinové a minimaxové strategie, hodnota hry

Na základě funkcí $\beta(\mathbf{x})$ a $\alpha(\mathbf{y})$ nyní můžeme přesně formulovat, co znamená, že je určitá smíšená strategie „optimální v nejhorším případě“.

Definice 10.10: Maximinová strategie, minimaxová strategie a hodnota hry

Smíšenou strategii $\mathbf{x}^* \in S_n$ nazveme *maximinovou strategií sloupcového hráče*, pokud maximalizuje jeho nejhorší možnou výplatu, tj.

$$\beta(\mathbf{x}^*) = \min_{\mathbf{y} \in S_m} \mathbf{y}^\top \mathbf{A} \mathbf{x}^* = \max_{\mathbf{x} \in S_n} \beta(\mathbf{x}).$$

Analogicky smíšenou strategii $\mathbf{y}^* \in S_m$ nazveme *minimaxovou strategií řádkového hráče*, pokud minimalizuje jeho nejhorší možnou ztrátu:

$$\alpha(\mathbf{y}^*) = \max_{\mathbf{x} \in S_n} (\mathbf{y}^*)^\top A \mathbf{x} = \min_{\mathbf{y} \in S_m} \alpha(\mathbf{y}).$$

Číslo $v \in \mathbb{R}$ nazveme *hodnota hry*, pokud existují strategie $\mathbf{x}^*, \mathbf{y}^*$ takové, že

$$\beta(\mathbf{x}^*) = v = \alpha(\mathbf{y}^*).$$

Intuitivně:

- sloupcový hráč si volbou $\mathbf{x}^*$ zaručí, že ani při nejhorší možné odpovědi protivníka neklesne jeho očekávaná výplata pod v ,
- řádkový hráč si volbou $\mathbf{y}^*$ zaručí, že ani při nejhorším možném postupu soupeře nepřesáhne jeho ztráta v .

Z již zavedených definic plyne pro libovolnou hru vždy

$$\max_{\mathbf{x} \in S_n} \beta(\mathbf{x}) \leq \min_{\mathbf{y} \in S_m} \alpha(\mathbf{y}).$$

V maticových hrách s nulovým součtem však nastává překvapivě silná vlastnost: *obě hodnoty jsou ve skutečnosti stejné*. Formuluje to následující věta o minimaxu, jejíž klasický důkaz pochází od Johna von Neumanna (von Neumann, 1928). Moderní výklad viz také (Osborne and Rubinstein, 1994).

Věta 10.11: Věta o minimaxu pro maticové hry s nulovým součtem

Nechť $A \in \mathbb{R}^{m \times n}$ je výplatní matice dvouhráčové hry s nulovým součtem. Potom platí

$$\max_{\mathbf{x} \in S_n} \beta(\mathbf{x}) = \min_{\mathbf{y} \in S_m} \alpha(\mathbf{y}),$$

tedy existuje číslo $v \in \mathbb{R}$ takové, že

$$\max_{\mathbf{x} \in S_n} \min_{\mathbf{y} \in S_m} \mathbf{y}^\top A \mathbf{x} = v = \min_{\mathbf{y} \in S_m} \max_{\mathbf{x} \in S_n} \mathbf{y}^\top A \mathbf{x}.$$

Číslo v nazýváme *hodnota hry*. Navíc existují smíšené strategie $\mathbf{x}^* \in S_n$ a $\mathbf{y}^* \in S_m$ splňující

$$\beta(\mathbf{x}^*) = v = \alpha(\mathbf{y}^*),$$

tj. $\mathbf{x}^*$ je maximinová strategie sloupcového hráče a $\mathbf{y}^*$ je minimaxová strategie řádkového hráče.

Číslo v je tedy hodnota hry a dvojice $(\mathbf{x}^*, \mathbf{y}^*)$ tvoří optimální smíšené strategie obou hráčů. Každý z nich si svou strategií hlídá právě tutéž hodnotu – jeden zdola a druhý shora.

Poznámka 10.12: Co přesně znamená rovnováha?

Význam optimálních strategií lze zapsat sedlovými nerovnostmi

$$(\mathbf{y}^*)^T \mathbf{A} \mathbf{x} \leq (\mathbf{y}^*)^T \mathbf{A} \mathbf{x}^* \leq \mathbf{y}^T \mathbf{A} \mathbf{x}^* \quad \text{pro všechna } \mathbf{x} \in S_n, \mathbf{y} \in S_m. \quad (10.1)$$

Levá nerovnost říká, že si sloupcový hráč jednostrannou změnou proti $\mathbf{y}^*$ nepolepší. Pravá říká, že řádkový hráč proti $\mathbf{x}^*$ jednostrannou změnou nesníží svou ztrátu. V řeči teorie her je tedy dvojice $(\mathbf{x}^*, \mathbf{y}^*)$ Nashovou rovnováhou. V řeči optimalizace jde o sedlový bod očekávané výplaty. Jsou-li strategie $\mathbf{x}^*$ a $\mathbf{y}^*$ čisté, najdeme tento sedlový bod přímo ve výplatní matici.

Tato vlastnost her s nulovým součtem plyne přímo z duality lineárního programování, jak nyní ukážeme.

4 Formulace maticové hry jako lineárního programu

Nyní ukážeme, že problém sloupcového hráče „zvolit strategii $\mathbf{x}$, která maximalizuje jeho nejhorší možnou výplatu“, lze přepsat jako lineární program. Analogicky problém řádkového hráče vede na duální lineární program.

4.1 LP pro sloupcového hráče (max–min)

Připomeňme, že sloupcový hráč chce maximalizovat funkci $\beta(\mathbf{x})$:

$$\max_{\mathbf{x} \in S_n} \beta(\mathbf{x}) = \max_{\mathbf{x} \in S_n} \min_{\mathbf{y} \in S_m} \mathbf{y}^T \mathbf{A} \mathbf{x}.$$

Pro fixovanou strategii $\mathbf{x}$ závisí výraz $\mathbf{y}^T \mathbf{A} \mathbf{x}$ na $\mathbf{y}$ jen lineárně. Minimum přes simplex S_m je proto dosaženo v některém jeho vrcholu, tj. v některé čisté strategii řádkového hráče. Označíme $\mathbf{e}_i$ i -tý vektor kanonické báze v $\mathbb{R}^m$. Vzpomeňme, že čisté strategie odpovídají právě těmto vektorům. Pak dostáváme

$$\min_{\mathbf{y} \in S_m} \mathbf{y}^T \mathbf{A} \mathbf{x} = \min_{i=1, \dots, m} \mathbf{e}_i^T \mathbf{A} \mathbf{x}.$$

Sloupcový hráč tedy řeší max–min problém

$$\max_{\mathbf{x} \in S_n} \min_{i=1, \dots, m} \mathbf{e}_i^T \mathbf{A} \mathbf{x}.$$

Zavedme proměnnou $t \in \mathbb{R}$, která bude dolní mezí všech hodnot $\mathbf{e}_i^T \mathbf{A} \mathbf{x}$. Podmínky

$$t \leq \mathbf{e}_i^T \mathbf{A} \mathbf{x} \quad \text{pro } i = 1, \dots, m$$

zaručují, že $t \leq \min_i \mathbf{e}_i^T \mathbf{A} \mathbf{x}$, a maximalizace t tedy skutečně maximalizuje nejhorší možnou výplatu sloupcového hráče.

Aby byl vektor $\mathbf{x}$ přípustnou smíšenou strategií, musí splňovat:

$$x_j \geq 0 \quad \text{pro všechna } j, \quad \sum_{j=1}^n x_j = 1.$$

Dostáváme tedy lineární program pro sloupcového hráče:

LP pro sloupcového hráče		
maximalizuj	t	
při podmínkách	$t \leq (\mathbf{Ax})_i = \mathbf{e}_i^T \mathbf{Ax}, \quad i = 1, \dots, m,$ $\sum_{j=1}^n x_j = 1,$ $x_j \geq 0 \quad \text{pro všechna } j.$	(LP _C)

Každé omezení $t \leq (\mathbf{Ax})_i$ říká: „Pokud řádkový hráč zvolí čistou strategii i , očekávaná výplata sloupcového hráče je alespoň t .“ Maximalizace t se tedy snaží, aby i při nejhorší odpovědi řádkového hráče byla výplata co největší.

