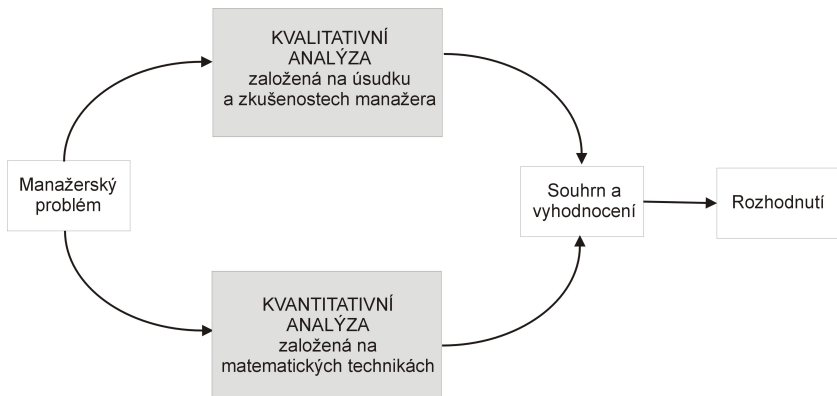


Kvantitativní metody v rozhodování

Každý manažer je ve své denodenní praxi vystaven řadě rozhodovacích situací a problémů, které může analyzovat ze dvou hledisek:

- buď na základě znalostí a zkušeností (kvalitativní analýza)
- nebo pomocí údajů v numerické podobě a jejich exaktního matematického zpracování (kvantitativní analýza), viz schéma.



Kvantitativní metody v rozhodování

Ve specifických situacích je samozřejmě možné provádět jen jednu ze zmíněných analýz. Spolehne-li se však pouze na kvalitativní analýzu bez číselných propočtů, závisí výsledek do značné míry na dobrém úsudku manažera. Naopak přílišná důvěra v numerické výsledky může být zavádějící: každé číselné řešení je přesné jen do té míry, jak přesně byl zkonstruován model. Navíc kvantitativní analýza problému může být zdlouhavá a neefektivní v situaci, kdy je třeba přijmout rozhodnutí rychle. Kdy by tedy měl manažer přizvat na pomoc kvantitativní metody?

Kvantitativní metody v rozhodování

Ve specifických situacích je samozřejmě možné provádět jen jednu ze zmíněných analýz. Spolehne-li se však pouze na kvalitativní analýzu bez číselných propočtů, závisí výsledek do značné míry na dobrém úsudku manažera. Naopak přílišná důvěra v numerické výsledky může být zavádějící: každé číselné řešení je přesné jen do té míry, jak přesně byl zkonstruován model. Navíc kvantitativní analýza problému může být zdlouhavá a neefektivní v situaci, kdy je třeba přijmout rozhodnutí rychle.

Kdy by tedy měl manažer přizvat na pomoc kvantitativní metody? Zejména, je-li problém:

- **složitý**, kdy specialisté mohou manažerovi pomoci prostřednictvím simulace reality vhodným modelem
- **velmi důležitý**, například jde-li o velké peníze a manažer chce mít pro rozhodování solidní podklady
- **nový**, chybí zkušenosti s řešením obdobných problémů
- **opakovaný**, takže použití ověřených kvantitativních procedur šetří čas i prostředky

Ekonomicko - matematický model

Modelem rozumíme určité zobrazení reálného systému. Nikdy nejde o dokonalý obraz skutečnosti, to ani není žádoucí! Správně zkonstruovaný model musí vystihovat pouze ty vlastnosti, které jsou z hlediska řešení problému důležité. Zahrneme-li do modelu všechny detaily, bude složitý, špatně řešitelný a nepřehledný. Na druhou stranu při přílišné snaze o zjednodušení můžou být opomenuty některé významné skutečnosti a vazby. Při modelování je klíčové právě dobré nastavení vztahu mezi reálným světem a modelem.

Ekonomicko - matematický model

Modelem rozumíme určité zobrazení reálného systému. Nikdy nejde o dokonalý obraz skutečnosti, to ani není žádoucí! Správně zkonstruovaný model musí vystihovat pouze ty vlastnosti, které jsou z hlediska řešení problému důležité. Zahrneme-li do modelu všechny detaily, bude složitý, špatně řešitelný a nepřehledný. Na druhou stranu při přílišné snaze o zjednodušení můžou být opomenuty některé významné skutečnosti a vazby. Při modelování je klíčové právě dobré nastavení vztahu mezi reálným světem a modelem. Manažer by měl umět problém dobře

- **formulovat** tak, aby bylo možné k jeho řešení využít kvantitativních metod, a následně výsledky
- **interpretovat** a implementovat do praxe.

S vlastním řešením matematického modelu mohou pomoci experti či specializovaný software. I při možnosti využití výpočetní techniky je však dobré mít přehled o dostupných metodách, abychom v konkrétní situaci uměli vybrat vhodný algoritmus a nastavit jeho parametry.

Co to je "optimalizace"?

Při optimalizaci řešíme problém výběru "nejlepšího řešení" mezi všemi "možnými řešeními". V každé konkrétní úloze je třeba pojmy uvedené v uvozovkách specifikovat. Všechna možná řešení budeme dále popisovat pomocí množiny M , kterou nazveme **množina přípustných řešení**, a míru kvality řešení budeme vyjadřovat prostřednictvím funkce $f : M \rightarrow \mathbb{R}$, která se označuje jako cílová nebo kriteriální nebo též **účelová funkce**. Zadání optimalizační úlohy pak zní:

Najděte prvek $x^* \in M$ takový, že platí: $f(x^*) \geq f(x), \forall x \in M,$

Co to je "optimalizace"?

Při optimalizaci řešíme problém výběru "nejlepšího řešení" mezi všemi "možnými řešeními". V každé konkrétní úloze je třeba pojmy uvedené v uvozovkách specifikovat. Všechna možná řešení budeme dále popisovat pomocí množiny M , kterou nazveme množina přípustných řešení, a míru kvality řešení budeme vyjadřovat prostřednictvím funkce $f : M \rightarrow \mathbb{R}$, která se označuje jako cílová nebo kriteriální nebo též účelová funkce. Zadání optimalizační úlohy pak zní:

Najděte prvek $x^* \in M$ takový, že platí: $f(x^*) \geq f(x), \forall x \in M$,

Pozn.: Maximalizační úlohu " $f \rightarrow \max$ " lze snadno převést na minimalizační úlohu " $-f \rightarrow \min$ ".

Co to je "optimalizace"?

Při optimalizaci řešíme problém výběru "nejlepšího řešení" mezi všemi "možnými řešeními". V každé konkrétní úloze je třeba pojmy uvedené v uvozovkách specifikovat. Všechna možná řešení budeme dále popisovat pomocí množiny M , kterou nazveme množina přípustných řešení, a míru kvality řešení budeme vyjadřovat prostřednictvím funkce $f : M \rightarrow \mathbb{R}$, která se označuje jako cílová nebo kriteriální nebo též účelová funkce. Zadání optimalizační úlohy pak zní:

Najděte prvek $x^* \in M$ takový, že platí: $f(x^*) \geq f(x), \forall x \in M$,

Pozn.: Maximalizační úlohu " $f \rightarrow \max$ " lze snadno převést na minimalizační úlohu " $-f \rightarrow \min$ ".

Příklady optimalizačních úloh v ekonomii:

Co to je "optimalizace"?

Při optimalizaci řešíme problém výběru "nejlepšího řešení" mezi všemi "možnými řešeními". V každé konkrétní úloze je třeba pojmy uvedené v uvozovkách specifikovat. Všechna možná řešení budeme dále popisovat pomocí množiny M , kterou nazveme množina přípustných řešení, a míru kvality řešení budeme vyjadřovat prostřednictvím funkce $f : M \rightarrow \mathbb{R}$, která se označuje jako cílová nebo kriteriální nebo též účelová funkce. Zadání optimalizační úlohy pak zní:

Najděte prvek $x^* \in M$ takový, že platí: $f(x^*) \geq f(x), \forall x \in M$,

Pozn.: Maximalizační úlohu " $f \rightarrow \max$ " lze snadno převést na minimalizační úlohu " $-f \rightarrow \min$ ".

Příklady optimalizačních úloh v ekonomii:

- Optimalizace výrobního programu
- Optimalizace portfolia
- Optimální rozdělení práce a řazení pracovních operací
- Minimalizace distribučních nákladů, plánování rozvozních tras a umístění distribučních center
- Minimalizace doby realizace při řízení projektů
- Optimální řízení zásob

Co je to "optimalizace"?

Z hlediska přípustné množiny rozlišujeme dva typy optimalizačních úloh:

- Je-li přípustným řešením každý bod x n -rozměrného Euklidova prostoru \mathbb{R}^n , tj. $M = \mathbb{R}^n$, hovoříme o **nepodmíněné optimalizaci**, resp. o volných extrémech. Postup analytického řešení takových úloh je znám ze základního kurzu matematiky
- V opačném případě, tedy je-li $M \subset \mathbb{R}^n$ hovoříme o **vázaných extrémech**. Jejich existenci pro spojitou funkci na omezené uzavřené množině zaručuje Weierstrassova věta.

Možnosti analytického řešení složitějších úloh (např. když je v úloze mnoho proměnných, komplikovaná hranice přípustné množiny nebo dostaneme nelineární rovnice pro určení stacionárního bodu) jsou však omezené. Proto byly vyvinuty speciální metody pro řešení určitých typů optimalizačních úloh.

Matematické programování

Pojem **matematické programování** označuje souhrn metod sloužících k optimalizaci předem definovaného kritéria vyjádřeného jako funkce n proměnných při současném splnění omezujících podmínek zadaných zpravidla ve formě rovností a nerovností. Úlohy matematického programování můžeme rozdělit na úlohy

- **lineárního programování** (dále jen LP), kdy účelová funkce i omezující podmínky jsou lineárními funkcemi proměnných
- **nelineárního programování** (NLP), když výše uvedená podmínka není splněna. Speciálním případem NLP je **kvadratické programování**, kdy účelová funkce je polynom druhého stupně, ale omezující podmínky jsou lineární.

Matematické programování

Pojem **matematické programování** označuje souhrn metod sloužících k optimalizaci předem definovaného kritéria vyjádřeného jako funkce n proměnných při současném splnění omezujících podmínek zadaných zpravidla ve formě rovností a nerovností. Úlohy matematického programování můžeme rozdělit na úlohy

- **lineárního programování** (dále jen LP), kdy účelová funkce i omezující podmínky jsou lineárními funkcemi proměnných
- **nelineárního programování** (NLP), když výše uvedená podmínka není splněna. Speciálním případem NLP je **kvadratické programování**, kdy účelová funkce je polynom druhého stupně, ale omezující podmínky jsou lineární.

Dále se zaměříme hlavně na modely LP, ty jsou jednoznačně nejrozšířenější. Proč? Hodně reálných problémů lze dobře formulovat jako úlohu LP, pro jejich rychlé řešení jsou dostupné programové prostředky, atp. V praxi se sice běžně vyskytují nelineární vztahy (např. **neproporcionalita**: když cena není konstantní, tak příjem není přímo úměrný prodanému množství, **neaditivita**: objem roztoku není roven součtu objemů výchozích látek, apod.), avšak kvůli nepoměrně větší složitosti postupů NLP bývá často výhodnější použít aproximaci lineárním modelem.

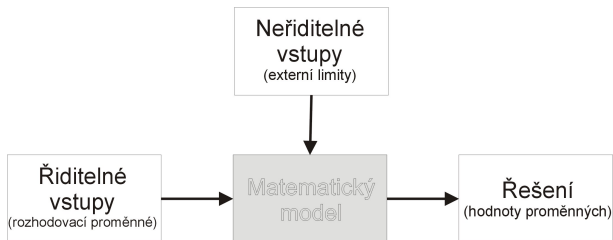
Lineární programování

Při formulaci úlohy matematického programování je třeba vycházet z dobře popsaného ekonomického modelu.

Lineární programování

Při formulaci úlohy matematického programování je třeba vycházet z dobře popsaného ekonomického modelu. Je tedy třeba znát:

- cíl, jehož chceme dosáhnout (tedy zvolit **kritérium**: zisk nebo náklady nebo objem výroby, atd. a určit, zda se jej budeme snažit minimalizovat nebo maximalizovat)
- říditelné vstupy, tj. jaké **proměnné** můžeme ovlivňovat za účelem dosažení cíle (počet vyrobených kusů různých typů produktu, velikost převáženého nákladu, atd.)
- neříditelné vstupy neboli **omezení**, která nás limitují (ceny nakupovaných surovin, dispoziční množství zdrojů, kapacita zařízení, atd.)



Úloha LP - Optimalizace výrobního programu

Veškerý další výklad problematiky lineárního programování bude ilustrován na následující optimalizační úloze převzaté z knihy Josefa Jablonského "Operační výzkum, Kvantitativní modely pro ekonomické rozhodování":

Balírný a pražírny kávy DE, a.s. plánují výrobu dvou směsí *Mocca* a *Standard*. Od dodavatelů mají k dispozici tři druhy kávových bobů K_1 , K_2 a K_3 v kapacitě 40, 60 a 25 tun. Technologický postup určující skladbu směsí shrňme v tabulce.

Komponenta	<i>Mocca</i>	<i>Standard</i>	Kapacita [t]
K_1	0,5	0,25	40
K_2	0,5	0,5	60
K_3		0,25	25

Vzhledem k výrobním nákladům a prodejní ceně směsí byl vykalkulován zisk, který činí 20000 Kč resp. 14000 Kč na jednu tunu směsi *Mocca* resp. *Standard*. Management firmy chce naplánovat produkci tak, aby její zisk byl maximální.

Formulace úlohy optimalizace výrobního programu

Označíme - li x_1 množství tun směsi *Mocca* a x_2 množství tun směsi *Standard*, můžeme problém formulovat matematicky jako úlohu maximalizovat účelovou funkci:

$$z = 20000x_1 + 14000x_2$$

za podmínek

$$0,5x_1 + 0,25x_2 \leq 40$$

$$0,5x_1 + 0,5x_2 \leq 60$$

$$0,25x_2 \leq 25$$

$$x_1, x_2 \geq 0$$

Formulace úlohy optimalizace výrobního programu

Označíme - li x_1 množství tun směsi *Mocca* a x_2 množství tun směsi *Standard*, můžeme problém formulovat matematicky jako úlohu maximalizovat **účelovou funkci**:

$$z = 20000x_1 + 14000x_2$$

za podmínek

$$0,5x_1 + 0,25x_2 \leq 40$$

$$0,5x_1 + 0,5x_2 \leq 60$$

$$0,25x_2 \leq 25$$

$$x_1, x_2 \geq 0$$

Je možný též maticový zápis úlohy:

$$z = \mathbf{c}^T \cdot \mathbf{x} \rightarrow \max \quad \text{za podmínek} \quad \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0},$$

kde $\mathbf{x} = (x_1, x_2)^T$ je vektor strukturních proměnných, $\mathbf{c} = (20, 14)$ je vektor cenových koeficientů v účelové funkci, $\mathbf{b} = (40, 60, 25)^T$ je vektor kapacitních

omezení a $\mathbf{A} = \begin{pmatrix} 0,5 & 0,25 \\ 0,5 & 0,5 \\ 0 & 0,25 \end{pmatrix}$ je matice strukturních koeficientů.

Matematická formulace obecné úlohy LP

Obecnou úlohu LP pro n proměnných a m omezení můžeme zapsat takto:
minimalizuj (maximalizuj) funkci

$$z = \sum_{j=1}^n c_j x_j$$

za podmínek

$$\sum_{j=1}^n a_{ij} x_j \text{ ? } b_i, \quad i = 1, \dots, m$$

$$x_j \geq 0, \quad j = 1, \dots, n,$$

kde na místě symbolů ? mohou být libovolná relační znaménka $\leq, =, \geq$.
Omezení se uvádějí v takové podobě, aby pravé strany b_i byly nezáporné.

Matematická formulace obecné úlohy LP

Obecnou úlohu LP pro n proměnných a m omezení můžeme zapsat takto: minimalizuj (maximalizuj) funkci

$$z = \sum_{j=1}^n c_j x_j$$

za podmínek

$$\sum_{j=1}^n a_{ij} x_j \ ? \ b_i, \ i = 1, \dots, m$$

$$x_j \geq 0, \ j = 1, \dots, n,$$

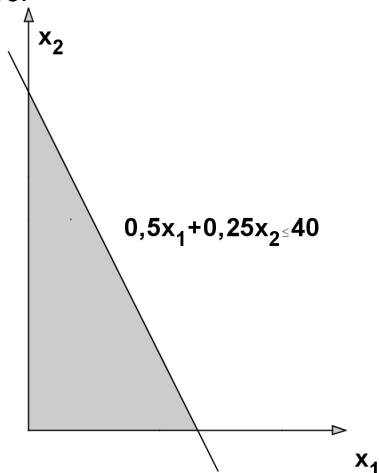
kde na místě symbolů ? mohou být libovolná relační znaménka $\leq, =, \geq$.
Omezení se uvádějí v takové podobě, aby pravé strany b_i byly nezáporné.

Je dobré si uvědomit, že jednu úlohu lze formulovat různými způsoby. Snadno lze převést úlohu minimalizační na úlohu maximalizace funkce

$-z = \sum_{j=1}^n (-c_j) x_j$. Omezení ve formě rovnosti lze přepsat jako dvě nerovnice typu \leq a \geq s týmiž koeficienty i pravou stranou jako původní rovnice. Převod omezení ve formě nerovnosti na rovnici se zase řešení zavedením dodatečných proměnných, jak si dále ukážeme.

Grafické řešení úlohy LP

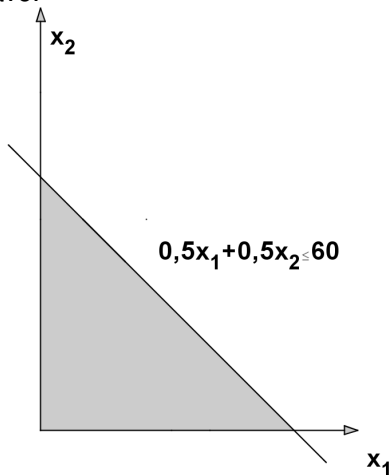
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Kvůli nezápornosti proměnných se omezíme pouze na první kvadrant. Znázorníme zde polorovinu tvořenou body splňujícími první omezující podmínku.

Grafické řešení úlohy LP

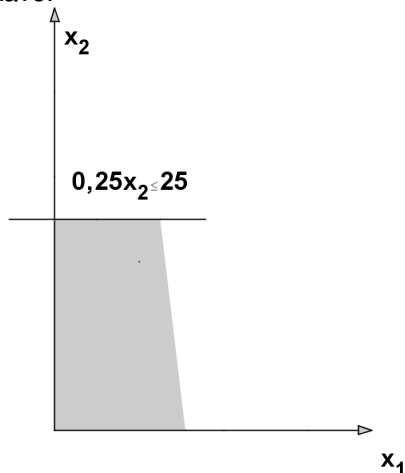
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Znázorníme také polorovinu tvořenou body splňujícími druhou omezující podmínku.

Grafické řešení úlohy LP

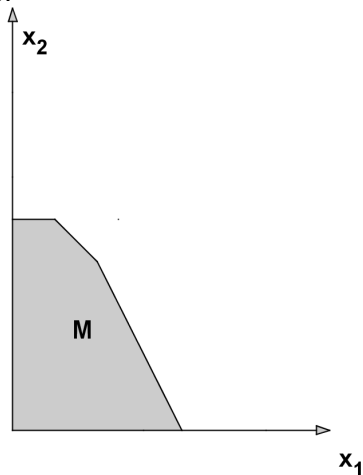
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Znázorníme ještě polorovinu tvořenou body splňujícími třetí omezující podmínku.

Grafické řešení úlohy LP

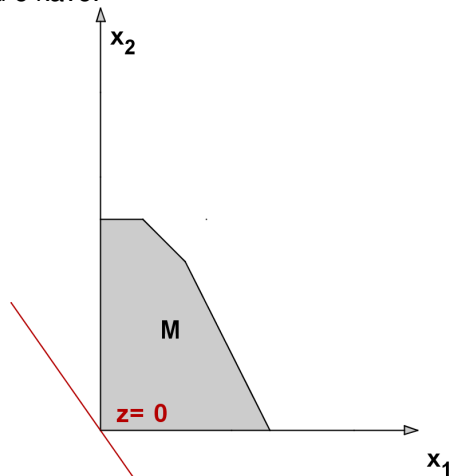
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Množina přípustných řešení M je tvořena body, které vyhovují všem omezením.

Grafické řešení úlohy LP

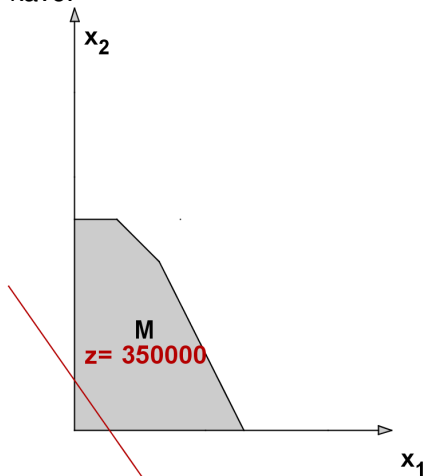
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Izokvanta účelové funkce $z = 20000x_1 + 14000x_2 = 0$

Grafické řešení úlohy LP

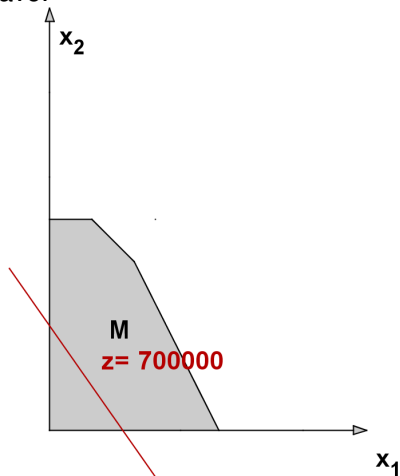
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Izokvanta účelové funkce $z = 20000x_1 + 14000x_2 = 350000$

Grafické řešení úlohy LP

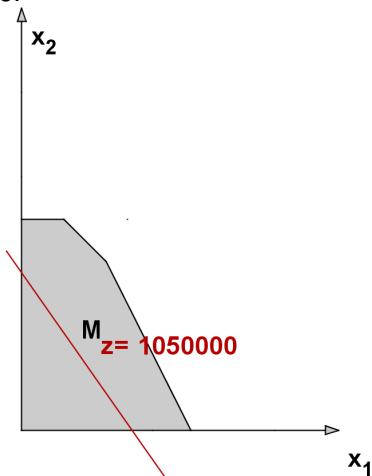
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Izokvanta účelové funkce $z = 20000x_1 + 14000x_2 = 700000$

Grafické řešení úlohy LP

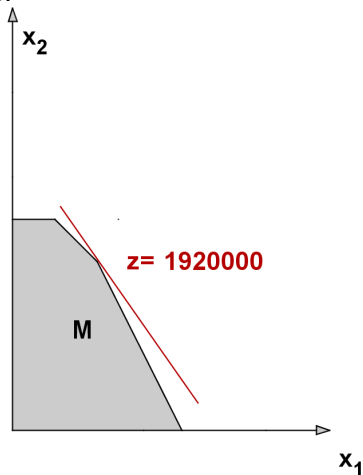
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Izokvanta účelové funkce $z = 20000x_1 + 14000x_2 = 1050000$

Grafické řešení úlohy LP

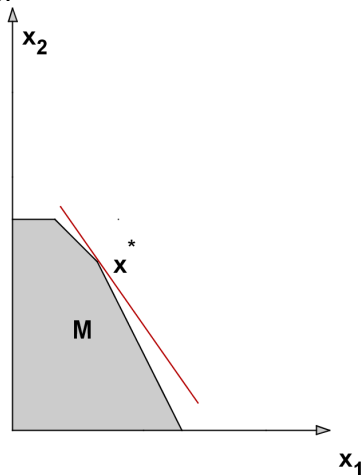
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Izokvanta účelové funkce $z = 20000x_1 + 14000x_2 = 1920000$

Grafické řešení úlohy LP

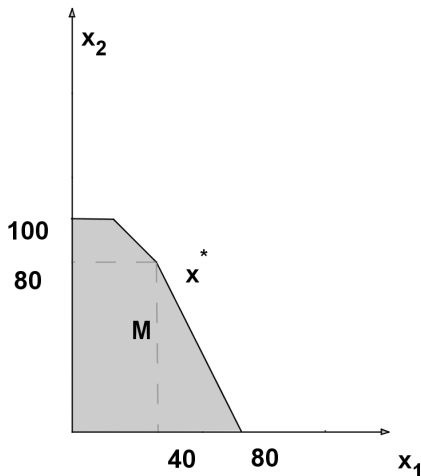
Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Nejvyšší izokvanta se dotýká množiny M v bodě x^*

Grafické řešení úlohy LP

Úlohy obsahující pouze dvě proměnné lze řešit graficky. Ukažme si postup pro naši úlohu o kávě.



Bod $x^* = [40, 80]$ je optimálním řešením.

Základní věta lineárního programování

Přípustná množina M je vymezena **obligátními** podmínkami (nezápornost) a omezujícími podmínkami $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. Ty lze vyjádřit pomocí rovností:

$$0,5x_1 + 0,25x_2 + x_3 = 40$$

$$0,5x_1 + 0,5x_2 + x_4 = 60$$

$$0,25x_2 + x_5 = 25$$

Základní věta lineárního programování

Přípustná množina M je vymezena **obligátními** podmínkami (nezápornost) a omezujícími podmínkami $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. Ty lze vyjádřit pomocí rovností:

$$0,5x_1 + 0,25x_2 + x_3 = 40$$

$$0,5x_1 + 0,5x_2 + x_4 = 60$$

$$0,25x_2 + x_5 = 25$$

Proměnné x_3 , x_4 , x_5 označujeme jako **přídavné** a lze je ekonomicky interpretovat jako nevyužitou kapacitu jednotlivých surovin.

Základní věta lineárního programování

Přípustná množina M je vymezena **obligátními** podmínkami (nezápornost) a omezujícími podmínkami $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. Ty lze vyjádřit pomocí rovností:

$$0,5x_1 + 0,25x_2 + x_3 = 40$$

$$0,5x_1 + 0,5x_2 + x_4 = 60$$

$$0,25x_2 + x_5 = 25$$

Proměnné x_3 , x_4 , x_5 označujeme jako **přídavné** a lze je ekonomicky interpretovat jako nevyužitou kapacitu jednotlivých surovin. Soustava obsahuje m rovnic pro $m + n$ proměnných, může mít obecně nekonečně mnoho řešení. Takové řešení soustavy, pro které je n proměnných rovno nule, nazýváme **základní** (Ve 2D odpovídají základní řešení průsečíkům hraničních přímk jednotlivých nerovností.) Nenulové proměnné pak označujeme jako základní, nulové jako nezákladní.

Základní věta lineárního programování

Přípustná množina M je vymezena **obligátními** podmínkami (nezápornost) a omezujícími podmínkami $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. Ty lze vyjádřit pomocí rovností:

$$0,5x_1 + 0,25x_2 + x_3 = 40$$

$$0,5x_1 + 0,5x_2 + x_4 = 60$$

$$0,25x_2 + x_5 = 25$$

Proměnné x_3 , x_4 , x_5 označujeme jako **přídavné** a lze je ekonomicky interpretovat jako nevyužitou kapacitu jednotlivých surovin. Soustava obsahuje m rovnic pro $m + n$ proměnných, může mít obecně nekonečně mnoho řešení. Takové řešení soustavy, pro které je n proměnných rovno nule, nazýváme **základní** (Ve 2D odpovídají základní řešení průsečíkům hraničních přímk jednotlivých nerovností.) Nenulové proměnné pak označujeme jako základní, nulové jako nezákladní. **Pozor!** Ne každé základní řešení je přípustné. Přípustná základní řešení odpovídají krajním bodům M . V našem příkladě je $m = 3$, $n = 2$; celkem dostaneme ? základních řešení, z toho ? přípustných.

Základní věta lineárního programování

Přípustná množina M je vymezena **obligátními** podmínkami (nezápornost) a omezujícími podmínkami $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. Ty lze vyjádřit pomocí rovností:

$$0,5x_1 + 0,25x_2 + x_3 = 40$$

$$0,5x_1 + 0,5x_2 + x_4 = 60$$

$$0,25x_2 + x_5 = 25$$

Proměnné x_3 , x_4 , x_5 označujeme jako **přídavné** a lze je ekonomicky interpretovat jako nevyužitou kapacitu jednotlivých surovin. Soustava obsahuje m rovnic pro $m + n$ proměnných, může mít obecně nekonečně mnoho řešení. Takové řešení soustavy, pro které je n proměnných rovno nule, nazýváme **základní** (Ve 2D odpovídají základní řešení průsečíkům hraničních přímk jednotlivých nerovností.) Nenulové proměnné pak označujeme jako základní, nulové jako nezákladní. **Pozor!** Ne každé základní řešení je přípustné. Přípustná základní řešení odpovídají krajním bodům M . V našem příkladě je $m = 3$, $n = 2$; celkem dostaneme ? základních řešení, z toho ? přípustných.

Hlavní věta lineárního programování:

Jestliže má úloha optimální řešení, pak má také optimální základní řešení.

Simplexová tabulka

Uvedenou soustavu rovnic můžeme zapsat maticově jako

$(\mathbf{A}, \mathbf{I}) \cdot (x_1, x_2, x_3, x_4, x_5)^\top = \mathbf{b}$, kde \mathbf{I} je jednotková matice řádu $m = 3$.

Každou takovou soustavu m rovnic pro $m + n$ neznámých, kde matice levé strany obsahuje všechny sloupce jednotkové matice řádu m , nazveme soustavou v **kanonickém tvaru**.

Simplexová tabulka

Uvedenou soustavu rovnic můžeme zapsat maticově jako

$(\mathbf{A}, \mathbf{I}) \cdot (x_1, x_2, x_3, x_4, x_5)^T = \mathbf{b}$, kde \mathbf{I} je jednotková matice řádu $m = 3$.

Každou takovou soustavu m rovnic pro $m + n$ neznámých, kde matice levé strany obsahuje všechny sloupce jednotkové matice řádu m , nazveme soustavou v **kanonickém tvaru**. Snadno vidíme jedno z řešení takové soustavy: $x_1 = 0$, $x_2 = 0$, $x_3 = b_1 = 40$, $x_4 = b_2 = 60$, $x_5 = b_3 = 25$, jde dokonce o řešení základní. Znázorníme vše do přehledné tabulky:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	b_i
x_3	$\frac{1}{2}$	$\frac{1}{4}$	1	0	0	40
x_4	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	60
x_5	0	$\frac{1}{4}$	0	0	1	25
Z_j	-20	-14	0	0	0	0

Simplexová tabulka

Uvedenou soustavu rovnic můžeme zapsat maticově jako

$(\mathbf{A}, \mathbf{I}) \cdot (x_1, x_2, x_3, x_4, x_5)^T = \mathbf{b}$, kde \mathbf{I} je jednotková matice řádu $m = 3$.

Každou takovou soustavu m rovnic pro $m + n$ neznámých, kde matice levé strany obsahuje všechny sloupce jednotkové matice řádu m , nazveme soustavou v **kanonickém tvaru**. Snadno vidíme jedno z řešení takové soustavy: $x_1 = 0$, $x_2 = 0$, $x_3 = b_1 = 40$, $x_4 = b_2 = 60$, $x_5 = b_3 = 25$, jde dokonce o řešení základní. Znázorníme vše do přehledné tabulky:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	b_i
x_3	$\frac{1}{2}$	$\frac{1}{4}$	1	0	0	40
x_4	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	60
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	-20	-14	0	0	0	0

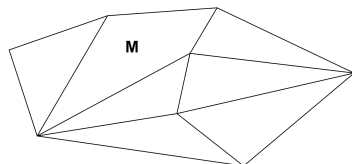
Poslední řádek odpovídá účelové funkci v tzv. **anulovaném tvaru**, původní vyjádření $z = 20x_1 + 14x_2$ [v tis. Kč] jsme převedli na tvar $z - 20x_1 - 14x_2 - 0x_3 - 0x_4 - 0x_5$, pro výchozí základní řešení dostaneme hodnotu účelové funkce $z = 0$, viz pravý dolní roh tabulky. Uvedené schéma nazveme výchozí **simplexovou tabulkou** úlohy.

Simplexová metoda

Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přídatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje.

Simplexová metoda

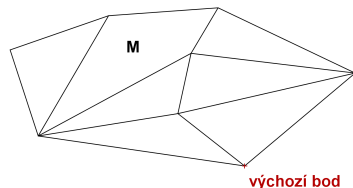
Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přidatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje. Na obrázku ukažme schematické znázornění postupu ve 3D.



Množina přípustných řešení

Simplexová metoda

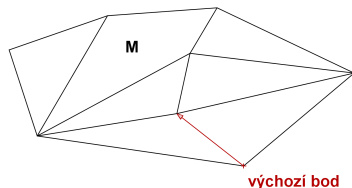
Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přidatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje. Na obrázku ukažme schematické znázornění postupu ve 3D.



Výchozí základní řešení

Simplexová metoda

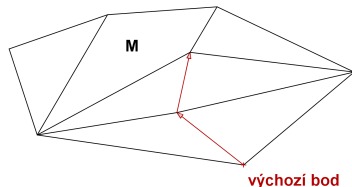
Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přidatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje. Na obrázku ukažme schematické znázornění postupu ve 3D.



Přesuneme se do sousedního bodu s lepší hodnotou účelové funkce

Simplexová metoda

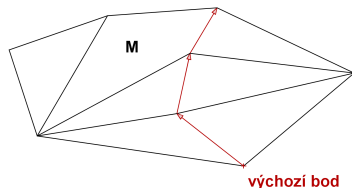
Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přidatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje. Na obrázku ukažme schematické znázornění postupu ve 3D.



Přesuneme se do sousedního bodu s ještě lepší hodnotou účelové funkce

Simplexová metoda

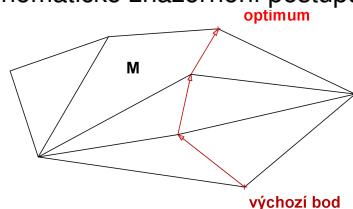
Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přidatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje. Na obrázku ukažme schematické znázornění postupu ve 3D.



Zase se přesuneme do sousedního bodu s lepší hodnotou účelové funkce

Simplexová metoda

Simplexová metoda je iterační postup k nalezení optimálního řešení úlohy LP. Úvodním krokem je nalezení výchozího základního řešení. U úloh obsahujících pouze nerovnice typu " \leq " je tento krok díky přídatným proměnným jednoduchý, u jiných typů úloh jej získáme řešením počáteční úlohy minimalizace pomocných proměnných vyjadřujících porušení omezujících podmínek, hovoříme pak o **dvoufázové simplexové metodě**. Dále metoda v jednotlivých krocích vypočte nové základní řešení s lepší hodnotou účelové funkce. Po konečném počtu kroků se nalezne řešení s nejlepší hodnotou účelové funkce (podle základní věty LP jde pak o optimální řešení celé úlohy) nebo se zjistí, že takové řešení neexistuje. Na obrázku ukažme schematické znázornění postupu ve 3D.



Nelze se přesunout do žádného lepšího bodu, byl nalezen bod optima

Iterační krok simplexové metody

Čísla z_j v spodním řádku simplexové tabulky nazýváme **redukované ceny**. Ukazují, jak se změní účelová funkce při přechodu k novému základnímu řešení. Stane-li se nezákladní proměnná x_k proměnnou základní, tj. změní-li hodnotu z 0 na $t > 0$, bude přírůstek účelové funkce $\Delta z = -t \cdot z_k$.

Iterační krok simplexové metody

Čísla z_j v spodním řádku simplexové tabulky nazýváme **redukované ceny**. Ukazují, jak se změní účelová funkce při přechodu k novému základnímu řešení. Stane-li se nezákladní proměnná x_k proměnnou základní, tj. změní-li hodnotu z 0 na $t > 0$, bude přírůstek účelové funkce $\Delta z = -t \cdot z_k$. Při maximalizaci chceme, aby toto Δz bylo kladné, tj. aby $z_k < 0$. Pokud jsou **všechny redukované ceny nezáporné**, již nejde zvýšit hodnotu účelové funkce, řešení je **optimální**.

Iterační krok simplexové metody

Čísla z_j v spodním řádku simplexové tabulky nazýváme **redukované ceny**. Ukazují, jak se změní účelová funkce při přechodu k novému základnímu řešení. Stane-li se nezákladní proměnná x_k proměnnou základní, tj. změní-li hodnotu z 0 na $t > 0$, bude přírůstek účelové funkce $\Delta z = -t \cdot z_k$. Při maximalizaci chceme, aby toto Δz bylo kladné, tj. aby $z_k < 0$. Pokud jsou **všechny redukované ceny nezáporné**, již nejde zvýšit hodnotu účelové funkce, řešení je **optimální**. Jinak vybereme x_k , pro které je z_k nejmenší (říkáme mu **vstupující** proměnná), a nahradíme s ním některou základní (**vystupující**) proměnnou.

Iterační krok simplexové metody

Čísla z_j v spodním řádku simplexové tabulky nazýváme **redukované ceny**. Ukazují, jak se změní účelová funkce při přechodu k novému základnímu řešení. Stane-li se nezákladní proměnná x_k proměnnou základní, tj. změní-li hodnotu z 0 na $t > 0$, bude přírůstek účelové funkce $\Delta z = -t \cdot z_k$. Při maximalizaci chceme, aby toto Δz bylo kladné, tj. aby $z_k < 0$. Pokud jsou **všechny redukované ceny nezáporné**, již nejde zvýšit hodnotu účelové funkce, řešení je **optimální**. Jinak vybereme x_k , pro které je z_k nejmenší (říkáme mu **vstupující** proměnná), a nahradíme s ním některou základní (**vystupující**) proměnnou. Tedy pro naši simplexovou tabulku

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_3	$\frac{1}{2}$	$\frac{1}{4}$	1	0	0	40
x_4	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	60
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	-20	-14	0	0	0	0

bude vstupující proměnnou x_1 , protože **-20** je nejmenší hodnota na posledním řádku.

Iterační krok simplexové metody

Volba vystupující proměnné vychází z nutnosti zachovat přípustnost řešení, tedy nezápornost všech základních proměnných. Zapišme tuto podmínku pro nové hodnoty původních základních proměnných x_3 , x_4 , x_5 , jestliže nově

$x_1 = t$. Z platnosti rovnic:

$$x_3 = 40 - \frac{1}{2}t \geq 0$$

$$x_4 = 60 - \frac{1}{2}t \geq 0$$

$$x_5 = 25 - 0 \geq 0$$

Zřejmě největší možné takové t je $t = 80$, pro ně dostaneme $x_3 = 0$. To se nyní stane vystupující proměnnou. Tabulku přepočteme elementárními úpravami tak, abychom vlevo nahoře dostali jedničku a pod ní samé nuly.

Iterační krok simplexové metody

Volba vystupující proměnné vychází z nutnosti zachovat přípustnost řešení, tedy nezápornost všech základních proměnných. Zapišme tuto podmínku pro nové hodnoty původních základních proměnných x_3 , x_4 , x_5 , jestliže nově

$x_1 = t$. Z platnosti rovnic:

$$x_3 = 40 - \frac{1}{2}t \geq 0$$

$$x_4 = 60 - \frac{1}{2}t \geq 0$$

$$x_5 = 25 - 0 \geq 0$$

Zřejmě největší možné takové t je $t = 80$, pro ně dostaneme $x_3 = 0$. To se nyní stane vystupující proměnnou. Tabulku přepočteme elementárními úpravami tak, abychom vlevo nahoře dostali jedničku a pod ní samé nuly.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_3	$\frac{1}{2}$	$\frac{1}{4}$	1	0	0	40
x_4	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	60
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	-20	-14	0	0	0	0

První řádek vynásobíme dvojkou

Iterační krok simplexové metody

Volba vystupující proměnné vychází z nutnosti zachovat přípustnost řešení, tedy nezápornost všech základních proměnných. Zapišme tuto podmínku pro nové hodnoty původních základních proměnných x_3 , x_4 , x_5 , jestliže nově

$x_1 = t$. Z platnosti rovnic:

$$x_3 = 40 - \frac{1}{2}t \geq 0$$

$$x_4 = 60 - \frac{1}{2}t \geq 0$$

$$x_5 = 25 - 0 \geq 0$$

Zřejmě největší možné takové t je $t = 80$, pro ně dostaneme $x_3 = 0$. To se nyní stane vystupující proměnnou. Tabulku přepočteme elementárními úpravami tak, abychom vlevo nahoře dostali jedničku a pod ní samé nuly.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	$\frac{1}{2}$	$\frac{1}{2}$	0	1	0	60
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	-20	-14	0	0	0	0

Odečteme jeho polovinu od druhého

Iterační krok simplexové metody

Volba vystupující proměnné vychází z nutnosti zachovat přípustnost řešení, tedy nezápornost všech základních proměnných. Zapišme tuto podmínku pro nové hodnoty původních základních proměnných x_3 , x_4 , x_5 , jestliže nově

$x_1 = t$. Z platnosti rovnic:

$$x_3 = 40 - \frac{1}{2}t \geq 0$$

$$x_4 = 60 - \frac{1}{2}t \geq 0$$

$$x_5 = 25 - 0 \geq 0$$

Zřejmě největší možné takové t je $t = 80$, pro ně dostaneme $x_3 = 0$. To se nyní stane vystupující proměnnou. Tabulku přepočteme elementárními úpravami tak, abychom vlevo nahoře dostali jedničku a pod ní samé nuly.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	-20	-14	0	0	0	0

Nakonec přičteme jeho dvacetinásobek k poslednímu.

Iterační krok simplexové metody

Volba vystupující proměnné vychází z nutnosti zachovat přípustnost řešení, tedy nezápornost všech základních proměnných. Zapišme tuto podmínku pro nové hodnoty původních základních proměnných x_3 , x_4 , x_5 , jestliže nově

$x_1 = t$. Z platnosti rovnic:

$$x_3 = 40 - \frac{1}{2}t \geq 0$$

$$x_4 = 60 - \frac{1}{2}t \geq 0$$

$$x_5 = 25 - 0 \geq 0$$

Zřejmě největší možné takové t je $t = 80$, pro ně dostaneme $x_3 = 0$. To se nyní stane vystupující proměnnou. Tabulku přepočteme elementárními úpravami tak, abychom vlevo nahoře dostali jedničku a pod ní samé nuly.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Dostali jsme novou tabulku.

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Nejvyšší hodnotou t , pro kterou můžeme položit $x_2 = t$, je $\min\{2.80, 4.20, 4.25\} = 80$. Vyjdou nám pak nové hodnoty základních proměnných $x_1 = 80 - \frac{1}{2}t = 40$, $x_4 = 20 - \frac{1}{4}t = 0$, $x_5 = 25 - \frac{1}{4}t = 5$. Tedy x_4 se stane nezákladní, je vystupující proměnnou.

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Nejvyšší hodnotou t , pro kterou můžeme položit $x_2 = t$, je $\min\{2 \cdot 80, 4 \cdot 20, 4 \cdot 25\} = 80$. Vyjdou nám pak nové hodnoty základních proměnných $x_1 = 80 - \frac{1}{2}t = 40$, $x_4 = 20 - \frac{1}{4}t = 0$, $x_5 = 25 - \frac{1}{4}t = 5$. Tedy x_4 se stane nezákladní, je vystupující proměnnou.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Vynásobíme druhý řádek 4.

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Nejvyšší hodnotou t , pro kterou můžeme položit $x_2 = t$, je $\min\{2 \cdot 80, 4 \cdot 20, 4 \cdot 25\} = 80$. Vyjdou nám pak nové hodnoty základních proměnných $x_1 = 80 - \frac{1}{2}t = 40$, $x_4 = 20 - \frac{1}{4}t = 0$, $x_5 = 25 - \frac{1}{4}t = 5$. Tedy x_4 se stane nezákladní, je vystupující proměnnou.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_2	0	1	-4	4	0	80
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Odečteme jeho polovinu od prvního.

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Nejvyšší hodnotou t , pro kterou můžeme položit $x_2 = t$, je $\min\{2 \cdot 80, 4 \cdot 20, 4 \cdot 25\} = 80$. Vyjdou nám pak nové hodnoty základních proměnných $x_1 = 80 - \frac{1}{2}t = 40$, $x_4 = 20 - \frac{1}{4}t = 0$, $x_5 = 25 - \frac{1}{4}t = 5$. Tedy x_4 se stane nezákladní, je vystupující proměnnou.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	4	0	0	40
x_2	0	1	-4	4	0	80
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Odečteme jeho čtvrtinu od třetího.

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Nejvyšší hodnotou t , pro kterou můžeme položit $x_2 = t$, je $\min\{2 \cdot 80, 4 \cdot 20, 4 \cdot 25\} = 80$. Vyjdou nám pak nové hodnoty základních proměnných $x_1 = 80 - \frac{1}{2}t = 40$, $x_4 = 20 - \frac{1}{4}t = 0$, $x_5 = 25 - \frac{1}{4}t = 5$. Tedy x_4 se stane nezákladní, je vystupující proměnnou.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	4	0	0	40
x_2	0	1	-4	4	0	80
x_5	0	0	1	-1	1	5
z_j	0	-4	40	0	0	1600

Nakonec přičteme jeho čtyřnásobek k poslednímu.

Další iterační krok

Redukovaná cena $z_2 = -4$ naznačuje, že lze ještě zvýšit účelovou funkci, jestliže zvolíme x_2 jako vstupující.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	$\frac{1}{2}$	2	0	0	80
x_4	0	$\frac{1}{4}$	-1	1	0	20
x_5	0	$\frac{1}{4}$	0	0	1	25
z_j	0	-4	40	0	0	1600

Nejvyšší hodnotou t , pro kterou můžeme položit $x_2 = t$, je $\min\{2 \cdot 80, 4 \cdot 20, 4 \cdot 25\} = 80$. Vyjdou nám pak nové hodnoty základních proměnných $x_1 = 80 - \frac{1}{2}t = 40$, $x_4 = 20 - \frac{1}{4}t = 0$, $x_5 = 25 - \frac{1}{4}t = 5$. Tedy x_4 se stane nezákladní, je vystupující proměnnou.

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	4	0	0	40
x_2	0	1	-4	4	0	80
x_5	0	0	1	-1	1	5
z_j	0	0	24	16	0	1920

Dostali jsme novou tabulku.

Ukončení výpočtu

Ve výsledné tabulce jsou již všechny redukované ceny nezáporné:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_j
x_1	1	0	4	0	0	40
x_2	0	1	-4	4	0	80
x_5	0	0	1	-1	1	5
z_j	0	0	24	16	0	1920

Nelze tedy již zvýšit hodnotu účelové funkce, maximální zisk je 1920 tisíc. Nezákladní proměnné jsou x_3 , x_4 , ty budou tedy nulové. Hodnoty základních proměnných vyčteme z tabulky: $x_1 = 40$, $x_2 = 80$, $x_5 = 5$. To nám říká, že optimálně máme vyrobit 40 tun směsi *Mocca* a 80 tun směsi *Standard*, přičemž zcela spotřebujeme první dvě suroviny a zbyde nám pět tun třetí suroviny.

Ukončení výpočtu

Ve výsledné tabulce jsou již všechny redukované ceny nezáporné:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_j
x_1	1	0	4	0	0	40
x_2	0	1	-4	4	0	80
x_5	0	0	1	-1	1	5
z_j	0	0	24	16	0	1920

Nelze tedy již zvýšit hodnotu účelové funkce, maximální zisk je 1920 tisíc. Nezákladní proměnné jsou x_3 , x_4 , ty budou tedy nulové. Hodnoty základních proměnných vyčteme z tabulky: $x_1 = 40$, $x_2 = 80$, $x_5 = 5$. To nám říká, že optimálně máme vyrobit 40 tun směsi *Mocca* a 80 tun směsi *Standard*, přičemž zcela spotřebujeme první dvě suroviny a zbyde nám pět tun třetí suroviny.

Pozn.: Pokud by dole zbyla záporná redukovaná cena, ale ve sloupci nad ní by už žádné číslo nebylo ≥ 0 , tak je úloha neomezená. Pro minimalizační úlohu by se obrátila role znamének v dolním řádku - vybírali bychom vstupující proměnnou podle nejvyšší redukované ceny a výpočet bychom ukončili až by všechny redukované ceny byly ≤ 0 .

Dvofázová simplexová metoda

Vyskytují-li se v úloze i jiná omezení než nerovnosti typu " \leq ", je nutné nejprve najít výchozí přípustné základní řešení. K tomu slouží **první fáze** simplexové metody. Ukažme si ji na ilustračním příkladě.

$$z = x_1 - x_2 \rightarrow \min$$

za podmínek

$$\begin{array}{rcl} & x_1 & \geq 2 \\ & x_2 & \geq 2 \\ 5x_1 & + & 10x_2 \leq 50 \end{array}$$

Dvofázová simplexová metoda

Vyskytují-li se v úloze i jiná omezení než nerovnosti typu " \leq ", je nutné nejprve najít výchozí přípustné základní řešení. K tomu slouží **první fáze** simplexové metody. Ukažme si ji na ilustračním příkladě.

$$z = x_1 - x_2 \rightarrow \min$$

za podmínek

$$\begin{array}{rcl} x_1 & \geq & 2 \\ x_2 & \geq & 2 \\ 5x_1 + 10x_2 & \leq & 50 \end{array}$$

Omezující podmínky lze opět zavedením nezáporných přídatných proměnných převést na rovnosti:

$$\begin{array}{rclclcl} x_1 & -x_3 & & & & = & 2 \\ x_2 & & -x_4 & & & = & 2 \\ 5x_1 + 10x_2 & & & +x_5 & & = & 50 \end{array}$$

Dvofázová simplexová metoda

Vyskytují-li se v úloze i jiná omezení než nerovnosti typu " \leq ", je nutné nejprve najít výchozí přípustné základní řešení. K tomu slouží **první fáze** simplexové metody. Ukažme si ji na ilustračním příkladě.

$$z = x_1 - x_2 \rightarrow \min$$

za podmínek

$$\begin{array}{rcl} x_1 & \geq & 2 \\ x_2 & \geq & 2 \\ 5x_1 + 10x_2 & \leq & 50 \end{array}$$

Omezující podmínky lze opět zavedením nezáporných přídatných proměnných převést na rovnosti:

$$\begin{array}{rclcl} x_1 & -x_3 & & & = 2 \\ x_2 & & -x_4 & & = 2 \\ 5x_1 + 10x_2 & & & +x_5 & = 50 \end{array}$$

Bohužel nejde o soustavu v kanonickém tvaru, protože koeficienty u x_3 a x_4 nejsou $= 1$. Proto přičteme k levým stranám příslušných omezení ještě nezáporné **pomocné proměnné** y_1, y_2 a tyto proměnné již budou spolu s x_5 základními. Pro výchozí bod platí $y_1 = y_2 = 2, x_5 = 50$, což však není přípustné pro původní úlohu.

Dvoufázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádříme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

Dvofázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádřeme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

zákl. prom.	x_1	x_2	x_3	x_4	x_5	y_1	y_2	β_i
y_1	1	0	-1	0	0	1	0	2
y_2	0	1	0	-1	0	0	1	2
x_5	5	10	0	0	1	0	0	50
z'_j	1	1	-1	-1	0	0	0	4

Jako vstupující proměnnou můžeme zvolit x_1 nebo x_2 , zvolme např. tu druhou.

Dvoufázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádřeme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

zákl. prom.	x_1	x_2	x_3	x_4	x_5	y_1	y_2	β_i
y_1	1	0	-1	0	0	1	0	2
y_2	0	1	0	-1	0	0	1	2
x_5	5	10	0	0	1	0	0	50
z'_j	1	1	-1	-1	0	0	0	4

Od třetího řádku odečteme desetinasobek druhého.

Dvoufázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádřeme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

zákl. prom.	x_1	x_2	x_3	x_4	x_5	y_1	y_2	β_i
y_1	1	0	-1	0	0	1	0	2
y_2	0	1	0	-1	0	0	1	2
x_5	5	0	0	10	1	0	-10	30
z'_j	1	1	-1	-1	0	0	0	4

Od čtvrtého řádku odečteme druhý.

Dvofázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádřeme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

zákl. prom.	x_1	x_2	x_3	x_4	x_5	y_1	y_2	β_i
y_1	1	0	-1	0	0	1	0	2
x_2	0	1	0	-1	0	0	1	2
x_5	5	0	0	10	1	0	-10	30
z'_j	1	0	-1	0	0	0	-1	2

Získali jsme novou tabulku, vstupující proměnnou bude x_1 . Od třetího řádku odečteme pětkrát první.

Dvofázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádřeme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

zákl. prom.	x_1	x_2	x_3	x_4	x_5	y_1	y_2	β_i
y_1	1	0	-1	0	0	1	0	2
x_2	0	1	0	-1	0	0	1	2
x_5	0	0	5	10	1	-5	-10	20
z'_j	1	0	-1	0	0	0	-1	2

Od posledního řádku odečteme první.

Dvoufázová simplexová metoda

K získání přípustného řešení původní úlohy je třeba zajistit, aby $y_1 = y_2 = 0$. To lze pomocí minimalizace pomocné účelové funkce $z' = y_1 + y_2$. (jestliže má tato funkce minimum >0 , pak výchozí úloha nemá žádné přípustné řešení). Vyjádřeme z' pomocí nezákladních proměnných a výsledné redukované ceny zapišme do simplexové tabulky:

$$z' = (2 - x_1 + x_3) + (2 - x_2 + x_4) = 4 - x_1 - x_2 + x_3 + x_4$$

zákl. prom.	x_1	x_2	x_3	x_4	x_5	y_1	y_2	β_i
x_1	1	0	-1	0	0	1	0	2
x_2	0	1	0	-1	0	0	1	2
x_5	0	0	5	10	1	-5	-10	20
z'_j	0	0	0	0	0	-1	-1	0

Nalezli jsme minimum pomocné fce $z' = 0$, můžeme tedy zahájit **2. fázi**: vynecháme y_1, y_2 a z bodu $[2, 2, 0, 0, 20]$ minimalizujeme funkci $z = x_1 - x_2 = (2 - x_3) - (2 - x_4) = -x_3 + x_4$, jejíž redukované ceny přidáme do tabulky.

Pozn.: kdyby vyšlo $z'_{opt} \neq 0$, nemá výchozí úloha žádné přípustné řešení.

Dvoufázová simplexová metoda

Druhou fázi již dořešíme běžným způsobem:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	-1	0	0	2
x_2	0	1	0	-1	0	2
x_5	0	0	5	10	1	20
z_j	0	0	-1	1	0	0

Jako vstupující volíme x_4 .

Dvufázová simplexová metoda

Druhou fázi již dořešíme běžným způsobem:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	-1	0	0	2
x_2	0	1	0	-1	0	2
x_5	0	0	5	10	1	20
z_j	0	0	-1	1	0	0

Třetí řádek vydělíme 10.

Dvufázová simplexová metoda

Druhou fázi již dořešíme běžným způsobem:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	-1	0	0	2
x_2	0	1	0	-1	0	2
x_5	0	0	$\frac{1}{2}$	1	$\frac{1}{10}$	2
z_j	0	0	-1	1	0	0

Přičteme třetí k druhému.

Dvufázová simplexová metoda

Druhou fázi již dořešíme běžným způsobem:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	-1	0	0	2
x_2	0	1	$\frac{1}{2}$	0	$\frac{1}{10}$	4
x_5	0	0	$\frac{1}{2}$	1	$\frac{1}{10}$	2
z_j	0	0	-1	1	0	0

Odečteme třetí od posledního.

Dvoufázová simplexová metoda

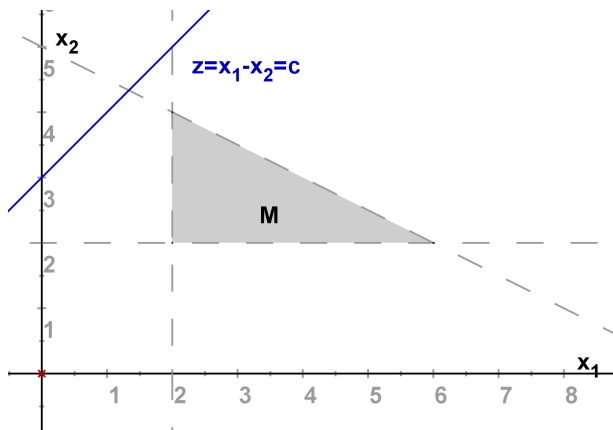
Druhou fázi již dořešíme běžným způsobem:

zákl. prom.	x_1	x_2	x_3	x_4	x_5	β_i
x_1	1	0	-1	0	0	2
x_2	0	1	$\frac{1}{2}$	0	$\frac{1}{10}$	4
x_4	0	0	$\frac{1}{2}$	1	$\frac{1}{10}$	2
z_j	0	0	$-\frac{3}{2}$	0	$-\frac{1}{10}$	-2

Dostali jsme optimální tabulku, je tedy $x_1 = 2$, $x_2 = 4$, $z_{opt} = -2$.

Dvoufázová simplexová metoda

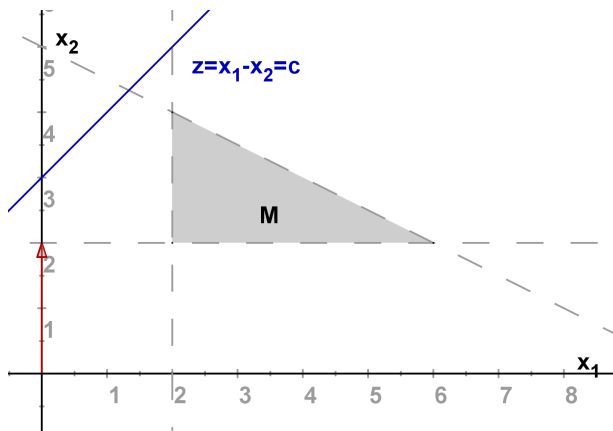
Celý postup v grafickém znázornění:



Množina přípustných řešení **M**. Při zahájení první fáze jsou obě proměnné x_1, x_2 nezákladní, vycházíme tedy z počátku.

Dvofázová simplexová metoda

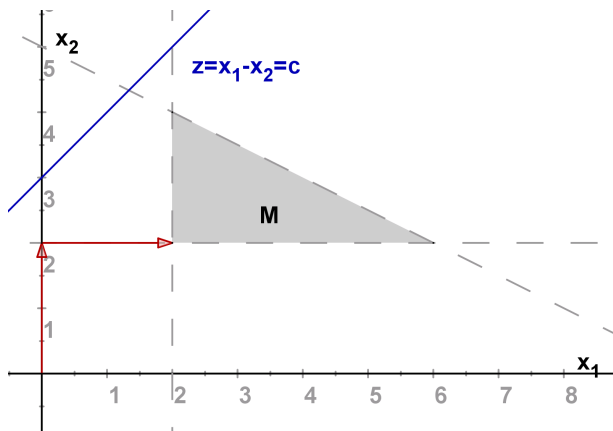
Celý postup v grafickém znázornění:



Po prvním kroku se proměnná x_2 stala základní s hodnotou $x_2 = 2$.

Dvufázová simplexová metoda

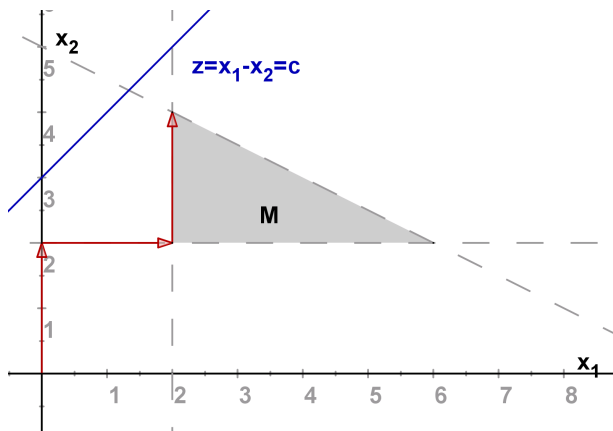
Celý postup v grafickém znázornění:



V dalším kroku se i proměnná x_1 stala základní s hodnotou $x_1 = 2$, dosáhli jsme přípustné množiny a startujeme druhou fází.

Dvufázová simplexová metoda

Celý postup v grafickém znázornění:



Po jednom kroku dosáhneme optima v bodě $[x_1, x_2] = [2, 4]$.

Dvoufázová simplexová metoda

Celý postup v grafickém znázornění:

Dualita úloh LP

Na původní úlohu lze nahlížet i jiným způsobem. Předpokládejme, že bychom suroviny nezpracovávali, ale rovnou prodali. Otázka zní, kdy se nám tento přímý prodej zdrojů vyplatí. To bude samozřejmě záviset na zisku z prodeje jednotlivých zdrojů - vyjádříme jej pomocí tzv. **duálních proměnných**, které označíme w_i (v naší úloze máme tři druhy kávových bobů, tedy $i = 1, 2, 3$). Můžeme pak formulovat tzv. **duální úlohu** k výchozímu problému: Jaký je minimální zisk z prodeje zdrojů, při kterém se nám nevyplatí vyrábět ani jeden výrobek?

Dualita úloh LP

Na původní úlohu lze nahlížet i jiným způsobem. Předpokládejme, že bychom suroviny nezpracovávali, ale rovnou prodali. Otázka zní, kdy se nám tento přímý prodej zdrojů vyplatí. To bude samozřejmě záviset na zisku z prodeje jednotlivých zdrojů - vyjádříme jej pomocí tzv. **duálních proměnných**, které označíme w_i (v naší úloze máme tři druhy kávových bobů, tedy $i = 1, 2, 3$).

Můžeme pak formulovat tzv. **duální úlohu** k výchozímu problému:

Jaký je minimální zisk z prodeje zdrojů, při kterém se nám nevyplatí vyrábět ani jeden výrobek? Tedy minimalizujeme zisk z prodeje zdrojů

$g(\mathbf{w}) = 40w_1 + 60w_2 + 25w_3$ za omezení, že se nevyplatí vyrábět ani směs

Mocca ani Standard, tedy, že platí nerovnosti $0,5w_1 + 0,5w_2 \geq 20$,

$0,5w_1 + 0,25w_2 + 0,5w_3 \geq 14$.

Dualita úloh LP

Na původní úlohu lze nahlížet i jiným způsobem. Předpokládejme, že bychom suroviny nezpracovávali, ale rovnou prodali. Otázka zní, kdy se nám tento přímý prodej zdrojů vyplatí. To bude samozřejmě záviset na zisku z prodeje jednotlivých zdrojů - vyjádříme jej pomocí tzv. **duálních proměnných**, které označíme w_i (v naší úloze máme tři druhy kávových bobů, tedy $i = 1, 2, 3$). Můžeme pak formulovat tzv. **duální úlohu** k výchozímu problému:

Jaký je minimální zisk z prodeje zdrojů, při kterém se nám nevyplatí vyrábět ani jeden výrobek? Tedy minimalizujeme zisk z prodeje zdrojů

$g(\mathbf{w}) = 40w_1 + 60w_2 + 25w_3$ za omezení, že se nevyplatí vyrábět ani směs

Mocca ani Standard, tedy, že platí nerovnosti $0,5w_1 + 0,5w_2 \geq 20$,

$0,5w_1 + 0,25w_2 + 0,5w_3 \geq 14$. Při použití označení zavedeného výše, kde

$\mathbf{c} = (20, 14)$ je vektor zisků z prodeje směsí, $\mathbf{b} = (40, 60, 25)^T$ je vektor kapacit surovin a \mathbf{A} strukturní matice, můžeme porovnat maticový zápis původní, tzv. **primární úlohy** a úlohy duální:

primární úloha

maximalizovat $z = \mathbf{c}^T \cdot \mathbf{x}$

za podm. $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$,

duální úloha

minimalizovat $g(\mathbf{w}) = \mathbf{b}^T \cdot \mathbf{w}$

za podm. $\mathbf{A}^T \cdot \mathbf{w} \geq \mathbf{c}, \mathbf{w} \geq \mathbf{0}$

Dualita úloh LP

Obecně lze pro formulaci duální úlohy k úloze LP použít následující pravidla:

Maximalizační úloha	↔	Minimalizační úloha
primární	↔	duální
duální	↔	primární
omezení typu \leq	↔	nezáporná proměnná
omezení typu \geq	↔	nekladná proměnná
omezení typu rovnice	↔	proměnná neomezená
nezáporná proměnná	↔	omezení typu \leq
nekladná proměnná	↔	omezení typu \geq
proměnná neomezená	↔	omezení typu rovnice

Dualita úloh LP

Vztah mezi vzájemně duálními úlohami lze vyjádřit **větou o dualitě**:

Existuje-li optimální řešení jedné z duálně sdružených úloh, potom existuje i optimální řešení druhé úlohy a navíc optimální hodnoty účelových funkcí se sobě rovnají!

Dualita úloh LP

Vztah mezi vzájemně duálními úlohami lze vyjádřit **větou o dualitě**:

Existuje-li optimální řešení jedné z duálně sdružených úloh, potom existuje i optimální řešení druhé úlohy a navíc optimální hodnoty účelových funkcí se sobě rovnají!

Z této věty logicky plyne, že pokud jedna ze sdružených úloh optimální řešení nemá, tak jej nemůže mít ani úloha druhá, lze ukázat, že pokud jedna úloha nemá žádné přípustné řešení, tak druhá úloha je neomezená a naopak. Dalším důsledkem je tzv. **slabá věta o dualitě**:

Hodnota účelové funkce maximalizační úlohy je vždy menší nebo rovna hodnotě účelové funkce minimalizační úlohy.

Dualita úloh LP

Vztah mezi vzájemně duálními úlohami lze vyjádřit **větou o dualitě**:

Existuje-li optimální řešení jedné z duálně sdružených úloh, potom existuje i optimální řešení druhé úlohy a navíc optimální hodnoty účelových funkcí se sobě rovnají!

Z této věty logicky plyne, že pokud jedna ze sdružených úloh optimální řešení nemá, tak jej nemůže mít ani úloha druhá, lze ukázat, že pokud jedna úloha nemá žádné přípustné řešení, tak druhá úloha je neomezená a naopak. Dalším důsledkem je tzv. **slabá věta o dualitě**:

Hodnota účelové funkce maximalizační úlohy je vždy menší nebo rovna hodnotě účelové funkce minimalizační úlohy.

Dále platí tzv. **věta o rovnováze**:

Je-li k -tá proměnná v řešení primární úlohy nenulová (tedy kladná), pak je k -tá podmínka v řešení duální úlohy splněna jako rovnost. Říkáme, že je k -tá podmínka **aktivní**.

Postoptimalizační analýza

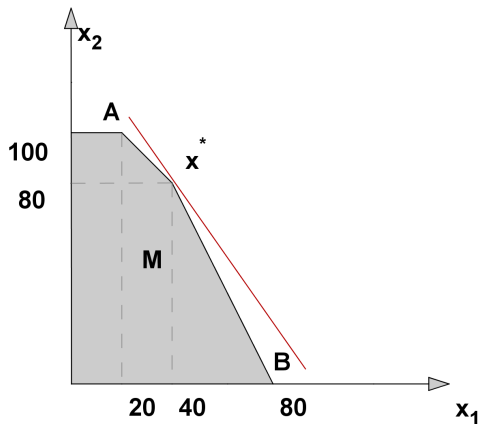
Analýza citlivosti primární úlohy zkoumá, do jaké míry ovlivní případné změny vstupních údajů původní optimální řešení. Zejména nás zajímají efekt při změně zisku z jednotlivého výrobku, případně při změně v jednotlivém kapacitním omezení. To lze zjistit bez nutnosti přepočítávat celou úlohu znovu. Určujeme tzv. **interval stability**, a to pro:

- koeficienty účelové funkce c_k , kdy zjišťujeme, v jakém rozmezí hodnot můžeme měnit jednotlivé c_k (při zachování hodnot ostatních koeficientů) tak, aby nedošlo ke změně optimálního řešení,
- kapacitní omezení b_i , kdy zjišťujeme v jakém rozmezí se může jednotlivé b_i pohybovat, aby nedošlo ke změně množiny základních proměnných, tedy byla zachována množina aktivních omezení.

Pro manažerské rozhodování je důležité zjistit, jaký je vliv změny kapacitního omezení na hodnotu účelové funkce. To nám prozradí optimální hodnoty duálních proměnných w_i . Tyto hodnoty se nazývají **stínové ceny** a vyjadřují hodnotu, o kterou se změní hodnota účelové funkce, jestliže zvýšíme kapacitu i - tého zdroje b_i o jednotku (za předpokladu že se touto změnou nedostaneme mimo interval stability).

Postoptimalizační analýza - intervaly stability pro ceny

Vlastní určení intervalů stability není složité a bývá nedílnou součástí softwarových výstupů. Dále si ukážeme grafickou interpretaci a odvození intervalů stability pro koeficienty účelové funkce v našem jednoduchém příkladě optimalizace výroby kávy. Na obrázku je vidět, jak lze optimální izokvantu účelové funkce naklánět, aby stále bylo optimálním řešením x^* .



Postoptimalizační analýza - intervaly stability pro ceny

Mezní hodnoty naklonění určíme tak, že přímka bude procházet body $x^* = [40, 80]$, $A = [20, 100]$ resp. $x^* = [40, 80]$, $B = [80, 0]$. Pro její směrnici q tedy musí platit nerovnosti

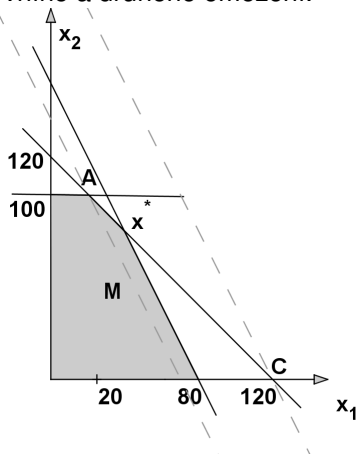
$$-2 = \frac{80 - 0}{40 - 80} \leq q \leq \frac{80 - 100}{40 - 20} = -1$$

Směrnici původní izokvanty $z = c_1 x_1 + c_2 x_2$ vyjádříme jako $q = \frac{-c_1}{c_2}$, přičemž původní hodnoty koeficientů jsou $c_1 = 20$, $c_2 = 14$. Interval stability pro c_1 tedy zjistíme po dosazení $q = \frac{-c_1}{14}$ do nerovností: $-2 \leq \frac{-c_1}{14} \leq -1$, tj.

$c_1 \in \langle 14, 28 \rangle$. Analogicky pro c_2 získáme interval stability dosazením $q = \frac{-20}{c_2}$ do nerovností: $-2 \leq \frac{-20}{c_2} \leq -1$ a dostaneme $c_2 \in \langle 10, 20 \rangle$.

Postoptimalizační analýza - intervaly stability pro kapacity

Ještě si ukažme ve stejné úloze grafické odvození intervalů stability pro pravé strany omezení. Na obrázku je vidět, jak můžeme posunout hranici prvního omezení, aby stále optimální řešení leželo v průsečíku hraničních přímek prvního a druhého omezení.



Postoptimalizační analýza - intervaly stability pro kapacity

Původní rovnice hraniční přímky prvního omezení byla $0,5x_1 + 0,25x_2 = 40$.
Její pravou stranu b_1 můžeme změnit maximálně tak, že by přímka procházela bodem A , resp. bodem C .

Dosazením souřadnic bodu $A = [20, 100]$ do levé strany omezení dostaneme $0,5 \cdot 20 + 0,25 \cdot 100 = 35$, což je dolní hranice pro b_1 .

Dosazením souřadnic bodu $C = [120, 0]$ do levé strany omezení dostaneme $0,5 \cdot 120 + 0,25 \cdot 0 = 60$, což je horní hranice pro b_1 .

Postoptimalizační analýza - intervaly stability pro kapacity

Původní rovnice hraniční přímky prvního omezení byla $0,5x_1 + 0,25x_2 = 40$. Její pravou stranu b_1 můžeme změnit maximálně tak, že by přímka procházela bodem A , resp. bodem C .

Dosazením souřadnic bodu $A = [20, 100]$ do levé strany omezení dostaneme $0,5 \cdot 20 + 0,25 \cdot 100 = 35$, což je dolní hranice pro b_1 .

Dosazením souřadnic bodu $C = [120, 0]$ do levé strany omezení dostaneme $0,5 \cdot 120 + 0,25 \cdot 0 = 60$, což je horní hranice pro b_1 .

Dostáváme tedy interval stability $b_1: \in \langle 35, 60 \rangle$. Podobně obdržíme intervaly stability pro ostatní omezení. Tyto intervaly jsou důležité při rozhodování o nákupu dalších zdrojů: pokud je stínová cena daného omezení větší než nákupní cena příslušné suroviny, vyplatí se v rozmezí intervalu stability navyšovat kapacitu. A jak určíme stínovou cenu pro b_1 ? Změnou na $b_1 + \Delta$ dostaneme nový bod optima jako průsečík přímek o rovnicích

$0,5x_1 + 0,25x_2 = 40 + \Delta$, $0,5x_1 + 0,5x_2 = 60$, tedy bod o souřadnicích $[40 + 4\Delta, 80 - 4\Delta]$. V tomto bodě je pak hodnota účelové funkce $z = 20(40 + 4\Delta) + 14(80 - 4\Delta) = 1920 + 24\Delta$. Stínová cena je $w_1 = 24$.

Stínové ceny najdeme v optimální tabulce pod sloupci přídatných proměnných!

Speciální úlohy lineárního programování

Mezi typickými úlohami LP lze najít úlohy s nějakými speciálními vlastnostmi. Tyto vlastnosti se mohou týkat struktury modelu, zejména strukturní matice, typu proměnných, dále způsobů řešení, apod. Významnou skupinu takových speciálních úloh tvoří **distribuční úlohy**. Z těchto úloh představíme dopravní problém, přiřazovací problém a okružní dopravní problém. Další problémy (kontejnerový či vícestupňový dopravní problém, úloha o pokrytí, stanovení rezných plánů apod.) viz literatura. Úlohy, ve kterých některé proměnné mohou nabývat pouze hodnot z množiny celých čísel souhrně nazýváme úlohami **celočíselného programování**. Proměnné v těchto úlohách zpravidla vyjadřují počty nedělitelných kusů, případně nabývají pouze hodnot 0 a 1, kterými se kóduje absence či přítomnost určitého spojení mezi zadanými objekty. Specifikům celočíselných úloh a základním přístupům k jejich řešení se budeme později také věnovat. V neposlední řadě stručně zmíníme alternativní přístup k řešení úloh LP, kdy je možné současně optimalizovat více kritérií, a to **cílové programování**.

Dopravní problém - formulace

V dopravní úloze se typicky řeší rozvržení rozvozu z dodavatelských míst k odběratelům tak, aby byly minimalizovány náklady související s rozvozem. Je definováno m dodavatelských míst - zdrojů V_1, V_2, \dots, V_m s omezenými **kapacitami** a_1, a_2, \dots, a_m a dále máme n cílových míst - odběratelů S_1, S_2, \dots, S_n se stanovenými **požadavky** b_1, b_2, \dots, b_n . Každá dvojice zdroj-cíl je nějak ohodnocena, typicky například náklady na přepravu jednotky zboží. Tyto **náklady** označíme c_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$. Cílem je naplánovat objemy přepravy mezi jednotlivými zdroji a cíli (označíme je x_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$) tak, aby byly uspokojeny požadavky odběratelů a nebyly překročeny kapacity zdrojů.

Dopravní problém - formulace

V dopravní úloze se typicky řeší rozvržení rozvozu z dodavatelských míst k odběratelům tak, aby byly minimalizovány náklady související s rozvozem. Je definováno m dodavatelských míst - zdrojů V_1, V_2, \dots, V_m s omezenými **kapacitami** a_1, a_2, \dots, a_m a dále máme n cílových míst - odběratelů S_1, S_2, \dots, S_n se stanovenými **požadavky** b_1, b_2, \dots, b_n . Každá dvojice zdroj-cíl je nějak ohodnocena, typicky například náklady na přepravu jednotky zboží. Tyto **náklady** označíme c_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$. Cílem je naplánovat objemy přepravy mezi jednotlivými zdroji a cíli (označíme je x_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$) tak, aby byly uspokojeny požadavky odběratelů a nebyly překročeny kapacity zdrojů. Úloha tedy obsahuje $m \cdot n$ proměnných x_{ij} , pro něž minimalizujeme účelovou funkci

$$z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \text{ za podmínek}$$

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

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

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

Účelová funkce i omezení jsou lineární, jde tedy o úlohu LP.

Dopravní problém - metody řešení

I když jde o úlohu LP, kterou lze řešit simplexovou metodou, vzhledem k velkému počtu proměnných a speciální struktuře matice omezení jsou většinou praktičtější jiné metody (jde o tzv. řídkou matici - obsahuje hodně nul, navíc zbylé jedničky mají blokovou strukturu).

Dopravní problém - metody řešení

I když jde o úlohu LP, kterou lze řešit simplexovou metodou, vzhledem k velkému počtu proměnných a speciální struktuře matice omezení jsou většinou praktičtější jiné metody (jde o tzv. řídkou matici - obsahuje hodně nul, navíc zbylé jedničky mají blokovou strukturu).

Příklad: Pro reálnou úlohu s 20 zdroji a 300 zákazníky máme ? proměnných a ? omezení, to znamená 1920000 polí v simplexové tabulce, což zabere cca 11 MB operační paměti

Dopravní problém - metody řešení

I když jde o úlohu LP, kterou lze řešit simplexovou metodou, vzhledem k velkému počtu proměnných a speciální struktuře matice omezení jsou většinou praktičtější jiné metody (jde o tzv. řídkou matici - obsahuje hodně nul, navíc zbylé jedničky mají blokovou strukturu).

Příklad: Pro reálnou úlohu s 20 zdroji a 300 zákazníky máme ? proměnných a ? omezení, to znamená 1920000 polí v simplexové tabulce, což zabere cca 11 MB operační paměti

Ze simplexové metody vychází **modifikovaná distribuční metoda** (MODI). Pro rychlé získání přibližného řešení bez záruky optimality lze využít **heuristické** metody, z nichž si ukážeme tři: metodu severozápadního rohu, metodu maticového minima (zvanou též indexní) a Vogelovu aproximační metodu (VAM).

Dopravní problém - vyrovnání úlohy

Zřejmě není možné uspokojit všechny spotřebitele, jestliže celková poptávka $\sum_{j=1}^n b_j$ převyšuje celkovou kapacitu $\sum_{i=1}^n a_i$, úloha pak nemá přípustné řešení. Úlohu, ve které platí rovnost $\sum_{j=1}^n b_j = \sum_{i=1}^n a_i$ označujeme jako **vyrovnaný dopravní problém**. Problém pak má přípustné řešení i pokud u omezení pro kapacity zdrojů nahradíme nerovnosti rovnostmi, spotřebují se tedy všechny jednotky. Nadále budeme pracovat jen s takovými vyrovnanými úlohami.

Dopravní problém - vyrovnání úlohy

Zřejmě není možné uspokojit všechny spotřebitele, jestliže celková poptávka $\sum_{j=1}^n b_j$ převyšuje celkovou kapacitu $\sum_{i=1}^n a_i$, úloha pak nemá přípustné řešení. Úlohu, ve které platí rovnost $\sum_{j=1}^n b_j = \sum_{i=1}^n a_i$ označujeme jako **vyrovnaný dopravní problém**. Problém pak má přípustné řešení i pokud u omezení pro kapacity zdrojů nahradíme nerovnosti rovnostmi, spotřebují se tedy všechny jednotky. Nadále budeme pracovat jen s takovými vyrovnanými úlohami.

Nevyrovnaná úloha s převisem poptávky se převede na vyrovnanou pomocí zavedení **fiktivního** zdroje s kapacitou $\sum_{j=1}^n b_j - \sum_{i=1}^n a_i$. V případě převisu nabídky se naopak zavede fiktivní zákazník s požadavkem $\sum_{i=1}^n a_i - \sum_{j=1}^n b_j$.

Pozor! **Přepravní náklady do fiktivních míst jsou vždy nulové.**

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1, S_2, S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1, V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky			
	x_{ij}	3	6	4	5
zbývá	5				
	7				
	6				

S_1 dostane z prvního zdroje 3 jednotky, ve zdroji pak zůstanou 2.

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1 , S_2 , S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1 , V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky				
		x_{ij}	0	6	4	5
zbyvá	2	3				
	7					
	6					

Dvě zbylé jednotky z V_1 dostane S_2 , přesuneme se na další zdroj.

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1 , S_2 , S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1 , V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky			
x_{ij}		0	4	4	5
zbyvá	0	3	2		
	7				
	6				

Čtyři zbylé jednotky pro S_2 dodá V_2 , zbydou mu tři jednotky.

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1 , S_2 , S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1 , V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky			
		0	0	4	5
zbývá	0	3	2		
	3		4		
	6				

Tři zbylé jednotky V_2 se dodají spotřebiteli S_3 .

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1 , S_2 , S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1 , V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky			
x_{ij}		0	0	1	5
zbývá	0	3	2		
	0		4	3	
	6				

Jednu chybějící jednotku dodá spotřebiteli S_3 zdroj V_3 .

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1 , S_2 , S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1 , V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky			
		0	0	0	5
zbývá	x_{ij}	0	2	3	
	0	3	4	1	
	5				

Posledních pět jednotek ve V_3 zbyde pro spotřebitele S_4 .

Dopravní problém - metoda SZ rohu

Metoda severozápadního rohu je zřejmě nejjednodušší metodou získání přípustného řešení dopravní úlohy, začíná se v levém horním rohu tabulky a požadavky zákazníků se uspokojují zleva doprava. Pokud je již zdroj vyčerpán, přejde se na další zdroj, tj. o řádek níž. Končí se v pravém dolním rohu. Metoda nebere v úvahu přepravní náklady.

Ukažme si metodu pro úlohu z "M. Plevný, M. Žižka: Modelování a optimalizace v manažerském rozhodování": Najděte přípustné řešení DÚ s požadavky odběratelů S_1, S_2, S_3 a S_4 postupně 3, 6, 4 a 5 jednotek zboží a zdroji V_1, V_2 a V_3 s kapacitou po řadě 5, 7 a 6 jednotek.

		požadavky			
		0	0	0	0
zbývá	x_{ij}	0	2	3	0
	0	3	4	1	5
	0				

Dostali jsme přípustné řešení.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy)

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
	x_{ij}	3	6	4	5
zbývá	5				
	7				
	6				

	přepravní náklady			
c_{ij}	S_1	S_2	S_3	S_4
V_1	2	1	3	4
V_2	6	2	6	1
V_3	7	3	3	3

Nejnižší náklady jsou $c_{12} = c_{24} = 1$. Vyberme například první z nich, V_1 může dodat S_2 nejvýše 5 jednotek. Tím se zcela vyčerpá a tudíž již dále neuvažujeme první řádek ani v matici nákladů.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
		3	1	4	5
zbývá	x_{ij}		5		
	0				
	7				
	6				

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1				
V_2	6	2	6	1
V_3	7	3	3	3

Nejnižší náklady jsou $c_{24} = 1$. S_4 může odebrat od V_2 nejvýše 5 jednotek, je plně spokojen a čtvrtý sloupec tudíž už dál nebereme v úvahu.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
		3	1	4	0
zbývá	x_{ij}		5		5
	0				
	2				
6					

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1				
V_2	6	2	6	
V_3	7	3	3	

Nejnižší náklady jsou $c_{22} = 2$. S_2 může odebrat od V_2 nejvýše 1 jednotku, druhý sloupec je tím vyčerpán.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
		3	0	4	0
zbývá	x_{ij}				
	0		5		
	1		1		5
	6				

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1				
V_2	6		6	
V_3	7		3	

Nejnižší náklady jsou $c_{33} = 3$. S_3 může odebrat od V_3 nejvýše 4 jednotky, třetí sloupec je tím vyčerpán.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
x_{ij}		3	0	0	0
zbývá	0		5		
	1		1		5
	2			4	

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1				
V_2	6			
V_3	7			

Nejnižší náklady jsou $c_{33} = 3$. S_3 může odebrat od V_3 nejvýše 4 jednotky, třetí sloupec je tím vyčerpán.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
		3	0	0	0
zbývá	x_{ij}		5		
	0		1		5
	1			4	
	2				

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1				
V_2	6			
V_3	7			

Nejnižší náklady jsou $c_{21} = 6$. V_2 může dodat S_1 nejvýše 1 jednotku, třetí řádek je tím vyčerpán.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
	x_{ij}	2	0	0	0
zbývá	0		5		
	0	1	1		5
	2			4	

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1				
V_2				
V_3	7			

V posledním kroku dodá V_3 spotřebiteli S_1 zbylé dvě jednotky.

Dopravní problém - indexová metoda

Indexová metoda dává většinou řešení s nižšími přepravními náklady než předchozí metoda. Jde o tzv. hladový algoritmus, vybíráme postupně vždy trasy s nejnižším cenovým ohodnocením. (vynecháme případné fiktivní trasy) Ukažme si postup metody pro již dříve řešený dopravní problém, přidáme tabulku s přepravními náklady c_{ij} , $i = 1, \dots, 3$, $j = 1, \dots, 4$.

		požadavky			
		x_{1j}	x_{2j}	x_{3j}	x_{4j}
zbývá	x_{11}	0	5		
	x_{12}	0	1	1	5
	x_{13}	0	2		4
	x_{14}				

přepravní náklady				
c_{ij}	S_1	S_2	S_3	S_4
V_1	2	1	3	4
V_2	6	2	6	1
V_3	7	3	3	3

Výsledné přepravní náklady jsou

$$x_{12} \cdot c_{12} + x_{21} \cdot c_{21} + x_{22} \cdot c_{22} + x_{24} \cdot c_{24} + x_{31} \cdot c_{31} + x_{33} \cdot c_{33} = 5 + 6 + 2 + 5 + 14 + 12 = 44.$$

Pro předchozí metodu bychom dostali náklady ve výši

$$6 + 2 + 6 + 2 + 5 + 14 + 12 = 47.$$

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší oproti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí.

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší proti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference d_i , $i = 1, \dots, 3$ a d_j , $j = 1, \dots, 4$.

		požadavky			
		3	6	4	5
zbývá	x_{ij}				
	5				
	7				
	6				

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1	2	1	3	4	1
V_2	6	2	6	1	1
V_3	7	3	3	3	0
d_j	4	1	0	2	

N největší difference je v prvním sloupečku, vybereme tedy jeho nejmenší náklad, což je $c_{11} = 2$.

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší oproti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference $d_i, i = 1, \dots, 3$ a $d_j, j = 1, \dots, 4$.

		požadavky			
		0	6	4	5
zbývá	x_{ij}				
	2	3			
	7				
	6				

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1		1	3	4	2
V_2		2	6	1	1
V_3		3	3	3	0
d_j		1	0	2	

Největší difference je v prvním řádku, vybereme tedy jeho nejmenší náklad, což je $c_{12} = 1$.

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší proti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference $d_i, i = 1, \dots, 3$ a $d_j, j = 1, \dots, 4$.

		požadavky			
		0	4	4	5
zbývá	x_{ij}	0	3	2	
	0	3	2		
	7				
	6				

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1					
V_2		2	6	1	1
V_3		3	3	3	0
d_j		1	3	2	

Největší difference je v třetím sloupci, vybereme tedy jeho nejmenší náklad, což je $c_{33} = 3$.

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší oproti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference $d_i, i = 1, \dots, 3$ a $d_j, j = 1, \dots, 4$.

		požadavky			
	x_{ij}	0	4	0	5
zbývá	0	3	2		
	7				
	2			4	

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1					
V_2		2		1	1
V_3		3		3	0
d_j		1		2	

Největší difference je ve čtvrtém sloupci, vybereme tedy jeho nejmenší náklad, což je $c_{24} = 1$.

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší proti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference d_i , $i = 1, \dots, 3$ a d_j , $j = 1, \dots, 4$.

		požadavky			
		0	4	0	0
zbývá	x_{ij}	0	2		5
	0	3			
	2			4	

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1					
V_2		2			
V_3		3			
d_j		1			

Největší difference je ve druhém sloupci, vybereme tedy jeho nejmenší náklad, což je $c_{22} = 2$.

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší oproti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference $d_i, i = 1, \dots, 3$ a $d_j, j = 1, \dots, 4$.

		požadavky			
		0	2	0	0
zbývá	x_{ij}	0	2	0	0
	0	3	2		
	0		2		5
	2			4	

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1					
V_2					
V_3					
d_j					

Zbývá přepravit dvě jednotky z V_3 zákazníkovi S_2 .

Dopravní problém - VAM

Vogelova aproximační metoda je třetí heuristickou metodou a její výhoda oproti metodě indexační spočívá v tom, že bere v každém kroku v úvahu "alternativní náklady" při volbě nejlevnější trasy. Pracuje s **diferencemi**, tedy rozdíly druhé nejlepší oproti nejlepší variantě v každém řádku a sloupci. Volí se pak vždy nejlevnější jízda v řádku či sloupci s největší diferencí. Použijme metodu pro náš známý příklad. K matici nákladů doplníme řádkové a sloupcové difference $d_i, i = 1, \dots, 3$ a $d_j, j = 1, \dots, 4$.

		požadavky			
		0	0	0	0
zbývá	x_{ij}	0	2		5
	0	3	2		
	0		2		
	0		2	4	

c_{ij}	S_1	S_2	S_3	S_4	d_i
V_1	2	1	3	4	1
V_2	6	2	6	1	1
V_3	7	3	3	3	0
d_j	4	1	0	2	

Výsledné přepravní náklady jsou

$x_{11} \cdot c_{11} + x_{12} \cdot c_{12} + x_{22} \cdot c_{22} + x_{24} \cdot c_{24} + x_{32} \cdot c_{32} + x_{33} \cdot c_{33} = 35$, což je výrazně méně než u předchozích metod.