4.2 LP pro řádkového hráče (min–max)

Zcela symetricky lze formulovat problém řádkového hráče:

$$\min_{\mathbf{y} \in S_m} \max_{\mathbf{x} \in S_n} \mathbf{y}^T \mathbf{Ax}.$$

Podobně jako dříve pro fixované $\mathbf{y}$ platí

$$\max_{\mathbf{x} \in S_n} \mathbf{y}^T \mathbf{Ax} = \max_{j=1, \dots, n} (A^T \mathbf{y})_j,$$

protože optimum lineární funkce v simplexu nastává v některém jeho vrcholu.

Zavedeme pomocnou proměnnou $u \in \mathbb{R}$, která bude horní mezí hodnot $(A^T \mathbf{y})_j$. Podmínky

$$(A^T \mathbf{y})_j \leq u \quad \text{pro } j = 1, \dots, n$$

zajišťují, že $u \geq \max_j (A^T \mathbf{y})_j$. Minimalizace u tedy minimalizuje nejhorší možnou výplatu sloupcového hráče, což je z pohledu řádkového hráče nejhorší možná ztráta.

Smíšená strategie řádkového hráče musí opět splňovat

$$y_i \geq 0 \quad \text{pro všechna } i, \quad \sum_{i=1}^m y_i = 1.$$

Lineární program pro řádkového hráče tedy je:

LP pro řádkového hráče		
minimalizuj	u	
při podmínkách	$(A^T \mathbf{y})_j \leq u, \quad j = 1, \dots, n,$ $\sum_{i=1}^m y_i = 1,$ $y_i \geq 0 \quad \text{pro všechna } i.$	(LP _R)

4.3 Dualita: každý hráč má svou úlohu

Primární úloha v maticovém tvaru

Máme k dispozici LP pro sloupcového hráče (LP_C). Ten budeme vnímat jako primární. Nejdříve zapíšeme tuto primární úlohu ve zkrácenějším tvaru a posléze pro ni odvodíme úlohu duální.

Zavedeme vektory

$$\mathbf{z} := \begin{pmatrix} \mathbf{x} \\ t \end{pmatrix} \in \mathbb{R}^{n+1}, \quad \mathbf{c} := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^{n+1},$$

takže účelovou funkci primární úlohy lze zapsat jako $\max \mathbf{c}^\top \mathbf{z}$.

Označíme-li i -tý řádek matice A jako $\mathbf{a}_i^\top$, pak nerovnosti $t \leq \mathbf{e}_i^\top A \mathbf{x} = \mathbf{a}_i^\top \mathbf{x}$ lze psát jako

$$(-\mathbf{a}_i^\top \ 1) \mathbf{z} \leq 0, \quad i = 1, \dots, m.$$

Definujeme

$$B := \begin{pmatrix} -\mathbf{a}_1^\top & 1 \\ \vdots & \vdots \\ -\mathbf{a}_m^\top & 1 \end{pmatrix}, \quad \mathbf{d} := \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \end{pmatrix} \in \mathbb{R}^{n+1},$$

a úlohu přepíšeme ve tvaru

$$\begin{aligned} & \text{maximalizuj} && \mathbf{c}^\top \mathbf{z} \\ & \text{při podmínkách} && B \mathbf{z} \leq \mathbf{0}, \\ & && \mathbf{d}^\top \mathbf{z} = 1, \\ & && z_1, \dots, z_n \geq 0, \ z_{n+1} = t \text{ volná.} \end{aligned} \tag{P}$$

Duální úloha v maticovém tvaru

Připomeňme velmi stručně pravidla duality známé z předchozích kapitol, která využijeme nyní:

- primární omezení typu „ $\leq$ “ $\longleftrightarrow$ duální proměnná ≥ 0 ,
- primární omezení typu „ $=$ “ $\longleftrightarrow$ duální proměnná bez omezení znaménka,
- primární proměnná ≥ 0 $\longleftrightarrow$ duální omezení typu „ $\geq$ “,
- primární proměnná volná $\longleftrightarrow$ duální omezení typu „ $=$ “,
- primárním cílem je „max“ $\longleftrightarrow$ duálním cílem je „min“.

Použitím těchto pravidel na (P) získáme:

- z nerovností $B \mathbf{z} \leq \mathbf{0}$ vektor duálních proměnných $\mathbf{y} \in \mathbb{R}^m$ s komponentami $y_i \geq 0$,
- z rovnosti $\mathbf{d}^\top \mathbf{z} = 1$ jednu duální proměnnou $u \in \mathbb{R}$ bez znaménkového omezení,
- duální účelovou funkci, kterou budeme minimalizovat.

Aby byl zápis duální úlohy co nejvíce symetrický s primární maticovou formulací, zavedeme vektory

$$\mathbf{w} := \begin{pmatrix} \mathbf{y} \\ u \end{pmatrix} \in \mathbb{R}^{m+1}, \quad \mathbf{r} := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^{m+1},$$

takže duální účelovou funkci můžeme psát jako skalární součin

$$\min \mathbf{r}^T \mathbf{w}.$$

Koeficienty u primárních proměnných v duální úloze vznikají jako lineární kombinace řádků B a vektoru $\mathbf{d}$ s váhami $\mathbf{y}$ a u . Kompaktně je lze zapsat jako vektor

$$B^T \mathbf{y} + u \mathbf{d} \in \mathbb{R}^{n+1},$$

který musí „odpovídat“ vektoru $\mathbf{c}$ a typu primárních proměnných:

- protože primární proměnné $z_1, \dots, z_n$ jsou nezáporné, musí v prvních n složkách platit nerovnosti typu „ $\geq$ “,
- poslední proměnná $z_{n+1} = t$ je volná, proto odpovídá duální rovnost.

Dostáváme tak kompaktní maticový tvar duální úlohy

$$\begin{aligned} &\text{minimalizuj} && \mathbf{r}^T \mathbf{w} \\ &\text{při podmínkách} && B^T \mathbf{y} + u \mathbf{d} \geq \mathbf{c} \quad (\text{v prvních } n \text{ složkách}), \\ & && (B^T \mathbf{y} + u \mathbf{d})_{n+1} = c_{n+1}, \\ & && \mathbf{y} \geq \mathbf{0}. \end{aligned} \tag{D}$$

V našem případě má výraz $B^T \mathbf{y} + u \mathbf{d}$ konkrétní tvar

$$B^T \mathbf{y} + u \mathbf{d} = \begin{pmatrix} -A^T \mathbf{y} + u \mathbf{1} \\ \sum_{i=1}^m y_i \end{pmatrix},$$

Duální podmínky tedy říkají

$$-A^T \mathbf{y} + u \mathbf{1} \geq \mathbf{0}, \quad \sum_{i=1}^m y_i = 1, \quad \mathbf{y} \geq \mathbf{0}.$$

To je přesná maticová verze duální úlohy. Její rozepsání po složkách vede k lineárnímu programu pro řádkového hráče odvozenému dříve z „min–max“ formulace.

A teď přichází hlavní pointa celé kapitoly. Obě úlohy jsou přípustné: lze například zvolit rovnoměrnou smíšenou strategii a dostatečně malé t , respektive dostatečně velké u . Jejich optimální hodnoty jsou konečné, protože očekávaná výplata leží mezi nejmenším a největším prvkem matice A . Věta o silné dualitě proto dává $t^* = u^*$. Primární optimum je maximinová strategie sloupceového hráče, duální optimum minimaxová strategie řádkového hráče a společná optimální hodnota je v . Tím dualita lineárního programování zároveň dokazuje větu o minimaxu v konečném maticovém případě.

5 Krátký ilustrační příklad

Ukažme formulaci obou lineárních programů na jednoduché hře se dvěma strategiemi a symetrickou výplatní maticí.