Dopravní problém - MODI

Modifikovaná distribuční metoda řešení vyrovnaných úloh spočívá ve využití duality. Duální úlohu k problému $z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$ za podmínek

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

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

formulujeme pomocí duálních proměnných $u_i, v_j, i = 1, \dots, m, j = 1, \dots, n$:

$$\sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \rightarrow \max$$

za omezení $u_i + v_j \leq c_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n$

Dopravní problém - MODI

Modifikovaná distribuční metoda řešení vyrovnaných úloh spočívá ve využití duality. Duální úlohu k problému $z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$ za podmínek

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

formulujeme pomocí duálních proměnných $u_i, v_j, i = 1, \dots, m, j = 1, \dots, n$:

$$\sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \rightarrow \max$$

za omezení $u_i + v_j \leq c_{ij}, i = 1, \dots, m, j = 1, \dots, n$

Z teorie duality plyne, že pro všechny základní proměnné primární úlohy ($x_{ij} > 0$) je příslušné omezení splněno jako rovnost. ($u_i + v_j = c_{ij}$). Pokud se nejedná o řešení degenerované, je v bázi právě $m + n - 1$ proměnných, a tedy dostaneme soustavu $m + n - 1$ rovnic pro $m + n$ duálních proměnných. Zvolíme-li libovolně hodnotu jedné duální proměnné, lze ostatní dopočítat. Ověřením duální přípustnosti zjistíme, zda je nalezené řešení optimální, tj. zda $u_i + v_j - c_{ij} \leq 0, i = 1, \dots, m, j = 1, \dots, n$

Dopravní problém - MODI

Určení hodnot duálních proměnných a ověření optimality lze provést přímo v tabulce, vraťme se k příkladu vyřešenému metodou VAM:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5	3	2		
	7		2		5
	6		2	4	

c_{ij}	S_1	S_2	S_3	S_4	u_j
V_1	2	1	3	4	
V_2	6	2	6	1	
V_3	7	3	3	3	
v_j					

Volíme u_i, v_j tak, aby jejich součty byly rovny číslům v červených polích. Jednu proměnnou volíme libovolně, tedy pro jednoduchost např. $v_2 = 0$. Ostatní postupně dopočítáme.

Dopravní problém - MODI

Určení hodnot duálních proměnných a ověření optimality lze provést přímo v tabulce, vraťme se k příkladu vyřešenému metodou VAM:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5	3	2		
	7		2		5
	6		2	4	

c_{ij}	S_1	S_2	S_3	S_4	u_j
V_1	2	1	3	4	1
V_2	6	2	6	1	2
V_3	7	3	3	3	3
v_j	1	0	0	-1	

Volíme u_i, v_j tak, aby jejich součty byly rovny číslům v červených polích. Jednu proměnnou volíme libovolně, tedy pro jednoduchost např. $v_2 = 0$. Ostatní postupně dopočítáme. Tabulka je optimální, součty duálních proměnných nepřesahují čísla uvnitř tabulky.

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5		5		
	7	1	1		5
	6	2		4	
	6				

c_{ij}	S_1	S_2	S_3	S_4	u_j
V_1	2	1	3	4	
V_2	6	2	6	1	
V_3	7	3	3	3	
v_j					

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
		x_{1j}	x_{2j}	x_{3j}	x_{4j}
kapacita	x_{1j}	3	6	4	5
	x_{2j}	5	5		
	x_{3j}	7	1	1	5
	x_{4j}	6	2	4	

c_{ij}	S_1	S_2	S_3	S_4	u_j
V_1	2	1	3	4	-1
V_2	6	2	6	1	0
V_3	7	3	3	3	1
v_j	6	2	2	1	

Volíme u_i, v_j tak, aby jejich součty byly rovny číslům v červených polích. Požadovných hodnot dosáhneme např. volbou $u_2 = 0$. Ostatní opět dopočítáme.

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5		5		
	7	1	1		5
	6	2		4	

c_{ij}	S_1	S_2	S_3	S_4	u_j
V_1	2	1	3	4	-1
V_2	6	2	6	1	0
V_3	7	3	3	3	1
v_j	6	2	2	1	

Je vidět, že v levém horním rohu je porušena podmínka optimality:

$$u_1 + v_1 = 5 > c_{11} = 2.$$

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5	+	5 -		
	7	1 -	1 +		5
	6	2		4	

c_{ij}	S_1	S_2	S_3	S_4	u_i
V_1	2	1	3	4	-1
V_2	6	2	6	1	0
V_3	7	3	3	3	1
v_j	6	2	2	1	

Je vidět, že v levém horním rohu je porušena podmínka optimality:

$$u_1 + v_1 = 5 > c_{11} = 2.$$

K základnímu řešení s lepší hodnotou primární účelové funkce přejdeme volbou x_{11} jako vstupující proměnné, pomocí které nahradíme některou z proměnných x_{12} , x_{22} , x_{21} ležících na tzv. **Dantzigově uzavřeném obvodu** vycházejícím z x_{11} . Vybíráme tak, aby se neporušily podmínky primární přípustnosti. Je vidět, že lze o 1 zmenšit a tudíž vyhodit z báze x_{21} . Současně musíme zvětšit x_{22} a zmenšit x_{12} (viz znaménka + a - v levé tabulce). Dostaneme nové základní řešení.

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
		x_{1j}	x_{2j}	x_{3j}	x_{4j}
kapacita	x_{1j}	3	6	4	5
	x_{2j}	1	4		
	x_{3j}	7	0	2	5
	x_{4j}	6	2		4

c_{ij}	S_1	S_2	S_3	S_4	u_i
V_1	2	1	3	4	-1
V_2	6	2	6	1	0
V_3	7	3	3	3	4
v_j	3	2	-1	1	

Znovu doplníme u_i , v_j a zkontrolujeme podmínku optimality. Ta je porušena pro c_{32} a c_{34} .

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5	1 +	4 -		
	7	0	2		5
	6	2 -	+	4	

c_{ij}	S_1	S_2	S_3	S_4	u_i
V_1	2	1	3	4	-1
V_2	6	2	6	1	0
V_3	7	3	3	3	4
v_j	3	2	-1	1	

Znovu doplníme u_i , v_j a zkontrolujeme podmínku optimality. Ta je porušena pro c_{32} a c_{34} .

Protože větší rozdíl je mezi $(u_3 + v_2) - c_{32} = 6 - 3 = 3$ než mezi $(u_3 + v_4) - c_{34} = 5 - 3 = 2$, získáme volbou x_{32} jako vstupující větší zlepšení účelové funkce. Musíme k ní doplnit Dantzigův obvod, budou jej tvořit x_{31} , x_{11} a x_{12} . Z báze vystoupí x_{31} , jehož 2 jednotky se přesunou do x_{32} . (na obvodu tudíž též o 2 zvětšíme x_{11} a zmenšíme x_{12} . Dostaneme novou tabulku.

Dopravní problém - MODI

Kritérium optimality však není splněno pro přípustné řešení nalezené indexovou metodou:

		požadavky			
	x_{ij}	3	6	4	5
kapacita	5	3	2		
	7		2		5
	6	0	2	4	

c_{ij}	S_1	S_2	S_3	S_4	u_j
V_1	2	1	3	4	1
V_2	6	2	6	1	2
V_3	7	3	3	3	3
v_j	1	0	0	-1	

Nalezené řešení je již optimální jak jsme ověřili dříve, při výpočtu metodou VAM. Hodnota účelové funkce $z = 35$ již dále nejde zlepšit. Při postupu metodou MODI se někdy setkáme s **degenerací**, kdy v bázi je méně než $m + n - 1$ proměnných. Pak se některé nulové x_{ij} uměle nahradí malou hodnotou $\varepsilon > 0$.

Dopravní problém - použití

Příklady možných aplikací dopravního problému ilustruje následující přehled.

druh činnosti	zdroje	cílová místa
rozvoz pohonných hmot	rafinérie, sklady	čerpací stanice
svoz poštovních zásilek	přepravní uzly	třídící centra
sběr fotozakázek	fotosběrny	spádová centra
distribuce léčiv	sklady distribučních firem	lékárny, nemocnice
zpracování cukrové řepy	produkční střediska	cukrovary

Vyjímečně se u dopravních úloh setkáme i s maximalizací účelové funkce.
Kdy?

Přiřazovací problém

Přiřazovací úlohu můžeme charakterizovat jako problém vytvoření párů z objektů ze dvou různých skupin, tak aby toto spárování přineslo co největší efekt. Typicky jde o přidělení jednotlivých projektů pracovníkům či pracovních činností strojům tak abychom minimalizovali náklady nebo maximalizovali zisk. Jde o úlohu příbuznou s dopravním problémem.

Ukažme příklad takové úlohy z knihy "M. Kavan: Výrobní a provozní management": Optimalizujte přidělení prací 1, 2, 3 strojům A, B, C, D, přičemž výrobní náklady jsou dány tabulkou:

		stroje			
	c_{ij}	A	B	C	D
práce	1	15	19	17	12
	2	12	10	15	9
	3	18	14	11	14

Musíme tedy vybrat jedno číslo v každém řádku, tak aby jejich součet byl minimální a přitom žádná dvě čísla neležela ve stejném sloupci.

Přiřazovací problém - matematická formulace

Přidělení i -tého úkolu j -tému pracovnímu místu můžeme reprezentovat zápisem $x_{ij} = 1$, ostatním proměnným přiřadíme hodnotu 0. Pokud by bylo úkolů více než pracovních míst ($m > n$), je úloha neřešitelná. V případě opačné nerovnosti dorovnáme úlohu zavedením fiktivních prací s nulovými náklady, tak aby $m = n$. Nadále předpokládejme, že je úloha vyrovnaná. Matematický model přiřazovacího problému zahrnuje podmínky, že řádkové a sloupcové součty v tabulce jsou rovny jedné, s tím že proměnné nabývají pouze hodnot 0 nebo 1. Úlohu můžeme zapsat takto:
Minimalizujme účelovou funkci

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

za podmínek

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

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

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

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	15	19	17	12
	2	12	10	15	9
	3	18	14	11	14

Přidáme tedy jednu fiktivní činnost s nulovými náklady.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	15	19	17	12
	2	12	10	15	9
	3	18	14	11	14
	4	0	0	0	0

Zredukujeme řádky odečtením řádkového minima, v každém pak bude aspoň jedna nula.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	3	7	5	0
	2	3	1	6	0
	3	7	3	0	3
	4	0	0	0	0

Nyní bychom totéž provedli i pro sloupce, aby v každém byla jedna nula. To už ale nemusíme dělat díky přidané činnosti.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	3	7	5	0
	2	3	1	6	0
	3	7	3	0	3
	4	0	0	0	0

K pokrytí všech nul stačí tři čáry, takže tabulka není optimální. Krok metody spočívá v přičtení nejmenšího nepokrytého prvku k hodnotám v průsečících čar a současně jeho odečtení od všech nepokrytých prvků.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	2	6	5	0
	2	2	0	6	0
	3	6	2	0	3
	4	0	0	1	1

Po provedení kroku metody jsme dostali novou nepokrytou nulu a zbavili se dvojitě pokrytých nul. Musíme setrojit nové pokrytí.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	2	6	5	0
	2	2	0	6	0
	3	6	2	0	3
	4	0	0	1	1

Ověřme, zda je nová tabulka optimální. K pokrytí všech nul jsou potřeba minimálně čtyři čáry. Výpočet končí, jinak bychom zopakovali krok.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	2	6	5	0
	2	2	0	6	0
	3	6	2	0	3
	4	0	0	1	1

Mezi těmi, které nejsou pokryty dvojitě vybereme nezávislé nuly. Ty nám říkají, které práce kterým strojům přiřadit. Tedy 1-D, 2-B, 3-C. Stroji A se přiřadí fiktivní práce, tedy se nevyužije.

Přiřazovací problém - řešení

Přiřazovací problém je možné řešit tzv. **mad'arskou metodou**. Jejím principem je převedení původní úlohy na úlohu redukovanou tak, aby v každé řadě (řadou označujeme souhrně řádky a sloupce) byla aspoň jedna nula a přitom ostatní sazby zůstaly kladné. Optimální tabulka je taková, která obsahuje n "nezávislých nul" (nezávislé jsou tehdy, když žádné dvě neleží ve stejné řadě - jako v sudoku). To poznáme podle toho, že nelze "přeškrtnout" všechny nuly pomocí méně než n vodorovných a svislých čar. Metodu ukážeme na řešení úvodní úlohy. Původní zadání je třeba vyrovnat, protože počet strojů je větší než počet prací.

		stroje			
	c_{ij}	A	B	C	D
práce	1	15	19	17	12
	2	12	10	15	9
	3	18	14	11	14

Optimální řešení znovu vyznačme v původní tabulce. Celkové náklady jsou $12 + 10 + 11 = 33$. Úloha nemusí mít jediné řešení - pokud je možné volit čáry pokrytí či vybírat nezávislé nuly více způsoby, všechna výsledná řešení budou mít stejnou hodnotu účelové funkce.

Okružní dopravní problém

Problém se často též nazývá jako **Úloha obchodního cestujícího** a připomíná přiřazovací úlohu. Cílem je vyjít z nějakého výchozího stanoviště (označme jej A_1), navštívit postupně každé jednou místa A_2, \dots, A_n a nakonec se vrátit zpět tak, aby délka trasy byla co nejmenší. Úloha má velké množství reálných aplikací - pravidelný rozvoz či svoz různých produktů (pekárny, mlékárny, popeláři, atd.).

Okružní dopravní problém

Problém se často též nazývá jako **Úloha obchodního cestujícího** a připomíná přiřazovací úlohu. Cílem je vyjít z nějakého výchozího stanoviště (označme jej A_1), navštívit postupně každé jednou místa A_2, \dots, A_n a nakonec se vrátit zpět tak, aby délka trasy byla co nejmenší. Úloha má velké množství reálných aplikací - pravidelný rozvoz či svoz různých produktů (pekárny, mlékárny, popeláři, atd.).

V matematickém modelu se také zavádějí **bivalentní proměnné** x_{ij} nabývající hodnoty 1 nebo 0 podle toho, zda cesta A_i, A_j bude na okružní trase zařazena či ne. Protože se každé místo projede právě jednou, tj. právě jednou bude koncovým a právě jednou výchozím bodem, máme stejně jako u přiřazovacího problému omezení ve formě jednotkových řádkových i sloupcových součtů. Okružní úloha je však o mnoho složitější, protože navíc obsahuje další omezení, které má zabránit tomu, aby se řešení rozpadlo do více samostatných cyklů (příklad viz tabulky).

Okružní dopravní problém

Problém se často též nazývá jako **Úloha obchodního cestujícího** a připomíná přiřazovací úlohu. Cílem je vyjít z nějakého výchozího stanoviště (označme jej A_1), navštívit postupně každé jednou místa A_2, \dots, A_n a nakonec se vrátit zpět tak, aby délka trasy byla co nejmenší. Úloha má velké množství reálných aplikací - pravidelný rozvoz či svoz různých produktů (pekárny, mlékárny, popeláři, atd.).

V matematickém modelu se také zavádějí **bivalentní proměnné** x_{ij} nabývající hodnoty 1 nebo 0 podle toho, zda cesta A_i, A_j bude na okružní trase zařazena či ne. Protože se každé místo projede právě jednou, tj. právě jednou bude koncovým a právě jednou výchozím bodem, máme stejně jako u přiřazovacího problému omezení ve formě jednotkových řádkových i sloupcových součtů. Okružní úloha je však o mnoho složitější, protože navíc obsahuje další omezení, které má zabránit tomu, aby se řešení rozpadlo do více samostatných cyklů (příklad viz tabulky).

Pro velká n je prakticky nemožné použití standartních optimalizačních algoritmů (s běžně dostupnou výpočetní technikou nenajdeme v reálném čase optimální řešení už pro 30 uzlů).

Okružní dopravní problém - přípustnost řešení

Ilustrujme si rozdíl mezi přiřazovací a okružní úlohou na následujícím příkladě. Cílem je najít nějakou okružní trasu (pro jednoduchost neuvádíme tabulku nákladů a tedy nepožadujeme optimalitu) mezi městy A_1, \dots, A_5 , může se nám tedy zdát, že je to stejné jako výběr dvojic měst, mezi kterými povede trasa. Znázorníme nějaký takový výběr v tabulce:

	A_1	A_2	A_3	A_4	A_5
A_1	0	0	1	0	0
A_2	0	0	0	1	0
A_3	0	1	0	0	0
A_4	0	0	0	0	1
A_5	1	0	0	0	0

Jedná se o přípustné řešení okružní úlohy, reprezentuje trasu $A_1 - A_3 - A_2 - A_4 - A_5 - A_1$.

Okružní dopravní problém - přípustnost řešení

Ilustrujme si rozdíl mezi přiřazovací a okružní úlohou na následujícím příkladě. Cílem je najít nějakou okružní trasu (pro jednoduchost neuvádíme tabulku nákladů a tedy nepožadujeme optimalitu) mezi městy A_1, \dots, A_5 , může se nám tedy zdát, že je to stejné jako výběr dvojic měst, mezi kterými povede trasa. Znáznorněme nějaký takový výběr v tabulce:

	A_1	A_2	A_3	A_4	A_5
A_1	0	1	0	0	0
A_2	0	0	1	0	0
A_3	1	0	0	0	0
A_4	0	0	0	0	1
A_5	0	0	0	1	0

Jedná se o nepřípustné řešení okružní úlohy, reprezentuje dva samostatné cykly $A_1 - A_2 - A_3 - A_1$ a $A_4 - A_5 - A_4$.

Okružní dopravní problém - přibližné řešení

K nalezení přibližného řešení lze využít heuristické metody, např. **hladový algoritmus**. Na úloze z knihy "J. Pelikán, V. Chýna: Kvantitativní management", ukážeme **metodu nejbližšího souseda**.

Pekárna v Kralupech vypravuje každé ráno dodávku, která zásobuje prodejny potravin v okolních městech a poté se vrací zpět. Doporučte, v jakém pořadí má města projet, aby ujel co nejméně km. Tabulka vzdáleností je následující:

Okružní dopravní problém - přibližné řešení

K nalezení přibližného řešení lze využít heuristické metody, např. **hladový algoritmus**. Na úloze z knihy "J. Pelikán, V. Chýna: Kvantitativní management", ukážeme **metodu nejbližšího souseda**.

Pekárna v Kralupech vypravuje každé ráno dodávku, která zásobuje prodejny potravin v okolních městech a poté se vrací zpět. Doporučte, v jakém pořadí má města projet, aby ujel co nejméně km. Tabulka vzdáleností je následující:

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza
Kralupy	0	4	16	8	18	25	17
Veltrusy	4	0	20	12	22	28	13
Slaný	16	20	0	12	7	14	17
Velvary	8	12	12	0	10	17	10
Zlonice	18	22	7	10	0	7	10
Vraný	25	28	14	17	7	0	15
Bříza	17	13	17	10	10	15	0

Okružní dopravní problém - přibližné řešení

K nalezení přibližného řešení lze využít heuristické metody, např. **hladový algoritmus**. Na úloze z knihy "J. Pelikán, V. Chýna: Kvantitativní management", ukážeme **metodu nejbližšího souseda**.

Pekárna v Kralupech vypravuje každé ráno dodávku, která zásobuje prodejny potravin v okolních městech a poté se vrací zpět. Doporučte, v jakém pořadí má města projet, aby ujel co nejméně km. Tabulka vzdáleností je následující:

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza
Kralupy	0	4	16	8	18	25	17
Veltrusy	4	0	20	12	22	28	13
Slaný	16	20	0	12	7	14	17
Velvary	8	12	12	0	10	17	10
Zlonice	18	22	7	10	0	7	10
Vraný	25	28	14	17	7	0	15
Bříza	17	13	17	10	10	15	0

Dle metody ze stávajícího města jedeme vždy do nejbližšího tak, aby se cesta nezacyklila. Dostaneme okruh Kralupy - Veltrusy - Velvary - Zlonice - Slaný - Vraný - Bříza - Kralupy o celkové délce $4+12+10+7+14+15+17 = 79$ km.

Okružní dopravní problém - přibližné řešení

K nalezení přibližného řešení lze využít heuristické metody, např. **hladový algoritmus**. Na úloze z knihy "J. Pelikán, V. Chýna: Kvantitativní management", ukážeme **metodu nejbližšího souseda**.

Pekárna v Kralupech vypravuje každé ráno dodávku, která zásobuje prodejny potravin v okolních městech a poté se vrací zpět. Doporučte, v jakém pořadí má města projet, aby ujel co nejméně km. Tabulka vzdáleností je následující:

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza
Kralupy	0	4	16	8	18	25	17
Veltrusy	4	0	20	12	22	28	13
Slaný	16	20	0	12	7	14	17
Velvary	8	12	12	0	10	17	10
Zlonice	18	22	7	10	0	7	10
Vraný	25	28	14	17	7	0	15
Bříza	17	13	17	10	10	15	0

Metodu je dobré zkusit nastartovat i odjinud. Start z Veltrus dává okruh Veltrusy - Kralupy - Velvary - Zlonice - Slaný - Vraný - Bříza - Veltrusy o celkové délce $4+8+10+7+14+15+13 = 71$ km.

Okružní dopravní problém - přibližné řešení

K nalezení přibližného řešení lze využít heuristické metody, např. **hladový algoritmus**. Na úloze z knihy "J. Pelikán, V. Chýna: Kvantitativní management", ukážeme **metodu nejbližšího souseda**.

Pekárna v Kralupech vypravuje každé ráno dodávku, která zásobuje prodejny potravin v okolních městech a poté se vrací zpět. Doporučte, v jakém pořadí má města projet, aby ujel co nejméně km. Tabulka vzdáleností je následující:

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza
Kralupy	0	4	16	8	18	25	17
Veltrusy	4	0	20	12	22	28	13
Slaný	16	20	0	12	7	14	17
Velvary	8	12	12	0	10	17	10
Zlonice	18	22	7	10	0	7	10
Vraný	25	28	14	17	7	0	15
Bříza	17	13	17	10	10	15	0

Start z Vraného dá okruh Vraný - Zlonice - Slaný - Velvary - Kralupy - Veltrusy - Bříza - Vraný o celkové délce $7+7+12+8+4+13+15 = 66$ km, což je dokonce nejkratší možná trasa.

Okružní dopravní problém - přibližné řešení

Další heuristikou je **metoda výhodnostních čísel**. Je založena na srovnání délky trasy mezi městy $1 - i - j - 1$ a součtu délek cest $1 - i - 1, 1 - j - 1$. Výhodnostní číslo S_{ij} pak definujeme jako rozdíl těchto délek, neboli výhodnost propojení i a j : $S_{ij} = c_{1i} + c_{1j} - c_{ij}$.
Spočtěme matici výhodnostních čísel pro úlohu o pekařství:

Okružní dopravní problém - přibližné řešení

Další heuristikou je **metoda výhodnostních čísel**. Je založena na srovnání délky trasy mezi městy $1 - i - j - 1$ a součtu délek cest $1 - i - 1, 1 - j - 1$. Výhodnostní číslo S_{ij} pak definujeme jako rozdíl těchto délek, neboli výhodnost propojení i a j : $S_{ij} = c_{1i} + c_{1j} - c_{ij}$.

Spočtíme matici výhodnostních čísel pro úlohu o pekařství:

	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza
Veltrusy	X	0	0	0	1	8
Slaný	0	X	12	27	27	16
Velvary	0	12	X	16	16	15
Zlonice	0	27	16	X	36	25
Vraný	1	27	16	36	X	27
Bříza	8	16	15	25	27	X

Trasu tvoříme postupně (hladově) od největších výhodnostních čísel, přičemž se bráníme předčasnému uzavření okruhu. Dostaneme trasu Slaný - Zlonice - Vraný - Bříza - Velvary - Veltrusy. Startovní a koncové zůstávají Kralupy, dohromady tedy máme $16+7+7+15+10+12+4 = 71$ km.

Úloha celočíselného programování

Mezi speciální úlohy lineárního programování patří i úlohy celočíselného programování (integer programming, IP). Jedná se o standardní úlohy LP doplněné o podmínky celočíselnosti u některých (**smíšené úlohy IP**), případně všech proměnných (**ryze celočíselné úlohy**). Tyto podmínky zpravidla plynou přímo z ekonomického modelu, kdy proměnné vyjadřují počty kusů nedělitelných produktů, počty opakování nějaké aktivity, apod. U řady úloh se pracuje dokonce jen s proměnnými, které vyjadřují určité rozhodnutí nebo alternativu, nabývají hodnot $\{0, 1\}$ a pak hovoříme o **bivalentních úlohách** .

Úloha celočíselného programování

Mezi speciální úlohy lineárního programování patří i úlohy celočíselného programování (integer programming, IP). Jedná se o standardní úlohy LP doplněné o podmínky celočíselnosti u některých (**smíšené úlohy IP**), případně všech proměnných (**ryze celočíselné úlohy**). Tyto podmínky zpravidla plynou přímo z ekonomického modelu, kdy proměnné vyjadřují počty kusů nedělitelných produktů, počty opakování nějaké aktivity, apod. U řady úloh se pracuje dokonce jen s proměnnými, které vyjadřují určité rozhodnutí nebo alternativu, nabývají hodnot $\{0, 1\}$ a pak hovoříme o **bivalentních úlohách** . Typickými představiteli takových úloh jsou například přiřazovací či okružní dopravní problém nebo též tzv. **"úloha o batohu"**:

Máme n různě cenných věcí s různou hmotností a batoh o omezené kapacitě. Úkolem je vybrat věci, které vložíme do batohu tak, aby nebyl přetížen a cena jeho obsahu byla co nejvyšší. (Řešení úlohy bez podmínky celočíselnosti by bylo triviální: Seřadili bychom věci dle klesajícího poměru cena/hmotnost a plnili batoh v tomto pořadí. U první věci, která by se nevešla celá, bychom vzali jen její poměrnou část do výše kapacity batohu.)

Celočíselné programování- příklad

Příklad: Oděvní firma Styl, s.r.o. se zabývá se výrobou pánské módy. Za týden pracovníce ušijí x_1 modelů "Marcel" a x_2 modelů "Filip", Přitom na výrobu jednoho modelu "Marcel" je potřeba 10 hodin a náklady na materiál jsou 400 Kč. Pro model "Filip" je to 20 hodin, materiál 300 Kč. Firma dostala nabídku na účast na zahraničním prodejním veletrhu, který se koná za týden. Očekávaný zisk z prodeje je 20 Euro, resp. 30 Euro pro jednotlivé modely. Navrhněte optimální plán výroby, jestliže firma má k dispozici jednu pracovníci na plný a jednu na poloviční úvazek (celkem 60 h) a zásobu materiálu za 1300Kč.

Celočíselné programování- příklad

Příklad: Oděvní firma Styl, s.r.o. se zabývá se výrobou pánské módy. Za týden pracovníce ušijí x_1 modelů "Marcel" a x_2 modelů "Filip", Přitom na výrobu jednoho modelu "Marcel" je potřeba 10 hodin a náklady na materiál jsou 400 Kč. Pro model "Filip" je to 20 hodin, materiál 300 Kč. Firma dostala nabídku na účast na zahraničním prodejním veletrhu, který se koná za týden. Očekávaný zisk z prodeje je 20 Euro, resp. 30 Euro pro jednotlivé modely. Navrhněte optimální plán výroby, jestliže firma má k dispozici jednu pracovníci na plný a jednu na poloviční úvazek (celkem 60 h) a zásobu materiálu za 1300Kč.

Matematický zápis úlohy z uvedeného příkladu je následující: Maximalizovat účelovou funkci

$$z = 20x_1 + 30x_2$$

za podmínek

$$400x_1 + 300x_2 \leq 1300$$

$$10x_1 + 20x_2 \leq 60$$

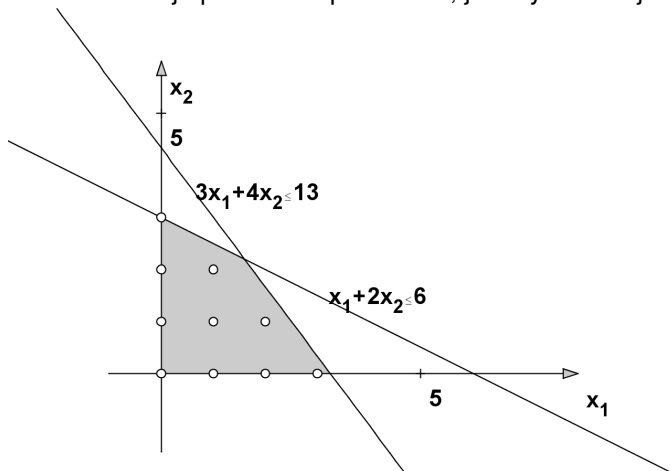
$$x_1, x_2 \in \{0, 1, 2, 3, \dots\}$$

Celočíselné programování- grafické znázornění

Úloha obsahuje pouze dvě proměnné, je tedy možné ji řešit graficky.

Celočíselné programování- grafické znázornění

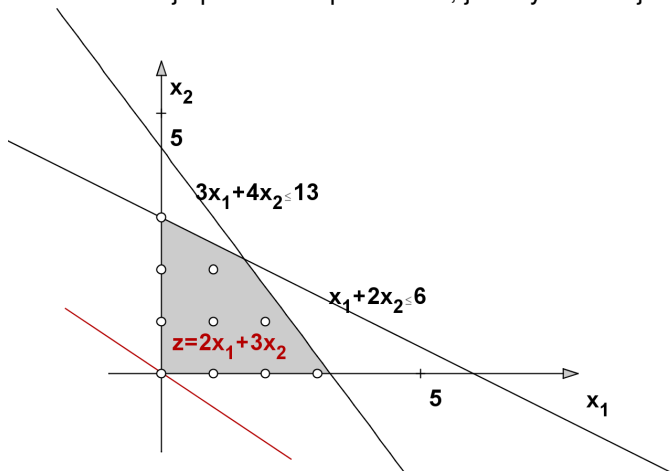
Úloha obsahuje pouze dvě proměnné, je tedy možné ji řešit graficky.



Znázorněme přípustnou množinu. Jsou to body o celočíselných souřadnicích ležící uvnitř čtyřúhelníka.

Celočíselné programování- grafické znázornění

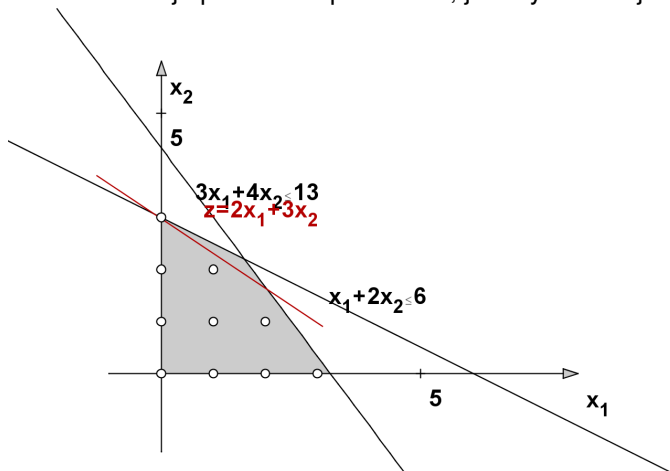
Úloha obsahuje pouze dvě proměnné, je tedy možné ji řešit graficky.



Přidáme nulovou izokvantu účelové funkce, vyjadřujeme v desítkách Eur.

Celočíselné programování- grafické znázornění

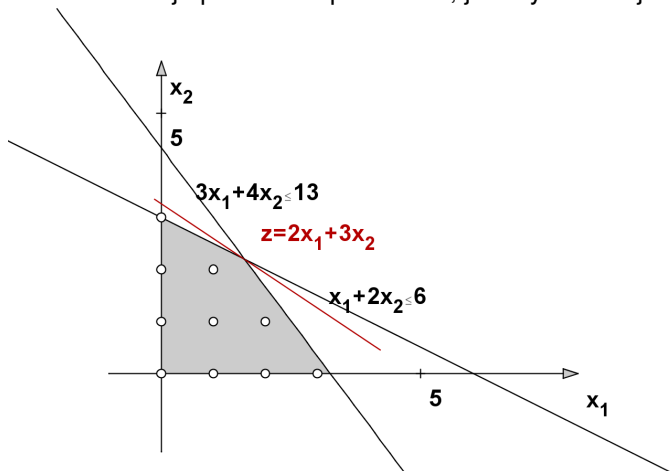
Úloha obsahuje pouze dvě proměnné, je tedy možné ji řešit graficky.



Nejvyšší izokvanta účelové funkce prochází bodem $[0, 3]$, což je hledané optimální řešení x^* . Získáme $3 \cdot 30 = 90$ Euro.

Celočíselné programování- grafické znázornění

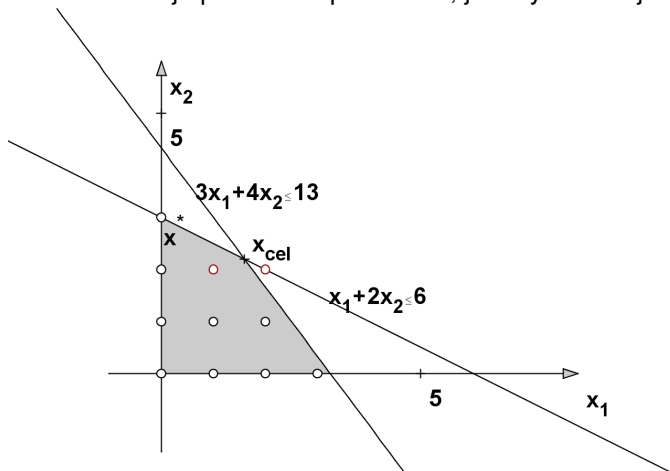
Úloha obsahuje pouze dvě proměnné, je tedy možné ji řešit graficky.



Optimální řešení úlohy bez omezení celočíselnosti je $x_{cel} = [1, 6; 2, 2]$, hodnota účelové funkce v tomto bodě je $Z_{cel} = 98$ Euro.

Celočíselné programování- grafické znázornění

Úloha obsahuje pouze dvě proměnné, je tedy možné ji řešit graficky.



Zakrouhlením $x_{cel} = [1, 6; 2, 2]$ dostaneme bod $[2, 2]$, který není přípustný. Při zakrouhlení dolů na $[1, 2]$ dostaneme neoptimální řešení, dává pouze $20 + 2 \cdot 30 = 80$ Euro oproti 90 Euro v bodě x^* .

Celočíselné programování- metody

V předchozím příkladě jsme viděli, že intuitivní přístup k řešení celočíselné úlohy zanedbat podmínky celočíselnosti, vyřešit získanou úlohu LP, tzv. **zrelaxovanou úlohu** a výsledné řešení zaokrouhlit, není vždy vhodný. Proto se používají jiné postupy vyvinuté speciálně pro celočíselné úlohy. Bývají však výpočetně náročnější, na běžných počítačích může výpočet zkolabovat i při počtu proměnných a omezení v řádu desítek.

Celočíselné programování- metody

V předchozím příkladě jsme viděli, že intuitivní přístup k řešení celočíselné úlohy zanedbat podmínky celočíselnosti, vyřešit získanou úlohu LP, tzv. **zrelaxovanou úlohu** a výsledné řešení zaokrouhlit, není vždy vhodný. Proto se používají jiné postupy vyvinuté speciálně pro celočíselné úlohy. Bývají však výpočetně náročnější, na běžných počítačích může výpočet zkolabovat i při počtu proměnných a omezení v řádu desítek.

Metody řezných nadrovin (např. Gomoryho algoritmus) spočívají v hledání optima zrelaxované úlohy a následném přidání dalšího omezení, které toto optimum "odřízne", ale přitom mu budou vyhovovat všechna přípustná řešení původní úlohy. Tyto metody jsou starší a méně používané.

Celočíselné programování- metody

V předchozím příkladě jsme viděli, že intuitivní přístup k řešení celočíselné úlohy zanedbat podmínky celočíselnosti, vyřešit získanou úlohu LP, tzv. **zrelaxovanou úlohu** a výsledné řešení zaokrouhlit, není vždy vhodný. Proto se používají jiné postupy vyvinuté speciálně pro celočíselné úlohy. Bývají však výpočetně náročnější, na běžných počítačích může výpočet zkolabovat i při počtu proměnných a omezení v řádu desítek.

Metody řezných nadrovin (např. Gomoryho algoritmus) spočívají v hledání optima zrelaxované úlohy a následném přidání dalšího omezení, které toto optimum "odřízne", ale přitom mu budou vyhovovat všechna přípustná řešení původní úlohy. Tyto metody jsou starší a méně používané.

Kombinatorické metody jsou založeny na efektivním prohledávání přípustné množiny, která pro ryze celočíselnou úlohu obsahuje konečně, avšak zpravidla velmi mnoho prvků. V programových systémech se nejčastěji využívá metody **větví a mezí** (též Branch and Bound, tj. B & B), kterou si blíže popíšeme.

Celočíselné programování- metody

V předchozím příkladě jsme viděli, že intuitivní přístup k řešení celočíselné úlohy zanedbat podmínky celočíselnosti, vyřešit získanou úlohu LP, tzv. **zrelaxovanou úlohu** a výsledné řešení zaokrouhlit, není vždy vhodný. Proto se používají jiné postupy vyvinuté speciálně pro celočíselné úlohy. Bývají však výpočetně náročnější, na běžných počítačích může výpočet zkolabovat i při počtu proměnných a omezení v řádu desítek.

Metody řezných nadrovin (např. Gomoryho algoritmus) spočívají v hledání optima zrelaxované úlohy a následném přidání dalšího omezení, které toto optimum "odřízne", ale přitom mu budou vyhovovat všechna přípustná řešení původní úlohy. Tyto metody jsou starší a méně používané.

Kombinatorické metody jsou založeny na efektivním prohledávání přípustné množiny, která pro ryze celočíselnou úlohu obsahuje konečně, avšak zpravidla velmi mnoho prvků. V programových systémech se nejčastěji využívá metody **větví a mezí** (též Branch and Bound, tj. B & B), kterou si blíže popíšeme.

Speciální metody se používají pro úlohy se speciální strukturou, jde např. o různé heuristiky pro okružní dopravní problém nebo Maďarskou metodu.

Celočíselné programování- metoda B & B

Množina přípustných řešení se postupně dělí na menší části (branching), kde sledujeme horní, (při minimalizaci dolní) hranici hodnot účelové funkce (bounding). To nám umožní vytipovat podmnožinu, kde nejpravděpodobněji nastane optimum a také podmnožiny, kde optimum určitě nebude.

Celočíselné programování- metoda B & B

Množina přípustných řešení se postupně dělí na menší části (branching), kde sledujeme horní, (při minimalizaci dolní) hranici hodnot účelové funkce (bounding). To nám umožní vytipovat podmnožinu, kde nejpravděpodobněji nastane optimum a také podmnožiny, kde optimum určitě nebude.

Větvení je možné provést pomocí řešení klasické úlohy LP získané relaxací celočíselné úlohy - přípustnou množinu označme M_0 . Je-li její bod optima x^0 celočíselný, našli jsme již optimální řešení původní úlohy. Jinak vybereme nějaké i , pro něž je i -tá složka řešení x_i^0 necelá, a přidáme dodatečné omezení $x_i \leq a$, resp. $x_i \geq b$, kde a , b jsou celá čísla obklopující x_i . Tím rozdělíme množinu M_0 na podmnožiny M_1 a M_2 .

Celočíselné programování- metoda B & B

Množina přípustných řešení se postupně dělí na menší části (branching), kde sledujeme horní, (při minimalizaci dolní) hranici hodnot účelové funkce (bounding). To nám umožní vytipovat podmnožinu, kde nejpravděpodobněji nastane optimum a také podmnožiny, kde optimum určitě nebude.

Větvení je možné provést pomocí řešení klasické úlohy LP získané relaxací celočíselné úlohy - přípustnou množinu označme M_0 . Je-li její bod optima x^0 celočíselný, našli jsme již optimální řešení původní úlohy. Jinak vybereme nějaké i , pro něž je i -tá složka řešení x_i^0 necelá, a přidáme dodatečné omezení $x_i \leq a$, resp. $x_i \geq b$, kde a , b jsou celá čísla obklopující x_i . Tím rozdělíme množinu M_0 na podmnožiny M_1 a M_2 .

Na každé z nich zase najdeme optimum účelové funkce x^1 a x^2 a určíme jejich hodnoty účelové funkce z^1 a z^2 . Celé části těchto hodnot nám dávají **horní mez** pro účelovou funkci na množinách M_1 a M_2 . Celý proces pokračuje tak dlouho, dokud se všechny větve neuzavřou tak, že:

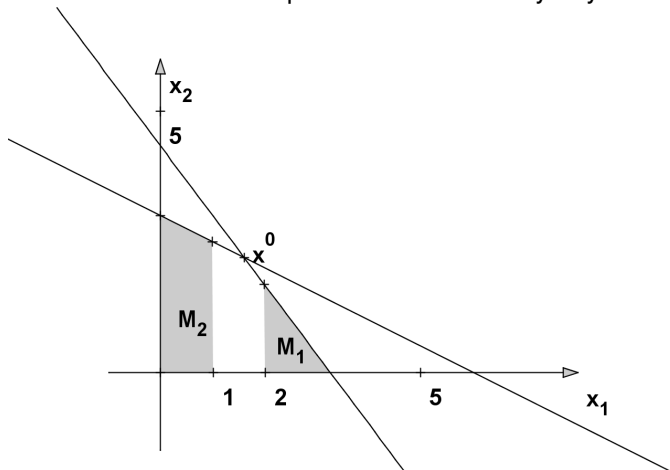
- Ve větvi je nalezeno celočíselné řešení nebo
- ve větvi neexistuje přípustné řešení nebo
- ve větvi je nalezeno necelé řešení, jehož hodnota je menší než hodnota celočíselného řešení z jiné větve

Celočíselné programování- použití metody B & B

Ukažme metodu B & B pro úlohu rozvržení výroby textilu

Celočíselné programování- použití metody B & B

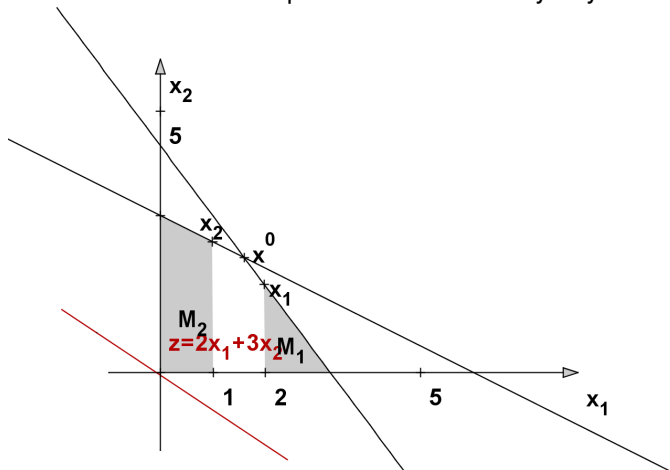
Ukažme metodu B & B pro úlohu rozvržení výroby textilu



Již máme řešení relaxace úvodní úlohy $x^0 = x_{cel} = [1, 6; 2, 2]$ Jeho první složka je necelá, přidáme tedy omezení $x_1 \geq 2$, resp. $x_1 \leq 1$, čímž vytvoříme množiny M_1 a M_2 .

Celočíselné programování- použití metody B & B

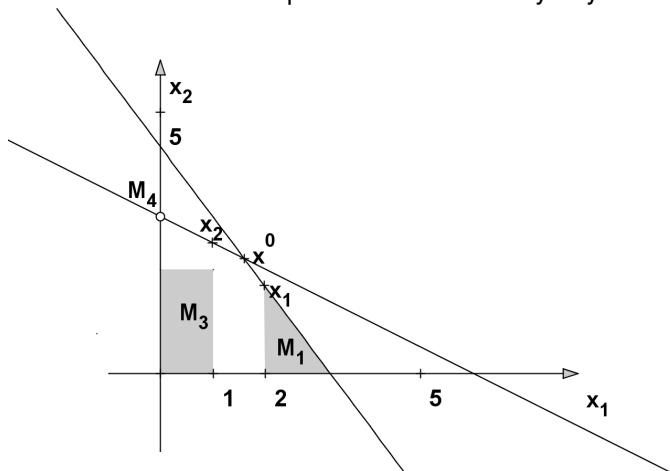
Ukažme metodu B & B pro úlohu rozvržení výroby textilu



Na množině M_1 dostaneme optimum $x^1 = [2; \frac{5}{3}]$ a na M_2 řešení $x^2 = [1; \frac{5}{2}]$. Jejich hodnoty jsou $z^1 = 90$ a $z^2 = 95$, což dává horní meze pro M_1 a M_2 .

Celočíselné programování- použití metody B & B

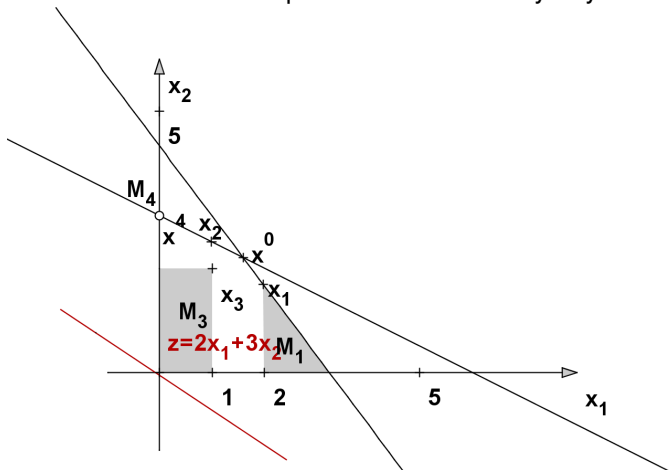
Ukažme metodu B & B pro úlohu rozvržení výroby textilu



Množina M_2 má vyšší horní mez, takže s dělením pokračujeme na ní. Druhá složka x^2 je rovna $\frac{5}{2}$, přidáme tedy omezení $x_2 \leq 2$, resp. $x_2 \geq 3$, čímž vytvoříme množiny M_3 a M_4 (ta je jen jednobodová).