Příklad 10.13: Hra se dvěma protikladnými strategiemi

Uvažujme maticovou hru se dvěma řádky a dvěma sloupci a výplatní maticí

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

Hodnota a_{ij} je zde výplata sloupcového hráče, tj. kladné číslo znamená zisk sloupcového hráče, záporné číslo zisk řádkového hráče.

LP pro sloupcového hráče. Označme $\mathbf{x} = (x_1, x_2)^T$ smíšenou strategii sloupcového hráče. Podle (LP_C) má řešit úlohu

$$\begin{array}{ll} \text{maximalizuj} & v \\ \text{při podmínkách} & v \leq (1, -1) \mathbf{x} = x_1 - x_2, \\ & v \leq (-1, 1) \mathbf{x} = -x_1 + x_2, \\ & x_1 + x_2 = 1, \\ & x_1, x_2 \geq 0. \end{array}$$

LP pro řádkového hráče. Označme $\mathbf{y} = (y_1, y_2)^T$ strategii řádkového hráče. Z (LP_R) dostáváme

$$\begin{array}{ll} \text{minimalizuj} & u \\ \text{při podmínkách} & \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}^T \mathbf{y} \leq u \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \\ & y_1 + y_2 = 1, \\ & y_1, y_2 \geq 0. \end{array}$$

Po rozepsání prvního omezení dostaneme

$$\begin{array}{ll} \text{minimalizuj} & u \\ \text{při podmínkách} & y_1 - y_2 \leq u, \\ & -y_1 + y_2 \leq u, \\ & y_1 + y_2 = 1, \\ & y_1, y_2 \geq 0. \end{array}$$

V tomto konkrétním příkladu musí optimální strategie vyrovnat obě protichůdné lineární výplaty. Proto oba hráči volí čisté strategie se stejnou pravděpodobností:

$$\mathbf{x}^* = \mathbf{y}^* = \left(\frac{1}{2}, \frac{1}{2}\right)^T,$$

a hodnota hry je $v = 0$. V rovnováze tedy žádný z hráčů nemá výhodu z hlediska očekávané výplaty.

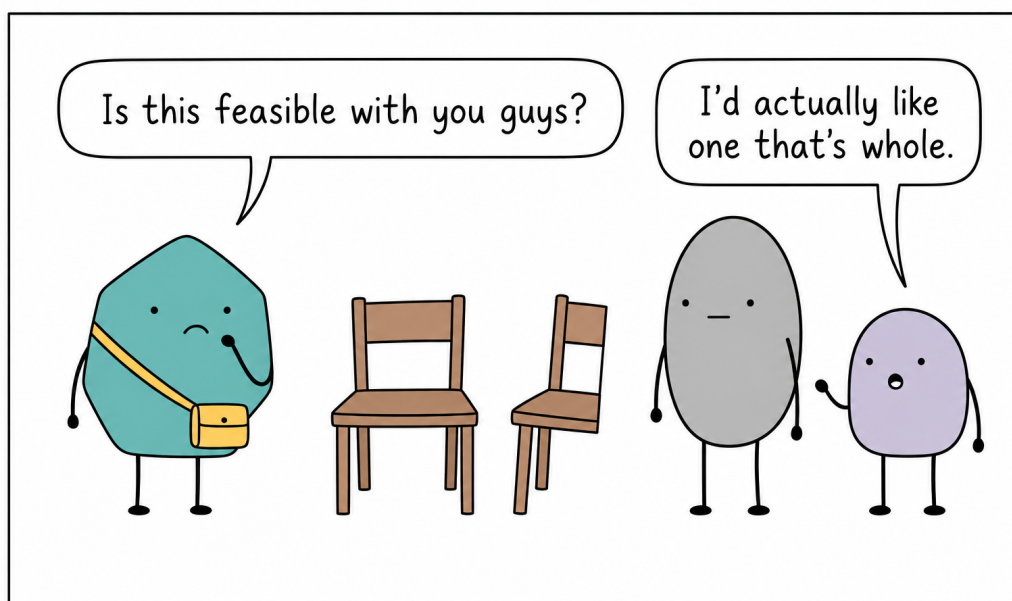
Pozor však na lákavou zkratku: samotná symetrie matice rovnoměrnou strategii obecně nezaručuje. Například symetrická výplatní matice $\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$ má optimální strategii $(1/3, 2/3)^T$, nikoli $(1/2, 1/2)^T$. Rozhodující je tedy vyřešení podmínek max–min a min–max, nikoli pouhá symetrie zápisu.

KAPITOLA 11

CELOČÍSELNÉ PROGRAMOVÁNÍ: VYBRANÉ METODY ŘEŠENÍ

1	Krátké zopakování pojmů ILP/MILP	114
2	Dva základní modelovací příklady	115
3	LP-relaxace, větvení a řezy: obecné schéma	115
4	Metoda větví a mezí	116
5	Gomoryho řezy	119

V této závěrečné kapitole se vrátíme k celočíselnému a smíšenému celočíselnému programování (ILP/MILP), které jsme dříve stručně motivovali. Nejprve krátce zopakujeme, co přesně znamená přidat celočíselné podmínky a jakou roli hraje LP-relaxace. Kapitola je dále věnována dvěma metodám používaným při řešení úloh celočíselného programování: metodě větví a mezí (*branch and bound*) a metodě Gomoryho řezů. Obě metody staví na LP-relaxaci, každá však jiným způsobem.

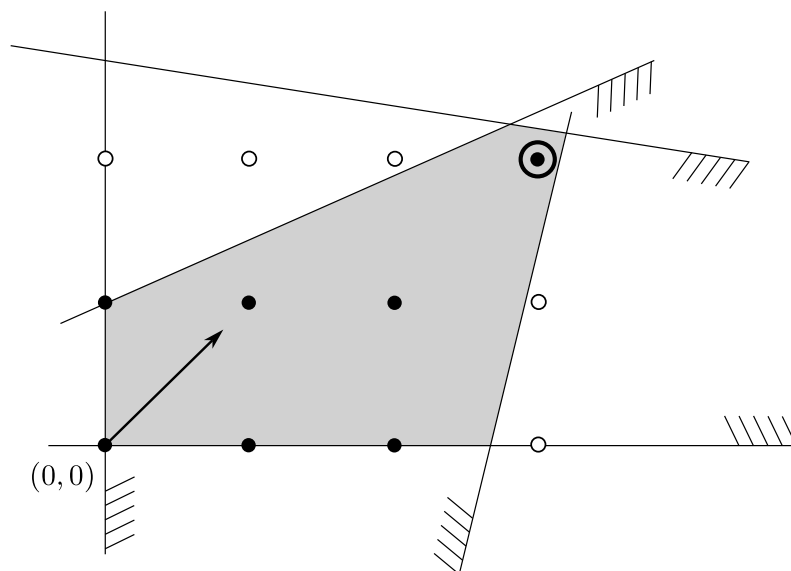


1 Krátké zopakování pojmů ILP/MILP

V úvodní kapitole jsme zavedli celočíselný lineární program (ILP) a smíšený celočíselný lineární program (MILP) jako lineární programy s dodatečnými celočíselnými podmínkami na vybrané proměnné. Zde připomeneme jen několik klíčových pozorování, která budou důležitá v této kapitole:

- V čistém ILP požadujeme celočíselnost všech rozhodovacích proměnných, tedy $\mathbf{x} \in \mathbb{Z}^n$ (často navíc $\mathbf{x} \geq \mathbf{0}$). V MILP je celočíselnost předepsána jen pro indexy z neprázdné množiny $I \subsetneq \{1, \dots, n\}$. Ostatní proměnné zůstávají reálné.
- LP-relaxace vznikne vypuštěním podmínek $x_i \in \mathbb{Z}$ pro $i \in I$. Všechna lineární omezení včetně mezí proměnných zachováme. Existují-li příslušné konečné optimální hodnoty, je u maximalizační úlohy optimum LP-relaxace horním odhadem optima ILP/MILP, u minimalizační úlohy dolním odhadem.
- Přípustná množina čistého ILP je tvořena celočíselnými body mnohostěnu a je obecně diskrétní a nekonvexní. Přípustná množina MILP může obsahovat spojitě části, protože některé proměnné jsou reálné. Ani ona však obecně není konvexní. LP-relaxace je v obou případech konvexní mnohostěn.

Geometricky vybíráme celočíselné body v mnohostěnu P_{LP} . Tento mnohostěn je přípustnou množinou LP-relaxace, kterou umíme řešit technikami z předchozích kapitol.



Obrázek 11.1: Přípustná množina LP-relaxace (šedý konvexní mnohoúhelník) a přípustné body čistého ILP (černé celočíselné body).

Základní strategie tedy bude stále stejná: nejprve si pomůžeme „levným“ řešením LP-relaxace a z něj získáme odhad na optimum. Potom budeme nechtěné neceločíselné části buď odřezávat, nebo úlohu větvit, dokud nenalezneme optimální řešení splňující předepsanou celočíselnost – anebo neprokážeme, že žádné takové řešení neexistuje.

2 Dva základní modelovací příklady

Celočíselné proměnné umožňují popsat rozhodnutí typu ano–ne i nedělitelné počty objektů. Následující dva příklady zároveň ukazují, že kvalita LP-relaxace závisí na struktuře konkrétního modelu.

Příklad 11.1: Úloha batohu s binárními proměnnými

Máme n předmětů. Předmět j má zisk $p_j > 0$ a hmotnost $w_j > 0$. Batoh má kapacitu $W > 0$. Binární proměnná x_j udává, zda předmět vybereme. Model je

$$\begin{aligned} &\text{maximalizovat} && \sum_{j=1}^n p_j x_j, \\ &\text{při} && \sum_{j=1}^n w_j x_j \leq W, \\ &&& x_j \in \{0, 1\}, \quad j = 1, \dots, n. \end{aligned}$$

LP-relaxace nahrazuje $x_j \in \{0, 1\}$ podmínkami $0 \leq x_j \leq 1$. Dovoluje tedy vybírat zlomkové části předmětů a její optimální hodnota může být ostře větší než hodnota původní úlohy.

Příklad 11.2: Přirazení pracovníků k úkolům

Máme m pracovníků, m úkolů a náklad c_{ij} přiřazení pracovníka i k úkolu j . Proměnná x_{ij} je rovna jedné právě tehdy, když pracovník i vykonává úkol j . Dostáváme model

$$\begin{aligned} &\text{minimalizovat} && \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_{ij}, \\ &\text{při} && \sum_{j=1}^m x_{ij} = 1, \quad i = 1, \dots, m, \\ &&& \sum_{i=1}^m x_{ij} = 1, \quad j = 1, \dots, m, \\ &&& x_{ij} \in \{0, 1\}. \end{aligned}$$

Na rozdíl od obecného ILP jsou všechny vrcholy LP-relaxace tohoto modelu permutační matice. Má-li tedy úloha přípustné řešení, existuje celočíselné optimum už mezi optimálními řešeními LP-relaxace. Tento výsledek, známý jako Birkhoffova–von Neumannova věta, je zvláštní vlastností přiřazovacího mnohostěnu, nikoli obecným pravidlem celočíselného programování (Schrijver, 1986).

3 LP-relaxace, větvení a řezy: obecné schéma

Většina známých metod pro obecné ILP/MILP stojí na dvou krocích:

1. **Uvolnění problému:** podmínky celočíselnosti dočasně vypustíme a vyřešíme LP-relaxaci, která poskytne odhad na optimum.
2. **Zpřesnění:** pokud optimální řešení relaxace nesplňuje předepsanou celočíselnost, úlohu zpřísníme – buď *lokálně* (větvením), nebo *globálně* (platným řezem).

Obecné schéma metody pro ILP/MILP

Mějme maximalizační úlohu

$$\max\{\mathbf{c}^T \mathbf{x} \mid \mathbf{x} \in X_I\}, \quad X_I := \{\mathbf{x} \in P \mid x_i \in \mathbb{Z} \text{ pro všechna } i \in I\},$$

kde $P \subseteq \mathbb{R}^n$ je mnohostěn a I je množina indexů celočíselných proměnných. Pro čisté ILP je $I = \{1, \dots, n\}$. Metoda typicky udržuje:

- *relaxovaný mnohostěn* $P^{(k)}$, který obsahuje všechna řešení z X_I splňující vět-vící omezení příslušného podproblému,
- nejlepší dosud nalezené řešení splňující předepsanou celočíselnost, $\mathbf{x}^{\text{best}}$, nebo informaci, že zatím žádné neznáme.

V k -tém kroku:

1. řeší LP nad $P^{(k)}$ a získá optimum $\mathbf{x}^{(k)}$ a horní odhad $z_{\text{LP}}^{(k)}$,
2. podle typu metody:
 - buď rozdělí problém na podproblémy s doplněnými celočíselnými omeze-ními (větvení),
 - nebo přidá do $P^{(k)}$ nové nerovnosti (řezy), které odříznou $\mathbf{x}^{(k)}$, ale zacho-vají všechna řešení z X_I .

Pokud je horní LP-odhad pro danou větev nejvýše roven hodnotě nejlepšího zná-mého řešení, lze větev bezpečně uzavřít. Jestliže žádná otevřená větev nezbyde, je nejlepší nalezené řešení globálně optimální. Pokud v této situaci žádné řešení nale-zeno nebylo, je původní úloha nepřipustná.

V následujících dvou sekcích uvidíme obě možnosti v akci. Metoda větví a mezí rozdělí úlohu na podproblémy ve stromové struktuře. Gomoryho řezy naopak ponechají jednu globální relaxaci a její nevhodné části budou postupně „odřezávat“.

4 Metoda větví a mezí

4.1 Základní myšlenka

Na první pohled by se nabízelo všechny kombinace celočíselných hodnot prostě vyzkou-šet. Takový hrubý přístup by však velmi rychle explodoval (pro binární vektor délky n je možností 2^n). Metoda větví a mezí (*branch and bound*) nahrazuje naivní úplný průchod

chytřejším stromovým prohledáváním. Horní odhady z LP-relaxací nám dovolí řadu podstromů vůbec neprozkoumat. Klasickou obecnou formulaci metody podaly Land and Doig (1960).

Každý uzel stromu odpovídá ILP/MILP úloze s dodatečnými omezeními (typicky $x_j \leq k$ nebo $x_j \geq k + 1$ pro $j \in I$). Po vyřešení LP-relaxace uzlu nastane jedna z následujících situací (případ neomezené relaxace probereme zvlášť níže):

- a) LP-relaxace je nepřipustná $\Rightarrow$ daný uzel ani jeho podstrom nemůže obsahovat řešení splňující předepsanou celočíselnost.
- b) Optimální hodnota LP-relaxace není lepší než hodnota nejlepšího známého řešení splňujícího celočíselnost $\Rightarrow$ uzel uzavřeme.
- c) Optimální řešení LP splňuje $x_i \in \mathbb{Z}$ pro všechna $i \in I \Rightarrow$ je optimální i pro podproblém ILP/MILP a případně aktualizujeme nejlepší známé řešení.
- d) Některá proměnná x_j , $j \in I$, je v optimu LP neceločíselná $\Rightarrow$ vytvoříme dva potomky pomocí větvících omezení.

4.2 Schematický algoritmus

Pro jednoduchost uvažujme maximalizační úlohu. Nechť z_{best} je hodnota nejlepšího dosud nalezeného řešení splňujícího předepsanou celočíselnost (počátečně $-\infty$).