Čeločíselné programování- použití metody B & B

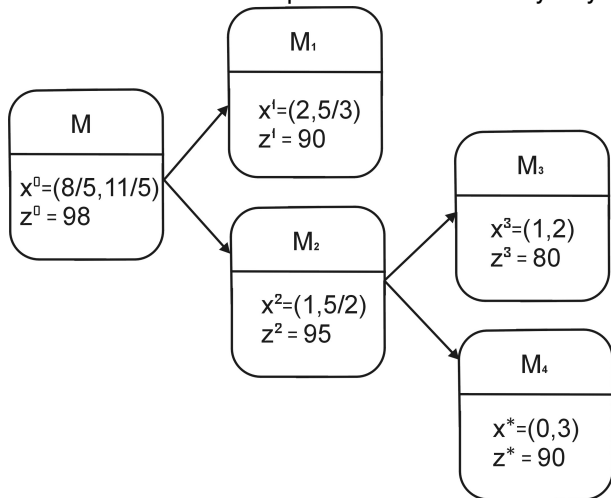
Ukažme metodu B & B pro úlohu rozvržení výroby textilu



Řešení x^3 a x^4 jsou již obě celočíselná, hodnoty účelové funkce v nich jsou $z^3 = 80$ a $z^4 = 90$. Výpočet končí, na žádné větvi (ani M_1) už nemůžeme dostat více než 90 Euro v bodě optima $x^4 = [0, 3]$.

Celočíselné programování- použití metody B & B

Ukažme metodu B & B pro úlohu rozvržení výroby textilu



Postup řešení můžeme znázornit schematicky.

Cílové programování

Představuje jiný přístup k řešení úlohy LP. Místo stanovení omezujících podmínek a kritéria optimality se zde stanoví **pevné a volné cíle** s přiřazenými **cílovými hodnotami**. U pevných cílů musí být tato hodnota splněna (analogie omezujících podmínek). U volných cílů můžeme dosáhnout vyšší i nižší hodnoty než je cíl (avšak ne moc odlišné). Protože cílových hodnot bývá několik a zpravidla není možné dosažení všech, volí se obvykle jeden ze dvou přístupů:

- pomocí **preferencí** - nejprve je optimalizován cíl s nejvyšší preferencí, atd. nebo
- pomocí **vah** - koeficientů vyjadřujících důležitost daného cíle, optimalizuje se pak vážený součet odchylek od všech cílů

Modely cílového programování jsou obecnější než standardní modely LP, protože v praxi zpravidla nemáme jediné kritérium optimality, ale sledujeme více hodnot.

Cílové programování - příklad

Příklad z J. Jablonský, Operační výzkum: Vedení penzijního fondu se rozhoduje o nákupu dvou druhů cenných papírů (akcie a obligace). Má k dispozici prostředky, které není nutné úplně vyčerpat, nelze je však překročit. Do akcií lze investovat maximálně 50 % celkového objemu prostředků a do obligací maximálně 75 % prostředků. Očekávaný výnos z **akcií je 15 %** a z **obligací 10 %** p.a., míra rizika investice je ohodnocena koeficientem **5 u akcií** a **2 u obligací**. Navrhněte takovou skladbu portfolia $[x_1, x_2]$, aby se dosáhlo průměrného **výnosu 12 %** p.a. a aby byla vážená míra **rizika rovna 3**.

Cílové programování - příklad

Příklad z J. Jablonský, Operační výzkum: Vedení penzijního fondu se rozhoduje o nákupu dvou druhů cenných papírů (akcie a obligace). Má k dispozici prostředky, které není nutné úplně vyčerpat, nelze je však překročit. Do akcií lze investovat maximálně 50 % celkového objemu prostředků a do obligací maximálně 75 % prostředků. Očekávaný výnos z **akcií je 15 %** a z **obligací 10 %** p.a., míra rizika investice je ohodnocena koeficientem **5 u akcií** a **2 u obligací**. Navrhněte takovou skladbu portfolia $[x_1, x_2]$, aby se dosáhlo průměrného **výnosu 12 %** p.a. a aby byla vážená míra **rizika rovna 3**.

Při standardním přístupu LP bychom museli zvolit jako účelovou funkci jen jedno z nich, např. výnos a k omezujícím podmínkám pro objem investic přidat další omezení, totiž že míra rizika nesmí překročit hodnotu 3. Lze spočítat, že optimálního řešení při tomto přístupu dosáhneme, dáme-li $\frac{1}{3}$ prostředků do akcií a $\frac{2}{3}$ do obligací. Při průměrném **riziku 3** tak získáme **11,67 %** p.a.

Cílové programování - příklad

Příklad z J. Jablonský, Operační výzkum: Vedení penzijního fondu se rozhoduje o nákupu dvou druhů cenných papírů (akcie a obligace). Má k dispozici prostředky, které není nutné úplně vyčerpat, nelze je však překročit. Do akcií lze investovat maximálně 50 % celkového objemu prostředků a do obligací maximálně 75 % prostředků. Očekávaný výnos z **akcií je 15 %** a z **obligací 10 %** p.a., míra rizika investice je ohodnocena koeficientem **5 u akcií** a **2 u obligací**. Navrhněte takovou skladbu portfolia $[x_1, x_2]$, aby se dosáhlo průměrného **výnosu 12 %** p.a. a aby byla vážená míra **rizika rovna 3**.

Při standardním přístupu LP bychom museli zvolit jako účelovou funkci jen jedno z nich, např. výnos a k omezujícím podmínkám pro objem investic přidat další omezení, totiž že míra rizika nesmí překročit hodnotu 3. Lze spočítat, že optimálního řešení při tomto přístupu dosáhneme, dáme-li $\frac{1}{3}$ prostředků do akcií a $\frac{2}{3}$ do obligací. Při průměrném **riziku 3** tak získáme **11,67 %** p.a. Nebo naopak budeme minimalizovat účelovou funkci vyjádřenou váženým rizikem a jako dodatečnou podmínku stanovíme omezení, aby průměrný výnos byl nejméně 12 % p.a. Pak je optimální investovat 40 % prostředků do akcií a 60 % do obligací. Při výsledném výnosu **12 %** p.a. pak bude míra **rizika 3,2**.

Cílové programování - formulace modelu

U volných cílů se používají **odchylkové proměnné** pro vyjádření kladných a záporných odchylek (značíme je d_i^+ , resp. d_i^-) od cílových hodnot. Je-li cíle splněn, platí $d_i^+ = d_i^- = 0$, dojde-li k přesáhnutí cíle, pak je $d_i^+ > 0$, $d_i^- = 0$ a není-li cíl dosažen, je $d_i^+ = 0$, $d_i^- > 0$. Pevné cíle musí být respektovány, žádné odchylky se nepřipouští.

Cílové programování - formulace modelu

U volných cílů se používají **odchylkové proměnné** pro vyjádření kladných a záporných odchylek (značíme je d_i^+ , resp. d_i^-) od cílových hodnot. Je-li cíle splněn, platí $d_i^+ = d_i^- = 0$, dojde-li k přesáhnutí cíle, pak je $d_i^+ > 0$, $d_i^- = 0$ a není-li cíl dosažen, je $d_i^+ = 0$, $d_i^- > 0$. Pevné cíle musí být respektovány, žádné odchylky se nepřipouští.

V modelu cílového programování je vždy účelová funkce vyjádřena jako **minimalizace odchylkových proměnných**, přičemž do ní lze zahrnout buď pouze kladné, pouze záporné nebo oba typy odchylek (pak se cílovým hodnotám blížíme shora, zdola nebo "oboustranně"). Zápis úlohy o optimalizaci portfolia by tedy byl:

$$d_2^+, d_1^- \rightarrow \min,$$

za podmíněk:

$$x_1 + x_2 \leq 1$$

$$x_1 \leq 0,5; x_2 \leq 0,75$$

$$15x_1 + 10x_2 + d_1^+ - d_1^- = 12$$

$$5x_1 + 2x_2 + d_2^+ - d_2^- = 3$$

$$x_1, x_2, d_1^+, d_1^-, d_2^+, d_2^- \geq 0$$

Cílové programování - odlišení cílů vahami

Při současné minimalizaci více odchylek můžeme odlišit jejich důležitost **vahami**. Abychom se vyhnuli problémům s různými jednotkami, je lepší pracovat s relativními odchylkami $\frac{d_i^+}{g_i}$, $\frac{d_i^-}{g_i}$, kde g_i značí i -tou cílovou hodnotu. Pokud je pro nás výnos pětkrát důležitější než riziko, stanovíme váhy 5 a 1 a účelová funkce bude mít podobu $z = 5 \frac{d_1^-}{12} + \frac{d_2^+}{3}$. Dále lze úlohu řešit běžnou simplexovou metodou.

Cílové programování - odlišení cílů vahami

Při současné minimalizaci více odchylek můžeme odlišit jejich důležitost **vahami**. Abychom se vyhnuli problémům s různými jednotkami, je lepší pracovat s relativními odchylkami $\frac{d_i^+}{g_i}$, $\frac{d_i^-}{g_i}$, kde g_i značí i -tou cílovou hodnotu. Pokud je pro nás výnos pětkrát důležitější než riziko, stanovíme váhy 5 a 1 a účelová funkce bude mít podobu $z = 5 \frac{d_1^-}{12} + \frac{d_2^+}{3}$. Dále lze úlohu řešit běžnou simplexovou metodou.

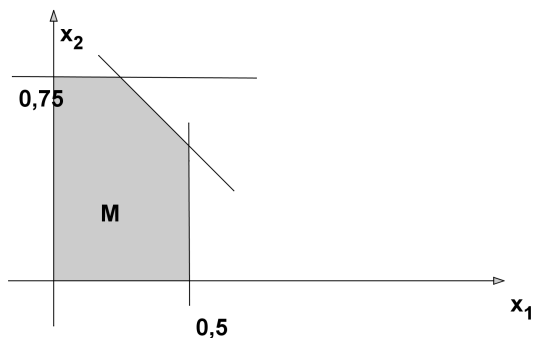
Řešením je dát $\frac{1}{3}$ prostředků do akcií a $\frac{2}{3}$ do obligací, přitom je zcela splněna cílová hodnota rizika a výnos je 11,67%.

Cílové programování - odlišení cílů preferencemi

Při odlišení cílů **preferencemi** minimalizujeme nejprve odchylku od důležitějšího cíle a pokud má množina optimálních řešení více prvků, minimalizujeme na této množině druhou nejzávažnější odchylku, atd. Má-li v naší úloze vyšší prioritu výnos, minimalizujeme nejprve d_1^- .

Cílové programování - odlišení cílů preferencemi

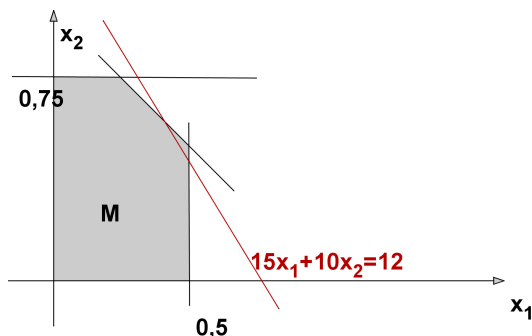
Při odlišení cílů **preferencemi** minimalizujeme nejprve odchylku od důležitějšího cíle a pokud má množina optimálních řešení více prvků, minimalizujeme na této množině druhou nejzávažnější odchylku, atd. Má-li v naší úloze vyšší prioritu výnos, minimalizujeme nejprve d_1^- . Znázorníme situaci graficky v rovině x_1, x_2 .



Nejprve si zobrazíme přípustnou množinu M , na níž jsou splněny všechny pevné cíle.

Cílové programování - odlišení cílů preferencemi

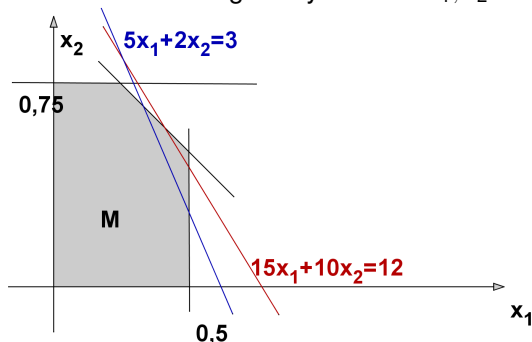
Při odlišení cílů **preferencemi** minimalizujeme nejprve odchylku od důležitějšího cíle a pokud má množina optimálních řešení více prvků, minimalizujeme na této množině druhou nejzávažnější odchylku, atd. Má-li v naší úloze vyšší prioritu výnos, minimalizujeme nejprve d_1^- . Znázorníme situaci graficky v rovině x_1, x_2 .



Přímka o rovnici $15x_1 + 10x_2 = 12$ vyjadřuje dosažení nejmenší možné hodnoty $d_1^- = 0$. Její průnik s M představuje všechna optimální řešení v prvním kroku.

Cílové programování - odlišení cílů preferencemi

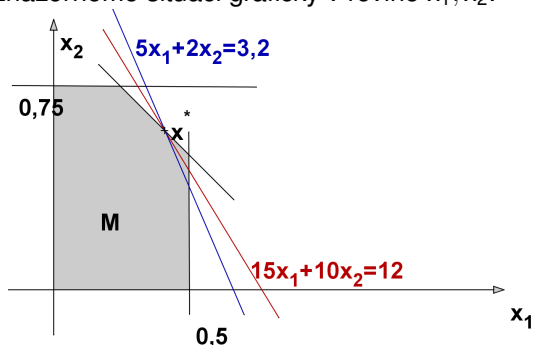
Při odlišení cílů **preferencemi** minimalizujeme nejprve odchylku od důležitějšího cíle a pokud má množina optimálních řešení více prvků, minimalizujeme na této množině druhou nejzávažnější odchylku, atd. Má-li v naší úloze vyšší prioritu výnos, minimalizujeme nejprve d_1^- . Znázorníme situaci graficky v rovině x_1, x_2 .



V druhém kroku minimalizujeme odchylku d_2^+ . Ideální by bylo, kdyby byla nulová, tj. kdyby platila rovnost $5x_1 + 2x_2 = 3$.

Cílové programování - odlišení cílů preferencemi

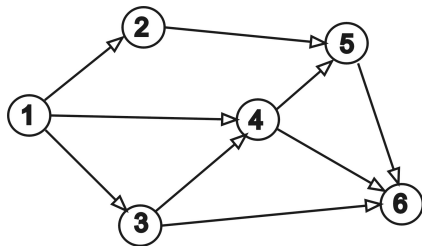
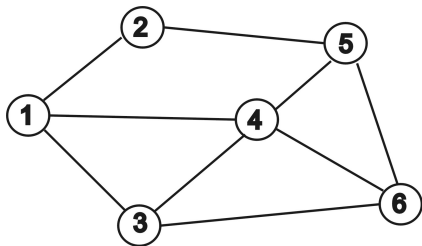
Při odlišení cílů **preferencemi** minimalizujeme nejprve odchylku od důležitějšího cíle a pokud má množina optimálních řešení více prvků, minimalizujeme na této množině druhou nejzávažnější odchylku, atd. Má-li v naší úloze vyšší prioritu výnos, minimalizujeme nejprve d_1^- . Znázorníme situaci graficky v rovině x_1, x_2 .



Protože se přímky protínají mimo přípustnou množinu, nelze dosáhnout rovnosti $d_2^+ = 0$. Musíme riziko zvýšit, tj. posunout modrou přímku tak, aby se dotkla optimální úsečky pro d_1^- . Našli jsme bod optima $x^* = [0, 4; 0, 6]$.

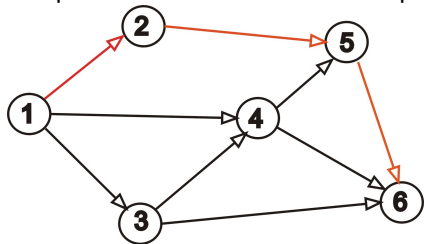
Optimalizace v grafech - základní pojmy

Řadu reálných systémů (např. distribuční síť) lze modelovat rovinnými grafy. Graf je tvořen **uzly**, které budeme značit u_1, \dots, u_n a **hranami**, přičemž hranu mezi uzly u_i a u_j označíme h_{ij} . V rovině můžeme znázornit graf pomocí bodů (koleček) a spojnic mezi nimi. Hranu, které umožňují pohyb v obou směrech nazýváme **neorientované**. Je-li povolen pouze jeden směr pohybu, znázorníme to na grafu šipkou a takové "jednosměrné"hrany nazýváme **orientované**. Neorientovaným grafem nazveme graf obsahující pouze neorientované hrany, jinak jej nazveme orientovaným. Na obrázku je znázorněn neorientovaný a orientovaný graf.



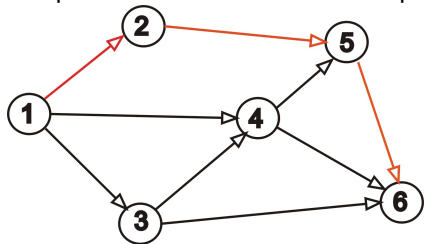
Optimalizace v grafech - základní pojmy

Cestou z uzlu u_i do uzlu u_j nazveme posloupnost na sebe navazujících hran, z nichž první začíná v u_i a poslední končí v u_j . Pokud cesta respektuje orientaci hran, nazývá se orientovaná (v opačném případě neorientovaná). Na obrázku je znázorněna jedna z orientovaných cest z uzlu 1 do uzlu 6. Naopak z uzlu 6 do uzlu 1 vedou pouze neorientované cesty.



Optimalizace v grafech - základní pojmy

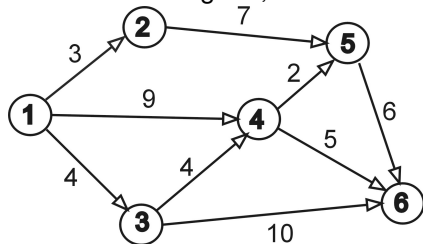
Cestou z uzlu u_i do uzlu u_j nazveme posloupnost na sebe navazujících hran, z nichž první začíná v u_i a poslední končí v u_j . Pokud cesta respektuje orientaci hran, nazývá se orientovaná (v opačném případě neorientovaná). Na obrázku je znázorněna jedna z orientovaných cest z uzlu 1 do uzlu 6. Naopak z uzlu 6 do uzlu 1 vedou pouze neorientované cesty.



Cestu, pro kterou $u_i = u_j$, nazveme **cyklus**, v případě neorientovaného grafu **kružnice**. Zobrazený graf obsahuje například kružnici 1 – 3 – 4 – 1. Graf, ve kterém mezi libovolnými dvěma uzly existuje aspoň jedna neorientovaná cesta, se nazývá **souvislý**. Každý souvislý neorientovaný graf, který neobsahuje kružnici, se nazývá **strom**.

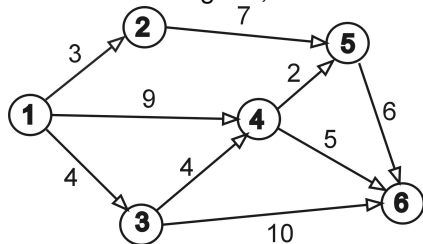
Optimalizace v grafech - základní pojmy

Při řešení optimalizačních úloh zpravidla pracujeme s **hranově ohodnocenými** grafy. Hranám jsou přiřazeny hodnoty y_{ij} podle ekonomického významu (např. vzdálenosti mezi distribučními centry či náklady na přepravu mezi centry, apod.) Souvislý orientovaný a nezáporně ohodnocený graf se dvěma speciálními uzly (vstupem a výstupem) nazveme **síť**. Přidáme-li ohodnocení hran do našeho grafu, získáme síť se vstupním uzlem 1 a výstupním uzlem 6.



Optimalizace v grafech - základní pojmy

Při řešení optimalizačních úloh zpravidla pracujeme s **hranově ohodnocenými** grafy. Hranám jsou přiřazeny hodnoty y_{ij} podle ekonomického významu (např. vzdálenosti mezi distribučními centry či náklady na přepravu mezi centry, apod.) Souvislý orientovaný a nezáporně ohodnocený graf se dvěma speciálními uzly (vstupem a výstupem) nazveme **sít'**. Přidáme-li ohodnocení hran do našeho grafu, získáme síť se vstupním uzlem 1 a výstupním uzlem 6.



Délkou cesty nazveme součet ohodnocení jejích hran. Například mezi délka orientované cesty 1-2-5-6 je $3+7+6=16$. **Pozor!** Graf je definován pomocí množiny uzlů a množiny hran, nikoliv zakreslením. Délky spojnic nemusí a často ani nemohou odpovídat ohodnocení hran.

Optimalizace v grafech - úlohy

Na grafech se řeší řada úloh:

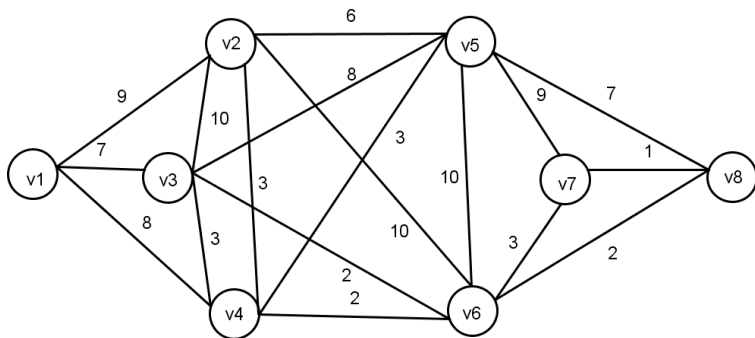
- Standartní optimalizační úlohou je hledání **nejkratší cesty** mezi dvěma uzly. Úloha se řeší v orientovaných i neorientovaných grafech. existuje více algoritmů, některé k určení celé matice vzdáleností. Jeden z nejznámějších algoritmů je Dijkstrův algoritmus.
- Hledání **minimální kostry** grafu - úkolem je vybrat takovou podmnožinu hran, aby mezi každými dvěma uzly existovala cesta a aby celkové ohodnocení bylo minimální (kostra nesmí obsahovat cyklus).
- Určení **maximálního toku** v síti (propustnosti sítě): Představuje - li ohodnocení v síti přepravní kapacitu hran, pak úkolem je určení maximálního počtu jednotek, které je možné přepravit ze vstupního do výstupního uzlu.
- Další úlohy, jako problém barvení grafu, problém čínského pošťáka, problém obchodního cestujícího, medián grafu, centr grafu, atd.

Optimalizace v grafech - minimální kostra

Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezněme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:

Optimalizace v grafech - minimální kostra

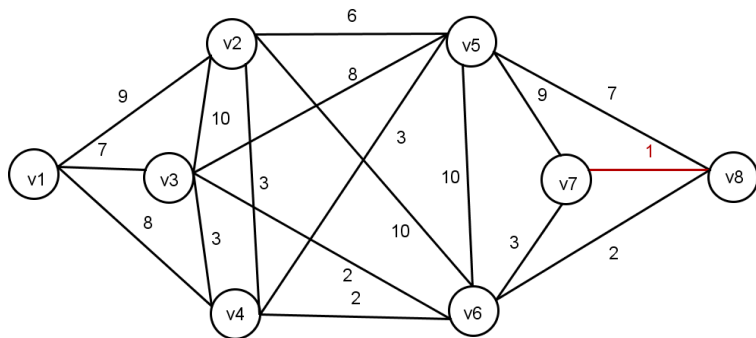
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezněme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Nejprve přidáme hranu s hodnotou 1.

Optimalizace v grafech - minimální kostra

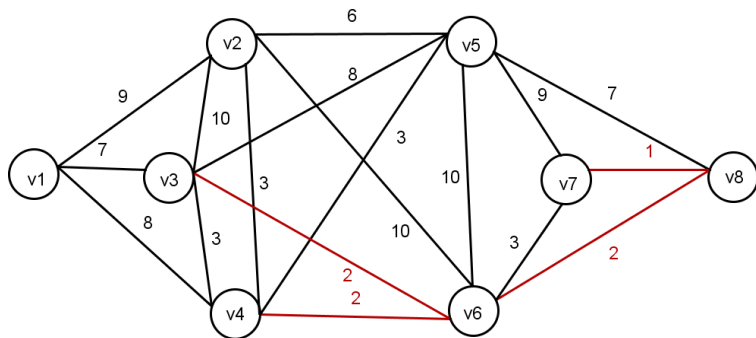
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezněme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Dále vybereme všechny hrany s hodnotou 2.

Optimalizace v grafech - minimální kostra

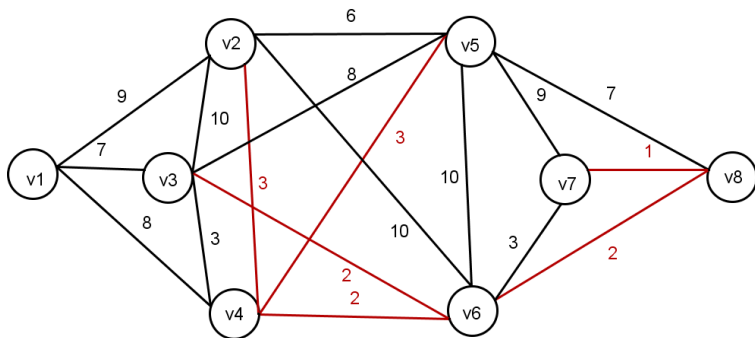
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezneme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Přidáme hrany s hodnotou 3 kromě v3-v4, která by uzavřela okruh.

Optimalizace v grafech - minimální kostra

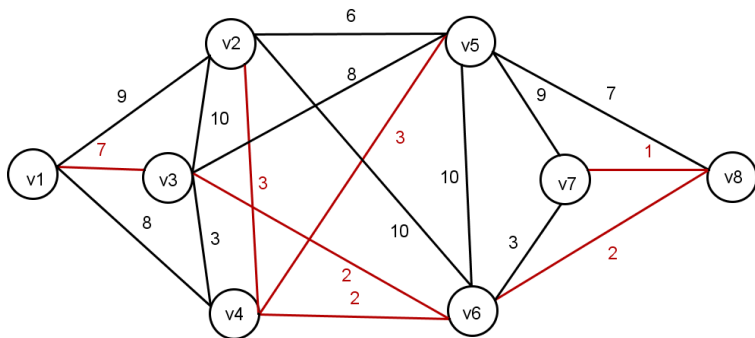
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezneme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Hrana s hodnotou 6 nemůže být do kostry zahrnuta, přidáme tedy hranu s hodnotou 7.

Optimalizace v grafech - minimální kostra

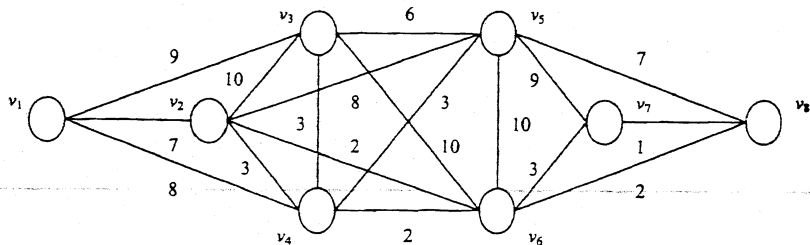
Pro nalezení kostry grafu s minimálním celkovým ohodnocením hran lze použít hladový (Kruskalův) algoritmus: do kostry postupně zařazujeme hrany s nejnižší hodnotou tak dlouho, dokud nejsou propojeny všechny uzly. Přitom nesmíme dopustit vznik cyklu: takové hrany, které by uzavřely okruh, do kostry nezahrneme. Nalezneme minimální kostru v grafu z knihy T. Šubrt: Ekonomicko-matematické metody:



Dostali jsem již souvislý podgraf, tedy kostru, celková hodnota jejích hran je 20.

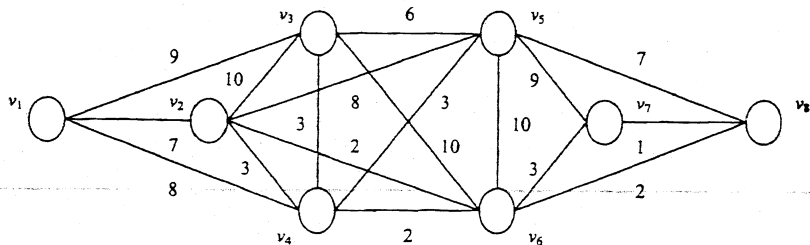
Optimalizace v grafech - nejkratší cesta

Pro hledání nejkratší cesty z uzlu v_1 do ostatních uzlů lze použít **Dijkstrův algoritmus**: předvedme na grafu z předchozího příkladu:



Optimalizace v grafech - nejkratší cesta

Pro hledání nejkratší cesty z uzlu v_1 do ostatních uzlů lze použít **Dijkstrův algoritmus**: předvedme na grafu z předchozího příkladu:



Algoritmus rozděljuje uzly podle toho, zda už do nich nejkratší cestu známe nebo ne. Začínáme s cestou délky 0 z v_1 do v_1 . Nejkratší cestu prodloužíme vždy o jednu hranu. Projdeme všechny uzly se známou délkou cesty z v_1 , sečteme tyto délky postupně s hodnotami hran z uzlů vycházejících a vybereme ze všech součtů nejmenší, tak získáme délku cesty do nějakého nového uzlu a postup dále opakujeme.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0							

V prvním kroku známe jen vzdálenost do v1, projdeme jeho sousedy a vybereme hranu s nejmenší hodnotou, což je v1-v2 s délkou 7. Vrchol v2 tedy zařadíme do skupiny se známou vzdáleností z v1.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7						

V druhém kroku procházíme kromě sousedů v1 také sousedy v2, délky cest přes v2 jsou 7+10, 7+3, 7+8, 7+2, takže nejkratší novou cestou bude v1-v4 délky 8. Přidáme v4 mezi uzly, do nichž nejkratší cestu známe.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7		8				

V dalším kroku procházíme dosud nezařazené sousedy uzlů v1, v2 a v4. Nejmenší celková délka cest přes tyto uzly do jejich sousedů je v1-v3 s hodnotou 9. To bude tedy vzdálenost do uzlu v3.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8				

Stejnou vzdálenost (také 9) má cesta přes v2 do v6.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8		9		

Z dosud nezařazených uzlů má nejmenší celkovou vzdálenost přes jednu hranu z již vytvořených cest uzel v5, který napojíme z v4, jeho vzdálenost z v1 je 11.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8	11	9		

Stejně vzdálenosti 11 dosáhneme napojením uzlu v8 přes v6.

Optimalizace v grafech - nejkratší cesta

Postup algoritmu se lépe sleduje v tabulce sousedů (v závorkách jsou uvedeny hodnoty hran):

sousedí	v1	v2	v3	v4	v5	v6	v7	v8
	v2 (7)	v1 (7)	v1 (9)	v1 (8)	v2 (8)	v2 (2)	v5 (10)	v5 (7)
	v3 (9)	v3 (10)	v2 (10)	v2 (3)	v3 (6)	v3 (10)	v6 (3)	v6 (2)
	v4 (8)	v4 (3)	v4 (3)	v3 (3)	v4 (3)	v4 (2)	v8 (1)	v7 (1)
		v5 (8)	v5 (6)	v5 (3)	v6 (10)	v5 (10)		
		v6 (2)	v6 (10)	v6 (2)	v7 (9)	v7 (3)		
					v8 (7)	v8 (2)		
vzd. z v1	0	7	9	8	11	9		11

Nakonec připojíme uzel v7, nejkratší možnost je přes v6, celková vzdálenost do něj je 12.

Optimalizace v grafech - medián grafu

Medián grafu minimalizuje součet vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění skladu pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	SUMA
Kralupy	0	4	16	8	18	25	17	279
Veltrusy	4	0	20	12	22	28	13	308
Slaný	16	20	0	12	7	14	17	184
Velvary	8	12	12	0	10	17	10	191
Zlonice	18	22	7	10	0	7	10	156
Vraný	25	28	14	17	7	0	15	231
Bříza	17	13	17	10	10	15	0	200

Optimalizace v grafech - medián grafu

Medián grafu minimalizuje součet vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění skladu pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	SUMA
Kralupy	0	4	16	8	18	25	17	279
Veltrusy	4	0	20	12	22	28	13	308
Slaný	16	20	0	12	7	14	17	184
Velvary	8	12	12	0	10	17	10	191
Zlonice	18	22	7	10	0	7	10	156
Vraný	25	28	14	17	7	0	15	231
Bříza	17	13	17	10	10	15	0	200

Aby se do skladu nejezdilo celkem co nejméně kilometrů, je nejlepší jej umístit ve Zlonici.

Pozn.: Při umístění dvou skladů hovoříme o dvoumediánu, atd.

Optimalizace v grafech - centr grafu

Centr grafu minimalizuje maximum vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění hasičské stanice pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	MAX
Kralupy	0	4	16	8	18	25	17	25
Veltrusy	4	0	20	12	22	28	13	28
Slaný	16	20	0	12	7	14	17	20
Velvary	8	12	12	0	10	17	10	17
Zlonice	18	22	7	10	0	7	10	22
Vraný	25	28	14	17	7	0	15	28
Bříza	17	13	17	10	10	15	0	17

Optimalizace v grafech - centr grafu

Centr grafu minimalizuje maximum vzdáleností od ostatních uzlů. Spočítejme ho pro úlohu o umístění hasičské stanice pro města uvažovaná ve výše řešené okružní úloze.

	Kralupy	Veltrusy	Slaný	Velvary	Zlonice	Vraný	Bříza	MAX
Kralupy	0	4	16	8	18	25	17	25
Veltrusy	4	0	20	12	22	28	13	28
Slaný	16	20	0	12	7	14	17	20
Velvary	8	12	12	0	10	17	10	17
Zlonice	18	22	7	10	0	7	10	22
Vraný	25	28	14	17	7	0	15	28
Bříza	17	13	17	10	10	15	0	17

Aby města byla pro hasiče co nejrychleji dostupná, nejlepší je umístit stanici ve Velvarech nebo Bříze.

Pozn.: Při umístění dvou stanic hovoříme o dvoucentru, atd.

Optimalizace v síťových grafech

Nejznámější úlohou je hledání maximálního toku v síti (např. vodovodní potrubí, silniční či datová síť,...). Pro řešení potřebujeme znát:

- popis potrubí (graf, zpravidla orientovaný)
- odkud voda vytéká (speciální uzel: zdroj)
- kam voda teče (speciální uzel: cíl, spotřebič)
- kolik vody může danou hranou protéct (kapacita = nazáporné ohodnocení hran)

Graf s vyjmenovanými vlastnostmi označujeme jako **síť**.

Optimalizace v síťových grafech

Nejznámější úlohou je hledání maximálního **toku v síti** (např. vodovodní potrubí, silniční či datová síť,...). Pro řešení potřebujeme znát:

- popis potrubí (graf, zpravidla orientovaný)
- odkud voda vytéká (speciální uzel: zdroj)
- kam voda teče (speciální uzel: cíl, spotřebič)
- kolik vody může danou hranou protéct (kapacita = nazáporné ohodnocení hran)

Graf s vyjmenovanými vlastnostmi označujeme jako **síť**.

Tokem nazveme funkci přiřazující každé hraně nezáporné číslo nepřesahující kapacitu splňující tzv. Kirchhoffovy zákony: **součet toků na hranách vstupujících je stejný jako na vystupujících**. Tyto podmínky musí platit ve všech uzlech kromě zdroje a cíle. **Velikost toku** je součet toků vycházejících ze zdroje. Pokud ze zdroje nic nevytéká, jde o nulový tok.

Optimalizace v síťových grafech

Pro hledání maximálního toku v síti se používají zlepšovací algoritmy: začne se zpravidla s nulovým tokem a hledá se cesta, na níž mají všechny hrany nenulovou rezervu (rozdíl kapacity a skutečného toku). Tok lze navýšit na dané cestě o minimální hodnotu rezervy. Postupujeme dále prohledáváním grafu a hledáním další zlepšující cesty. Prostou aplikací tohoto postupu bychom ale nemuseli nalézt nejlepší řešení, někdy je potřeba na některých hranách naopak tok zmenšit, aby se mohl vést lepší cestou a celková velikost toku se zvýšila.

Optimalizace v síťových grafech

Pro hledání maximálního toku v síti se používají zlepšovací algoritmy: začne se zpravidla s nulovým tokem a hledá se cesta, na níž mají všechny hrany nenulovou rezervu (rozdíl kapacity a skutečného toku). Tok lze navýšit na dané cestě o minimální hodnotu rezervy. Postupujeme dále prohledáváním grafu a hledáním další zlepšující cesty. Prostou aplikací tohoto postupu bychom ale nemuseli nalézt nejlepší řešení, někdy je potřeba na některých hranách naopak tok zmenšit, aby se mohl vést lepší cestou a celková velikost toku se zvýšila. Vylepšený algoritmus (**Ford-Fulkerson**) je následující:

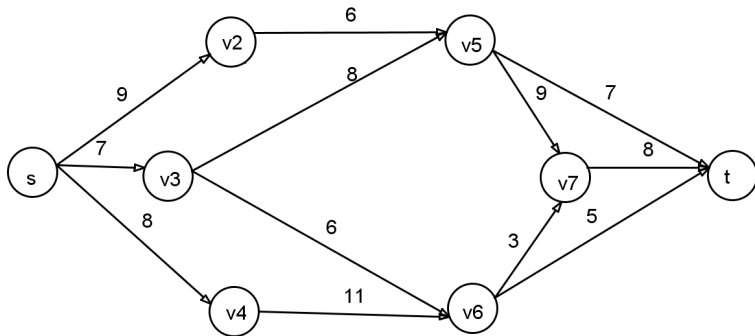
- U každé hrany evidujeme její rezervu i "rezervu v protisměru" (o kolik lze tok snížit). Na začátku u výchozího nulového toku jsou rezervy rovny kapacitám (rezervy v protisměru u orientovaných hran nulové)
- Vybereme zlepšující cestu a navýšíme tok o její rezervu. Současně o stejnou hodnotu snížíme rezervy všech hran na cestě a navýšíme jejich rezervy v protisměru.
- Postupujeme takto dál, dokud lze nalézt cestu s nenulovou rezervou.

Maximální tok v síti: Ford-Fulkerson

Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:

Maximální tok v síti: Ford-Fulkerson

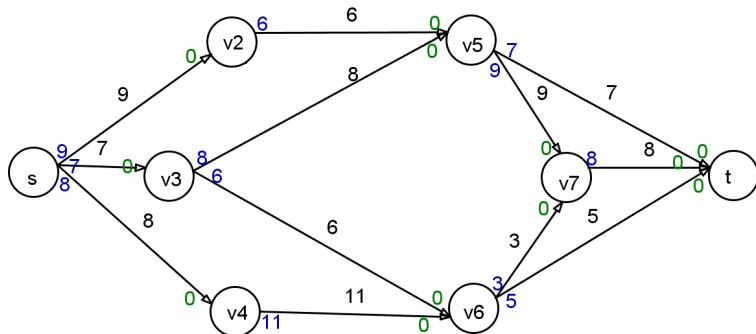
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Začneme s nulovým tokem, na hranách modře vyznačíme rezervy, zeleně "rezervy v protisměru".

Maximální tok v síti: Ford-Fulkerson

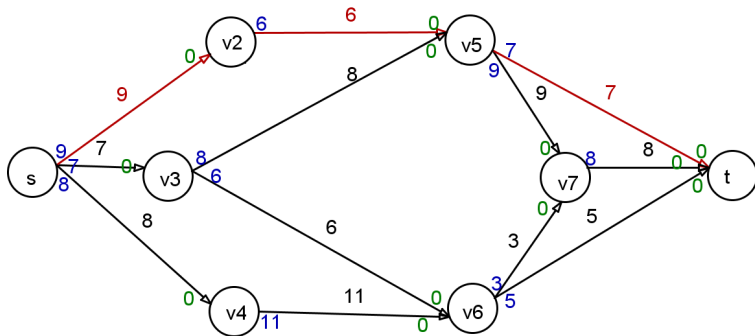
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Najdeme libovolnou zlepšující cestu.

Maximální tok v síti: Ford-Fulkerson

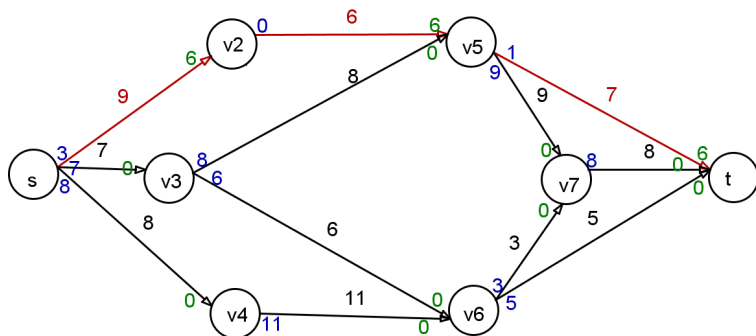
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Na zvolené cestě je nejmenší rezerva 6, o tuto hodnotu navýšíme tok na cestě. Rezervy na všech vybraných hranách tedy klesnou o šest a o stejnou hodnotu se zvýší protisměrné rezervy.

Maximální tok v síti: Ford-Fulkerson

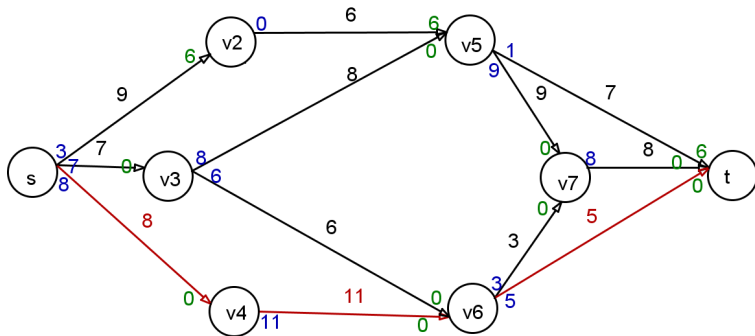
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Najdeme další zlepšující cestu.

Maximální tok v síti: Ford-Fulkerson

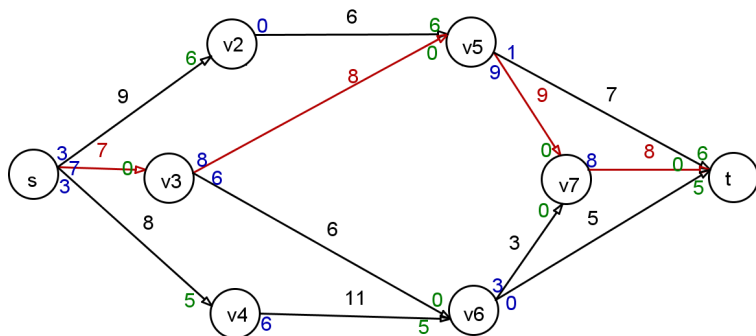
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Na této cestě je hrana s rezervou 5, nemůžeme tedy tok zvýšit o víc.

Maximální tok v síti: Ford-Fulkerson

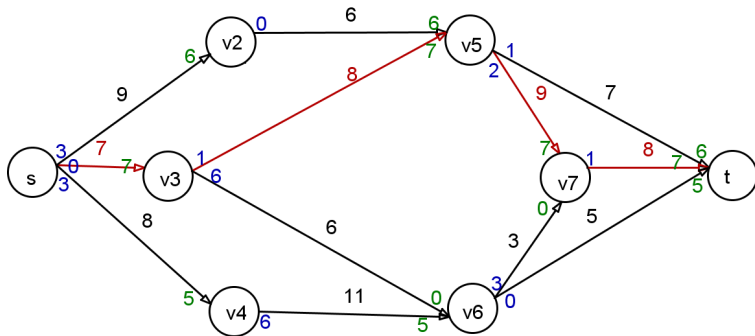
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Po navýšení najdeme další cestu s nenulovou rezervou.

Maximální tok v síti: Ford-Fulkerson

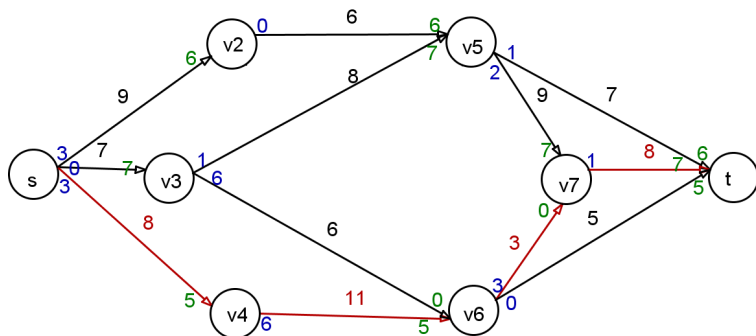
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Na této cestě jsme navýšili tok o 7.

Maximální tok v síti: Ford-Fulkerson

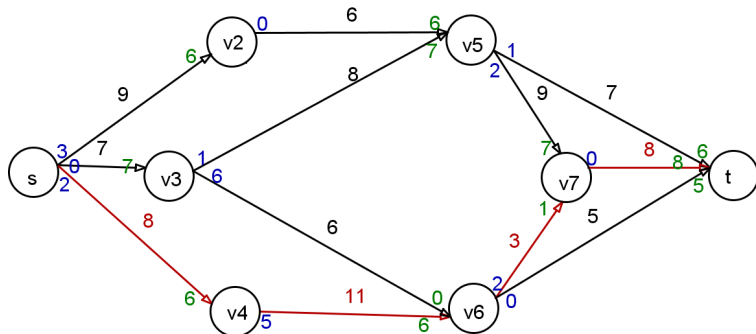
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Existuje ještě zlepšující cesta s rezervou 1.