Metoda větví a mezí – základní schéma
1. Založ <i>kořenový uzel</i> s původní ILP/MILP úlohou (bez dodatečných větvících omezení). 2. Dokud existují nevyřešené uzly: i) Vyber uzel (pomocí nějakého pravidla) a spočítej LP-relaxaci. ii) Je-li relaxace LP nepřipustná, uzel uzavřeme. iii) Je-li $z_{\text{LP}} \leq z_{\text{best}}$, uzel uzavřeme (nemůže obsahovat lepší řešení). iv) Splňuje-li LP optimum podmínky $x_i \in \mathbb{Z}$ pro všechna $i \in I$, případně aktualizujeme nejlepší řešení a uzel uzavřeme. v) Jinak zvolíme některou neceločíselnou proměnnou x_j s $j \in I$ a vytvoříme dva potomky s doplněnými omezeními $x_j \leq \lfloor \hat{x}_j \rfloor, \quad x_j \geq \lceil \hat{x}_j \rceil,$ kde $\hat{x}_j$ je hodnota proměnné x_j v optimu LP-relaxace. Uzly zařadíme k dalšímu zpracování. 3. Jakmile nejsou k dispozici další uzly, je $\mathbf{x}^{\text{best}}$ globálně optimální. Pokud jsme žádné řešení splňující celočíselnost nenašli, je úloha nepřipustná.

Poznámka 11.3: Výběr uzlu a proměnné

V praxi existuje celá řada heuristik, jak vybírat uzel (hledání do hloubky, do šířky, *best-bound* atd.) i větvící proměnnou. Volba těchto pravidel může dramaticky ovlivnit velikost prohledávaného stromu, přesto však nemění korektnost metody: všechny uzly se buď navštíví, nebo se jejich podstrom správně ořízne pomocí odhadů z LP.

Poznámka 11.4: Konečnost a neomezené relaxace

Uvedené schéma předpokládá, že řešené LP-relaxace mají konečné optimum. Neomezená relaxace neposkytuje konečný odhad a vyžaduje samostatnou analýzu přípustnosti celočíselného podproblému a neomezeného směru. Konečnost stromu je bezprostředně zaručena například tehdy, když mají všechny celočíselné proměnné konečné dolní a horní meze. Bez takových předpokladů je popsané schéma korektním principem, nikoli samo o sobě důkazem konečného ukončení každé možné implementace.

4.3 Jednoduchý ilustrativní příklad**Příklad 11.5: Metoda větví a mezí ve dvou proměnných**

Uvažujme úlohu

$$\begin{aligned} \max \quad & 3x_1 + 2x_2 \\ \text{při} \quad & 4x_1 + 2x_2 \leq 9, \\ & 2x_1 + 4x_2 \leq 9, \\ & x_1, x_2 \in \mathbb{Z}_{\geq 0}. \end{aligned}$$

LP-relaxace (s $x_1, x_2 \geq 0$ reálnými) má optimum v průsečíku

$$4x_1 + 2x_2 = 9, \quad 2x_1 + 4x_2 = 9,$$

tedy v bodě

$$(\hat{x}_1, \hat{x}_2) = \left(\frac{3}{2}, \frac{3}{2}\right), \quad z_{\text{LP}} = 3 \cdot \frac{3}{2} + 2 \cdot \frac{3}{2} = \frac{15}{2} = 7,5.$$

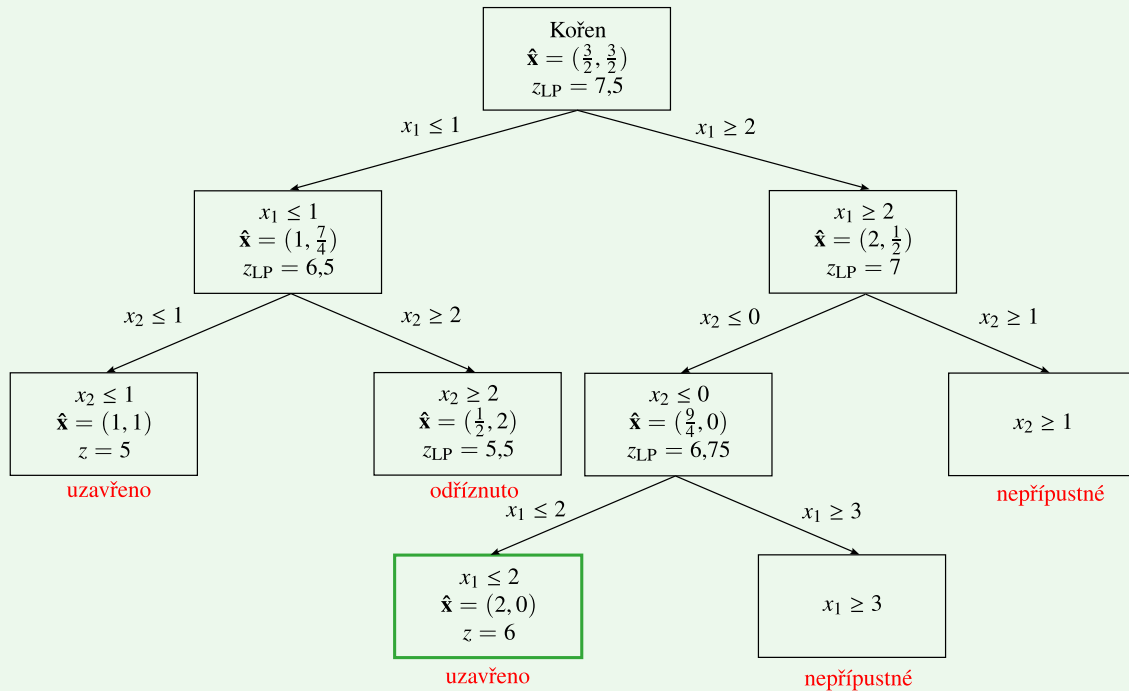
Tento bod není celočíselný, proto větvíme například podle x_1 :

$$x_1 \leq 1 \quad \text{a} \quad x_1 \geq 2.$$

Při volbě uzlů například strategií *best-bound* dostaneme následující průběh:

- V uzlu $x_1 \geq 2$ vyjde LP optimum $(2, \frac{1}{2})$ s odhadem $z_{\text{LP}} = 7$, větvíme tedy podle x_2 na $x_2 \leq 0$ a $x_2 \geq 1$. Uzel $x_2 \geq 1$ je nepřipustný (protože z $4x_1 + 2x_2 \leq 9$ a $x_1 \geq 2$ plyne $x_2 \leq \frac{1}{2}$). V uzlu $x_2 \leq 0$ dostaneme LP optimum $(\frac{9}{4}, 0)$ s odhadem $z_{\text{LP}} = \frac{27}{4} = 6,75$ a větvíme podle x_1 . Z větve $x_1 \leq 2$ dostaneme celočíselné řešení $(2, 0)$ s hodnotou $z_{\text{best}} = 6$. Sourozenecká větev $x_1 \geq 3$ je nepřipustná, protože při $x_2 \geq 0$ by platilo $4x_1 + 2x_2 \geq 12 > 9$.

- V uzlu $x_1 \leq 1$ vyjde LP optimum $(1, \frac{7}{4})$ s odhadem $z_{LP} = \frac{13}{2} = 6,5$, a tedy ještě nelze uzel odříznout (protože $6,5 > 6$). Větvíme podle x_2 na $x_2 \leq 1$ a $x_2 \geq 2$. Uzel $x_2 \leq 1$ dá celočíselné optimum $(1, 1)$ s hodnotou 5 (dosud nejlepší řešení nezlepší), a uzel $x_2 \geq 2$ má horní odhad $z_{LP} = \frac{11}{2} = 5,5 \leq 6$, takže jej odřízneme podle odhadu.



Obrázek 11.2: Strom metody větví a mezí pro úlohu z Příkladu 11.5. U každého řešeného uzlu je uvedeno optimum jeho LP-relaxace a horní odhad z_{LP} . Listy jsou uzavřeny celočíselným řešením, nepřípustností nebo porovnáním odhadu s dosud nejlepším řešením.

Výsledkem je globální optimum

$$z^* = 6 \quad \text{v bodě} \quad (x_1, x_2) = (2, 0).$$

5 Gomoryho řezy

5.1 Základní myšlenka

V metodách využívajících rovin jako řezů (v literatuře *cutting-plane methods*) pracujeme s jedinou „globální“ LP-relaxací, kterou postupně zpřisňujeme přidáváním dalších omezení. Místo rozvětveného stromu tedy sledujeme jeden mnohostěn, z něhož postupně odkrajujeme části bez celočíselných řešení. Začínáme z relaxace původní celočíselné úlohy a v každém kroku hledáme nerovnost, která

- je *platná* pro všechna celočíselná řešení (tj. nevyřazuje žádný bod z $P \cap \mathbb{Z}^n$),
- ale je *porušena* optimálním řešením aktuální LP-relaxace.

Taková nerovnost se nazývá *řez* (*cut*), protože „odřízne“ aktuální optimum, ale zachová všechny celočíselné přípustné body. Po přidání řezu úlohu znovu řešíme jako relaxovanou úlohu LP (například simplexovou metodou) a proces opakujeme, dokud nedostaneme celočíselné optimum. V praxi se může stát, že přidávání řezů začne být numericky problematické (degenerace, špatná podmíněnost, kumulace omezení apod.), proto moderní řešiče řezy obvykle kombinují s již zmíněným větvením.

Existuje mnoho typů řezů (Chvátal–Gomory, pokrývací řezy, subtour-eliminační řezy pro TSP, ...). Zde se omezíme na klasické *Gomoryho frakční řezy* pro čisté ILP (Gomory, 1958). Pro MILP se používají jejich zobecnění, zejména Gomoryho smíšené celočíselné řezy (GMI). Jejich odvození je nad rámec tohoto textu.

5.2 Gomoryho řez

Uvažujme celočíselný program s racionálními daty v rovnicovém tvaru

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{při} \quad & A\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq \mathbf{0}, \mathbf{x} \in \mathbb{Z}^n. \end{aligned}$$

kde $A \in \mathbb{Q}^{m \times n}$ má hodnot m a $\mathbf{b} \in \mathbb{Q}^m$, $\mathbf{c} \in \mathbb{Q}^n$. V tomto zápisu jsou celočíselné všechny proměnné, nejen původní rozhodovací proměnné. Vznikl-li rovnicový tvar z nerovností s celočíselnými koeficienty a pravými stranami přidáním doplňkových proměnných, jsou i tyto doplňkové proměnné celočíselné v každém celočíselném řešení. Pokud by některá proměnná v řádku byla spojitá, následující frakční řez by obecně nebyl platný.

Poznámka 11.6: Racionalita dat

Konstrukce Gomoryho řezů je založena na rozkladu čísel na celou a zlomkovou část. Formálně proto předpokládáme, že data úlohy jsou racionální, což zaručuje racionalitu všech koeficientů simplexové tabulky.

Po vyřešení LP-relaxace máme simplexovou tabulku odpovídající optimálnímu řešení ve tvaru

$$\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \quad z = z_0 + \mathbf{r}^\top \mathbf{x}_N,$$

tj. po složkách (pro $B = \{j_1 < \dots < j_m\}$ a $N = \{\ell_1 < \dots < \ell_{n-m}\}$)

$$x_{j_i} = p_i + \sum_{k=1}^{n-m} q_{ik} x_{\ell_k}.$$

Bázické přípustné řešení odpovídající této tabulce je dáno volbou $\mathbf{x}_N = \mathbf{0}$, tedy $\mathbf{x}_B = \mathbf{p}$.

Předpokládejme, že pro některý řádek i je $p_i \notin \mathbb{Z}$. Označme zlomkovou část čísla t jako $\{t\} = t - \lfloor t \rfloor \in [0, 1)$. Položme

$$f_i = \{p_i\} \in (0, 1), \quad f_{ik} = \{-q_{ik}\} \in [0, 1).$$

Potom platí, že nerovnost

$$\sum_{k=1}^{n-m} f_{ik} x_{\ell_k} \geq f_i$$

je platná pro všechna celočíselná řešení a současné frakční bázické optimum ji porušuje (protože v něm je $\mathbf{x}_N = \mathbf{0}$, takže levá strana je $0 < f_i$). Právě touto nerovností definujeme Gomoryho řez, jak je shrnuto v následující definici.

Definice 11.7: Gomoryho frakční řez

Nechť v optimální simplexové tabulce LP-relaxace čistého ILP existuje báze $B = \{j_1, \dots, j_m\}$ a množina nebázických indexů $N = \{\ell_1, \dots, \ell_{n-m}\}$. Předpokládejme, že řádek s indexem i má tvar

$$x_{j_i} = p_i + \sum_{k=1}^{n-m} q_{ik} x_{\ell_k}, \quad p_i \notin \mathbb{Z}.$$

Všechny proměnné v tomto řádku musejí být v uvažovaném rovnicovém modelu celočíselné. To zahrnuje i zavedené doplňkové proměnné. Označme $f_i = \{p_i\}$ a $f_{ik} = \{-q_{ik}\}$. Potom nerovnost

$$\sum_{k=1}^{n-m} f_{ik} x_{\ell_k} \geq f_i$$

nazýváme *Gomoryho řez*.

Poznámka 11.8: Proč Gomoryho řez platí a proč je porušen

Uvažujme zvolený řádek simplexové tabulky

$$x_{j_i} = p_i + \sum_{k=1}^{n-m} q_{ik} x_{\ell_k}, \quad p_i \notin \mathbb{Z},$$

a předpokládejme, že pracujeme s ILP, takže v libovolném celočíselném řešení jsou $x_{j_i} \in \mathbb{Z}$ a zároveň i všechny nebázické proměnné $x_{\ell_k} \in \mathbb{Z}$. Připomeňme, že stále musí platit i $\mathbf{x} \geq \mathbf{0}$.

Zaveďme pomocné koeficienty $a_{ik} = -q_{ik}$, takže řádek přepíšeme do tvaru

$$x_{j_i} = p_i - \sum_{k=1}^{n-m} a_{ik} x_{\ell_k}.$$

Rozložme nyní pravou stranu na celou a zlomkovou část:

$$p_i = \lfloor p_i \rfloor + f_i, \quad a_{ik} = \lfloor a_{ik} \rfloor + f_{ik},$$

kde $f_i = \{p_i\} \in (0, 1)$ a $f_{ik} = \{a_{ik}\} = \{-q_{ik}\} \in [0, 1)$. Po dosazení dostaneme

$$x_{j_i} = \underbrace{\left(\lfloor p_i \rfloor - \sum_{k=1}^{n-m} \lfloor a_{ik} \rfloor x_{\ell_k} \right)}_{\in \mathbb{Z}} + \left(f_i - \sum_{k=1}^{n-m} f_{ik} x_{\ell_k} \right).$$

V celočíselném řešení je levá strana $x_{j_i} \in \mathbb{Z}$ a první závorka je také celočíselná, proto i druhá závorka musí být celé číslo. Zároveň ale platí

$$f_i - \sum_{k=1}^{n-m} f_{ik} x_{\ell_k} \leq f_i < 1,$$

tedy tato hodnota nemůže být kladné celé číslo. Musí proto být ≤ 0 , a dostáváme

$$\sum_{k=1}^{n-m} f_{ik} x_{\ell_k} \geq f_i.$$

To ukazuje, že nerovnost (Gomoryho řez) je *platná* pro všechna celočíselná řešení. Naopak, pro současné bázecké řešení odpovídající tabulce platí $\mathbf{x}_N = \mathbf{0}$, tedy $x_{\ell_k} = 0$ pro všechna k a levá strana řezu je rovna 0. Protože $f_i \in (0, 1)$, vyjde $0 < f_i$ a řez je v tomto bázeckém optimu *porušen*.

Poznámka 11.9: Řezy vs. větve

Gomoryho řezy a metodu větví a mezí lze chápat jako dvě komplementární strategie:

- metoda větví a mezí dělí prostor řešení na mnoho podproblémů (větví), které se prořezávají pomocí LP-odhadů,
- řezové metody ponechávají jediný globální mnohostěn a zpřesňují jej přidáváním platných nerovností.

Moderní řešiče obvykle kombinují obě myšlenky: nejprve pomocí řezů výrazně zpřesní kořenovou relaxaci a poté v takto posílené relaxaci spouští metodu větví a mezí. Tato kombinace se nazývá *branch-and-cut*.