Maximální tok v síti: Ford-Fulkerson

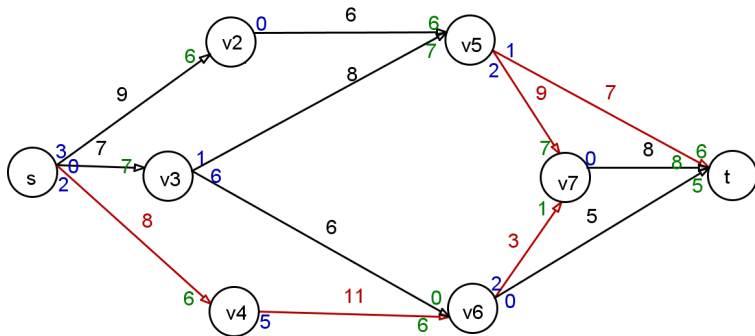
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Provedli jsme navýšení.

Maximální tok v síti: Ford-Fulkerson

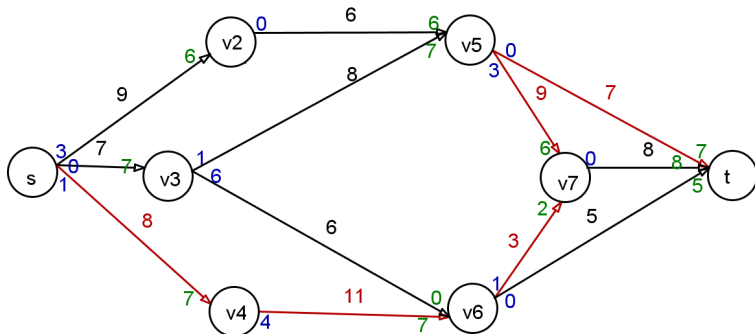
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Můžeme tok ještě zlepšit po cestě, která mezi v_5 a v_7 vede "v protisměru".

Maximální tok v síti: Ford-Fulkerson

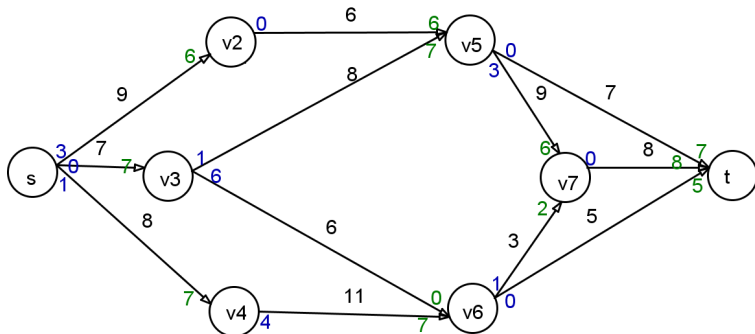
Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Provedeme navýšení o nejmenší rezervu (1, na poslední hraně). Mezi v_5 a v_7 se tok fakticky snížil o 1.

Maximální tok v síti: Ford-Fulkerson

Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:



Nalezený tok má velikost $7+8+5=20$ (viz zelené hodnoty u cíle). Evidentně je to optimální řešení, do cíle už není volná další kapacita.

Maximální tok v síti: Ford-Fulkerson

Aplikujme Ford-Fulkersonův algoritmus na nalezení maximálního toku v zadané síti:

Řízení projektů

Řízení projektů je jednou z typických aplikací teorie grafů. Projekt můžeme obecně chápat jako soubor činností. Tyto činnosti lze charakterizovat předpokládanou dobou trvání, náklady na realizaci, požadavky na zajištění, výčtem činností, které realizaci musí předcházet, atd. Nejčastěji hledáme odpověď na tyto otázky:

- Jaká je nejkratší možná doba realizace projektu?
- Které činnosti jsou z hlediska dodržení termínu klíčové, tzv. **kritické činnosti**?
- Jaké jsou rezervy u nekritických činností
- Jaký je časový rozvrh pro realizaci jednotlivých činností?

Kromě **časové analýzy** projektů nás zajímají též náklady na projekt v závislosti na čase, hovoříme pak o **nákladové analýze**. Dále můžeme sledovat úroveň a rozložení zdrojů potřebných pro jednotlivé činnosti, tedy provádět **zdrojovou analýzu** projektu.

Řízení projektů - konstrukce síťového grafu

Projekt znázorňujeme pomocí ohodnoceného síťového grafu, kde hrany reprezentují činnosti, ohodnocení většinou dobu jejich trvání a uzly představují momenty zahájení či ukončení jednotlivých činností. Při analýze nejprve musíme vymezit jednotlivé činnosti, odhadnout délky jejich realizace a definovat **návaznosti** pomocí výčtu bezprostředně předcházejících činností.

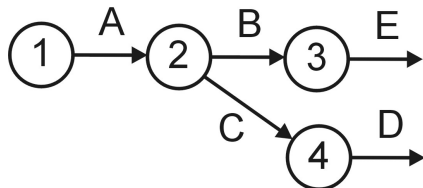
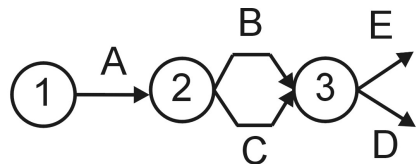
Řízení projektů - konstrukce síťového grafu

Projekt znázorňujeme pomocí ohodnoceného síťového grafu, kde hrany reprezentují činnosti, ohodnocení většinou dobu jejich trvání a uzly představují momenty zahájení či ukončení jednotlivých činností. Při analýze nejprve musíme vymezit jednotlivé činnosti, odhadnout délky jejich realizace a definovat **návaznosti** pomocí výčtu bezprostředně předcházejících činností. Ukažme si síťový graf pro projekt vytvoření nového obchodního střediska firmy Q-mark, a.s. (J.Jablonský: Operační výzkum). Před vlastním sestavením grafu definujeme elementární činnosti a jejich vlastnosti, viz. tabulka:

činnost	popis činnosti	trvání [týdny]	předchází
A	výběr a nákup projektu	6	-
B	zpracování projektu	4	A
C	obsazení pozice manažera	3	A
D	výběr personálu	3	B,C
E	rekonstrukce a vybavení objektu	8	B
F	školení personálu	2	D
G	výběr sortimentu zboží	2	B,C
H	uzavření smluv s dodavateli	5	G
I	nákup zboží	3	E,F,H
J	reklama	2	H

Řízení projektů - konstrukce síťového grafu

Při sestavování grafu můžeme narazit na problém u definice uzlů. Například činnosti D musí předcházet B,C, ale činnosti E předchází pouze B. Jak tedy správně znázornit návaznost? Příklady **nesprávného** znázornění, viz obr.:



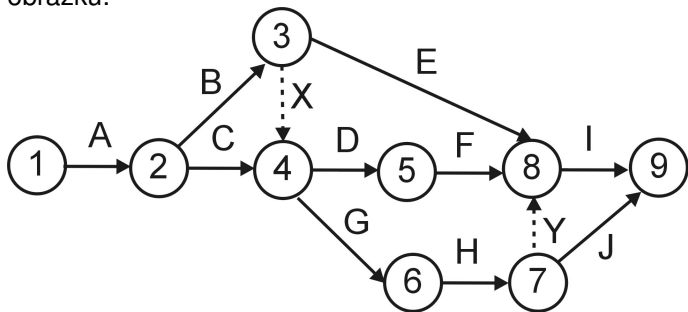
Podobný problém je u činností I,J, které mají společného předchůdce H, ale činnosti I navíc předchází i E,F.

Řízení projektů - konstrukce síťového grafu

Problém s návazností lze řešit zavedením **fiktivních činností**, jimž odpovídající fiktivní hrany doplní chybějící spoje. Znázorňují se přerušovanou čarou. Doba fiktivních činností je vždy nulová. Do našeho příkladu tedy doplníme dvě fiktivní činnosti: X zprostředkující návaznost mezi B a D a činnost Y pro návaznost mezi H a I.

Řízení projektů - konstrukce síťového grafu

Problém s návazností lze řešit zavedením **fiktivních činností**, jimž odpovídající fiktivní hrany doplní chybějící spoje. Znárodnují se přerušovanou čarou. Doba fiktivních činností je vždy nulová. Do našeho příkladu tedy doplníme dvě fiktivní činnosti: X zprostředkující návaznost mezi B a D a činnost Y pro návaznost mezi H a I. Jedno z možných znázornění sítě projektu je na obrázku:



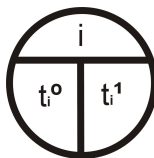
Při sestavování grafu je dobré držet se pravidla, aby index počátečního uzlu každé hrany byl nižší než index jejího koncového uzlu. Potom graf nebude obsahovat žádné orientované cykly.

Critical Path Method (CPM)

Metoda se používá od 50. let minulého století. Jejím principem je určit pro každou činnost tyto 4 charakteristiky:

- Nejdříve možný začátek provádění činnosti začínající v uzlu u_i , značíme jej: t_i^0
- Nejdříve možný konec provádění činnosti reprezentované hranou h_{ij} získáme přičtením doby trvání činnosti: $t_i^0 + y_{ij}$
- Nejpozději přípustný konec provádění činnosti končící v uzlu u_j , značíme jej: t_j^1
- Nejpozději přípustný začátek provádění činnosti reprezentované hranou h_{ij} získáme odečtením doby trvání činnosti: $t_j^1 - y_{ij}$

V síťovém grafu vyznačíme do jednotlivých uzlů i nejdříve možné začátky a nejpozději přípustné konce činností, které v něm začínají, resp. končí:



Critical Path Method (CPM)

Vlastní algoritmus metody probíhá ve čtyřech fázích:

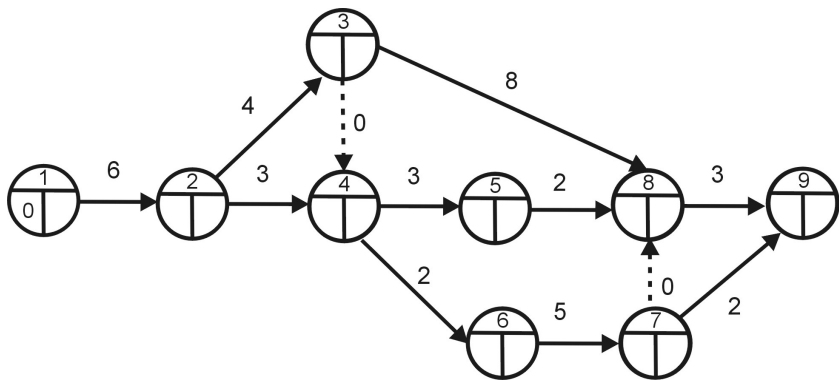
- 1 Výpočet nejprve možných začátků: Pro činnosti začínající v uzlu u_j se spočte jako maximum nejdříve možných konců činností, které do něj vstupují. $t_j^0 = \max_i(t_i^0 + y_{ij})$ Pro výstupní uzel tak dostaneme nejkratší možnou dobu realizace projektu.
- 2 Výpočet nejpozději přípustných konců: Pro činnosti končící v uzlu u_i se spočte jako minimum z nejpozději přípustných začátků činností z uzlu vystupujících. $t_i^1 = \min_j(t_j^1 - y_{ij})$
- 3 Výpočet celkových časových rezerv: Činnost reprezentovaná hranou h_{ij} má stanoveno, kdy může nejdříve začít (t_i^0) a kdy musí nejpozději skončit (t_j^1). Doba, během které se musí realizovat, je tedy $t_j^1 - t_i^0$ a protože její realizace trvá y_{ij} , dostaneme její pro časovou rezervu vztah:
 $R_{ij} = t_j^1 - t_i^0 - y_{ij}$
- 4 Sestavení harmonogramu činností

Critical Path Method (CPM) - 1. fáze

V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.

Critical Path Method (CPM) - 1. fáze

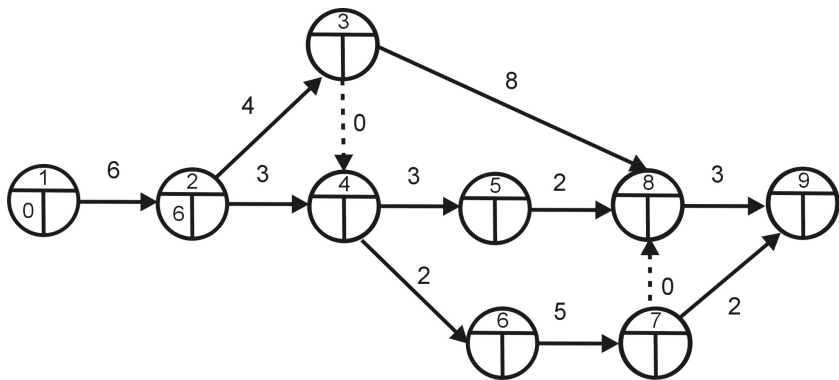
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejprve si zobrazíme graf projektu s dobami trvání činností.

Critical Path Method (CPM) - 1. fáze

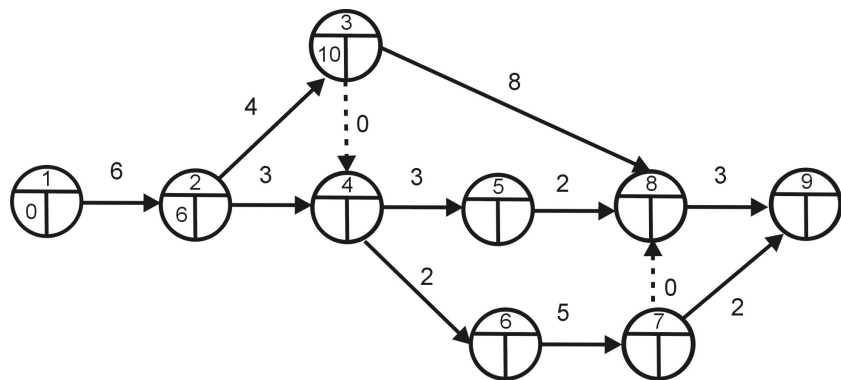
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činností B,C je $t_2^0 = 0 + 6 = 6$.

Critical Path Method (CPM) - 1. fáze

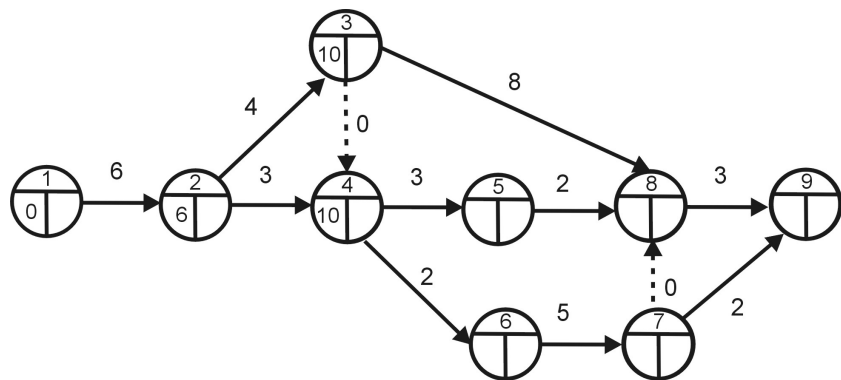
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činnosti E je $t_3^0 = 6 + 4 = 10$.

Critical Path Method (CPM) - 1. fáze

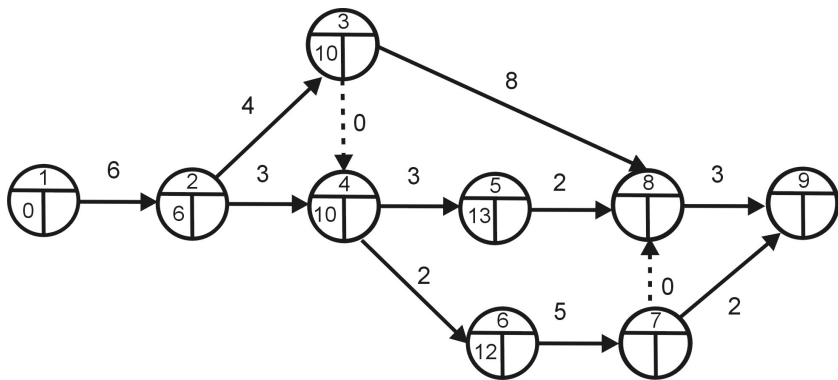
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činností D,G je $t_4^0 = \max(10 + 0, 6 + 3) = 10$.

Critical Path Method (CPM) - 1. fáze

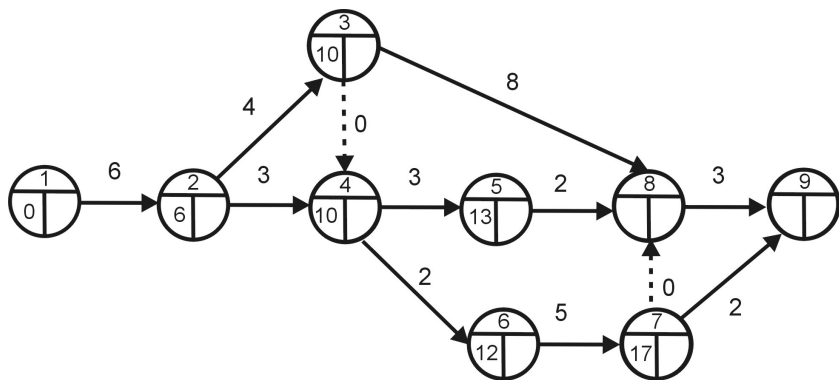
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činností F, resp. H je $t_5^0 = 10 + 3 = 13$, resp. $t_6^0 = 10 + 2 = 12$.

Critical Path Method (CPM) - 1. fáze

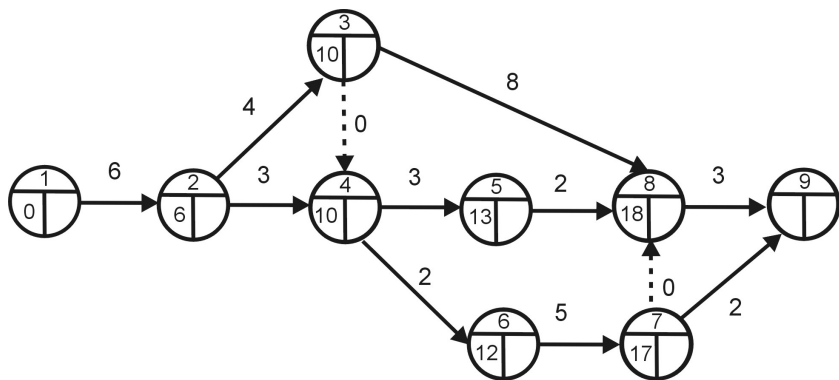
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činnosti J je $t_7^0 = 12 + 5 = 17$.

Critical Path Method (CPM) - 1. fáze

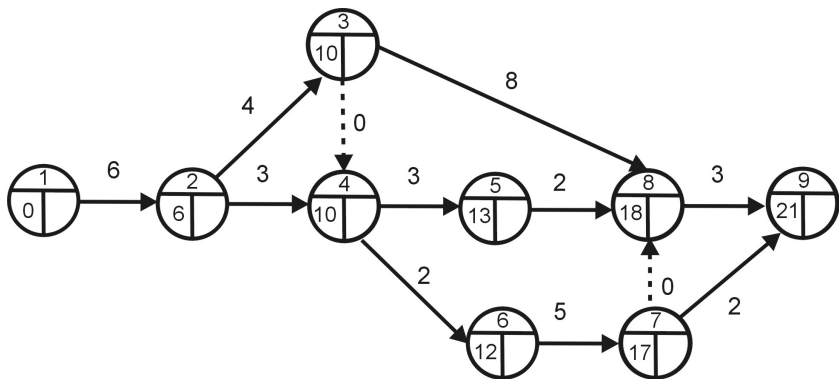
V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



Nejdříve možný začátek činnosti I je $t_8^0 = \max(10 + 8, 13 + 2, 17 + 2) = 18$.

Critical Path Method (CPM) - 1. fáze

V první fázi metody CPM postupně zleva doprava počítáme nejdříve možné začátky činností.



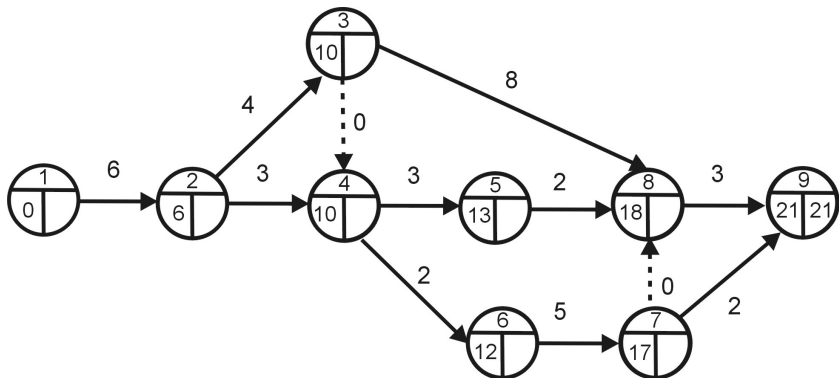
Nejdříve možná doba ukončení projektu je v čase
 $t_9^0 = \max(18 + 3, 17 + 2) = 21$ týdnů.

Critical Path Method (CPM) - 2. fáze

Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.

Critical Path Method (CPM) - 2. fáze

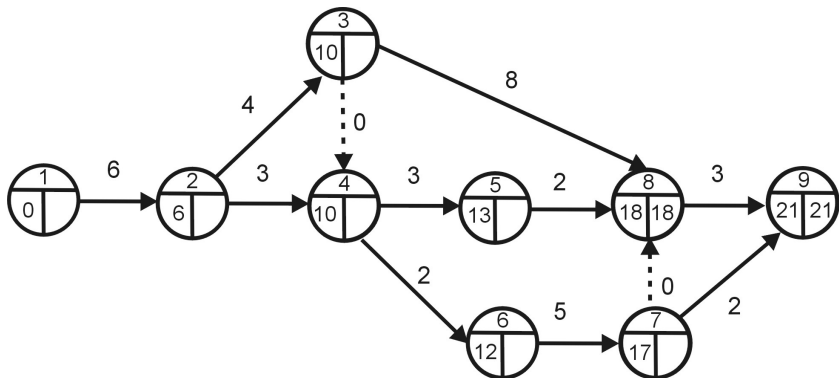
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Stanovili jsme minimální dobu trvání projektu na 21 týdnů. To bude i nejpozději přípustný konec činností I,J: $t_9^1 = 21$.

Critical Path Method (CPM) - 2. fáze

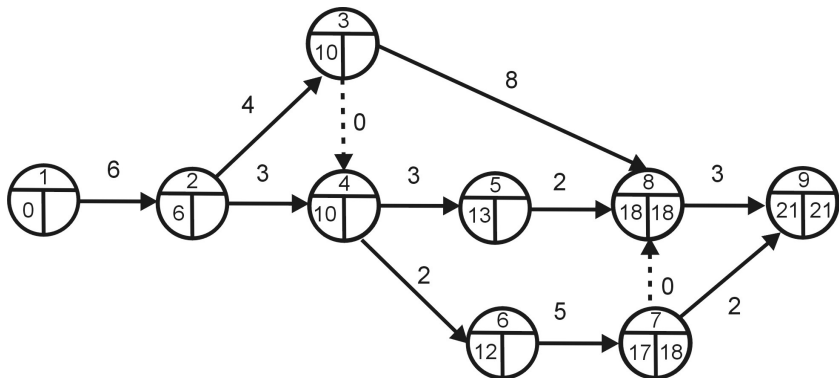
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činností F,E je $t_8^1 = 21 - 3 = 18$.

Critical Path Method (CPM) - 2. fáze

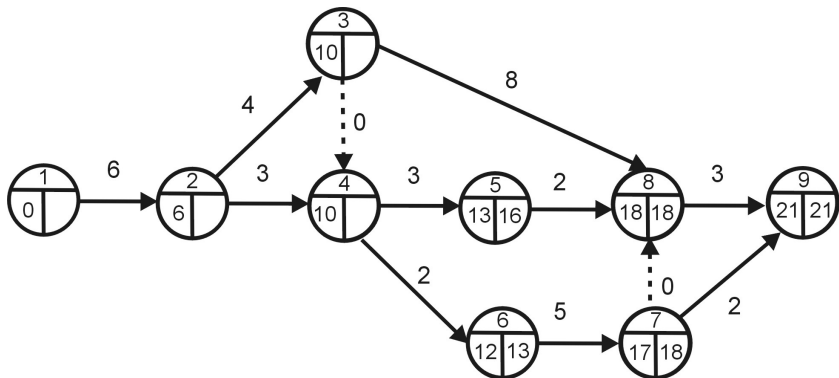
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti H je $t_7^1 = \min(21 - 2, 18 - 0) = 18$.

Critical Path Method (CPM) - 2. fáze

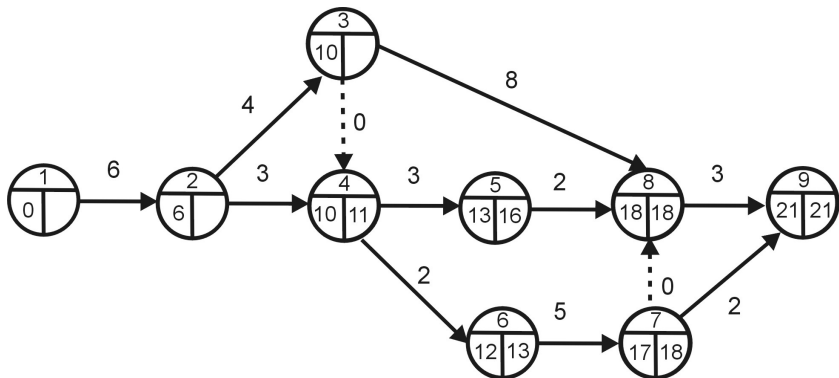
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činností D, resp. G je $t_5^1 = 18 - 2 = 16$, resp. $t_6^1 = 18 - 5 = 13$.

Critical Path Method (CPM) - 2. fáze

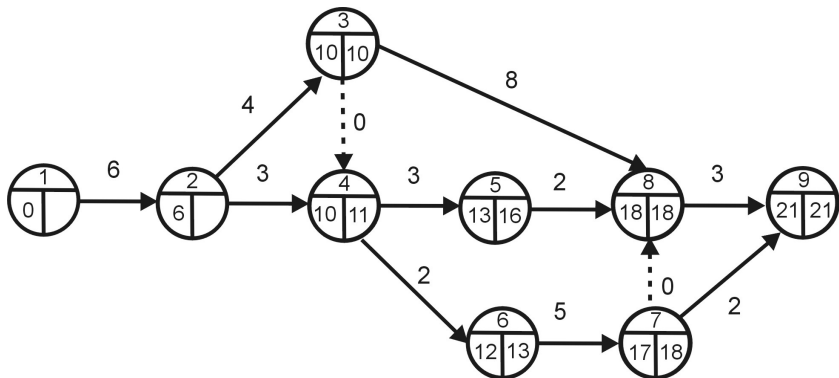
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti C je $t_4^1 = \min(13 - 2, 16 - 3) = 11$.

Critical Path Method (CPM) - 2. fáze

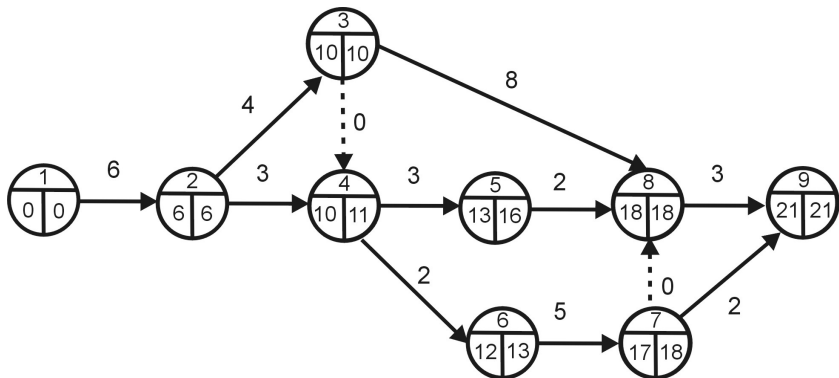
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti B je $t_3^1 = \min(18 - 8, 11 - 0) = 10$.

Critical Path Method (CPM) - 2. fáze

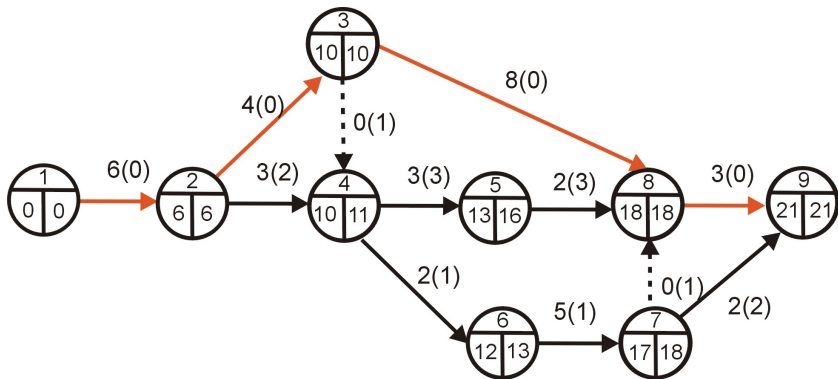
Ve druhé fázi postupujeme zprava doleva a doplňujeme nejpozději přípustné konce. Předpokládejme, že chceme stihnout projekt v nejkratším možném čase.



Nejpozději přípustný konec činnosti A je $t_2^1 = \min(10 - 4, 11 - 3) = 6$. Protože jsme vycházeli z nejrychlejší možné realizace projektu, samozřejmě vyšlo $t_1^1 = 6 - 6 = 0$.

Critical Path Method (CPM) - 3. fáze

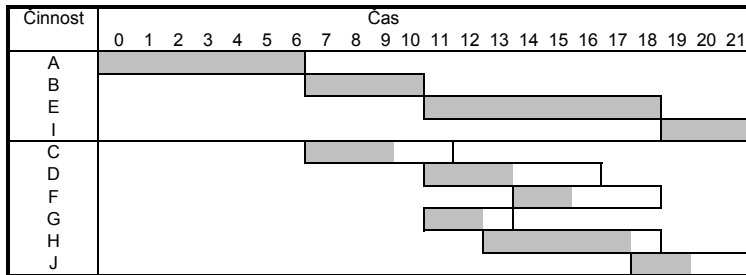
Dopočítáme rezervy jednotlivých činností podle vztahu $R_{ij} = t_j^1 - t_i^0 - y_{ij}$.
Například činnost J musí proběhnout mezi 17. a 21. týdnem a trvá 2 týdny, její rezerva je tedy $R_{79} = 21 - 17 - 2 = 2$ týdny, apod.



Rezervy jednotlivých činností jsou uvedeny v závorkách. Červeně je znázorněna kritická cesta sestávající z činností A,B,E,I, které nemají žádnou časovou rezervu.

Critical Path Method (CPM) - 4. fáze

Poslední, ale velmi důležitou fází je rozvržení realizace činností v čase. Je nutné určit, které činnosti mohou probíhat paralelně a které na sebe musí navazovat. To nám umožní dále rozvrhovat zdroje potřebné pro jednotlivé činnosti. Ukažme si rozvrh činností v diagramu, kde rámečky tvoří nejdříve možný začátek a nejpozději přípustný konec činností, stínování naznačuje dobu jejich trvání. V horní části tabulky jsou kritické činnosti s nulovými rezervami - tyto musí na sebe bezprostředně navazovat, aby nedošlo ke zpoždění projektu.



Metoda PERT

Metoda **PERT** (Program Evaluation and Review Technique) je pravděpodobnostním rozšířením metody CPM. V praxi často není reálné stanovit realizační doby činností, proto jsou hodnoty y_{ij} nahrazeny náhodnými veličinami, které se realizují na nějakém intervalu $\langle a_{ij}, b_{ij} \rangle$. Kromě tohoto mezního **optimistického**, resp. **pesimistického** odhadu se pracuje s nejpravděpodobnější dobou trvání činnosti m_{ij} , ta se označuje jako **modální** odhad.

Metoda PERT

Metoda **PERT** (Program Evaluation and Review Technique) je pravděpodobnostním rozšířením metody CPM. V praxi často není reálné stanovit realizační doby činností, proto jsou hodnoty y_{ij} nahrazeny náhodnými veličinami, které se realizují na nějakém intervalu $\langle a_{ij}, b_{ij} \rangle$. Kromě tohoto mezního **optimistického**, resp. **pesimistického** odhadu se pracuje s nejpravděpodobnější dobou trvání činnosti m_{ij} , ta se označuje jako **modální** odhad. Skutečné rozložení pravděpodobnosti náhodné veličiny není obecně známo, ale často se aproximuje β -rozdělením. Lze ukázat, že pro její střední hodnotu a rozptyl platí:

$$\mu_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}$$

$$\sigma_{ij} = \frac{b_{ij} - a_{ij}}{6}$$

Metoda PERT

Metoda **PERT** (Program Evaluation and Review Technique) je pravděpodobnostním rozšířením metody CPM. V praxi často není reálné stanovit realizační doby činností, proto jsou hodnoty y_{ij} nahrazeny náhodnými veličinami, které se realizují na nějakém intervalu $\langle a_{ij}, b_{ij} \rangle$. Kromě tohoto mezního **optimistického**, resp. **pesimistického** odhadu se pracuje s nejpravděpodobnější dobou trvání činnosti m_{ij} , ta se označuje jako **modální odhad**. Skutečné rozložení pravděpodobnosti náhodné veličiny není obecně známo, ale často se aproximuje β -rozdělením. Lze ukázat, že pro její střední hodnotu a rozptyl platí:

$$\mu_{ij} = \frac{a_{ij} + 4m_{ij} + b_{ij}}{6}$$

$$\sigma_{ij} = \frac{b_{ij} - a_{ij}}{6}$$

Vlastní výpočet metodou PERT se neliší od metody CPM, jen se místo hodnot y_{ij} pracuje se středními hodnotami μ_{ij} . Za určitých předpokladů se dá dle centrální limitní věty aproximovat rozložení celkové délky projektu normální náhodnou veličinou a získat tak odpověď na otázky : Jaká je pravděpodobnost, že projekt bude ukončen v čase T , resp. v jakém čase bude projekt ukončen se stanovenou pravděpodobností p , apod.

Vícekriteriální rozhodování

V reálných rozhodovacích situacích je často důležité vzít do úvahy více optimalizačních kritérií. Tato však zřídka bývají ve vzájemném souladu, takže není možné najít řešení, které bude nejlepší podle všech kritérií. Úlohy vícekriteriálního rozhodování se dělí podle způsobu stanovení rozhodovacích variant.

- Jsou-li stanoveny výčtem, mluvíme o **vícekriteriálním hodnocení variant (VHV)**.
- Jiný přístup je vymezení variant soustavou omezujících podmínek, mluvíme pak o úlohách **vícekriteriálního programování**.

Vícekritériální hodnocení variant

V úlohách VHV je dána množina variant $X = \{X_1, \dots, X_n\}$, které jsou hodnoceny podle kritérií Y_1, \dots, Y_k . Každá varianta je pak popsána vektorem kritériálních hodnot, tyto lze pak jako řádky pro jednotlivé varianty shrnout do **kritériální matice**.

	Y_1	Y_2	\dots	Y_k
X_1	y_{11}	y_{12}	\dots	y_{1k}
X_2	y_{21}	y_{22}	\dots	y_{2k}
\vdots	\vdots	\vdots	\vdots	\vdots
X_n	y_{n1}	y_{n2}	\dots	y_{nk}

Součástí modelu úlohy musí být u kvantitativních kritérií i určení jejich typu (maximalizační či minimalizační). Protože některé metody vyžadují, aby všechna kritéria byla stejného typu, někdy je nutné provést transformaci (např. u hodnocení hospodářské vyspělosti je HDP na hlavu kritériem maximalizačním a míra nezaměstnanosti minimalizačním). Další oblasti aplikace VHV jsou velmi široké: různá výběrová řízení, hodnocení podniků, výrobků či služeb, výběr lokality pro investiční akci, atd.

Vícekriteriální hodnocení variant - příklad

Dále uvedené postupy budeme ilustrovat na příkladu výběru tabletu (z knihy Tomáš Šubrt a kol.: Ekonomicko - matematické metody): Uživatel definoval pět relevantních hledisek, podle kterých bude jednotlivé nabídky hodnotit: cena [Kč], velikost operační paměti RAM [MB], výdrž baterie [hod.], hmotnost [g] a souhrnně kombinace OS, procesoru a velikosti displeje. V úvahu připadá pět konkrétních tabletů T1 - T5, uveďme přehled jejich charakteristik:

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display
T1	12000	1000	9,5	680	Android 3.0, Tegra 1 GHz, 10,1"
T2	12000	1000	10	600	Apple 1054, 1 GHz Dual, 9,7"
T3	5000	512	7	380	Android 2.2, 1 GHz, 7"
T4	20000	4000	3	1160	Windows 7, iCore5 1 GHz, 12,1"
T5	5000	256	4	400	Android 2.1, 800 MHz, 7"

První čtyři kritéria jsou kvantitativní (cena a hmotnost minimalizační, ostatní maximalizační), u pátého kritéria máme alespoň ordinální informaci v podobě pořadí tabletů dle odborníka: 1,3,4,2,5.

Vícekritériální hodnocení variant

Mezi základní cíle při analýze VHV patří:

- výběr jedné, tzv. **kompromisní** varianty (např. při výběrovém řízení)
- uspořádání variant (např. spotřebitelské žebříčky)
- klasifikace variant (např. při přijímačkách: přijatí/nepřijatí nebo hodnocení bonity klientů bankou, atd.)

Vzájemný vztah mezi variantami může při VHV být následující:

- Varianta X_i **dominuje** variantu X_j , jestliže $(y_{i1}, \dots, y_{ik}) \geq (y_{j1}, \dots, y_{jk})$, ale vektory se nerovnají.
- Varianta X_j dominuje variantu X_i
- Varianty X_i, X_j jsou vzájemně **nedominované**.

Řekneme, že varianta X_i je nedominovaná, jestliže neexistuje žádná jiná varianta, která ji dominuje. Zřejmě kompromisní varianta musí být vždy nedominovaná. Dále definujeme pojmy **bazální** a **ideální** varianta, což je označení pro zpravidla reálně neexistující variantu nabývající nejhorších (resp. nejlepších) hodnot podle všech kritérií.

VHV - vztahy mezi variantami, příklad

V úloze o výběru tabletu stanovte nedominované varianty a vyberte bazální a ideální variantu.

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display
Tablet 1	12000	1000	9,5	680	1
Tablet 2	12000	1000	10	600	3
Tablet 3	5000	512	7	380	4
Tablet 4	20000	4000	3	1160	2
Tablet 5	5000	256	4	400	5

V tabulce jsou vyznačeny nejlepší hodnoty dosažené u jednotlivých kritérií.

VHV - vztahy mezi variantami, příklad

V úloze o výběru tabletu stanovte nedominované varianty a vyberte bazální a ideální variantu.

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display
Tablet 1	12000	1000	9,5	680	1
Tablet 2	12000	1000	10	600	3
Tablet 3	5000	512	7	380	4
Tablet 4	20000	4000	3	1160	2
Tablet 5	5000	256	4	400	5

V tabulce jsou vyznačeny nejlepší hodnoty dosažené u jednotlivých kritérií. Ideální varianta by tedy hypoteticky byla (5000 Kč, 4000 MB, 10 hod., 380g a 1. pořadí dle experta). Obdobně bazální varianta (20000 Kč, 256 MB, 3 hod., 1160g a 5. pořadí dle experta). Nedominované jsou téměř všechny varianty, dominovaný je pouze Tablet 5 (ve všem horší než Tablet 3)

VHV - vyjádření preference kritérií

U většiny metod VHV je nutné, aby rozhodovatel vyjádřil své preference ve vztahu k jednotlivým kritériím. Tyto preference mohou být stanoveny pomocí

- **aspiračních úrovní kritérií**, tedy stanovením minimálních hodnot, kterých má být dosaženo u jednotlivých maximalizačních kritérií (resp. maximálních hodnot pro minimalizační kritéria). Preference kritérií je tak vyjádřena nepřímo, důležitějším kritériím nastavíme přísnější limity.
- **pořadí kritérií** (ordinální informace o kritériích)
- **vah kritérií**: $\mathbf{v} = (v_1, \dots, v_k)$, $\sum v_i = 1$, $v_i > 0$, $i = 1, \dots, k$. (kardinální informace o kritériích)
- **míry substituce** mezi kriteriálními hodnotami, na níž jsou založeny kompenzační metody VHV

VHV - metody odhadu vah kritérií

Získání vah od rozhodovatele přímo v numerické podobě bývá problematické, proto je vhodné usnadnit mu situaci pomocí nějakého jednoduchého nástroje.

- **Metoda pořadí** vyžaduje pouze seřazení kritérií od nejméně důležitého po nejdůležitější. Přiřazené pořadí p_i tedy bude nabývat hodnot $1, \dots, k$ a odhad vah lze získat jejich normalizací:
$$v_i = \frac{p_i}{\sum_{i=1}^k p_i}.$$
- **Bodovací metoda** spočívá v tom, že rozhodovatel přiřadí každému kritériu body p_i z nějaké předem zvolené škály. Přepočtení bodů na váhy je stejné jako výše.
- **Fullerův trojúhelník** je založen na párovém porovnávání kritérií. Jednotlivým kritériím se přiřadí tolik bodů p_i , kolikrát je zvolen jako důležitější nebo stejně důležitý mezi všemi dvojicemi různých kritérií. (takových dvojic je $k(k-1)/2$ a lze je uspořádat do trojúhelníkového schématu, odtud název metody)
- Poněkud propracovanější přístup představuje **Saatyho metoda**, při níž se projdou všechny dvojice kritérií a každé se přiřadí číslo $s_{ij} \approx \frac{v_i}{v_j}$, které odhaduje poměr mezi důležitostmi jednotlivých kritérií. Matice $S = (s_{ij})_{i,j=1,\dots,k}$ se nazývá Saatyho matice.

VHV - Saatyho metoda odhadu vah kritérií

Saatyho metoda umožňuje formulovat preference verbálně a pak vyjádřit numericky pomocí stupnice:

- kritéria Y_i a Y_j jsou stejně důležitá, pak $s_{ij} = s_{ji} = 1$,
- kritérium Y_i je slabě důležitější než Y_j , pak $s_{ij} = 3$, $s_{ji} = 1/3$,
- kritérium Y_i je silně důležitější než Y_j , pak $s_{ij} = 5$, $s_{ji} = 1/5$,
- kritérium Y_i je velmi silně důležitější než Y_j , pak $s_{ij} = 7$, $s_{ji} = 1/7$,
- kritérium Y_i je absolutně důležitější než Y_j , pak $s_{ij} = 9$, $s_{ji} = 1/9$.

Jestliže uvedená stupnice nepostačuje, lze použít i mezistupně 2,4,6,8.

Pokud je Saatyho matice tzv. **konzistentní**, stačí váhy spočítat jako řešení soustavy rovnic $\frac{v_i}{v_j} = s_{ij}$, $i, j = 1, \dots, k$, $\sum v_i = 1$. Pro nekonzistentní matici soustava nemá řešení a váhy se pak odhadují například normalizací

geometrických průměrů řádků matice S: $p_i = \sqrt[k]{\prod_{j=1}^k s_{ij}}$.

VHV - metody odhadu vah kritérií, příklad

Pro úlohu o tabletu ukažme různé způsoby stanovení vah kritérií.

Nejjednodušší metodou **pořadí** by uživatel při preferenci „1.cena, 2.RAM, 3.názor experta, 4.výdrž baterie, 5.hmotnost“ dospěl k vahám

$$\mathbf{v} = \left(\frac{5}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15}, \frac{3}{15} \right).$$

VHV - metody odhadu vah kritérií, příklad

Pro úlohu o tabletu ukažme různé způsoby stanovení vah kritérií.

Nejjednodušší metodou **pořadí** by uživatel při preferenci „1.cena, 2.RAM, 3.názor experta, 4.výdrž baterie, 5.hmotnost“ dospěl k vahám

$$\mathbf{v} = \left(\frac{5}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15}, \frac{3}{15} \right).$$

Jiným způsobem je označení preference kritéria z daného řádku oproti kritériím v jednotlivých sloupcích vyznačením hodnoty 1 ve **Fullerově trojúhelníku**:

	cena	RAM	baterie	hmotnost	expert	skóre	váhy
cena		1	1	1	1		
RAM			1	1	1		
baterie				1	0		
hmotnost					0		
expert							

Pro výpočet skóre doplníme spodek tabulky symetricky opačnými hodnotami

VHV - metody odhadu vah kritérií, příklad

Pro úlohu o tabletu ukažme různé způsoby stanovení vah kritérií.

Nejjednodušší metodou **pořadí** by uživatel při preferenci „1.cena, 2.RAM, 3.názor experta, 4.výdrž baterie, 5.hmotnost“ dospěl k vahám

$$\mathbf{v} = \left(\frac{5}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15}, \frac{3}{15} \right).$$

Jiným způsobem je označení preference kritéria z daného řádku oproti kritériím v jednotlivých sloupcích vyznačením hodnoty 1 ve **Fullerově trojúhelníku**:

	cena	RAM	baterie	hmotnost	expert	skóre	váhy
cena		1	1	1	1		
RAM	0		1	1	1		
baterie	0	0		1	0		
hmotnost	0	0	0		0		
expert	0	0	1	1			

Nyní vypočteme skóre jako součet počtu preferencí v daných řádcích.