5.3 Schematický algoritmus

Opět pro jednoduchost uvažujme maximalizační čisté ILP v rovnicovém tvaru a předpokládejme, že každá řešená LP-relaxace je buď nepřipustná, nebo má konečné optimum. Metoda řezů udržuje posloupnost relaxací $P^{(k)}$ tak, aby vždy platilo

$$P^{(k)} \supseteq P \cap \mathbb{Z}^n,$$

tj. žádný řez nikdy nevyřadí celočíselné řešení.

Gomoryho řezová metoda – základní schéma

1. Polož $k \leftarrow 0$ a $P^{(0)} \leftarrow P_{LP}$ (počáteční LP-relaxace).
2. Opakuj:
 - i) Vyřeš LP nad $P^{(k)}$ simplexovou metodou. Je-li LP nepřipustné, pak (protože všechny přidané řezy jsou platné) neexistuje ani celočíselné řešení. Konec.
 - ii) Jinak vezmi simplexovou tabulku odpovídající optimálnímu řešení ve tvaru

$$\mathbf{x}_B = \mathbf{p} + Q\mathbf{x}_N, \quad z = z_0 + \mathbf{r}^T \mathbf{x}_N.$$

V bázeckém přípustném řešení odpovídajícím tabulce je $\mathbf{x}_N = \mathbf{0}$, tedy $\mathbf{x}_B = \mathbf{p}$. Je-li $\mathbf{p} \in \mathbb{Z}^m$, máme celočíselné optimum. Konec.

- iii) Jinak vyber řádek i s neceločíselným p_i a sestroj Gomoryho řez (viz Definice 11.7) z tohoto řádku.
- iv) Vynásob řez společným jmenovatelem tak, aby měl celočíselné koeficienty a pravou stranu. Přidej jej do relaxace. Případná nová doplňková proměnná je celočíselná. Polož $P^{(k+1)} \leftarrow P^{(k)} \cap \{\text{nový řez}\}$, $k \leftarrow k + 1$ a pokračuj.

Poznámka 11.10: Ukončení řezové metody

Popsaný postup vystihuje hlavní myšlenku, ale věta „vždy vyber libovolný frakční řádek“ ještě sama o sobě nezaručuje konečné ukončení. Klasické konečné varianty Gomoryho metody předpokládají čisté ILP s racionálními daty, přesnou aritmetiku, předepsanou volbu frakčního řádku a lexikografickou duální simplexovou metodu pro obnovení přípustnosti. Za těchto podmínek metoda po konečně mnoha iteracích nalezne celočíselné optimum, nebo prokáže nepřípustnost (Gomory, 1958, Schrijver, 1986, Gade and Küçükyavuz, 2013). Technické provedení tohoto důkazu už přesahuje rámec našeho výkladu. V numerické aritmetice navíc mohou vznikat potíže se zaokrouhlováním a téměř celočíselnými hodnotami. Praktické řešiče proto řezy obvykle kombinují s větvením. Podrobný moderní výklad uvádějí Conforti et al. (2014).

5.4 Ilustrativní příklad Gomoryho řezů

Příklad 11.11: Dvě iterace Gomoryho řezů

Uvažujme čistý celočíselný lineární program

$$\begin{aligned} \mathbf{max} \quad & z = x_1 + x_2, \\ \mathbf{při} \quad & 2x_1 + x_2 \leq 4, \\ & x_1 + 2x_2 \leq 4, \\ & x_1, x_2 \in \mathbb{Z}_{\geq 0}. \end{aligned}$$

Po zavedení doplňkových proměnných $s_1, s_2 \geq 0$ dostaneme rovnicový tvar

$$2x_1 + x_2 + s_1 = 4, \quad x_1 + 2x_2 + s_2 = 4.$$

Protože koeficienty i pravé strany jsou celočíselné, jsou pro každé celočíselné (x_1, x_2) celočíselné také s_1 a s_2 . Lze je tedy při odvození Gomoryho frakčního řezu považovat za celočíselné proměnné.

LP-relaxace. Řešení LP-relaxace má optimum v průsečíku obou omezení,

$$(\hat{x}_1, \hat{x}_2) = \left(\frac{4}{3}, \frac{4}{3}\right), \quad z_{\text{LP}} = \frac{8}{3},$$

které není celočíselné.

Jedna optimální simplexová tabulka (s bází $B = \{x_1, x_2\}$ a nebází $N = \{s_1, s_2\}$) má tvar

$$\begin{aligned}x_1 &= \frac{4}{3} - \frac{2}{3}s_1 + \frac{1}{3}s_2, \\x_2 &= \frac{4}{3} + \frac{1}{3}s_1 - \frac{2}{3}s_2, \\z &= \frac{8}{3} - \frac{1}{3}s_1 - \frac{1}{3}s_2.\end{aligned}$$

Bázické přípustné řešení odpovídá volbě $s_1 = s_2 = 0$.

První Gomoryho řez. Zvolíme řádek pro proměnnou x_1 , kde

$$p_1 = \frac{4}{3}, \quad f_1 = \{p_1\} = \frac{1}{3}.$$

Koeficienty u nebázických proměnných jsou

$$q_{1,s_1} = -\frac{2}{3}, \quad q_{1,s_2} = \frac{1}{3},$$

tedy

$$-q_{1,s_1} = \frac{2}{3}, \quad -q_{1,s_2} = -\frac{1}{3},$$

proto

$$f_{1,s_1} = \{-q_{1,s_1}\} = \frac{2}{3}, \quad f_{1,s_2} = \{-q_{1,s_2}\} = -\frac{1}{3} - (-1) = \frac{2}{3}.$$

Gomoryho řez má pak tvar

$$\frac{2}{3}s_1 + \frac{2}{3}s_2 \geq \frac{1}{3},$$

neboli po vynásobení třemi

$$2s_1 + 2s_2 \geq 1.$$

Po dosazení $s_1 = 4 - 2x_1 - x_2$ a $s_2 = 4 - x_1 - 2x_2$ dostáváme ekvivalentní omezení

$$2x_1 + 2x_2 \leq 5.$$

Druhá iterace. Po přidání řezu a opětovném vyřešení LP-relaxace vyjde nové optimum

$$(x_1, x_2) = (1, \frac{3}{2}), \quad z_{\text{LP}} = \frac{5}{2}.$$

Jedna odpovídající optimální simplexová tabulka (například s bází $B = \{x_1, x_2, s_1\}$ a nebází $N = \{s_2, s_3\}$, kde celočíselná doplňková proměnná s_3 splňuje $2x_1 + 2x_2 + s_3 = 5$) má tvar

$$\begin{aligned}x_1 &= 1 + s_2 - s_3, \\x_2 &= \frac{3}{2} - s_2 + \frac{1}{2}s_3, \\s_1 &= \frac{1}{2} - s_2 + \frac{3}{2}s_3, \\z &= \frac{5}{2} - \frac{1}{2}s_3.\end{aligned}$$

Z řádku pro proměnnou x_2 máme

$$q_{2,s_2} = -1, \quad q_{2,s_3} = \frac{1}{2},$$

tedy

$$f_{2,s_2} = \{-q_{2,s_2}\} = 0, \quad f_{2,s_3} = \{-q_{2,s_3}\} = \frac{1}{2}.$$

Gomoryho řez má tvar

$$\frac{1}{2}s_3 \geq \frac{1}{2}, \quad \text{tj.} \quad s_3 \geq 1.$$

Protože $s_3 = 5 - 2x_1 - 2x_2$, dostáváme nové omezení

$$x_1 + x_2 \leq 2.$$

Po přidání tohoto řezu je optimální množinou LP-relaxace celá úsečka

$$\{(x_1, x_2) \in \mathbb{R}_{\geq 0}^2 \mid x_1 + x_2 = 2\}, \quad z_{\text{LP}} = 2.$$

Celočíselná optimální řešení jsou $(2, 0)$, $(1, 1)$ a $(0, 2)$. Simplexová metoda může jako optimální bázecké přípustné řešení vrátit například $(2, 0)$ nebo $(0, 2)$. Takové řešení je celočíselné, a Gomoryho metoda proto končí s optimální hodnotou $z^* = 2$. Existence neceločíselných optimálních bodů uvnitř úsečky tomu nevádí: postačuje nalézt jedno celočíselné optimum.