VHV - metody odhadu vah kritérií, příklad

Pro úlohu o tabletu ukažme různé způsoby stanovení vah kritérií.

Nejjednodušší metodou **pořadí** by uživatel při preferenci „1.cena, 2.RAM, 3.názor experta, 4.výdrž baterie, 5.hmotnost“ dospěl k vahám

$$\mathbf{v} = \left(\frac{5}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15}, \frac{3}{15} \right).$$

Jiným způsobem je označení preference kritéria z daného řádku oproti kritériím v jednotlivých sloupcích vyznačením hodnoty 1 ve **Fullerově trojúhelníku**:

	cena	RAM	baterie	hmotnost	expert	skóre	váhy
cena		1	1	1	1	4	
RAM	0		1	1	1	3	
baterie	0	0		1	0	1	
hmotnost	0	0	0		0	0	
expert	0	0	1	1		2	

Nyní vypočteme váhy vydělením bodových zisků jejich celkovým součtem, tj. 10.

VHV - metody odhadu vah kritérií, příklad

Pro úlohu o tabletu ukažme různé způsoby stanovení vah kritérií.

Nejjednodušší metodou **pořadí** by uživatel při preferenci „1.cena, 2.RAM, 3.názor experta, 4.výdrž baterie, 5.hmotnost“ dospěl k vahám

$$\mathbf{v} = \left(\frac{5}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15}, \frac{3}{15} \right).$$

Jiným způsobem je označení preference kritéria z daného řádku oproti kritériím v jednotlivých sloupcích vyznačením hodnoty 1 ve **Fullerově trojúhelníku**:

	cena	RAM	baterie	hmotnost	expert	skóre	váhy
cena		1	1	1	1	4	0,4
RAM	0		1	1	1	3	0,3
baterie	0	0		1	0	1	0,1
hmotnost	0	0	0		0	0	0
expert	0	0	1	1		2	0,2

Bohužel váha nejméně důležitého kritéria vyjde při konzistenci v preferencích nulová. Postup je možné modifikovat přidáním jedniček na diagonále (jako by každé kritérium bylo ve srovnání se sebou samým důležitější).

VHV - metody odhadu vah kritérií, příklad

Pro úlohu o tabletu ukažme různé způsoby stanovení vah kritérií.

Nejjednodušší metodou **pořadí** by uživatel při preferenci „1.cena, 2.RAM, 3.názor experta, 4.výdrž baterie, 5.hmotnost“ dospěl k vahám

$$\mathbf{v} = \left(\frac{5}{15}, \frac{4}{15}, \frac{2}{15}, \frac{1}{15}, \frac{3}{15} \right).$$

Jiným způsobem je označení preference kritéria z daného řádku oproti kritériím v jednotlivých sloupcích vyznačením hodnoty 1 ve **Fullerově trojúhelníku**:

	cena	RAM	baterie	hmotnost	expert	skóre	váhy
cena	1	1	1	1	1	5	$\frac{5}{15}$
RAM	0	1	1	1	1	4	$\frac{4}{15}$
baterie	0	0	1	1	0	2	$\frac{2}{15}$
hmotnost	0	0	0	1	0	1	$\frac{1}{15}$
expert	0	0	1	1	1	3	$\frac{3}{15}$

Dostaneme stejné váhy jako u metody pořadí.

VHV - metody odhadu vah kritérií, příklad

Ukažme ještě pro stejné zadání jedno z možných uživatelsky vyplněných **Saatyho matice**.

	cena	RAM	baterie	hmotnost	expert	b_i	v_i
cena	1	4	2	9	2		
RAM	1/4	1	1/2	3	1/2		
baterie	1/2	2	1	7	1		
hmotnost	1/9	1/3	1/7	1	1/7		
expert	1/2	2	1	7	1		

Matice je dostatečně konzistentní. (lze vypočítat index konzistence 0,005)
Doplníme geometrické průměry řádků b_i a z nich odvozené váhy v_i .

VHV - metody odhadu vah kritérií, příklad

Ukažme ještě pro stejné zadání jedno z možných uživatelských vyplnění
Saatyho matice.

	cena	RAM	baterie	hmotnost	expert	b_i	v_i
cena	1	4	2	9	2	2,7	0,41
RAM	1/4	1	1/2	3	1/2	0,72	0,11
baterie	1/2	2	1	7	1	1,48	0,22
hmotnost	1/9	1/3	1/7	1	1/7	0,24	0,04
expert	1/2	2	1	7	1	1,48	0,22

U Saatyho metody vycházejí váhy většinou více diferencované než u ostatních metod.

VHV - klasifikace metod

Existuje celá řada přístupů k řešení úloh VHV, my zmíníme pouze ty jednodušší z nich. Metody lze klasifikovat podle typu informace o preferencích mezi jednotlivými kritérii a variantami, viz následující stručný přehled.

Informace o preferencích mezi variantami					
aspirační úrovně	ordinální informace	kardinální informace			
		funkce užitku	vzdálenost variant od bazální resp. ideální	preferenční relace	mezní míra substituce
Metody					
PRIAM	Lexikografická ORESTE Permutační	WSA	TOPSIS PROMETHEE ELECTRE	AHP	Kompenzační

VHV - metody nevyžadující váhy kritérií

Pokud nejsou známy preference mezi kritérii nebo jsou kritéria rovnocenně důležitá, můžeme pro výběr kompromisní varianty použít pouhá pořadí hodnot u jednotlivých kritérií (v případě shodných hodnot se přiřadí průměrné pořadí). Vybere se ta varianta, která bude mít nejnižší součet pořadí přes všechna kritéria.

V příkladu s tablety pracujeme s pořadími rovnou v posledním sloupci (expertní názor). V ostatních sloupcích nahradíme jednotlivé hodnoty pořadím v rámci sloupce a doplníme jejich řádkové součty:

	cena	RAM	výdrž baterie	hmotnost	OS, procesor display	součet
Tablet 1	3	2,5	2	4	1	12,5
Tablet 2	3	2,5	1	3	3	12,5
Tablet 3	1	4	3	1	4	13
Tablet 4	5	1	5	5	2	18
Tablet 5	1	5	4	2	5	17

Podle uvedeného postupu by se jako nejvýhodnější jevíly tablety 1 a 2.

VHV - metody nevyžadující váhy kritérií

Další metodou nevyžadující apriori váhy kritérií je metoda **PRIAM**, která pracuje s aspiračními úrovněmi kritérií. Varianty jsou pro konkrétní nastavení aspiračních úrovní rozděleny na akceptovatelné a neakceptovatelné. Může nastat situace, kdy nevyhovuje žádná varianta, pak je nutné úrovně některých kritérií uvolnit. Naopak, vyhovuje-li mnoho variant, je možné jejich počet zredukovat zpřísněním některých úrovní. Metoda PRIAM je interaktivním přístupem postupného přizpůsobování aspiračních mezí k dosažení určitého počtu akceptovatelných variant (v krajním případě jediné varianty, kterou pak zvolíme jako kompromisní).

VHV - metody nevyžadující váhy kritérií

Další metodou nevyžadující apriori váhy kritérií je metoda **PRIAM**, která pracuje s aspiračními úrovněmi kritérií. Varianty jsou pro konkrétní nastavení aspiračních úrovní rozděleny na akceptovatelné a neakceptovatelné. Může nastat situace, kdy nevyhovuje žádná varianta, pak je nutné úrovně některých kritérií uvolnit. Naopak, vyhovuje-li mnoho variant, je možné jejich počet zredukovat zpřísněním některých úrovní. Metoda PRIAM je interaktivním přístupem postupného přizpůsobování aspiračních mezí k dosažení určitého počtu akceptovatelných variant (v krajním případě jediné varianty, kterou pak zvolíme jako kompromisní).

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	vyhovuje
Tablet 1	12000	1000	9,5	680	1	ANO
Tablet 2	12000	1000	10	600	3	ANO
Tablet 3	5000	512	7	380	4	ANO
Tablet 4	20000	4000	3	1160	2	ANO
Tablet 5	5000	256	4	400	5	ANO

Pro úlohu s tablety můžeme nastavit výchozí aspirační úrovně na hodnoty bazální varianty $z^0 = (20000, 256, 3, 1160, 5)$, pak vyhovují všechny tablety.

VHV - metody nevyžadující váhy kritérií

Další metodou nevyžadující apriori váhy kritérií je metoda **PRIAM**, která pracuje s aspiračními úrovněmi kritérií. Varianty jsou pro konkrétní nastavení aspiračních úrovní rozděleny na akceptovatelné a neakceptovatelné. Může nastat situace, kdy nevyhovuje žádná varianta, pak je nutné úrovně některých kritérií uvolnit. Naopak, vyhovuje-li mnoho variant, je možné jejich počet zredukovat zpřísněním některých úrovní. Metoda PRIAM je interaktivním přístupem postupného přizpůsobování aspiračních mezí k dosažení určitého počtu akceptovatelných variant (v krajním případě jediné varianty, kterou pak zvolíme jako kompromisní).

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	vyhovuje
Tablet 1	12000	1000	9,5	680	1	ANO
Tablet 2	12000	1000	10	600	3	ANO
Tablet 3	5000	512	7	380	4	ANO
Tablet 4	20000	4000	3	1160	2	NE
Tablet 5	5000	256	4	400	5	ANO

Zpřísněním požadavku na cenu max. 12000Kč dostaneme $z^1 = (12000, 256, 3, 1160, 5)$, takže tablet 4 již nevyhovuje.

VHV - metody nevyžadující váhy kritérií

Další metodou nevyžadující apriori váhy kritérií je metoda **PRIAM**, která pracuje s aspiračními úrovněmi kritérií. Varianty jsou pro konkrétní nastavení aspiračních úrovní rozděleny na akceptovatelné a neakceptovatelné. Může nastat situace, kdy nevyhovuje žádná varianta, pak je nutné úrovně některých kritérií uvolnit. Naopak, vyhovuje-li mnoho variant, je možné jejich počet zredukovat zpřísněním některých úrovní. Metoda PRIAM je interaktivním přístupem postupného přizpůsobování aspiračních mezí k dosažení určitého počtu akceptovatelných variant (v krajním případě jediné varianty, kterou pak zvolíme jako kompromisní).

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	vyhovuje
Tablet 1	12000	1000	9,5	680	1	ANO
Tablet 2	12000	1000	10	600	3	ANO
Tablet 3	5000	512	7	380	4	NE
Tablet 4	20000	4000	3	1160	2	NE
Tablet 5	5000	256	4	400	5	NE

Zpřísněním požadavku na baterii a expertní názor,
 $z^2 = (12000, 256, 7, 1160, 3)$, vyřadíme dále tablety 3 a 5.

VHV - metody nevyžadující váhy kritérií

Další metodou nevyžadující apriori váhy kritérií je metoda **PRIAM**, která pracuje s aspiračními úrovněmi kritérií. Varianty jsou pro konkrétní nastavení aspiračních úrovní rozděleny na akceptovatelné a neakceptovatelné. Může nastat situace, kdy nevyhovuje žádná varianta, pak je nutné úrovně některých kritérií uvolnit. Naopak, vyhovuje-li mnoho variant, je možné jejich počet zredukovat zpřísněním některých úrovní. Metoda PRIAM je interaktivním přístupem postupného přizpůsobování aspiračních mezí k dosažení určitého počtu akceptovatelných variant (v krajním případě jediné varianty, kterou pak zvolíme jako kompromisní).

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	vyhovuje
Tablet 1	12000	1000	9,5	680	1	NE
Tablet 2	12000	1000	10	600	3	ANO
Tablet 3	5000	512	7	380	4	NE
Tablet 4	20000	4000	3	1160	2	NE
Tablet 5	5000	256	4	400	5	NE

Budeme-li požadovat ještě nižší hmotnost, $z^3 = (12000, 256, 7, 600, 3)$, zůstane přijatelný pouze tablet 2.

VHV - metody vyžadující pořadí kritérií

Nejpoužívanější a současně zřejmě nejjednodušší metodou z třídy postupů vyžadujících pouze ordinální informaci o kritériích je metoda **lexikografická**. Řídíme se dle nejdůležitějšího kritéria a je-li nejlépe hodnocená varianta dle tohoto kritéria jediná, je zvolena jako kompromisní. V případě, že by nejlepší hodnoty dosáhlo více variant, vybere se ta z nich, která má lepší hodnocení dle druhého nejdůležitějšího kritéria, atd.

VHV - metody vyžadující pořadí kritérií

Nejpoužívanější a současně zřejmě nejjednodušší metodou z třídy postupů vyžadujících pouze ordinální informaci o kritériích je metoda **lexikografická**.

Řídíme se dle nejdůležitějšího kritéria a je-li nejlépe hodnocená varianta dle tohoto kritéria jediná, je zvolena jako kompromisní. V případě, že by nejlepší hodnoty dosáhlo více variant, vybere se ta z nich, která má lepší hodnocení dle druhého nejdůležitějšího kritéria, atd.

Použitím lexikografické metody na výběr tabletu, je-li prioritním kritériem cena a druhým kritériem výdrž baterie, se rozhodneme pro tablet 3, protože je spolu s tabletem 5 nejlevnější a jeho baterie vydrží 7 hodin oproti 4 hodinám tabletu 5:

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	vyhovuje
Tablet 1	12000	1000	9,5	680	1	NE
Tablet 2	12000	1000	10	600	3	NE
Tablet 3	5000	512	7	380	4	ANO
Tablet 4	20000	4000	3	1160	2	NE
Tablet 5	5000	256	4	400	5	NE

VHV - metody vyžadující kardinální informaci o kritériích

Existují tři základní kategorie přístupů k hodnocení variant využívající vah kritérií, a to:

- maximalizace užitku
- minimalizace vzdálenosti od ideální varianty
- preferenční relace

Z každé skupiny metod uvedeme aspoň jednoho zástupce. První z možností je vyčíslení užitku jednotlivých variant na škále od 0 do 1. Pro vyjádření celkového užitku varianty je třeba nejprve vyjádřit dílčí funkce užitku u_j dle jednotlivých kritérií $j = 1, \dots, k$. Kardinální hodnoty y_{ij} jsou tedy nahrazeny pomocí hodnot $u_{ij} = u_j(y_{ij})$, $j = 1, \dots, k$. Dílčí funkce užitku jsou nastaveny tak, aby ideální varianta měla dílčí užitek dle všech kritérií roven 1 a bazální varianta nulový. Rozlišujeme tři typy funkcí užitku:

- 1 lineární (růst užitku je proporcionální růstu hodnot kritéria)
- 2 progresivní (tempo růstu užitku se při zlepšování hodnot zvyšuje)
- 3 degresivní (tempo růstu užitku se při zlepšování hodnot snižuje)

VHV - metody založené na funkci užitku

Metoda **váženého součtu** (WSA) je založena na konstrukci lineární funkce užitku se stupnicí od 0 do 1. Jestliže pro Y_j označíme D_j nejnižší a H_j nejvyšší kriteriální hodnotu, pak lze při maximalizaci nahradit prvky y_{ij} standardizovanými hodnotami

$$y'_{ij} = \frac{y_{ij} - D_j}{H_j - D_j}.$$

Potom bude mít tedy nejhorší varianta D_j užitek 0 a nejlepší H_j užitek 1. Pro minimalizační kritéria použijeme vztah

$$y'_{ij} = \frac{H_j - y_{ij}}{H_j - D_j}.$$

Tentokrát bude nejhorší varianta H_j a nejlepší D_j a bude jim opět odpovídat užitek 0, resp. 1. Celkový užitek varianty X_i pak spočteme jako vážený součet dílčích užiteků $u(X_i) = \sum_{j=1}^k v_j y'_{ij}$ a varianty pak seřadíme podle klesajících hodnot užitku.

VHV - metoda WSA, příklad

Použijme metodu WSA na problém výběru tabletu, přičemž převezmeme váhy stanovené dle Saatyho metody:

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	užitek
Tablet 1	12000	1000	9,5	680	1	
Tablet 2	12000	1000	10	600	3	
Tablet 3	5000	512	7	380	4	
Tablet 4	20000	4000	3	1160	2	
Tablet 5	5000	256	4	400	5	
váhy	0,41	0,12	0,22	0,03	0,22	
D_j	5000	256	3	380	1	
H_j	20000	4000	10	1160	5	
typ	min	max	max	min	min	

Přepočteme na standardizované hodnoty y'_{ij} .

VHV - metoda WSA, příklad

Použijme metodu WSA na problém výběru tabletu, přičemž převezmeme váhy stanovené dle Saatyho metody:

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	užitek
Tablet 1	0,53	0,2	0,93	0,62	1	
Tablet 2	0,53	0,2	1	0,72	0,5	
Tablet 3	1	0,07	0,57	1	0,25	
Tablet 4	0	1	0	0	0,75	
Tablet 5	1	0	0,14	0,97	0	
váhy	0,41	0,12	0,22	0,03	0,22	
D_j	5000	256	3	380	1	
H_j	20000	4000	10	1160	5	
typ	min	max	max	min	min	

Pro každou variantu spočteme hodnotu agregované funkce užitku.

VHV - metoda WSA, příklad

Použijme metodu WSA na problém výběru tabletu, přičemž převezmeme váhy stanovené dle Saatyho metody:

	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display	užitek
Tablet 1	0,53	0,2	0,93	0,62	1	0,69
Tablet 2	0,53	0,2	1	0,72	0,5	0,59
Tablet 3	1	0,07	0,57	1	0,25	0,63
Tablet 4	0	1	0	0	0,75	0,29
Tablet 5	1	0	0,14	0,97	0	0,47
váhy	0,41	0,12	0,22	0,03	0,22	
D_j	5000	256	3	380	1	
H_j	20000	4000	10	1160	5	
typ	min	max	max	min	min	

Největší užitek nám přinese tablet 1 (v závěsu s tabletem 3).

VHV - metody založené na optimalizaci vzdálenosti od bazální a ideální varianty

Metoda **TOPSIS** spočívá ve výběru varianty, která je co nejbližší tzv. **ideální variantě** a současně je co nejdál od tzv. **bazální varianty**. Popišme její postup pro případ, kdy že jsou všechna kritéria maximalizační. Postupuje se v následujících krocích:

- 1 Normalizace: hodnoty y_{ij} se transformují na r_{ij} podle vztahu

$$r_{ij} = \frac{y_{ij}}{\sqrt{\sum_{i=1}^n y_{ij}^2}}.$$

- 2 Vypočte se vážená kritériální matice $W = (w_{ij})$ jako $w_{ij} = v_j r_{ij}$.
- 3 Z prvků W se vybere ideální varianta $H = (H_1, \dots, H_k)$ a bazální varianta $D = (D_1, \dots, D_k)$, kde $H_j = \max_i(w_{ij})$, $D_j = \min_i(w_{ij})$. (Pozor! Hodnoty H_j a D_j se neshodují s těmi dříve zavedenými - jsou spočteny po normalizaci a zvážení sloupců!)
- 4 Vypočtou se vzdálenosti variant od ideální a bazální varianty

$$d_i^+ = \sqrt{\sum_{j=1}^k (w_{ij} - H_j)^2}, \quad d_i^- = \sqrt{\sum_{j=1}^k (w_{ij} - D_j)^2}$$

- 5 Nakonec se vypočte relativní vzdálenost od bazální varianty $c_i = \frac{d_i^-}{d_i^+ + d_i^-}$ a varianty se podle ní sestupně uspořádají.

VHV - metoda TOPSIS, příklad

Použijme metodu TOPSIS pro výběr tabletu, jako váhy opět vezmeme Saatyho odhady.

	cena	RAM	výdrž baterie	hmotnost	expert display	d_i^+	d_i^-	c_i
T1	0,442	0,234	0,584	0,432	0,135			
T2	0,442	0,234	0,615	0,382	0,405			
T3	0,184	0,120	0,431	0,242	0,539			
T4	0,736	0,934	0,185	0,738	0,270			
T5	0,184	0,06	0,246	0,254	0,674			
v_j	0,41	0,12	0,22	0,03	0,22			
typ	min	max	max	min	min			
D_j								
H_j								

Přepočteme na normalizované hodnoty r_{ij} .

VHV - metoda TOPSIS, příklad

Použijme metodu TOPSIS pro výběr tabletu, jako váhy opět vezmeme Saatyho odhady.

	cena	RAM	výdrž baterie	hmotnost	expert	d_i^+	d_i^-	c_i
T1	0,181	0,028	0,129	0,013	0,03			
T2	0,181	0,028	0,135	0,011	0,089			
T3	0,075	0,014	0,095	0,007	0,119			
T4	0,302	0,112	0,041	0,022	0,059			
T5	0,075	0,007	0,054	0,008	0,148			
v_j	0,41	0,12	0,22	0,03	0,22			
typ	min	max	max	min	min			
D_j	0,302	0,007	0,041	0,022	0,148			
H_j	0,075	0,112	0,135	0,007	0,030			

Spočteme matici W_{ij} a určíme ideální a bazální variantu: H , D .

VHV - metoda TOPSIS, příklad

Použijme metodu TOPSIS pro výběr tabletu, jako váhy opět vezmeme Saatyho odhady.

	cena	RAM	výdrž baterie	hmotnost	expert	d_i^+	d_i^-	c_i
T1	0,181	0,028	0,129	0,013	0,03	0,135	0,192	0,587
T2	0,181	0,028	0,135	0,011	0,089	0,148	0,166	0,53
T3	0,075	0,014	0,095	0,007	0,119	0,138	0,235	0,63
T4	0,302	0,112	0,041	0,022	0,059	0,248	0,138	0,357
T5	0,075	0,007	0,054	0,008	0,148	0,178	0,227	0,56
v_j	0,41	0,12	0,22	0,03	0,22			
typ	min	max	max	min	min			
D_j	0,302	0,007	0,041	0,022	0,148			
H_j	0,075	0,112	0,135	0,007	0,030			

Vypočteme vzdálenosti od ideální a bazální varianty d_i^+ a d_i^- a relativní index vzdálenosti od bazální varianty c_i . Nejlépe se jeví tablet 3.

VHV - metody třídy ELECTRE

Významnou skupinu metod tvoří metody založené na vyhodnocování podle **preferenční relace**. Z dílčích preferencí podle jednotlivých kritérií se vhodnými agregačními procedurami odvodí celková preference. Ta však bohužel nemusí být tranzitivní, takže je nutný další postup, který stanoví celkové pořadí variant nebo je alespoň rozdělí do několika indifferenčních tříd. U metody **ELECTRE I** jde o rozdělení na tzv. **efektivní a neefektivní** varianty.

VHV - metody třídy ELECTRE

Významnou skupinu metod tvoří metody založené na vyhodnocování podle **preferenční relace**. Z dílčích preferencí podle jednotlivých kritérií se vhodnými agregačními procedurami odvodí celková preference. Ta však bohužel nemusí být tranzitivní, takže je nutný další postup, který stanoví celkové pořadí variant nebo je alespoň rozdělí do několika indifferenčních tříd. U metody

ELECTRE I jde o rozdělení na tzv. **efektivní a neefektivní** varianty.

Předpokládejme, že všechna kritéria jsou maximalizační. Pro každou dvojici variant X_i, X_j určíme množiny

$$C_{ij} = \{h \in \{1, \dots, k\}; y_{ih} \geq y_{jh}\}$$

indexů kritérií, podle nichž je varianta X_i alespoň tak dobrá jako X_j a

$$D_{ij} = \{h \in \{1, \dots, k\}; y_{ih} < y_{jh}\}$$

indexů kritérií, podle nichž je varianta X_i horší než X_j

VHV - metody třídy ELECTRE

Pro stanovené váhy kritérií v_1, \dots, v_k pak vypočteme pro každou dvojici variant **stupeň preference** vztahem

$$c_{ij} = \sum_{h \in G_{ij}} v_h$$

Platí $c_{ij} \in \langle 0, 1 \rangle$.

VHV - metody třídy ELECTRE

Pro stanovené váhy kritérií v_1, \dots, v_k pak vypočteme pro každou dvojici variant **stupeň preference** vztahem

$$c_{ij} = \sum_{h \in C_{ij}} v_h$$

Platí $c_{ij} \in \langle 0, 1 \rangle$.

V dalším kroku se pro každou dvojici variant vypočte **stupeň dispreference** vztahem

$$d_{ij} = \frac{\max_{h \in D_{ij}} |y_{ih} - y_{jh}|}{\max_{h=1, \dots, k} |y_{ih} - y_{jh}|},$$

v případě $D_{ij} = \emptyset$ se definuje $d_{ij} = 0$. Opět platí $d_{ij} \in \langle 0, 1 \rangle$.

VHV - metody třídy ELECTRE

Pro stanovené váhy kritérií v_1, \dots, v_k pak vypočteme pro každou dvojici variant **stupeň preference** vztahem

$$c_{ij} = \sum_{h \in C_{ij}} v_h$$

Platí $c_{ij} \in \langle 0, 1 \rangle$.

V dalším kroku se pro každou dvojici variant vypočte **stupeň dispreference** vztahem

$$d_{ij} = \frac{\max_{h \in D_{ij}} |y_{ih} - y_{jh}|}{\max_{h=1, \dots, k} |y_{ih} - y_{jh}|},$$

v případě $D_{ij} = \emptyset$ se definuje $d_{ij} = 0$. Opět platí $d_{ij} \in \langle 0, 1 \rangle$.

Pomocí vhodně stanovených konstant **prahu preference** c^* a **prahu dispreference** d^* se rozhodne o celkové preferenci mezi X_i a X_j :

X_i preferujeme před X_j , platí-li $c_{ij} \geq c^* \wedge d_{ij} \leq d^*$.

VHV - metody třídy ELECTRE

Pro stanovené váhy kritérií v_1, \dots, v_k pak vypočteme pro každou dvojici variant **stupeň preference** vztahem

$$c_{ij} = \sum_{h \in C_{ij}} v_h$$

Platí $c_{ij} \in \langle 0, 1 \rangle$.

V dalším kroku se pro každou dvojici variant vypočte **stupeň dispreference** vztahem

$$d_{ij} = \frac{\max_{h \in D_{ij}} |y_{ih} - y_{jh}|}{\max_{h=1, \dots, k} |y_{ih} - y_{jh}|},$$

v případě $D_{ij} = \emptyset$ se definuje $d_{ij} = 0$. Opět platí $d_{ij} \in \langle 0, 1 \rangle$.

Pomocí vhodně stanovených konstant **prahu preference** c^* a **prahu dispreference** d^* se rozhodne o celkové preferenci mezi X_i a X_j :

X_i preferujeme před X_j , platí-li $c_{ij} \geq c^* \wedge d_{ij} \leq d^*$.

Za **efektivní** považujeme ty varianty, které jsou preferovány před alespoň jednou variantou, ale současně není žádná varianta preferována před nimi. Výsledek závisí na volbě prahů, postupnými změnami jejich hodnot můžeme dostat pouze jedinou efektivní variantu a tu pak brát jako "nejlepší".

VHV - metoda ELECTRE I - příklad

Pro naši úlohu o tabletech určíme množiny C_{ij}

C_{ij}	T1	T2	T3	T4	T5
T1		{1, 2, 5}	{2, 3, 5}	{1, 3, 4, 5}	{2, 3, 5}
T2	{1, 2, 3, 4}		{2, 3, 5}	{1, 3, 4}	{2, 3, 5}
T3	{1, 4}	{1, 4}		{1, 4}	{1, 2, 3, 4, 5}
T4	{2}	{2, 5}	{2, 3, 5}		{2, 5}
T5	{1, 4}	{1, 4}	{1}	{1, 3, 4}	

VHV - metoda ELECTRE I - příklad

Pro naši úlohu o tabletech určíme množiny C_{ij}

C_{ij}	T1	T2	T3	T4	T5
T1		{1, 2, 5}	{2, 3, 5}	{1, 3, 4, 5}	{2, 3, 5}
T2	{1, 2, 3, 4}		{2, 3, 5}	{1, 3, 4}	{2, 3, 5}
T3	{1, 4}	{1, 4}		{1, 4}	{1, 2, 3, 4, 5}
T4	{2}	{2, 5}	{2, 3, 5}		{2, 5}
T5	{1, 4}	{1, 4}	{1}	{1, 3, 4}	

Pro Saatyho váhy (0,41; 0,12; 0,22; 0,03; 0,22)
dopočteme tabulku stupňů preference:

C_{ij}	T1	T2	T3	T4	T5
T1		0,75	0,56	0,88	0,56
T2	0,78		0,56	0,66	0,56
T3	0,44	0,44		0,44	1
T4	0,12	0,34	0,56		0,34
T5	0,44	0,44	0,41	0,66	

VHV - metoda ELECTRE I - příklad

Určíme také množiny D_{ij}

D_{ij}	T1	T2	T3	T4	T5
T1		{3, 4}	{1, 4}	{2}	{1, 4}
T2	{5}		{1, 4}	{2, 5}	{1, 4}
T3	{2, 3, 5}	{2, 3, 5}		{2, 3, 5}	{}
T4	{1, 3, 4, 5}	{1, 3, 4}	{1, 4}		{1, 3, 4}
T5	{2, 3, 5}	{2, 3, 5}	{2, 3, 4, 5}	{2, 5}	

VHV - metoda ELECTRE I - příklad

Určíme také množiny D_{ij}

D_{ij}	T1	T2	T3	T4	T5
T1		{3, 4}	{1, 4}	{2}	{1, 4}
T2	{5}		{1, 4}	{2, 5}	{1, 4}
T3	{2, 3, 5}	{2, 3, 5}		{2, 3, 5}	{}
T4	{1, 3, 4, 5}	{1, 3, 4}	{1, 4}		{1, 3, 4}
T5	{2, 3, 5}	{2, 3, 5}	{2, 3, 4, 5}	{2, 5}	

Dopočteme tabulku stupňů dispreference, vyjdeme z standardizovaných hodnot y'_{ij} :

d_{ij}	T1	T2	T3	T4	T5
T1		0,1/0,5	0,47/0,75	0,8/0,93	0,47/1
T2	0,5/0,5		0,47/0,47	0,8/1	0,47/0,86
T3	0,75/0,75	0,28/0,47		0,93/1	0/0,43
T4	0,93/0,93	1/1	1/1		1/1
T5	1/1	0,86/0,86	0,43/0,43	1/1	

VHV - metoda ELECTRE I - příklad

Stanovíme práh preference $c^* = 0,5$ a vyznačíme $c_{ij} \geq c^*$.

c_{ij}	T1	T2	T3	T4	T5
T1		0,75	0,56	0,88	0,56
T2	0,78		0,56	0,66	0,56
T3	0,44	0,44		0,44	1
T4	0,12	0,34	0,56		0,34
T5	0,44	0,44	0,41	0,66	

VHV - metoda ELECTRE I - příklad

Stanovíme práh preference $c^* = 0,5$ a vyznačíme $c_{ij} \geq c^*$.

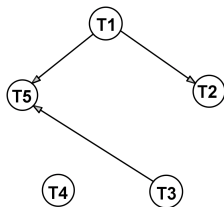
c_{ij}	T1	T2	T3	T4	T5
T1		0,75	0,56	0,88	0,56
T2	0,78		0,56	0,66	0,56
T3	0,44	0,44		0,44	1
T4	0,12	0,34	0,56		0,34
T5	0,44	0,44	0,41	0,66	

Stanovíme práh dispreference $d^* = 0,5$ a vyznačíme $d_{ij} \leq d^*$.

d_{ij}	T1	T2	T3	T4	T5
T1		0,2	0,62	0,86	0,47
T2	1		1	0,8	0,55
T3	1	0,6		0,93	0
T4	1	1	1		1
T5	1	1	1	1	

VHV - metoda ELECTRE I - příklad

Relaci preference můžeme znázornit v matici nebo v grafu (zahrneme pouze ty hrany, kde jsou splněny obě podmínky pro preferenci: dosažení prahu preference c^* a nepřekročení prahu dispreference d^*):



Získali jsme dvě efektivní varianty 1 a 3 (uzly, ze kterých vycházejí šipky, ale žádné nevstupují).

VHV - metody třídy PROMETHEE

Patří mezi nejčastěji používané postupy VHV. Základem je párové porovnávání variant postupně podle všech kritérií. Pro varianty X_i, X_j definujeme **intenzitu preference** podle kritéria Y_h :

$$P_h(X_i, X_j)$$

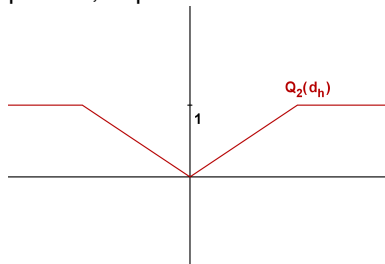
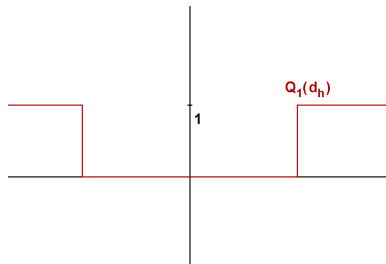
jako funkci nabývající hodnoty z intervalu $\langle 0, 1 \rangle$, takovou, že $P_h(X_i, X_j) = 0$ není-li X_i preferováno před X_j a $P_h(X_i, X_j) = 1$ v případě absolutní preference.

VHV - metody třídy PROMETHEE

Patří mezi nejčastěji používané postupy VHV. Základem je párové porovnávání variant postupně podle všech kritérií. Pro varianty X_i, X_j definujeme **intenzitu preference** podle kritéria Y_h :

$$P_h(X_i, X_j)$$

jako funkci nabývající hodnoty z intervalu $\langle 0, 1 \rangle$, takovou, že $P_h(X_i, X_j) = 0$ není-li X_i preferováno před X_j a $P_h(X_i, X_j) = 1$ v případě absolutní preference. Intenzita preference závisí na diferenci $d_h = (y_{ih} - y_{jh})$, transformace Q převádějící diferenci d_h na intenzitu preference $P_h(X_i, X_j)$ se nazývá **zobecněné kritérium** a může mít různý průběh, například viz obrázek:



VHV - metody třídy PROMETHEE

Jsou-li stanoveny váhy kritérií, můžeme spočítat **globální preferenční indexy** podle vztahu

$$P(X_i, X_j) = \sum_{h=1}^k v_h P_h(X_i, X_j).$$

Pro získání výsledné relace vypočteme pro každou variantu pozitivní (výstupní) a negativní (vstupní) **tok** jako

$$F^+(X_i) = \sum_{j=1}^n P(X_i, X_j)/(n-1),$$

$$F^-(X_i) = \sum_{j=1}^n P(X_j, X_i)/(n-1),$$

VHV - metody třídy PROMETHEE

Jsou-li stanoveny váhy kritérií, můžeme spočítat **globální preferenční indexy** podle vztahu

$$P(X_i, X_j) = \sum_{h=1}^k v_h P_h(X_i, X_j).$$

Pro získání výsledné relace vypočteme pro každou variantu pozitivní (výstupní) a negativní (vstupní) **tok** jako

$$F^+(X_i) = \sum_{j=1}^n P(X_i, X_j)/(n-1),$$

$$F^-(X_i) = \sum_{j=1}^n P(X_j, X_i)/(n-1),$$

Jednotlivé metody se v postupu dále liší, PROMETHEE II například uspořádává varianty podle tzv. **čistého toku**

$$F(X_i) = F^+(X_i) - F^-(X_i).$$

VHV - metoda PROMETHEE, příklad

Pro úlohu s tablety vyjdeme při výpočtu diferencí d_h ze standardizovaných hodnot y'_{ij} :

y'_{ij}	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display
Tablet 1	0,53	0,2	0,93	0,62	1
Tablet 2	0,53	0,2	1	0,72	0,5
Tablet 3	1	0,07	0,57	1	0,25
Tablet 4	0	1	0	0	0,75
Tablet 5	1	0	0,14	0,97	0

VHV - metoda PROMETHEE, příklad

Pro úlohu s tablety vyjdeme při výpočtu diferencí d_h ze standardizovaných hodnot y'_{ij} :

y'_{ij}	cena	RAM	výdrž baterie	hmotnost	OS, procesor, display
Tablet 1	0,53	0,2	0,93	0,62	1
Tablet 2	0,53	0,2	1	0,72	0,5
Tablet 3	1	0,07	0,57	1	0,25
Tablet 4	0	1	0	0	0,75
Tablet 5	1	0	0,14	0,97	0

Nejprve spočteme všechny difference a pak s využitím zobecněného kritéria Q_1 s prahem $d^* = 0,5$ přepočítáme na intenzity preference $P_h(X_i, X_j)$. Výpočet ukážeme jen pro d_1 .

VHV - metoda PROMETHEE, příklad

$d_1(X_i, X_j)$	T1	T2	T3	T4	T5
T1	0	0	-0,47	0,53	-0,47
T2	0	0	-0,47	0,53	-0,47
T3	0,47	0,47	0	1	0
T4	-0,53	-0,53	-1	0	1
T5	0,47	0,47	0	-1	0

VHV - metoda PROMETHEE, příklad

$d_1(X_i, X_j)$	T1	T2	T3	T4	T5
T1	0	0	-0,47	0,53	-0,47
T2	0	0	-0,47	0,53	-0,47
T3	0,47	0,47	0	1	0
T4	-0,53	-0,53	-1	0	1
T5	0,47	0,47	0	-1	0

Intenzity preference jsou:

$P_1(X_i, X_j)$	T1	T2	T3	T4	T5
T1	0	0	0	1	0
T2	0	0	0	1	0
T3	0	0	0	1	0
T4	0	0	0	0	1
T5	0	0	0	0	0

VHV - metoda PROMETHEE, příklad

Proces opakujeme pro všechna další kritéria a nakonec pomocí vah (0,41; 0,12; 0,22; 0,03; 0,22) agregujeme preferenční intenzity a získáme globální preferenční indexy:

$P(X_i, X_j)$	T1	T2	T3	T4	T5	$F^+(X_j)$
T1	0	0	0,22	0,66	0,22	
T2	0	0	0	0,44	0,22	
T3	0	0	0	0,66	0,22	
T4	0,12	0,12	0,12	0	0,75	
T5	0	0	0	0	0	
$F^-(X_i)$						

Zprůměrováním řádků a sloupců získáme pozitivní a negativní toky.

VHV - metoda PROMETHEE, příklad

Proces opakujeme pro všechna další kritéria a nakonec pomocí vah (0,41; 0,12; 0,22; 0,03; 0,22) agregujeme preferenční intenzity a získáme globální preferenční indexy:

$P(X_i, X_j)$	T1	T2	T3	T4	T5	$F^+(X_i)$
T1	0	0	0,22	0,66	0,22	0,275
T2	0	0	0	0,44	0,22	0,165
T3	0	0	0	0,66	0,22	0,22
T4	0,12	0,12	0,12	0	0,75	0,2775
T5	0	0	0	0	0	0
$F^-(X_i)$	0,03	0,03	0,085	0,44	0,3525	

Rozdílem pozitivních a negativních toků získáme celkový tok:

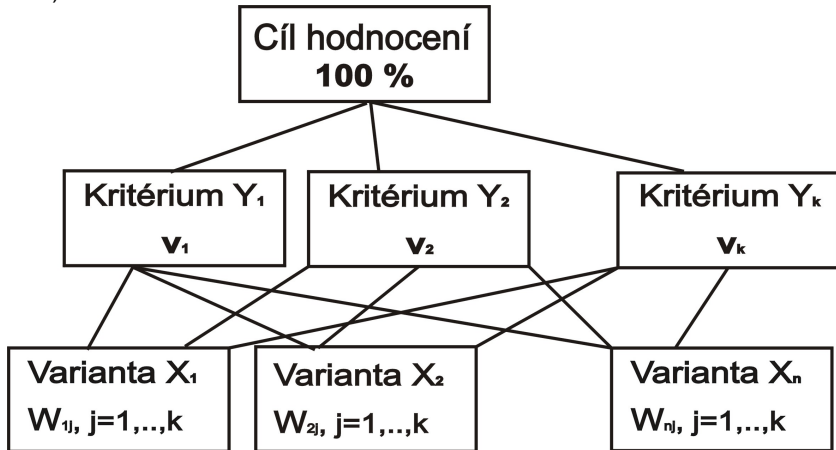
$$F(T1) = 0,245, F(T2) = 0,135, F(T3) = 0,135,$$

$$F(T4) = -0,1625, F(T5) = -0,3525$$

Nejlépe vychází tablet 1.

VHV - metoda AHP

Metoda **AHP** (Analytic Hierarchy Process) modeluje rozhodovací problém pomocí hierarchické struktury, která má pro nejjednodušší úlohy tři úrovně (viz obr.)



Intenzitu vztahu mezi jednotlivými prvky hierarchie můžeme vyjádřit pomocí dělení počáteční jednotky (100 %) podle preferencí rozhodovatele na další úrovně. Nejprve jsou na druhé úrovni přiřazeny kritériím váhy v_j , $j = 1, \dots, k$. Tyto váhy se dále rozdělují jednotlivým variantám podle toho, jak dobře či špatně jsou tyto varianty dle daného kritéria ohodnoceny, čímž pro dané kritérium dostaneme preferenční indexy w_{ij} , $i = 1, \dots, n$. Z konstrukce modelu vyplývá, že platí vztahy $\sum_{j=1}^k v_j = 1$, $\sum_{i=1}^n w_{ij} = v_j$, $j = 1, \dots, k$ Celkový užitek, podle nějž lze varianty uspořádat, se vypočte jako $u(X_i) = \sum_{j=1}^k w_{ij}$.

VHV - metoda AHP

Intenzitu vztahu mezi jednotlivými prvky hierarchie můžeme vyjádřit pomocí dělení počáteční jednotky (100 %) podle preferencí rozhodovatele na další úrovni. Nejprve jsou na druhé úrovni přiřazeny kritériím váhy v_j , $j = 1, \dots, k$. Tyto váhy se dále rozdělují jednotlivým variantám podle toho, jak dobře či špatně jsou tyto varianty dle daného kritéria ohodnoceny, čímž pro dané kritérium dostaneme preferenční indexy w_{ij} , $i = 1, \dots, n$. Z konstrukce modelu vyplývá, že platí vztahy $\sum_{j=1}^k v_j = 1$, $\sum_{i=1}^n w_{ij} = v_j$, $j = 1, \dots, k$ Celkový

užitek, podle nějž lze varianty uspořádat, se vypočte jako $u(X_i) = \sum_{j=1}^k w_{ij}$.

Vlastní numerická realizace je založena na párovém porovnávání prvků podobně jako u Saatyho metody: Pro nejvyšší uzel se sestaví porovnávací matice ($k \times k$) a z ní se odvodí váhy kritérií. Následně se pro každé kritérium určí preference variant párovým porovnáváním v matici ($n \times n$). Nevýhodou metody je tedy očividně velký objem porovnávání, což je na druhou stranu vyváženo její univerzální použitelností a možností použití verbální stupnice pro vyjádření preferencí.

Vícekriteriální programování

Ve vícekriteriálním programování jde o optimalizaci více účelových funkcí na přípustné množině definované sadou omezení. Narozdíl od úloh VHV je množina variant v úlohách nekonečná a kritéria jsou definována v podobě funkcí. Jsou-li všechny účelové funkce i omezující podmínky lineární, mluvíme o vícekriteriálním lineárním programování (VLP). Problém VLP tedy můžeme formulovat jako úlohu "optimalizovat"

$$z_1 = \mathbf{c}^1 \cdot \mathbf{x}, z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \dots, z_k = \mathbf{c}^k \cdot \mathbf{x},$$

za podmínek

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

kde \mathbf{c}^i je cenový vektor i -té účelové funkce.

Vícekriteriální programování

Ve vícekriteriálním programování jde o optimalizaci více účelových funkcí na přípustné množině definované sadou omezení. Narozdíl od úloh VHV je množina variant v úlohách nekonečná a kritéria jsou definována v podobě funkcí. Jsou-li všechny účelové funkce i omezující podmínky lineární, mluvíme o vícekriteriálním lineárním programování (VLP). Problém VLP tedy můžeme formulovat jako úlohu "optimalizovat"

$$z_1 = \mathbf{c}^1 \cdot \mathbf{x}, z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \dots, z_k = \mathbf{c}^k \cdot \mathbf{x},$$

za podmínek

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

kde \mathbf{c}^i je cenový vektor i -té účelové funkce.

Pomocí ekvivalence minimalizační úlohy pro z_i s maximalizační úlohou pro $-z_i$ můžeme převést úlohu do takové podoby, aby všechna kritéria měla maximalizační charakter. Úlohu je pak možné zapsat pomocí maticového zápisu, označíme-li $\mathbf{z} = (z_1, z_2, \dots, z_k)$ vektor účelových funkcí a \mathbf{C} matici vytvořenou z cenových vektorů $\mathbf{c}^1, \mathbf{c}^2, \dots, \mathbf{c}^k$:

$$\mathbf{z} = \mathbf{C} \cdot \mathbf{x} \rightarrow \text{MAX}, \mathbf{x} \in \mathbf{X}.$$

Model VLP - základní pojmy

Podobně jako u VHV je typicky cílem nalézt v množině všech přípustných řešení pomocí vhodného postupu nějaké prakticky přijatelné, tzv. **kompromisní** řešení. Je dobré si uvědomit, že při hledání kompromisního řešení se opět stačí omezit na **nedominovaná** řešení. Řešení $\mathbf{x} \in \mathbf{X}$ je nedominované, pokud neexistuje žádné přípustné řešení, jehož vektor kritériálních hodnot by byl ve všech složkách větší nebo roven vektoru $\mathbf{C} \cdot \mathbf{x}$ (s vyloučením případu rovnosti vektorů).

Model VLP - základní pojmy

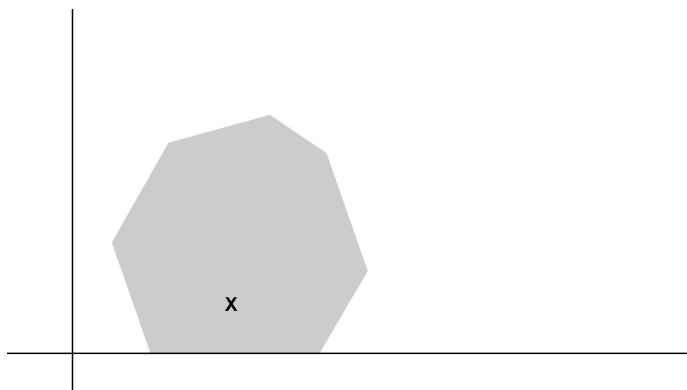
Podobně jako u VHV je typicky cílem nalézt v množině všech přípustných řešení pomocí vhodného postupu nějaké prakticky přijatelné, tzv.

kompromisní řešení. Je dobré si uvědomit, že při hledání kompromisního řešení se opět stačí omezit na **nedominovaná** řešení. Řešení $\mathbf{x} \in \mathbf{X}$ je nedominované, pokud neexistuje žádné přípustné řešení, jehož vektor kritériálních hodnot by byl ve všech složkách větší nebo roven vektoru $\mathbf{C} \cdot \mathbf{x}$ (s vyloučením případu rovnosti vektorů).

Většina principů pro hledání kompromisního řešení je založena na řešení dílčích úloh lineárního programování $z_i = \mathbf{c}^i \cdot \mathbf{x} \rightarrow \max, \mathbf{x} \in \mathbf{X}$ standardní simplexovou metodou. Vektor \mathbf{x}_H , ve kterém všechny účelové funkce nabývají svých optimálních hodnot nazýváme **ideální řešení**. Analogicky bychom mohli zavést **bazální řešení** \mathbf{x}_D . Optimální a bazální řešení zpravidla neleží v přípustné množině.

Model VLP - grafické znázornění

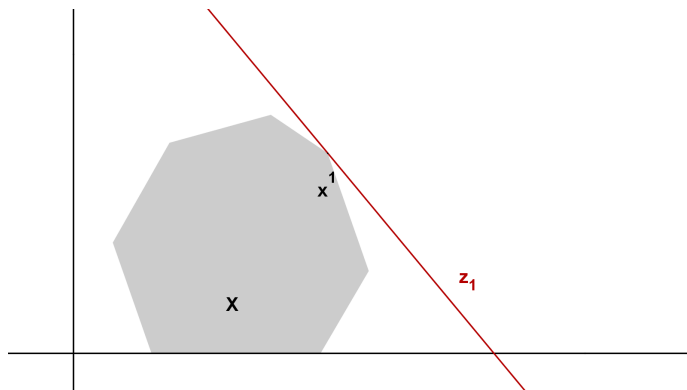
Vícekritériální lineární model je možné zobrazovat v rozhodovacím nebo kritériálním prostoru. Nejprve schematicky znázorníme úlohu se dvěma proměnnými v rozhodovacím prostoru, kde souřadné osy budou reprezentovat hodnoty proměnných.



Vyznačíme přípustnou množinu X .

Model VLP - grafické znázornění

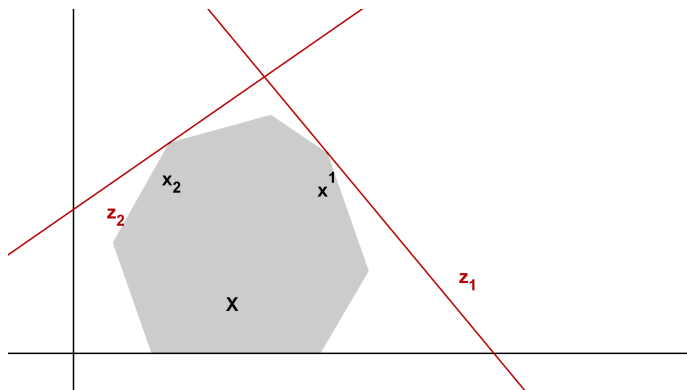
Vícekriteriální lineární model je možné zobrazovat v rozhodovacím nebo kriteriálním prostoru. Nejprve schematicky znázorníme úlohu se dvěma proměnnými v rozhodovacím prostoru, kde souřadné osy budou reprezentovat hodnoty proměnných.



Dílčí úloha LP pro účelovou funkci $z_1 = c^1 \cdot x$ má optimum v bodě x^1 .

Model VLP - grafické znázornění

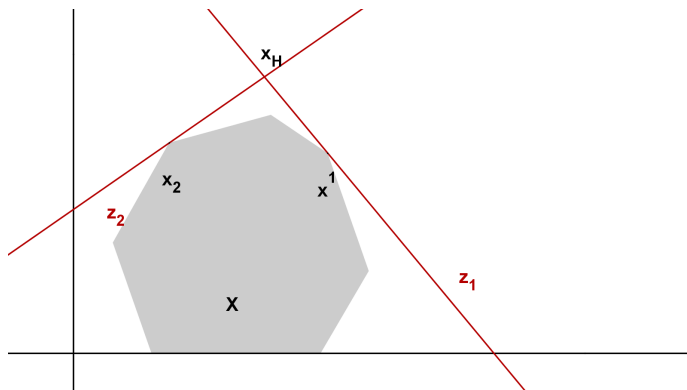
Vícekriteriální lineární model je možné zobrazovat v rozhodovacím nebo kriteriálním prostoru. Nejprve schematicky znázorníme úlohu se dvěma proměnnými v rozhodovacím prostoru, kde souřadné osy budou reprezentovat hodnoty proměnných.



Dílčí úloha LP pro účelovou funkci $\mathbf{z}_2 = \mathbf{c}^2 \cdot \mathbf{x}$ má optimum v bodě \mathbf{x}^2 .

Model VLP - grafické znázornění

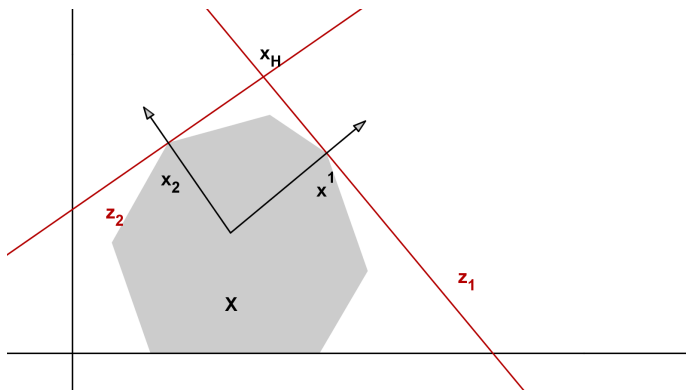
Vícekriteriální lineární model je možné zobrazovat v rozhodovacím nebo kriteriálním prostoru. Nejprve schematicky znázorníme úlohu se dvěma proměnnými v rozhodovacím prostoru, kde souřadné osy budou reprezentovat hodnoty proměnných.



Průsečík přímek znázorňujících účelové funkce vedené v dílčích optimech představuje ideální řešení x_H .

Model VLP - grafické znázornění

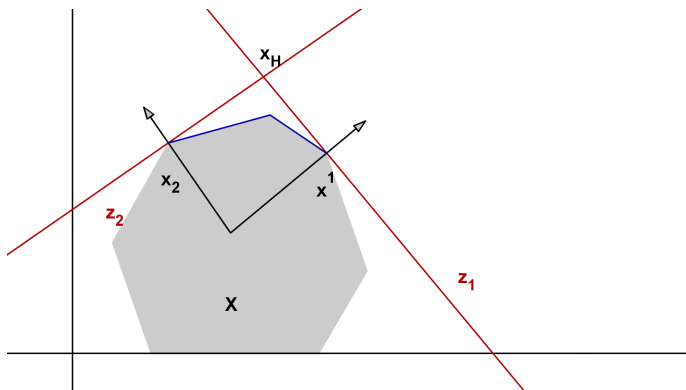
Vícekriteriální lineární model je možné zobrazovat v rozhodovacím nebo kriteriálním prostoru. Nejprve schematicky znázorníme úlohu se dvěma proměnnými v rozhodovacím prostoru, kde souřadné osy budou reprezentovat hodnoty proměnných.



Normálové vektory k těmto přímkám (tedy vektory cenových koeficientů c^i) vyjadřují směr růstu účelových funkcí. Jejich nezáporné lineární kombinace definují tzv. **kriteriální kužel**.

Model VLP - grafické znázornění

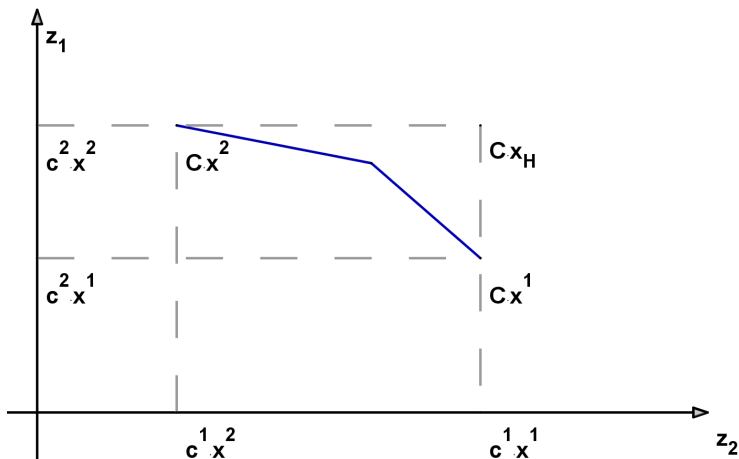
Vícekriteriální lineární model je možné zobrazovat v rozhodovacím nebo kriteriálním prostoru. Nejprve schematicky znázorníme úlohu se dvěma proměnnými v rozhodovacím prostoru, kde souřadné osy budou reprezentovat hodnoty proměnných.



Na hranici množiny přípustných řešení vymezené kriteriálním kuželem leží všechna nedominovaná řešení.

Model VLP - grafické znázornění

Při zobrazování v kriteriálním prostoru vynášíme na jednotlivé osy přímo hodnoty účelových funkcí.



Modře jsou vyznačeny nedominované hodnoty

Vícekriteriální programování - klasifikace metod

Při řešení úloh VLP můžeme požadovat výsledky v podobě úplného popisu množiny nedominovaných řešení nebo nalezení nějaké její reprezentativní podmnožiny nebo výběru několika kompromisních variant. Chce-li uživatel vybrat jedinou kompromisní variantu, bude jeho rozhodnutí značně záviset na preferencích jednotlivých kritérií. Ty mohou být zadávány v různých fázích výpočtu:

- **před** započítím výpočtu,
- interaktivně **v průběhu** výpočtu,
- **po** skončení výpočtu,

a to opět prostřednictvím aspiračních úrovní kritérií, pořadí kritérií, váhami kritérií nebo mírou kompenzace kriteriálních hodnot.

Vícekriteriální programování - klasifikace metod

Při řešení úloh VLP můžeme požadovat výsledky v podobě úplného popisu množiny nedominovaných řešení nebo nalezení nějaké její reprezentativní podmnožiny nebo výběru několika kompromisních variant. Chce-li uživatel vybrat jedinou kompromisní variantu, bude jeho rozhodnutí značně záviset na preferencích jednotlivých kritérií. Ty mohou být zadávány v různých fázích výpočtu:

- před započítáním výpočtu,
- interaktivně v průběhu výpočtu,
- po skončení výpočtu,

a to opět prostřednictvím aspiračních úrovní kritérií, pořadí kritérií, váhami kritérií nebo mírou kompenzace kritériálních hodnot. Podle toho, kdy je informace o kritériích do modelu zapracována, dělíme výpočetní metody na:

- metody s preferenční informací **a priori**
- metody s preferenční informací **a posteriori**
- metody **s postupným zpřesňováním** preferenční informace
- metody kombinované

Vícekriteriální programování - klasifikace metod

Metody s informací **apriori** dělíme do několika skupin:

- metoda lexikografická
- metoda minimální komponenty
- řešení jednokriteriální úlohy s agregovanými kritérii
- záměna kritérií za omezení
- "minimalizace odchylek" od ideálních hodnot (při vhodně zvolené metrice)

Vícekriteriální programování - klasifikace metod

Metody s informací **apriori** dělíme do několika skupin:

- metoda lexikografická
- metoda minimální komponenty
- řešení jednokriteriální úlohy s agregovanými kritérii
- záměna kritérií za omezení
- "minimalizace odchylek" od ideálních hodnot (při vhodně zvolené metrice)

Metody s informací **aposteriori** spočívají v popisu množiny nedominovaných řešení, ve které následně uživatel vybírá kompromisní řešení. Patří sem:

- metoda parametrická (agregujeme kritéria s parametricky zadaným vektorem vah)
- metoda omezení (hledá nedominovaná řešení, kde hodnoty kritérií dosahují parametricky zadaných cílových hodnot)
- vícekriteriální simplexový algoritmus (postupně určuje nedominovaná bazická řešení)

Vícekriteriální programování - klasifikace metod

Metody s informací **apriori** dělíme do několika skupin:

- metoda lexikografická
- metoda minimální komponenty
- řešení jednokriteriální úlohy s agregovanými kritérii
- záměna kritérií za omezení
- "minimalizace odchylek" od ideálních hodnot (při vhodně zvolené metrice)

Metody s informací **aposteriori** spočívají v popisu množiny nedominovaných řešení, ve které následně uživatel vybírá kompromisní řešení. Patří sem:

- metoda parametrická (agregujeme kritéria s parametricky zadaným vektorem vah)
- metoda omezení (hledá nedominovaná řešení, kde hodnoty kritérií dosahují parametricky zadaných cílových hodnot)
- vícekriteriální simplexový algoritmus (postupně určuje nedominovaná bazická řešení)

Metody **interaktivní** probíhají iterativně a jsou založeny na komunikaci mezi uživatelem a rozhodovatelem.

Vícekriteriální programování - příklad

Příklad z knihy Tomáš Šubrt, Ekonomicko-matematické metody: Vedení půjčovny lyžařského a snowboardového vybavení zvažuje optimalizaci sortimentu zboží. Do kalkulací zahrnuje obvyklou cenu za půjčení vybavení i riziko, že utrpí ztrátu, protože vybavení nebude půjčeno. Denní zisk [v Kč] při půjčení jednotlivých kompletů i riziko ztráty při nepůjčení [v bodech] jsou uvedeny v tabulce.

	lyžařský komplet sjezdový dospělý	lyžařský komplet sjezdový dětský	lyžařský komplet běžecký	snowboardový komplet
zisk	300	200	170	250
riziko	10	15	25	5

Společnost chce investovat nejvýše 1 mil. Kč do nákupu kompletů, přičemž snowboardové komplety chce nakoupit alespoň za 200 tis. Kč. Pořizovací cena každého z kompletů je 10 tis. Kč. Navrhněte, jak má společnost investovat, aby maximalizovala zisk a minimalizovala ztrátu.

VLP - příklad, matematický model

Označme x_1, \dots, x_4 proměnné vyjadřující počty nakoupených kompletů. Kriteriační funkce tedy jsou

$$z_1 = 300x_1 + 200x_2 + 170x_3 + 250x_4 \rightarrow \max$$

$$z_2 = 10x_1 + 15x_2 + 25x_3 + 5x_4 \rightarrow \min$$

Při pořizovací ceně kompletů 10 000 Kč budou mít omezení následující podobu: celkem lze koupit maximálně 100 ks kompletů, z toho nejméně 20 ks musí být snowboardové komplety:

$$x_1 + x_2 + x_3 + x_4 \leq 100$$

$$x_4 \geq 20$$

Dále musíme přidat obligátní podmínky :

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

Správně bychom měli přidat ještě podmínky celočíselnosti, ale nebudeme pro zjednodušení zatím uvažovat.

Řešením zjednodušených úloh modelu, kdy uvažujeme vždy pouze jednu kriteriační funkci dostaneme **dílčí optimální řešení**.

VLP - dílčí optimální řešení, příklad

Snadno lze zjistit, že **minimálního rizika** půjčovna dosáhne, jestliže nakoupí pouze požadované snowboardové komplety, tj. 20 ks po 10 000 Kč a ostatní komplety nebude kupovat vůbec.

VLP - dílčí optimální řešení, příklad

Snadno lze zjistit, že **minimálního rizika** půjčovna dosáhne, jestliže nakoupí pouze požadované snowboardové komplety, tj. 20 ks po 10 000 Kč a ostatní komplety nebude kupovat vůbec.

Podobně **maximálního zisku** půjčovna dosáhne, jestliže po zakoupení požadovaného množství snowboardových kompletů (tj. 20 ks) utratí všechny zbývající peníze za nejziskovější sjezdové komplety pro dospělé. Prostředky postačí pro zakoupení 80 ks těchto kompletů.

VLP - dílčí optimální řešení, příklad

Snadno lze zjistit, že **minimálního rizika** půjčovna dosáhne, jestliže nakoupí pouze požadované snowboardové komplety, tj. 20 ks po 10 000 Kč a ostatní komplety nebude kupovat vůbec.

Podobně **maximálního zisku** půjčovna dosáhne, jestliže po zakoupení požadovaného množství snowboardových kompletů (tj. 20 ks) utratí všechny zbývající peníze za nejziskovější sjezdové komplety pro dospělé. Prostředky postačí pro zakoupení 80 ks těchto kompletů.

Dílčí optimální řešení můžeme znázornit v **kriteriální tabulce**:

	Kriteriální funkce	
	výnos	riziko
80 dospělých sjezdových + 20 snowboardových	29000	880
20 snowboardových	5000	100

Z hlavní diagonály tabulky vidíme, že ideální řešení je ohodnoceno kriteriálními hodnotami zisku a rizika 29000 a 100, ale reálně neexistuje.

VLP - lexikografická metoda

Popišme si některé z metod s apriorní preferenční informací. Při použití **lexikografické** metody rozhodovatel určí pořadí významnosti kriteriálních funkcí (předpokládejme, že již jsou funkce označeny tak, že z_1 je nejvýznamnější a z_k nejméně významné, jejich optimální hodnoty označme $z_1^{opt}, \dots, z_k^{opt}$). Hledání kompromisního řešení spočívá v postupném řešení sekvence optimalizačních úloh

$max. z_1 = \mathbf{c}^1 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}$. Pokud má úloha více optimálních řešení, řešíme další úlohu:

$max. z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^{opt}$. Opět je-li řešení více, postupujeme dále, až nalezneme kompromisní řešení jako bod optima úlohy

$max. z_k = \mathbf{c}^k \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^{opt}, \dots, \mathbf{c}^{k-1} \mathbf{x} \geq z_{k-1}^{opt}$.

VLP - lexikografická metoda

Popišme si některé z metod s apriorní preferenční informací. Při použití **lexikografické** metody rozhodovatel určí pořadí významnosti kriteriálních funkcí (předpokládejme, že již jsou funkce označeny tak, že z_1 je nejdůležitější a z_k nejméně významné, jejich optimální hodnoty označme $z_1^{opt}, \dots, z_k^{opt}$). Hledání kompromisního řešení spočívá v postupném řešení sekvence optimalizačních úloh

$max. z_1 = \mathbf{c}^1 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}$. Pokud má úloha více optimálních řešení, řešíme další úlohu:

$max. z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^{opt}$. Opět je-li řešení více, postupujeme dále, až nalezneme kompromisní řešení jako bod optima úlohy

$max. z_k = \mathbf{c}^k \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^{opt}, \dots, \mathbf{c}^{k-1} \mathbf{x} \geq z_{k-1}^{opt}$.

Zmírnit předpoklad, že jednotlivé úlohy mají více než jedno řešení, je možné připuštěním odchylky δ_i , o kterou se můžou v jednotlivých krocích lišit hodnoty preferovaných účelových funkcí od svých optim. Například v druhém kroku bychom řešili úlohu

$max. z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^{opt} - \delta_1$, atd.

Popsaná metoda v podstatě kopíruje reálné uvažování manažerů.

VLP - lexikografická metoda, příklad

Již víme, že při preferenci rizika je nejvýhodnější nakoupit pouze 20 snowboardových kompletů a při preferenci zisku 20 snowboardových a 80 dospělých sjezdových kompletů.

Jaké by měla úloha řešení, jestliže budeme preferovat riziko, ale připustíme jeho odchylku z ideální hodnoty 100 na 120? Maximalizujeme tedy zisk

$$z_1 = 300x_1 + 200x_2 + 170x_3 + 250x_4 \text{ za omezení}$$

$$z_2 = 10x_1 + 15x_2 + 25x_3 + 5x_4 \leq 120, \text{ (tj. že riziko nepřesáhne hodnotu 120) a dalších omezení modelu}$$

$$x_1 + x_2 + x_3 + x_4 \leq 100 \quad x_4 \geq 20, \quad x_1, \dots, x_4 \geq 0.$$

VLP - lexikografická metoda, příklad

Již víme, že při preferenci rizika je nejvýhodnější nakoupit pouze 20 snowboardových kompletů a při preferenci zisku 20 snowboardových a 80 dospělých sjezdových kompletů.

Jaké by měla úloha řešení, jestliže budeme preferovat riziko, ale připustíme jeho odchylku z ideální hodnoty 100 na 120? Maximalizujeme tedy zisk

$$z_1 = 300x_1 + 200x_2 + 170x_3 + 250x_4 \text{ za omezení}$$

$$z_2 = 10x_1 + 15x_2 + 25x_3 + 5x_4 \leq 120, \text{ (tj. že riziko nepřesáhne hodnotu 120) a dalších omezení modelu}$$

$$x_1 + x_2 + x_3 + x_4 \leq 100 \quad x_4 \geq 20, \quad x_1, \dots, x_4 \geq 0.$$

Snadno vypočteme optimální řešení $x_1 = 2, x_2 = 0, x_3 = 0, x_4 = 20$, tedy k požadovaným dvaceti snowboardovým přikoupíme ještě 2 dospělé sjezdové komplety. Získáme tak **5600 Kč** a riziko bude rovno limitním **120 bodům**.

VLP - metoda minimální komponenty

Jiným přístupem je stanovení kompromisního řešení podle **minimální komponenty**, kdy maximalizujeme nejmenší, tedy nejhorší komponentu vektoru hodnot účelových funkcí. Dostaneme tedy úlohu LP maximalizovat

$z = \delta$ za podmínek $\mathbf{c}^1 \cdot \mathbf{x} \geq \delta, \mathbf{c}^2 \cdot \mathbf{x} \geq \delta, \dots, \mathbf{c}^k \cdot \mathbf{x} \geq \delta, \mathbf{x} \in \mathbf{X}$.

VLP - metoda minimální komponenty

Jiným přístupem je stanovení kompromisního řešení podle **minimální komponenty**, kdy maximalizujeme nejmenší, tedy nejhorší komponentu vektoru hodnot účelových funkcí. Dostaneme tedy úlohu LP maximalizovat

$z = \delta$ za podmínek $\mathbf{c}^1 \cdot \mathbf{x} \geq \delta, \mathbf{c}^2 \cdot \mathbf{x} \geq \delta, \dots, \mathbf{c}^k \cdot \mathbf{x} \geq \delta, \mathbf{x} \in \mathbf{X}$.

Pokud mají kritéria rozdílnou povahu, je třeba je nejprve převést všechna na maximalizační a upravit na bezrozměrné, aby byla zajištěna jejich srovnatelnost.

VLP - metoda agregace kriteriálních funkcí

Pomocí vhodně zvoleného operátoru je možné sloučit všechny kriteriální funkce do jediné. Obecně se při agregaci používají různé operátory, zde se jeví nejvhodnější lineární kombinace jednotlivých funkcí za použití normovaného vektoru vah $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_k)$. Místo původní optimalizační úlohy

$\mathbf{z} = \mathbf{C} \cdot \mathbf{x} \rightarrow \text{MAX}, \mathbf{x} \in \mathbf{X}.$ řešíme jednorozměrnou úlohu

$z_v = \mathbf{v} \cdot \mathbf{C} \cdot \mathbf{x} \rightarrow \text{max}, \mathbf{x} \in \mathbf{X}.$

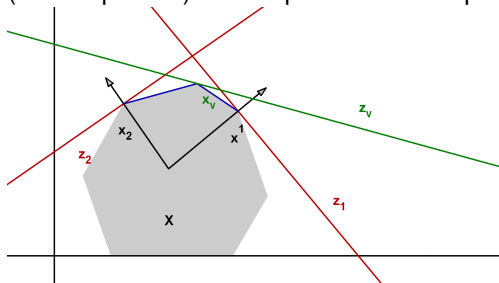
VLP - metoda agregace kritériálních funkcí

Pomocí vhodně zvoleného operátoru je možné sloučit všechny kritériální funkce do jediné. Obecně se při agregaci používají různé operátory, zde se jeví nejvhodnější lineární kombinace jednotlivých funkcí za použití normovaného vektoru vah $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_k)$. Místo původní optimalizační úlohy

$\mathbf{z} = \mathbf{C} \cdot \mathbf{x} \rightarrow \text{MAX}, \mathbf{x} \in \mathbf{X}$. řešíme jednorozměrnou úlohu

$z_v = \mathbf{v} \cdot \mathbf{C} \cdot \mathbf{x} \rightarrow \text{max}, \mathbf{x} \in \mathbf{X}$.

Pozor! Při nastavení vektoru vah je třeba vzít v úvahu, že hodnoty funkcí se pohybují na různých škálách. Vyznačme agregované kritérium graficky (zelená přímka). Nemá praktickou interpretaci, je pouze kritériem pomocným.



VLP - metoda agragace, příklad

Při agregaci funkce zisku z_1 a funkce rizika z_2 je třeba vzít v úvahu odlišnou povahu kritérií, například můžeme vynásobit z_2 koeficientem (-1).

Jestliže nastavíme vektor vah

$$\mathbf{v} = \left(\frac{5}{100}, \frac{95}{100} \right),$$

pak agregovaná účelová funkce bude mít tvar

$$z_v = 5,5x_1 - 4,25x_2 - 15,25x_3 + 7,75x_4.$$

VLP - metoda agragace, příklad

Při agregaci funkce zisku z_1 a funkce rizika z_2 je třeba vzít v úvahu odlišnou povahu kritérií, například můžeme vynásobit z_2 koeficientem (-1).

Jestliže nastavíme vektor vah

$$\mathbf{v} = \left(\frac{5}{100}, \frac{95}{100} \right),$$

pak agregovaná účelová funkce bude mít tvar

$$z_v = 5,5x_1 - 4,25x_2 - 15,25x_3 + 7,75x_4.$$

Snadno nahlédneme, že optimální řešení tedy bude

$x_1 = 0, x_2 = 0, x_3 = 0, x_4 = 100$, všechny peníze utratíme za snowboardové komplety. Optimální hodnota účelové funkce $z_v = 775$, což nemá žádnou vypovídací schopnost, ale můžeme pro nalezené řešení dopočítat hodnotu zisku 25000 Kč a hodnotu rizika 500 bodů.

VLP - převod kritériálních funkcí na omezení

Při **postupném** převodu kritérií na omezení postupujeme podobně jako u lexikografické metody. Předpokládáme opět, že jsou funkce označeny od nejvýznamnějšího po nejméně významné). Hledání kompromisního řešení spočívá v postupném řešení sekvence optimalizačních úloh

$\max. z_1 = \mathbf{c}^1 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}.$ Optimální hodnotu označíme z_1^* a řešíme další úlohu, kde připustíme určitou odchylku od z_1^* :

$\max. z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^* - \delta_1.$ Opět zjistíme z_2^* a postupujeme dále, až nalezneme kompromisní řešení jako bod optima úlohy

$\max. z_k = \mathbf{c}^k \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^* - \delta_1, \dots, \mathbf{c}^{k-1} \mathbf{x} \geq z_{k-1}^* - \delta_k.$

VLP - převod kritériálních funkcí na omezení

Při **postupném** převodu kritérií na omezení postupujeme podobně jako u lexikografické metody. Předpokládáme opět, že jsou funkce označeny od nejdůležitějšího po nejméně významné). Hledání kompromisního řešení spočívá v postupném řešení sekvence optimalizačních úloh

$\max. z_1 = \mathbf{c}^1 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}.$ Optimální hodnotu označíme z_1^* a řešíme další úlohu, kde připustíme určitou odchylku od z_1^* :

$\max. z_2 = \mathbf{c}^2 \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^* - \delta_1.$ Opět zjistíme z_2^* a postupujeme dále, až nalezneme kompromisní řešení jako bod optima úlohy

$\max. z_k = \mathbf{c}^k \cdot \mathbf{x}, \mathbf{x} \in \mathbf{X}, \mathbf{c}^1 \mathbf{x} \geq z_1^* - \delta_1, \dots, \mathbf{c}^{k-1} \mathbf{x} \geq z_{k-1}^* - \delta_k.$

Kritéria lze také převést na omezení **najednou** tak, že maximalizujeme pouze nejdůležitější kritérium z_1 a přidáme do modelu současně všechna omezení $z_i \geq au_i, i = 2, \dots, k$ (aspirační úrovně všech kritérií $au_i, i = 2, \dots, k$ nastavíme někde mezi bazální a ideální hodnotu pro dané kritérium $\langle z_i^{\min}, z_i^{\max} \rangle$.) Nevýhodou tohoto přístupu je, že může být obtížné nastavení aspiračních úrovní, aby nebyla přípustná množina prázdná nebo aby nebyl vliv kritérií eliminován úplně.

Lineární lomené programování

Skupinou úloh, které lze převést na lineární problém, jsou úlohy **lineárního lomeného programování** optimalizovat funkci $f(\mathbf{x}) = \frac{\mathbf{d}^\top \cdot \mathbf{x} + d_0}{\mathbf{c}^\top \cdot \mathbf{x} + c_0}$ za omezení $\mathbf{x} \geq \mathbf{0}$, $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. Tyto úlohy sice nespádají do kategorie konvexních úloh, avšak umožňují pro $\mathbf{c}^\top \cdot \mathbf{x} + c_0 > 0$ zavedení substituce $r = \frac{1}{\mathbf{c}^\top \cdot \mathbf{x} + c_0}$.

Dostaneme účelovou funkci ve tvaru $f = \mathbf{d}^\top \cdot \mathbf{x} \cdot r + d_0 r$, kterou můžeme vyjádřit jako funkci nových proměnných $r, y_i = x_i \cdot r, i = 1, \dots, n$:

$$f(r, y_1, \dots, y_n) = \mathbf{d}^\top \cdot \mathbf{y} + d_0 r, \text{ podobně vyjádříme i omezení}$$

$$r \geq 0, \mathbf{y} \geq \mathbf{0}, \mathbf{A} \cdot \mathbf{y} \leq \mathbf{b} \cdot r. \text{ Je však třeba přidat i další omezující podmínku}$$

$$1 = r(\mathbf{c}^\top \cdot \mathbf{x} + c_0), \text{ tedy } 1 = \mathbf{c}^\top \cdot \mathbf{y} + c_0 r.$$

Dostali jsme obyčejnou úlohu LP, kterou můžeme dále řešit například simplexovou metodou.

Lineární lomené programování - příklad

V ekonomii se setkáme s řadou ukazatelů poměrového typu. Jsou-li výrazy v čitateli a jmenovateli ukazatele lineárními funkcemi proměnných, podle kterých se snažíme ukazatel optimalizovat, jedná se o problém lineárního lomeného programování, viz úloha z knihy I. Gros: "Kvantitativní metody v manažerském rozhodování":

Lineární lomené programování - příklad

V ekonomii se setkáme s řadou ukazatelů poměrového typu. Jsou-li výrazy v čitateli a jmenovateli ukazatele lineárními funkcemi proměnných, podle kterých se snažíme ukazatel optimalizovat, jedná se o problém lineárního lomeného programování, viz úloha z knihy I. Gros: "Kvantitativní metody v manažerském rozhodování":

Příklad: Uvažujme firmu, která má ve výrobním programu tři výrobky A, B, C s následujícími charakteristikami:

Výrobek	Variabilní náklady [Kč/t]	Cena [Kč/t]
A	11 000	12 000
B	15 000	18 000
C	14 000	16 000
fixní náklady 150 000 Kč		

Lineární lomené programování - příklad

V ekonomii se setkáme s řadou ukazatelů poměrového typu. Jsou-li výrazy v čitateli a jmenovateli ukazatele lineárními funkcemi proměnných, podle kterých se snažíme ukazatel optimalizovat, jedná se o problém lineárního lomeného programování, viz úloha z knihy I. Gros: "Kvantitativní metody v manažerském rozhodování":

Příklad: Uvažujme firmu, která má ve výrobním programu tři výrobky A, B, C s následujícími charakteristikami:

Výrobek	Variabilní náklady [Kč/t]	Cena [Kč/t]
A	11 000	12 000
B	15 000	18 000
C	14 000	16 000
fixní náklady 150 000 Kč		

Označíme-li si x_1 , x_2 , x_3 množství prodaných výrobků, můžeme definovat například následující poměrové ukazatele:

$$\text{nákladovost tržeb } z = \frac{11000x_1 + 15000x_2 + 14000x_3 + 150000}{12000x_1 + 18000x_2 + 16000x_3},$$

$$\text{rentabilitu tržeb } z = \frac{1000x_1 + 3000x_2 + 2000x_3 - 150000}{12000x_1 + 18000x_2 + 16000x_3}.$$

Lineární lomené programování - příklad

Příklad: Minimalizujte nákladovost tržeb z uvedeného příkladu za podmínky, že celkové náklady nepřesáhnou 200 000 Kč.

Lineární lomené programování - příklad

Příklad: Minimalizujte nákladovost tržeb z uvedeného příkladu za podmínky, že celkové náklady nepřesáhnou 200 000 Kč.

Matematický zápis úlohy: minimalizujte $z = \frac{11x_1 + 15x_2 + 14x_3 + 150}{12x_1 + 18x_2 + 16x_3}$
za podmínek $11x_1 + 15x_2 + 14x_3 + 150 \leq 200$, $x_1, x_2, x_3 \geq 0$.

Lineární lomené programování - příklad

Příklad: Minimalizujte nákladovost tržeb z uvedeného příkladu za podmínky, že celkové náklady nepřesáhnou 200 000 Kč.

Matematický zápis úlohy: minimalizujte $z = \frac{11x_1 + 15x_2 + 14x_3 + 150}{12x_1 + 18x_2 + 16x_3}$
za podmínek $11x_1 + 15x_2 + 14x_3 + 150 \leq 200$, $x_1, x_2, x_3 \geq 0$.

Úlohu linearizujeme zavedením substituce $r = \frac{1}{12x_1 + 18x_2 + 16x_3}$, dostaneme tak problém minimalizovat funkci

$$f(y_1, y_2, y_3, r) = 11y_1 + 15y_2 + 14y_3 + 150r$$

za omezení $11y_1 + 15y_2 + 14y_3 - 50r \leq 0$, $y_1, y_2, y_3, r \geq 0$ a doplňující podmínky $12y_1 + 18y_2 + 16y_3 = 1$

Lineární lomené programování - příklad

Příklad: Minimalizujte nákladovost tržeb z uvedeného příkladu za podmínky, že celkové náklady nepřesáhnou 200 000 Kč.

Matematický zápis úlohy: minimalizujte $z = \frac{11x_1 + 15x_2 + 14x_3 + 150}{12x_1 + 18x_2 + 16x_3}$
za podmínek $11x_1 + 15x_2 + 14x_3 + 150 \leq 200$, $x_1, x_2, x_3 \geq 0$.

Úlohu linearizujeme zavedením substituce $r = \frac{1}{12x_1 + 18x_2 + 16x_3}$, dostaneme tak problém minimalizovat funkci

$$f(y_1, y_2, y_3, r) = 11y_1 + 15y_2 + 14y_3 + 150r$$

za omezení $11y_1 + 15y_2 + 14y_3 - 50r \leq 0$, $y_1, y_2, y_3, r \geq 0$ a doplňující podmínky $12y_1 + 18y_2 + 16y_3 = 1$

Modely datových obalů (DEA) slouží k hodnocení technické efektivity produkčních jednotek na základě velikosti vstupů a výstupů (bez nutnosti cenového vyjádření).

Modely datových obalů (DEA) slouží k hodnocení technické efektivity produkčních jednotek na základě velikosti vstupů a výstupů (bez nutnosti cenového vyjádření).

Historie:

1957: Farrell: model měření efektivity jednotek s jedním vstupem a výstupem

1978: Charnes, Cooper, Rhodes: **CCR model**: vícenásobné vstupy a výstupy, konstantní výnosy z rozsahu

1984: Banker, Charnes, Cooper: **BCC model**: proměnný výnos z rozsahu

Modely datových obalů (DEA) slouží k hodnocení technické efektivity produkčních jednotek na základě velikosti vstupů a výstupů (bez nutnosti cenového vyjádření).

Historie:

1957: Farrell: model měření efektivity jednotek s jedním vstupem a výstupem

1978: Charnes, Cooper, Rhodes: **CCR model**: vícenásobné vstupy a výstupy, konstantní výnosy z rozsahu

1984: Banker, Charnes, Cooper: **BCC model**: proměnný výnos z rozsahu

Uvažujme homogenní produkční jednotky, spotřebovávající stejný typ zdrojů (materiál, podlahová plocha, pracovníci, atd.), budeme je označovat **vstupy**, k produkci ekvivalentních efektů (tržby, zisk, počet obslužených klientů, atp.), dále jen **výstupy**. Pokud v činnosti jednotek dominuje pouze jeden vstup a jeden výstup, lze snadno vyjádřit efektivitu pomocí poměrového ukazatele

efektivita = výstup/vstup

Modely datových obalů (DEA) slouží k hodnocení technické efektivity produkčních jednotek na základě velikosti vstupů a výstupů (bez nutnosti cenového vyjádření).

Historie:

1957: Farrell: model měření efektivity jednotek s jedním vstupem a výstupem

1978: Charnes, Cooper, Rhodes: **CCR model**: vícenásobné vstupy a výstupy, konstantní výnosy z rozsahu

1984: Banker, Charnes, Cooper: **BCC model**: proměnný výnos z rozsahu

Uvažujme homogenní produkční jednotky, spotřebovávající stejný typ zdrojů (materiál, podlahová plocha, pracovníci, atd.), budeme je označovat **vstupy**, k produkci ekvivalentních efektů (tržby, zisk, počet obslužených klientů, atp.), dále jen **výstupy**. Pokud v činnosti jednotek dominuje pouze jeden vstup a jeden výstup, lze snadno vyjádřit efektivitu pomocí poměrového ukazatele

$$\text{efektivita} = \text{výstup/vstup}$$

Pro vícenásobné vstupy a výstupy je možné jejich agregováním ukazatel modifikovat:

$$\text{efektivita} = \text{vážené výstupy/vážené vstupy}$$

Model s jedním vstupem a výstupem

Uvažujme 8 poboček obchodní firmy, které charakterizujeme jedním vstupem (počet zaměstnanců) a jedním výstupem (počet uzavřených smluv s klienty). Efektivitu lze vyjádřit pomocí ukazatele "počet smluv na zaměstnance", údaje jsou zaznamenány v tabulce:

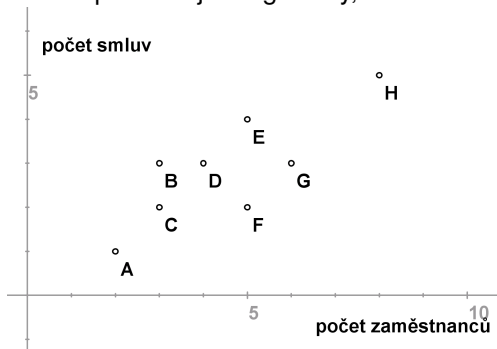
Pobočka	A	B	C	D	E	F	G	H
zaměstnanců	2	3	3	4	5	5	6	8
smluv	1	3	2	3	4	2	3	5
efektivita	0,5	1	0,667	0,75	0,8	0,4	0,5	0,625

Model s jedním vstupem a výstupem

Data reprezentujeme graficky, viz:

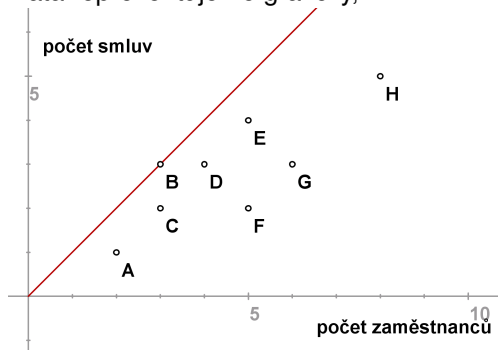
Model s jedním vstupem a výstupem

Data reprezentujeme graficky, viz:



Model s jedním vstupem a výstupem

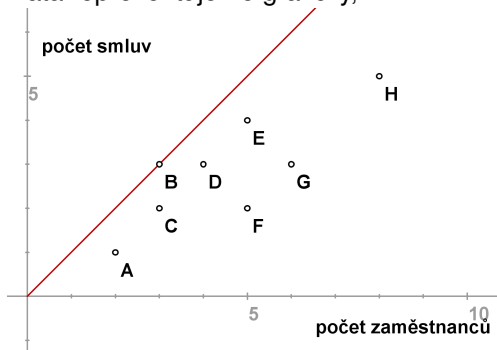
Data reprezentujeme graficky, viz:



Ukazatel efektivity dané pobočky udává sklon přímky spojující příslušný bod s počátkem, pro pobočku B nabývá nejvyšší hodnoty - tuto přímku budeme dále nazývat **efektivní hranicí**.

Model s jedním vstupem a výstupem

Data reprezentujeme graficky, viz:

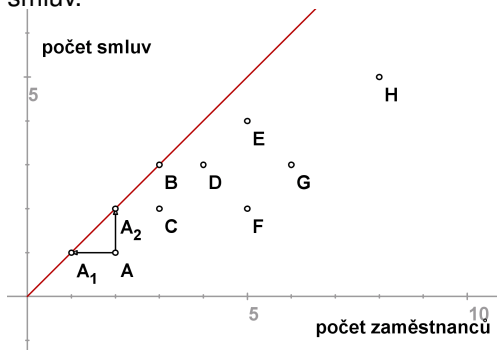


Ukazatel efektivity dané pobočky udává sklon přímky spojující příslušný bod s počátkem, pro pobočku B nabývá nejvyšší hodnoty - tuto přímku budeme dále nazývat **efektivní hranicí**. Nejlepší pobočkou je tedy pobočka B, efektivitu ostatních můžeme pak vyjádřit také v relativní míře. Například

efektivita A / efektivita B = 0,5 což znamená, že A dosahuje pouze 50% efektivity B. Tato relativní míra nabývá i pro ostatní pobočky hodnot z intervalu $[0, 1]$ a je nezávislá na jednotkách, ve kterých jsme vstup, resp. výstup měřili.

Model s jedním vstupem a výstupem

Jakým způsobem může jednotka A dosáhnout stoprocentní hodnoty, tj. efektivní hranice? Může snížit vstupy při zachování výstupů (**model orientovaný na vstupy**, v grafickém znázornění reprezentuje bod A_1) nebo zvýšit výstup při zachování vstupů (**model orientovaný na výstupy**, v grafickém znázornění reprezentuje bod A_2) nebo změnit obojí. Jednotky A_1 , A_2 nazýváme **virtuální**, neodpovídají žádné reálné pobočce. Úsečka A_1A_2 reprezentuje všechny dostupné body na efektivní hranici, pro jejichž dosažení A nemusí zvyšovat počet zaměstnanců nebo snižovat počet uzavřených smluv.



Model s jedním vstupem a výstupem

Označíme-li souřadnice bodů $A[x, y]$, $A_1[x_1, y_1]$, $A_2[x_2, y_2]$, vzhledem k tomu, že A_1, A_2 leží na efektivní hranici, lze ve vyjádření relativní efektivity A použít ve jmenovateli místo jednotky B libovolnou z těchto virtuálních jednotek.

Dostaneme pak

$$\frac{x/y}{x_1/y_1} = x/x_1 \quad \text{neboť } y = y_1$$

nebo

$$\frac{x/y}{x_2/y_2} = y_2/y \quad \text{neboť } x = x_2.$$

Model s jedním vstupem a výstupem

Označíme-li souřadnice bodů $A[x, y]$, $A_1[x_1, y_1]$, $A_2[x_2, y_2]$, vzhledem k tomu, že A_1, A_2 leží na efektivní hranici, lze ve vyjádření relativní efektivity A použít ve jmenovateli místo jednotky B libovolnou z těchto virtuálních jednotek.

Dostaneme pak

$$\frac{x/y}{x_1/y_1} = x/x_1 \quad \text{neboť } y = y_1$$

nebo

$$\frac{x/y}{x_2/y_2} = y_2/y \quad \text{neboť } x = x_2.$$

V modelu orientovaném na výstupy můžeme relativní efektivitu interpretovat jako **potřebné navýšení výstupu**, $\frac{y_2}{y} = \frac{2}{1} = 2$, tedy efektivní hranice by A dosáhla zdvojnásobením počtu uzavřených smluv.