LITERATURA

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993. ISBN 978-0-13-617549-0.

Donald J. Albers and Constance Reid. An interview with George B. Dantzig: The father of linear programming. *The College Mathematics Journal*, 17(4):292–314, 1986. doi: 10.1080/07468342.1986.11972971.

Xavier Allamigeon, Pascal Benchimol, Stéphane Gaubert, and Michael Joswig. What tropical geometry tells us about the complexity of linear programming. *SIAM Review*, 63(1):123–164, 2021. doi: 10.1137/20M1380211.

Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Hoboken, NJ, 4 edition, 2010. ISBN 978-0-470-46272-0.

Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997. ISBN 978-1-886529-19-9.

Robert G. Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, 2(2):103–107, 1977. doi: 10.1287/moor.2.2.103.

Ivan Boldyrev and Till Düppe. Programming the USSR: Leonid V. Kantorovich in context. *The British Journal for the History of Science*, 53(2):255–278, 2020. doi: 10.1017/S0007087420000059.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. URL <https://web.stanford.edu/~boyd/cvxbook/>.

Abraham Charnes and William W. Cooper. The stepping stone method of explaining linear programming calculations in transportation problems. *Management Science*, 1(1):49–69, 1954. doi: 10.1287/mnsc.1.1.49.

Michele Conforti, Gérard Cornuéjols, and Giacomo Zambelli. *Integer Programming*, volume 271 of *Graduate Texts in Mathematics*. Springer, Cham, 2014. ISBN 978-3-319-11007-3. doi: 10.1007/978-3-319-11008-0.

Richard W. Cottle, Ellis L. Johnson, and Roger J.-B. Wets. George B. Dantzig (1914–2005). *Notices of the American Mathematical Society*, 54(3):344–362, 2007. URL <https://www.ams.org/notices/200703/fea-cottle.pdf>.

- George B. Dantzig. Application of the simplex method to a transportation problem. In Tjalling C. Koopmans, editor, *Activity Analysis of Production and Allocation*, number 13 in Cowles Commission Monograph, pages 359–373. John Wiley & Sons, New York, 1951. URL <https://cowles.yale.edu/sites/default/files/2022-09/m13-all.pdf>.
- George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963. doi: 10.1515/9781400884179. DOI odkazuje na pozdější digitální vydání.
- George B. Dantzig. The diet problem. *Interfaces*, 20(4):43–47, 1990. doi: 10.1287/inte.20.4.43.
- George B. Dantzig. Linear programming. *Operations Research*, 50(1):42–47, 2002. doi: 10.1287/opre.50.1.42.17798.
- Yann Disser, Georg Loho, Matthew Maat, and Nils Mosis. Lower bounds for ranking-based pivot rules. In *43rd International Symposium on Theoretical Aspects of Computer Science (STACS 2026)*, volume 364 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi: 10.4230/LIPIcs.STACS.2026.31. URL <https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.STACS.2026.31>.
- Julius Farkas. Theorie der einfachen ungleichungen. *Journal für die reine und angewandte Mathematik*, 124:1–27, 1902. doi: 10.1515/crll.1902.124.1.
- Jana Friebelová. *Operační analýza*. Jihočeská univerzita v Českých Budějovicích, České Budějovice, 2009. ISBN 978-80-7394-193-2. URL <https://omp.ef.jcu.cz/index.php/EF/catalog/view/44/43/104-1>.
- Dinakar Gade and Simge Küçükyavuz. Pure cutting-plane algorithms and their convergence. In James J. Cochran, Louis Anthony Cox, Pinar Keskinocak, Jeffrey P. Kharoufeh, and J. Cole Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, 2013. doi: 10.1002/9780470400531.eorms1084.
- Saul I. Gass. The first linear-programming shoppe. *Operations Research*, 50(1):61–68, 2002. doi: 10.1287/opre.50.1.61.17781.
- Donald Goldfarb and John K. Reid. A practicable steepest-edge simplex algorithm. *Mathematical Programming*, 12:361–371, 1977. doi: 10.1007/BF01593804.
- Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958. doi: 10.1090/S0002-9904-1958-10224-4.
- Frank L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of Mathematics and Physics*, 20:224–230, 1941. doi: 10.1002/sapm1941201224.
- Josef Jablonský and Milada Lagová. *Lineární modely*. Oeconomica, Praha, 3 edition, 2014. ISBN 978-80-245-2020-9. URL <https://oeconomica.vse.cz/publikace/linearni-modely/>.

- Leonid V. Kantorovich. Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422, 1960. doi: 10.1287/mnsc.6.4.366. Anglický překlad práce z roku 1939.
- Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984. doi: 10.1007/BF02579150.
- Leonid G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979. URL <https://www.mathnet.ru/eng/dan42319>. Anglický překlad. Odkaz vede na záznam ruského originálu.
- Victor Klee and George J. Minty. How good is the simplex algorithm? In Oved Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, New York, 1972.
- Ailsa H. Land and Alison G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960. doi: 10.2307/1910129.
- Carlton E. Lemke. The dual method of solving the linear programming problem. *Naval Research Logistics Quarterly*, 1(1):36–47, 1954. doi: 10.1002/nav.3800010107.
- David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, Cham, 4 edition, 2016. doi: 10.1007/978-3-319-18842-3.
- Jiří Matoušek. Lineární programování: Úvod pro informatiky. Technical Report 2006-311, Katedra aplikované matematiky, MFF UK, 2006. URL <https://iti.mff.cuni.cz/series/2006/311.pdf>.
- Jiří Matoušek and Bernd Gärtner. *Understanding and Using Linear Programming*. Universitext. Springer, Berlin and Heidelberg, 2007. ISBN 978-3-540-30697-9. doi: 10.1007/978-3-540-30717-4.
- Alexander I. Nazarov and Galina I. Sinkevich. History of Leningrad mathematics in the first half of the 20th century. *Notices of the International Congress of Chinese Mathematicians*, 7(2):47–63, 2019. doi: 10.4310/ICCM.2019.v7.n2.a5.
- George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, 1988. doi: 10.1002/9781118627372.
- Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994. ISBN 978-0-262-65040-3.
- Christos H. Papadimitriou. On the complexity of integer programming. *Journal of the ACM*, 28(4):765–768, 1981. doi: 10.1145/322276.322287.
- R. Tyrrell Rockafellar. *Convex Analysis*. Number 28 in Princeton Mathematical Series. Princeton University Press, Princeton, NJ, 1970. ISBN 978-0-691-08069-7. doi: 10.1515/9781400873173.
- Jan Rohn. Lineární a nelineární programování. Učební text, MFF UK, 1997. URL https://kam.mff.cuni.cz/~hladik/doc/Rohn-LP_NLP-1997.pdf.
- Royal Swedish Academy of Sciences. The prize in economic sciences 1975: Press release. NobelPrize.org, 1975. URL <https://www.nobelprize.org/prizes/economic-sciences/1975/press-release/>.

- Francisco Santos. A counterexample to the Hirsch conjecture. *Annals of Mathematics*, 176(1):383–412, 2012. doi: 10.4007/annals.2012.176.1.7.
- Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Chichester, 1986. ISBN 0-471-90854-1. URL <https://ir.cwi.nl/pub/10206>.
- George J. Stigler. The cost of subsistence. *Journal of Farm Economics*, 27(2):303–314, 1945. doi: 10.2307/1231810.
- Robert J. Vanderbei. *Linear Programming: Foundations and Extensions*, volume 285 of *International Series in Operations Research & Management Science*. Springer, Cham, 5 edition, 2020. ISBN 978-3-030-39415-8. doi: 10.1007/978-3-030-39415-8.
- John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100: 295–320, 1928. doi: 10.1007/BF01448847.
- Tomáš Werner. Optimalizace. Skripta předmětu B0B33OPT, FEL ČVUT, 2026. URL https://cw.fel.cvut.cz/wiki/_media/courses/b0b33opt/opt-2up.pdf.
- Günter M. Ziegler. *Lectures on Polytopes*. Springer, New York, 1995. doi: 10.1007/978-1-4613-8431-1.