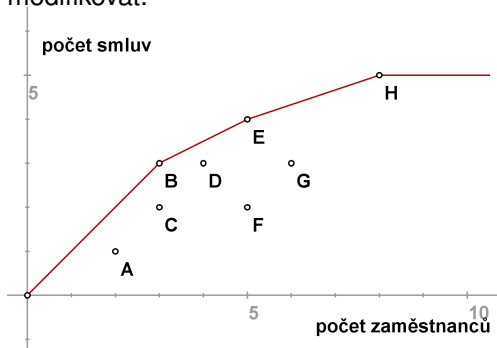
Pro model orientovaný na vstupy převrácená hodnota relativní efektivity reprezentuje **potřebnou redukci vstupů**, $\frac{x_1}{x} = \frac{1}{2} = 0,5$, tedy efektivní hranice by A dosáhla s polovinou zaměstnanců.

Model s jedním vstupem a výstupem

V předchozích úvahách jsme pracovali s předpokladem **konstantních výnosů z rozsahu**, kdy efektivní hranice byla tvořena polopřímkou, tj. pro každou pobočku s kombinací vstupu a výstupu $[x, y]$ se předpokládá za dosažitelnou i kombinace $[\alpha x, \alpha y]$ pro lib. $\alpha > 0$. Ukazatel relativní efektivity pak vychází shodně, ať použijeme model orientovaný na vstupy či na výstupy.

Model s jedním vstupem a výstupem

V předchozích úvahách jsme pracovali s předpokladem **konstantních výnosů z rozsahu**, kdy efektivní hranice byla tvořena polopřímkou, tj. pro každou pobočku s kombinací vstupu a výstupu $[x, y]$ se předpokládá za dosažitelnou i kombinace $[\alpha x, \alpha y]$ pro lib. $\alpha > 0$. Ukazatel relativní efektivity pak vychází shodně, ať použijeme model orientovaný na vstupy či na výstupy. Za předpokladu **variabilních výnosů z rozsahu** je třeba efektivní hranici modifikovat.



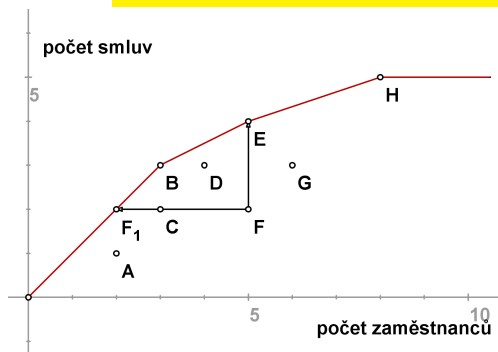
Nyní tvoří hranice obal dat, tak že efektivní se jeví též jednotky E a H.

Model s jedním vstupem a výstupem

Ukazatel relativní efektivity se může měnit podle použitého modelu. Například pro jednotku F při orientaci na vstupy dostaneme hodnotu

efektivita F_1 / efektivita $F = 2/5 = 0,4$, kdežto při orientaci na výstupy

hodnotu efektivita F /efektivita $E = 1/2 = 0,5$.



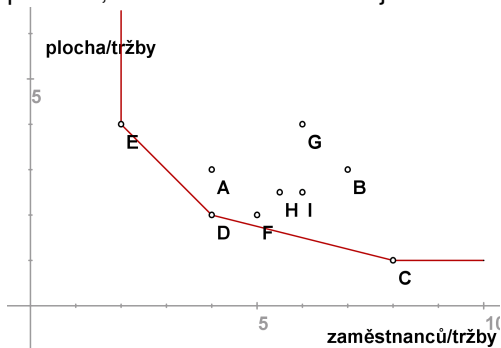
Dva vstupy a jeden výstup

Uvažujme příklad 9 supermarketů , kde jako vstupy bereme počet zaměstnanců (v desítkách) a podlahovou plochu (v 1000 m^2), výstupem rozumíme roční tržby. Za předpokladu konstantních výnosů z rozsahu můžeme dále pracovat s hodnotami vstupů přepočtenými na jednotku výstupu. Normované hodnoty jsou uvedeny v tabulce.

Obchod	A	B	C	D	E	F	G	H	I
zaměstnanců	4	7	8	4	2	5	6	5,5	6
plocha	3	3	1	2	4	2	4	2,5	2,5
tržby	1	1	1	1	1	1	1	1	1

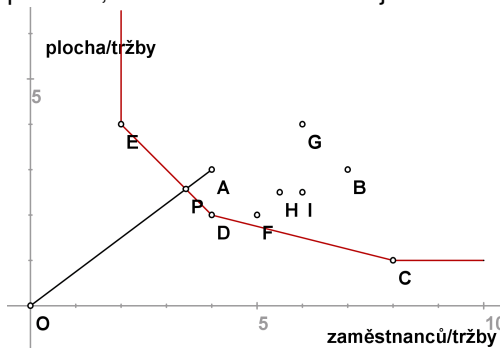
Dva vstupy a jeden výstup

V grafickém znázornění se efektivnější jeví ty obchody, které se nalézají blíž k počátku, efektivní hranice obaluje data následujícím způsobem:



Dva vstupy a jeden výstup

V grafickém znázornění se efektivnější jeví ty obchody, které se nalézají blíže k počátku, efektivní hranice obaluje data následujícím způsobem:

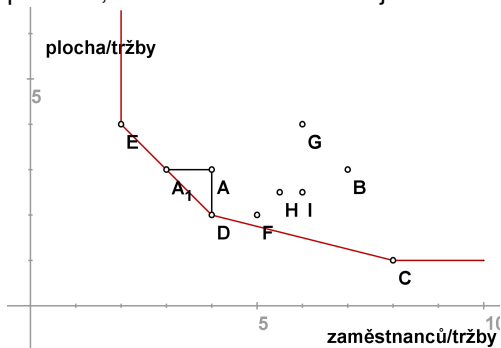


Obchod A je neefektivní, míru jeho efektivity můžeme měřit radiálně jako

$\frac{|OP|}{|OA|} = 0,8571$. Protože virtuální jednotka P je kombinací jednotek D,E, nazýváme je **referenčními jednotkami pro A**.

Dva vstupy a jeden výstup

V grafickém znázornění se efektivnější jeví ty obchody, které se nalézají blíže k počátku, efektivní hranice obaluje data následujícím způsobem:



Obchod A je neefektivní, míru jeho efektivity můžeme měřit radiálně jako

$$\frac{|OP|}{|OA|} = 0,8571$$

. Protože virtuální jednotka P je kombinací jednotek D, E, nazýváme je **referenčními jednotkami pro A**. Efektivní hranice lze dosáhnout též jinak než proporčním snížením obou vstupů o 15%, snížení pouze jednoho vstupu při zachování úrovně druhého demonstrují body A_1 , D.

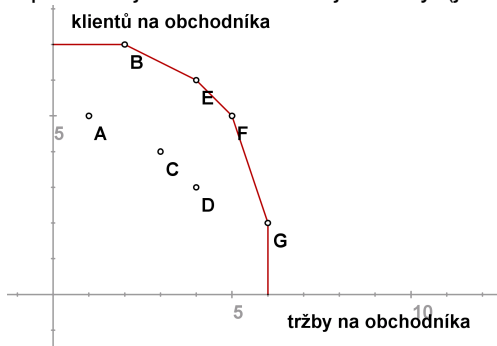
Jeden vstup a dva výstupy

Nyní naopak uvažujme případ, kdy u 7 obchodních kanceláří sledujeme 1 vstup (počet obchodníků) a dva výstupy (počet obslužených zákazníků a tržby). Hodnoty výstupů u jednotlivých poboček přepočtené na 1 obchodníka jsou uvedeny v tabulce.

Kancelář	A	B	C	D	E	F	G
obchodníků	1	1	1	1	1	1	1
zákazníků	1	2	3	4	4	5	6
tržby	5	7	4	3	6	5	2

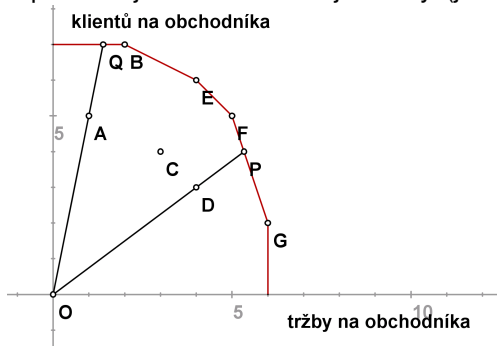
Jeden vstup a dva výstupy

Můžeme znázornit jednotkové výstupy jednotlivých kanceláří, efektivní hranice bude obalovat data z opačné strany, protože body ležící blíž k počátku reprezentují méně efektivní jednotky. (jde o jednotky A,C,D)



Jeden vstup a dva výstupy

Můžeme znázornit jednotkové výstupy jednotlivých kanceláří, efektivní hranice bude obalovat data z opačné strany, protože body ležící blíž k počátku reprezentují méně efektivní jednotky. (jde o jednotky A,C,D)



Míru neefektivity lze opět měřit radiálně, například pro jednotku D jako

$\frac{|OD|}{|OP|} = 0,75$. Pro bod A by analogická míra vyjadřovala pouze tzv. **technickou neefektivitu**, z bodu Q lze ještě zvýšit tržby bez ztráty klientů až na úroveň bodu B.

CCR model

Pro n jednotek U_1, \dots, U_n u nichž sledujeme m vstupů a r výstupů zavedeme označení x_{iq} pro i -tý vstup q -té jednotky a y_{jq} pro j -tý výstup q -té jednotky. Pro U_q značíme $\mathbf{x}_q = (x_{1q}, \dots, x_{mq})'$, $\mathbf{y}_q = (y_{1q}, \dots, y_{rq})'$. Hodnoty lze uspořádat do matic

$$\mathbf{X} = [x_{iq}]_{q=1, \dots, n}^{i=1, \dots, m}, \quad \mathbf{Y} = [y_{jq}]_{q=1, \dots, n}^{j=1, \dots, r}.$$

CCR model

Pro n jednotek U_1, \dots, U_n u nichž sledujeme m vstupů a r výstupů zavedeme označení x_{iq} pro i -tý vstup q -té jednotky a y_{jq} pro j -tý výstup q -té jednotky. Pro U_q značíme $\mathbf{x}_q = (x_{1q}, \dots, x_{mq})'$, $\mathbf{y}_q = (y_{1q}, \dots, y_{rq})'$. Hodnoty lze uspořádat do matic

$$\mathbf{X} = [x_{iq}]_{q=1, \dots, n}^{i=1, \dots, m}, \quad \mathbf{Y} = [y_{jq}]_{q=1, \dots, n}^{j=1, \dots, r}.$$

Pomocí vektorů nezáporných vah $\mathbf{v} = (v_1, \dots, v_m)$, $\mathbf{u} = (u_1, \dots, u_r)$ můžeme pro libovolnou jednotku U_q definovat

virtuální vstup = $v_1 x_{1q} + \dots + v_m x_{mq} = \mathbf{v} \mathbf{x}_q$ a

virtuální výstup = $u_1 y_{1q} + \dots + u_r y_{rq} = \mathbf{u} \mathbf{y}_q$.

Míru efektivnosti dané jednotky pak vyjádříme jako podíl jejího virtuálního výstupu a vstupu.

CCR model

Pro n jednotek U_1, \dots, U_n u nichž sledujeme m vstupů a r výstupů zavedeme označení x_{iq} pro i -tý vstup q -té jednotky a y_{jq} pro j -tý výstup q -té jednotky. Pro U_q značíme $\mathbf{x}_q = (x_{1q}, \dots, x_{mq})'$, $\mathbf{y}_q = (y_{1q}, \dots, y_{rq})'$. Hodnoty lze uspořádat do matic

$$\mathbf{X} = [x_{iq}]_{q=1, \dots, n}^{i=1, \dots, m}, \quad \mathbf{Y} = [y_{jq}]_{q=1, \dots, n}^{j=1, \dots, r}$$

Pomocí vektorů nezáporných vah $\mathbf{v} = (v_1, \dots, v_m)$, $\mathbf{u} = (u_1, \dots, u_r)$ můžeme pro libovolnou jednotku U_q definovat

virtuální vstup = $v_1 x_{1q} + \dots + v_m x_{mq} = \mathbf{v} \mathbf{x}_q$ a

virtuální výstup = $u_1 y_{1q} + \dots + u_r y_{rq} = \mathbf{u} \mathbf{y}_q$.

Míru efektivity dané jednotky pak vyjádříme jako podíl jejího virtuálního výstupu a vstupu.

CCR model optimalizuje váhy vstupů a výstupů tak, aby míra efektivity dané jednotky U_q ,
$$z = \frac{u_1 y_{1q} + \dots + u_r y_{rq}}{v_1 x_{1q} + \dots + v_m x_{mq}} = \frac{\mathbf{u} \mathbf{y}_q}{\mathbf{v} \mathbf{x}_q}$$
 byla maximální za podmínky, že efektivity ostatních jednotek jsou nejvýše jednotkové. Existuje-li kladné řešení s optimální hodnotou účelové funkce $z^* = 1$, pak se jednotka U_q označuje jako **CCR efektivní**.

CCR model orientovaný na vstupy

Celý model pro jednotku U_q lze formulovat jako úlohu lineárního lomeného programování

$$z = \frac{u_1 y_{1q} + \dots + u_r y_{rq}}{v_1 x_{1q} + \dots + v_m x_{mq}} \rightarrow \max_{u,v}$$

za omezení

$$\frac{u_1 y_{1k} + \dots + u_r y_{rk}}{v_1 x_{1k} + \dots + v_m x_{mk}} \leq 1, \quad k = 1, \dots, n,$$

$$u_j \geq 0, \quad v_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, r.$$

CCR model orientovaný na vstupy

Celý model pro jednotku U_q lze formulovat jako úlohu lineárního lomeného programování

$$z = \frac{u_1 y_{1q} + \dots + u_r y_{rq}}{v_1 x_{1q} + \dots + v_m x_{mq}} \rightarrow \max_{u,v}$$

za omezení

$$\frac{u_1 y_{1k} + \dots + u_r y_{rk}}{v_1 x_{1k} + \dots + v_m x_{mk}} \leq 1, \quad k = 1, \dots, n,$$
$$u_i \geq 0, \quad v_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, r.$$

Úlohu lze snadno linearizovat pomocí Charnes-Cooperovy transformace:

$$z = u_1 y_{1q} + \dots + u_r y_{rq} \rightarrow \max_{u,v}$$

za omezení

$$v_1 x_{1q} + \dots + v_m x_{mq} = 1$$
$$u_1 y_{1k} + \dots + u_r y_{rk} \leq v_1 x_{1k} + \dots + v_m x_{mk}, \quad k = 1, \dots, n,$$
$$u_i \geq 0, \quad v_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, r.$$

CCR model orientovaný na vstupy

Celý model pro jednotku U_q lze formulovat jako úlohu lineárního lomeného programování

$$z = \frac{u_1 y_{1q} + \dots + u_r y_{rq}}{v_1 x_{1q} + \dots + v_m x_{mq}} \rightarrow \max_{u,v}$$

za omezení

$$\frac{u_1 y_{1k} + \dots + u_r y_{rk}}{v_1 x_{1k} + \dots + v_m x_{mk}} \leq 1, \quad k = 1, \dots, n,$$
$$u_i \geq 0, \quad v_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, r.$$

Úlohu lze snadno linearizovat pomocí Charnes-Cooperovy transformace:

$$z = u_1 y_{1q} + \dots + u_r y_{rq} \rightarrow \max_{u,v}$$

za omezení

$$v_1 x_{1q} + \dots + v_m x_{mq} = 1$$
$$u_1 y_{1k} + \dots + u_r y_{rk} \leq v_1 x_{1k} + \dots + v_m x_{mk}, \quad k = 1, \dots, n,$$
$$u_i \geq 0, \quad v_j \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, r.$$

Tento model nazýváme **CCR modelem orientovaným na vstupy**. Množina takových indexů $k \in \{1, \dots, n\}$ efektivních jednotek, pro které jsou omezující podmínky v úloze pro U_q aktivní, definuje tzv. **referenční množinu** pro U_q .

CCR model - příklad

Uvažujme úlohu s dvěma vstupy a jedním výstupem, hodnoty pro 6 jednotek jsou uvedeny v tabulce:

Jednotka	A	B	C	D	E	F
x_1	4	7	8	4	2	10
x_2	3	3	1	2	4	1
y	1	1	1	1	1	1

CCR model - příklad

Uvažujme úlohu s dvěma vstupy a jedním výstupem, hodnoty pro 6 jednotek jsou uvedeny v tabulce:

Jednotka	A	B	C	D	E	F
x_1	4	7	8	4	2	10
x_2	3	3	1	2	4	1
y	1	1	1	1	1	1

Linearizovaná úloha pro jednotku A bude mít podobu:

$$z = u \rightarrow \max$$

za omezení

$$4v_1 + 3v_2 = 1, \quad u, v_1, v_2 \geq 0$$

$$u \leq 4v_1 + 3v_2 \text{ (A)} \quad u \leq 7v_1 + 3v_2 \text{ (B)}$$

$$u \leq 8v_1 + v_2 \text{ (C)} \quad u \leq 4v_1 + 2v_2 \text{ (D)}$$

$$u \leq 2v_1 + 4v_2 \text{ (E)} \quad u \leq 10v_1 + v_2 \text{ (F)}$$

CCR model - příklad

Uvažujme úlohu s dvěma vstupy a jedním výstupem, hodnoty pro 6 jednotek jsou uvedeny v tabulce:

Jednotka	A	B	C	D	E	F
x_1	4	7	8	4	2	10
x_2	3	3	1	2	4	1
y	1	1	1	1	1	1

Linearizovaná úloha pro jednotku A bude mít podobu:

$$z = u \rightarrow \max$$

za omezení

$$4v_1 + 3v_2 = 1, \quad u, v_1, v_2 \geq 0$$

$$u \leq 4v_1 + 3v_2 \text{ (A)} \quad u \leq 7v_1 + 3v_2 \text{ (B)}$$

$$u \leq 8v_1 + v_2 \text{ (C)} \quad u \leq 4v_1 + 2v_2 \text{ (D)}$$

$$u \leq 2v_1 + 4v_2 \text{ (E)} \quad u \leq 10v_1 + v_2 \text{ (F)}$$

Úlohu lze vyřešit standartními postupy lineárního programování, pro A dostaneme řešení $z^* = u^* = 6/7, v_1^* = v_2^* = 1/7$. Jednotka A není efektivní, protože $z^* = 6/7 \leq 1$, omezující nerovnice pro jednotky D, E jsou aktivní, tedy **D, E jsou kandidáty na referenční jednotky pro A.**

CCR model - příklad

Úlohy pro ostatní jednotky budou podobné, lišit se budou pouze v omezující rovnosti, pro jednotku B to bude podmínka $7v_1 + 3v_2 = 1$, atd. V tabulce jsou shrnuty výsledky úloh pro všechny jednotky:

jednotka	x_1	x_2	y	v_1^*	v_2^*	$z^* = u^*$	referenční množina
A	4	3	1	1/7	1/7	6/7	D,E
B	7	3	1	1/19	4/19	12/19	C,D
C	8	1	1	1/12	1/3	1	C
D	4	2	1	1/6	1/6	1	D
E	2	4	1	3/14	1/7	1	E
F	10	1	1	0	1	1	C

CCR model - příklad

Úlohy pro ostatní jednotky budou podobné, lišit se budou pouze v omezující rovnosti, pro jednotku B to bude podmínka $7v_1 + 3v_2 = 1$, atd. V tabulce jsou shrnuty výsledky úloh pro všechny jednotky:

jednotka	x_1	x_2	y	v_1^*	v_2^*	$z^* = u^*$	referenční množina
A	4	3	1	1/7	1/7	6/7	D,E
B	7	3	1	1/19	4/19	12/19	C,D
C	8	1	1	1/12	1/3	1	C
D	4	2	1	1/6	1/6	1	D
E	2	4	1	3/14	1/7	1	E
F	10	1	1	0	1	1	C

Vidíme, že jednotky C,D,E jsou efektivní. Optimální hodnota účelové funkce pro F je sice také rovna 1, ale referenční jednotkou pro F je C, protože má při stejném vstupu v_2 nižší vstup v_1 . V definici CCR efektivity je tedy třeba zdůraznit: **jednotka je CCR efektivní, je-li její $z^* = 1$ a existuje aspoň jedno řešení, pro něž $u^* > 0$, $v^* > 0$** . Požadavek existence kladných vah lze zapracovat přímo do modelu, kdy v podmínkách nezápornosti proměnných změním pravou stranu na nějaké $\varepsilon > 0$.

CCR model - duální úloha

Zapišme CCR model pro jednotku U_q maticově:

$$z = \mathbf{u}\mathbf{y}_q \rightarrow \max_{\mathbf{u}, \mathbf{v}}$$

za omezení

$$\mathbf{v}\mathbf{x}_q = 1$$

$$\mathbf{u}\mathbf{Y} \leq \mathbf{v}\mathbf{X},$$

$$\mathbf{u}, \mathbf{v} \geq 0.$$

CCR model - duální úloha

Zapišme CCR model pro jednotku U_q maticově:

$$z = \mathbf{u}\mathbf{y}_q \rightarrow \max_{\mathbf{u}, \mathbf{v}}$$

za omezení

$$\mathbf{v}\mathbf{X}_q = 1$$

$$\mathbf{u}\mathbf{Y} \leq \mathbf{v}\mathbf{X},$$

$$\mathbf{u}, \mathbf{v} \geq 0.$$

K úloze formulujme její duální problém, duální proměnné označíme θ a

$$\lambda = (\lambda_1, \dots, \lambda_n)'$$

$$z = \theta \rightarrow \min_{\theta, \lambda}$$

za omezení

$$\theta\mathbf{X}_q \geq \mathbf{X}\lambda,$$

$$\mathbf{Y}\lambda \geq \mathbf{y}_q,$$

$$\lambda \geq 0$$

Použití duálního modelu je výhodné z výpočetního i interpretačního hlediska.

Duální CCR model - interpretace

Model vlastně hledá virtuální jednotku se vstupy a výstupy $\mathbf{X}\lambda$, $\mathbf{Y}\lambda$, která je lepší nebo alespoň srovnatelná s radiální projekcí hodnocené jednotky U_q na efektivní hranici:

$$\theta \mathbf{x}_q \geq \mathbf{X}\lambda, \mathbf{Y}\lambda \geq \mathbf{y}_q$$

Duální CCR model - interpretace

Model vlastně hledá virtuální jednotku se vstupy a výstupy $\mathbf{X}\lambda, \mathbf{Y}\lambda$, která je lepší nebo alespoň srovnatelná s radiální projekcí hodnocené jednotky U_q na efektivní hranici:

$$\theta \mathbf{x}_q \geq \mathbf{X}\lambda, \mathbf{Y}\lambda \geq \mathbf{y}_q$$

Hodnocená jednotka U_q leží přímo na efektivní hranici, je-li totožná s nalezenou virtuální jednotkou. Nutně tedy musí být optimální hodnota účelové funkce modelu θ^* (tzv. **Farellova efektivita**) vyjadřující potřebnou radiální redukci vstupů k dosažení efektivní hranice jednotková, $\theta^* = 1$. Při splnění této podmínky je jednotka U_q též nazvána **technicky efektivní**.

Duální CCR model - interpretace

Model vlastně hledá virtuální jednotku se vstupy a výstupy $\mathbf{X}\lambda, \mathbf{Y}\lambda$, která je lepší nebo alespoň srovnatelná s radiální projekcí hodnocené jednotky U_q na efektivní hranici:

$$\theta \mathbf{x}_q \geq \mathbf{X}\lambda, \mathbf{Y}\lambda \geq \mathbf{y}_q$$

Hodnocená jednotka U_q leží přímo na efektivní hranici, je-li totožná s nalezenou virtuální jednotkou. Nutně tedy musí být optimální hodnota účelové funkce modelu θ^* (tzv. **Farellova efektivita**) vyjadřující potřebnou radiální redukci vstupů k dosažení efektivní hranice jednotková, $\theta^* = 1$. Při splnění této podmínky je jednotka U_q též nazvána **technicky efektivní**.

K dosažení CCR-efektivity však současně musí být nulové všechny přídavné proměnné převádějící omezující nerovnosti na rovnosti:

$$\mathbf{s}^- = \theta \mathbf{x}_q - \mathbf{X}\lambda = \mathbf{0}, \mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_q = \mathbf{0}.$$

Duální CCR model - interpretace

Model vlastně hledá virtuální jednotku se vstupy a výstupy $\mathbf{X}\lambda, \mathbf{Y}\lambda$, která je lepší nebo alespoň srovnatelná s radiální projekcí hodnocené jednotky U_q na efektivní hranici:

$$\theta \mathbf{x}_q \geq \mathbf{X}\lambda, \mathbf{Y}\lambda \geq \mathbf{y}_q$$

Hodnocená jednotka U_q leží přímo na efektivní hranici, je-li totožná s nalezenou virtuální jednotkou. Nutně tedy musí být optimální hodnota účelové funkce modelu θ^* (tzv. **Farellova efektivita**) vyjadřující potřebnou radiální redukci vstupů k dosažení efektivní hranice jednotková, $\theta^* = 1$. Při splnění této podmínky je jednotka U_q též nazvána **technicky efektivní**.

K dosažení CCR-efektivity však současně musí být nulové všechny přídavné proměnné převádějící omezující nerovnosti na rovnosti:

$$\mathbf{s}^- = \theta \mathbf{x}_q - \mathbf{X}\lambda = \mathbf{0}, \mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_q = \mathbf{0}.$$

Při splnění všech těchto podmínek vyhovuje hodnocená jednotka tzv. **Pareto-Koopmansově definici** efektivity, tj. není možné zlepšit žádný z jejích vstupů a výstupů, aniž by současně nedošlo ke zhoršení jiného.

Duální CCR model - alternativní formulace

Uvažujme primární CCR model s požadavkem nenulovosti vah $\mathbf{u}, \mathbf{v} \geq \varepsilon > 0$ a označme $\mathbf{e} = (1, \dots, 1)^\top$. Duální úlohu lze též formulovat ve tvaru

$$z = \theta - \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \theta \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_q,$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

Duální CCR model - alternativní formulace

Uvažujme primární CCR model s požadavkem nenulovosti vah $\mathbf{u}, \mathbf{v} \geq \varepsilon > 0$ a označme $\mathbf{e} = (1, \dots, 1)^\top$. Duální úlohu lze též formulovat ve tvaru

$$z = \theta - \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \theta \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_q,$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

Optimální hodnoty modelu dávají jednotce U_q návod pro zlepšení vstupů a výstupů na $\mathbf{x}'_q, \mathbf{y}'_q$ pomocí tzv. **CCR - projekce**:

$$\mathbf{x}'_q = \theta^* \mathbf{x}_q - \mathbf{s}^{-*}, \mathbf{y}'_q = \mathbf{y}_q + \mathbf{s}^{+*},$$

nebo též

$$\mathbf{x}'_q = \mathbf{X}\lambda^*, \mathbf{y}'_q = \mathbf{Y}\lambda^*.$$

Přitom ty indexy $j \in \{1, \dots, n\}$, pro něž jsou hodnoty λ_j^* kladné, určují **referenční jednotky** pro U_q .

CCR model orientovaný na výstupy

Pro jednotku U_q můžeme také formulovat CCR model orientovaný na výstupy, zapišme jej rovnou v upravené duální podobě:

$$z = \Theta + \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\Theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \Theta\mathbf{y}_q,$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

CCR model orientovaný na výstupy

Pro jednotku U_q můžeme také formulovat CCR model orientovaný na výstupy, zapišme jej rovnou v upravené duální podobě:

$$z = \Theta + \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\Theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \Theta \mathbf{y}_q,$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

Pokud je $\Theta^* > 1$, jednotka U_q není efektivní a hodnota Θ^* vyjadřuje potřebnou míru proporcionalního navýšení vstupů. Opět lze analogicky definovat CCR projekci na efektivní hranici jako

$$\mathbf{x}'_q = \mathbf{X}\lambda^*, \mathbf{y}'_q = \mathbf{Y}\lambda^*.$$

CCR model orientovaný na výstupy

Pro jednotku U_q můžeme také formulovat CCR model orientovaný na výstupy, zapišme jej rovnou v upravené duální podobě:

$$z = \Theta + \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\Theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \Theta \mathbf{y}_q,$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

Pokud je $\Theta^* > 1$, jednotka U_q není efektivní a hodnota Θ^* vyjadřuje potřebnou míru proporcionalního navýšení vstupů. Opět lze analogicky definovat CCR projekci na efektivní hranici jako

$$\mathbf{x}'_q = \mathbf{X}\lambda^*, \mathbf{y}'_q = \mathbf{Y}\lambda^*.$$

V CCR modelech platí, že míry efektivity jednotky U_q při orientaci na vstupy či výstupy (tj. optimální hodnoty účelových funkcí modelů) jsou vzájemně převrácenými hodnotami $\theta^* \cdot \Theta^* = 1$

BCC model orientovaný na vstupy

Jako modifikaci modelu CCR, který předpokládá konstantní výnosy z rozsahu a definuje tak kónický obal dat, navrhli Banker, Charnes a Cooper model využívající variabilní výnosy z rozsahu, tzv. **BCC model**. Při tomto přístupu jsou data obalována konvexním obalem, jako virtuální jednotky se neuvažují libovolné nezáporné kombinace $\mathbf{X}\lambda$, $\mathbf{Y}\lambda$, ale pouze kombinace splňující podmínku $\mathbf{e}\lambda = 1$. Díky této dodatečné podmínce vychází zpravidla jako efektivní větší počet hodnocených jednotek.

BCC model orientovaný na vstupy

Jako modifikaci modelu CCR, který předpokládá konstantní výnosy z rozsahu a definuje tak kónický obal dat, navrhli Banker, Charnes a Cooper model využívající variabilní výnosy z rozsahu, tzv. **BCC model**. Při tomto přístupu jsou data obalována konvexním obalem, jako virtuální jednotky se neuvažují libovolné nezáporné kombinace $\mathbf{X}\lambda$, $\mathbf{Y}\lambda$, ale pouze kombinace splňující podmínku $\mathbf{e}\lambda = 1$. Díky této dodatečné podmínce vychází zpravidla jako efektivní větší počet hodnocených jednotek.

Uveďme formulaci duálního BCC modelu orientovaného na vstupy:

$$z = \theta - \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \theta \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_q,$$

$$\mathbf{e}\lambda = 1$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

BCC model orientovaný na vstupy

Jako modifikaci modelu CCR, který předpokládá konstantní výnosy z rozsahu a definuje tak kónický obal dat, navrhli Banker, Charnes a Cooper model využívající variabilní výnosy z rozsahu, tzv. **BCC model**. Při tomto přístupu jsou data obalována konvexním obalem, jako virtuální jednotky se neuvažují libovolné nezáporné kombinace $\mathbf{X}\lambda$, $\mathbf{Y}\lambda$, ale pouze kombinace splňující podmínku $\mathbf{e}\lambda = 1$. Díky této dodatečné podmínce vychází zpravidla jako efektivní větší počet hodnocených jednotek.

Uveďme formulaci duálního BCC modelu orientovaného na vstupy:

$$z = \theta - \varepsilon \cdot (\mathbf{e} \cdot \mathbf{s}^+ + \mathbf{e} \cdot \mathbf{s}^-) \rightarrow \min_{\theta, \lambda, \mathbf{s}^+, \mathbf{s}^-}$$

za omezení

$$\mathbf{s}^- = \theta \mathbf{x}_q - \mathbf{X}\lambda,$$

$$\mathbf{s}^+ = \mathbf{Y}\lambda - \mathbf{y}_q,$$

$$\mathbf{e}\lambda = 1$$

$$\lambda, \mathbf{s}^+, \mathbf{s}^- \geq 0$$

Jako **BCC efektivní** jsou identifikovány ty jednotky, pro něž je

$$\theta^* = 1, \mathbf{s}^{+*} = \mathbf{0}, \mathbf{s}^{-*} = \mathbf{0}.$$

Další DEA modely

Kromě základních DEA modelů byla navržena řada jejich modifikací, mimo jiné:

- aditivní, též **SBM (Slack-Based Measure) model**: nenutí uživatele rozlišovat mezi orientací na vstupy nebo výstupy, měří efektivnost přímo pomocí přídatných proměnných s^- , s^+
- DEA modely s nekontrolovatelnými výstupy a vstupy
- DEA modely s nežádoucími výstupy a vstupy
- Modely **superefektivnosti**
- diskrétní modely, např. **FDH (Free Disposable Hull) model**
- pro hodnocení změn efektivnosti v čase navržený **Malmquistův index**

Metody jsou podrobněji popsány v použité literatuře:

- J. Jablonský, M. Dlouhý: Modely hodnocení produkčních jednotek, Professional Publishing, Praha 2004
- W. W. Cooper, L. M. Seiford, K. Tone: Data Envelopment Analysis, Springer, New York 2007

Použití DEA k hodnocení efektivnosti dopravních podniků

Údaje z výroční zprávy Sdružení dopravních podniků za rok 2012 (<http://www.sdp-cr.cz/o-nas/vyrocní-zpravy/>) byly použity k hodnocení efektivnosti podniků z 19 měst. V první tabulce jsou uvedeny údaje v následujícím pořadí:

- počet přepravených osob (tis. osob)
- tržby (tis. Kč)
- vozové kilometry (tis.)
- počet zaměstnanců
- z toho řidičů

Druhá tabulka popisuje vozový park.

město	převprav.	tržby	vozokm	zaměst.	řidičů
1. Brno	352 052	994 040	38 118	2 727	1 375
2. České Budějovice	38 091	129 059	5 673	384	187
3. Děčín	8 938	42 731	3 678	207	132
4. Hradec Králové	35 162	120 854	6 242	401	226
5. Chomutov-Jirkov	5 223	50 367	1 838	253	163
6. Jihlava	13 530	50 232	2 821	170	94
7. Karlovy Vary	13 436	65 174	2 624	260	148
8. Liberec	32 656	192 236	8 648	377	169
9. Mariánské Lázně	3 844	11 439	495	30	19
10. Most-Litvínov	27 418	110 059	4 908	471	216
11. Olomouc	52 737	143 318	5 902	432	255
12. Opava	10 750	50 476	3 046	180	115
13. Ostrava	96 389	519 873	33 773	2 008	1 026
14. Pardubice	27 178	119 280	5 721	406	189
15. Plzeň	99 154	300 097	15 102	1 030	570
16. Praha	1 383 124	4 508 422	167 760	10 595	4 207
17. Teplice	15 039	94 159	5 726	265	176
18. Ústí nad Labem	47 091	203 278	7 347	501	253
19. Zlín-Otrokovice	32 335	119 506	4 812	340	184

město	autobusy	tramvaje	trolejbusy	metro	celkem
1. Brno	298	309	151	0	758
2. České Budějovice	83	0	58	0	141
3. Děčín	55	0	0	0	55
4. Hradec Králové	95	0	36	0	131
5. Chomutov-Jirkov	29	0	19	0	48
6. Jihlava	32	0	32	0	64
7. Karlovy Vary	67	0	0	0	67
8. Liberec	139	68	0	0	207
9. Mariánské Lázně	4	0	9	0	30
10. Most-Litvínov	80	57	0	0	137
11. Olomouc	77	60	0	0	137
12. Opava	34	0	34	0	68
13. Ostrava	297	273	62	0	632
14. Pardubice	73	0	55	0	128
15. Plzeň	113	122	88	0	323
16. Praha	1214	920	0	738	2872
17. Teplice	65	0	38	0	103
18. Ústí n. Labem	73	0	68	0	141
19. Zlín-Otrokovice	37	0	57	0	94

Použití DEA k hodnocení efektivnosti dopravních podniků

Jako vstupy pro analýzu byly použity celkové počty zaměstnanců a celkový vozový park, jako výstupy počty přepravených osob, tržby a ujeté vozové kilometry. Byl zvolen model CCR orientovaný na vstupy, výpočty byly realizovány prostřednictvím doplňku Solver pro MS Excel a aplikace pro DEA (<http://nb.vse.cz/jablon/>)

Použití DEA k hodnocení efektivnosti dopravních podniků

Jako vstupy pro analýzu byly použity celkové počty zaměstnanců a celkový vozový park, jako výstupy počty přepravených osob, tržby a ujeté vozové kilometry. Byl zvolen model CCR orientovaný na vstupy, výpočty byly realizovány prostřednictvím doplňku Solver pro MS Excel a aplikace pro DEA (<http://nb.vse.cz/jablon/>)

Výsledky jsou uvedeny v následující tabulce, červeně jsou vyznačeny podniky, které vyšly jako efektivní.

město	efektivita	čísla	refer.	jednot.
1. Brno	0,9889	16		
2. České Budějovice	0,833	8	9	16
3. Děčín	1	3		
4. Hradec Králové	0,8546	8	16	17
5. Chomutov-Jirkov	0,6684	16		
6. Jihlava	0,8493	8	16	17
7. Karlovy Vary	0,6592	3	16	
8. Liberec	1	8		
9. Mariánské Lázně	1	9		
10. Most-Litvínov	0,6092	3	16	17
11. Olomouc	0,9351	16		
12. Opava	0,8209	8	16	17
13. Ostrava	0,8838	3	16	17
14. Pardubice	0,7743	3	16	17
15. Plzeň	0,8426	8	16	17
16. Praha	1	16		
17. Teplice	1	17		
18. Ústí n. Labem	0,9465	8	16	
19. Zlín-Otrokovice	0,8695	3	16	17

Matematická analýza funkcí více proměnných

Předpokládejme dále, že funkce f je spojitě diferencovatelná až do řádu 2.

Gradient funkce

Matematická analýza funkcí více proměnných

Předpokládejme dále, že funkce f je spojitě diferencovatelná až do řádu 2.

Gradient funkce

Bud' $X \subseteq \mathbb{R}^n$, $f : X \rightarrow \mathbb{R}$ funkce n proměnných. Gradientem funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme vektor

$$\nabla f(\mathbf{t}) = (f'_{x_1}(\mathbf{t}), \dots, f'_{x_n}(\mathbf{t}))^\top.$$

Matematická analýza funkcí více proměnných

Předpokládejme dále, že funkce f je spojitě diferencovatelná až do řádu 2.

Gradient funkce

Bud' $X \subseteq \mathbb{R}^n$, $f : X \rightarrow \mathbb{R}$ funkce n proměnných. Gradientem funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme vektor

$$\nabla f(\mathbf{t}) = (f'_{x_1}(\mathbf{t}), \dots, f'_{x_n}(\mathbf{t}))^\top.$$

Hessova matice

Hessovou maticí funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme symetrickou matici řádu n :

$$H(\mathbf{t}) = (f''_{x_i, x_j}(\mathbf{t}))_{i,j=1}^n$$

Matematická analýza funkcí více proměnných

Předpokládejme dále, že funkce f je spojitě diferencovatelná až do řádu 2.

Gradient funkce

Bud' $X \subseteq \mathbb{R}^n$, $f : X \rightarrow \mathbb{R}$ funkce n proměnných. Gradientem funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme vektor

$$\nabla f(\mathbf{t}) = (f'_{x_1}(\mathbf{t}), \dots, f'_{x_n}(\mathbf{t}))^\top.$$

Hessova matice

Hessovou maticí funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme symetrickou matici řádu n :

$$H(\mathbf{t}) = (f''_{x_i, x_j}(\mathbf{t}))_{i,j=1}^n$$

Příklad : Určete gradient a Hessovu matici funkce $f(x, y) = x \cdot y^2$ v bodě $(5, 3)$.

Matematická analýza funkcí více proměnných

Předpokládejme dále, že funkce f je spojitě diferencovatelná až do řádu 2.

Gradient funkce

Bud' $X \subseteq \mathbb{R}^n$, $f : X \rightarrow \mathbb{R}$ funkce n proměnných. Gradientem funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme vektor

$$\nabla f(\mathbf{t}) = (f'_{x_1}(\mathbf{t}), \dots, f'_{x_n}(\mathbf{t}))^\top.$$

Hessova matice

Hessovou maticí funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme symetrickou matici řádu n :

$$H(\mathbf{t}) = (f''_{x_i, x_j}(\mathbf{t}))_{i,j=1}^n$$

Příklad : Určete gradient a Hessovu matici funkce $f(x, y) = x \cdot y^2$ v bodě $(5, 3)$.

Řešení: Parciální derivace prvního řádu jsou: $f'_x(x, y) = y^2$, $f'_y(x, y) = 2x \cdot y$, tedy $\nabla f(5, 3) = (9, 30)^\top$.

Matematická analýza funkcí více proměnných

Předpokládejme dále, že funkce f je spojitě diferencovatelná až do řádu 2.

Gradient funkce

Bud' $X \subseteq \mathbb{R}^n$, $f : X \rightarrow \mathbb{R}$ funkce n proměnných. Gradientem funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme vektor

$$\nabla f(\mathbf{t}) = (f'_{x_1}(\mathbf{t}), \dots, f'_{x_n}(\mathbf{t}))^\top.$$

Hessova matice

Hessovou maticí funkce f v bodě $\mathbf{t} = (t_1, \dots, t_n) \in X$ nazýváme symetrickou matici řádu n :

$$H(\mathbf{t}) = (f''_{x_i, x_j}(\mathbf{t}))_{i,j=1}^n$$

Příklad : Určete gradient a Hessovu matici funkce $f(x, y) = x \cdot y^2$ v bodě $(5, 3)$.

Řešení: Parciální derivace prvního řádu jsou: $f'_x(x, y) = y^2$, $f'_y(x, y) = 2x \cdot y$, tedy $\nabla f(5, 3) = (9, 30)^\top$. Parciální derivace druhého řádu jsou:

$$f''_{xx}(x, y) = 0, \quad f''_{xy}(x, y) = 2y, \quad f''_{yy}(x, y) = 2x, \quad \text{tedy } H(5, 3) = \begin{pmatrix} 0 & 6 \\ 6 & 10 \end{pmatrix}$$

Směrové derivace

Uvažujme $X \subseteq \mathbb{R}^n$, funkci $f : X \rightarrow \mathbb{R}$, bod $\mathbf{t} \in X$ a jednotkový vektor $\mathbf{s} \in \mathbb{R}^n$. Vytvoříme funkci jedné proměnné $\varphi(x) = f(\mathbf{t} + x \cdot \mathbf{s})$. Hodnotu $\varphi'(0)$ nazveme **derivací f ve směru \mathbf{s}** a značíme $f'_s(\mathbf{t})$. (pozn.: obdobně definujeme $f''_s(\mathbf{t})$ jako $\varphi''(0)$.)

Směrové derivace

Uvažujme $X \subseteq \mathbb{R}^n$, funkci $f : X \rightarrow \mathbb{R}$, bod $\mathbf{t} \in X$ a jednotkový vektor $\mathbf{s} \in \mathbb{R}^n$. Vytvoříme funkci jedné proměnné $\varphi(x) = f(\mathbf{t} + x \cdot \mathbf{s})$. Hodnotu $\varphi'(0)$ nazveme **derivací f ve směru \mathbf{s}** a značíme $f'_s(\mathbf{t})$. (pozn.: obdobně definujeme $f''_s(\mathbf{t})$ jako $\varphi''(0)$.) Dá se ukázat, že platí

$$f'_s(\mathbf{t}) = \mathbf{s} \cdot \nabla f(\mathbf{t}), \quad f''_s(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^\top.$$

Směrové derivace

Uvažujme $X \subseteq \mathbb{R}^n$, funkci $f : X \rightarrow \mathbb{R}$, bod $\mathbf{t} \in X$ a jednotkový vektor $\mathbf{s} \in \mathbb{R}^n$. Vytvoříme funkci jedné proměnné $\varphi(x) = f(\mathbf{t} + x \cdot \mathbf{s})$. Hodnotu $\varphi'(0)$ nazveme **derivací f ve směru \mathbf{s}** a značíme $f'_s(\mathbf{t})$. (pozn.: obdobně definujeme $f''_s(\mathbf{t})$ jako $\varphi''(0)$.) Dá se ukázat, že platí

$$f'_s(\mathbf{t}) = \mathbf{s} \cdot \nabla f(\mathbf{t}), \quad f''_s(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^\top.$$

Příklad : Určete první a druhou derivaci funkce $f(x, y) = x \cdot y^2$ ve směru $\mathbf{s} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.

Směrové derivace

Uvažujme $X \subseteq \mathbb{R}^n$, funkci $f : X \rightarrow \mathbb{R}$, bod $\mathbf{t} \in X$ a jednotkový vektor $\mathbf{s} \in \mathbb{R}^n$. Vytvoříme funkci jedné proměnné $\varphi(x) = f(\mathbf{t} + x \cdot \mathbf{s})$. Hodnotu $\varphi'(0)$ nazveme **derivací f ve směru \mathbf{s}** a značíme $f'_s(\mathbf{t})$. (pozn.: obdobně definujeme $f''_s(\mathbf{t})$ jako $\varphi''(0)$.) Dá se ukázat, že platí

$$f'_s(\mathbf{t}) = \mathbf{s} \cdot \nabla f(\mathbf{t}), \quad f''_s(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^\top.$$

Příklad : Určete první a druhou derivaci funkce $f(x, y) = x \cdot y^2$ ve směru $\mathbf{s} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.

Řešení: Víme, že $\nabla f(x, y) = (y^2, 2xy)^\top$, tedy $f'_s(x, y) = \frac{y^2 + 2xy}{\sqrt{2}}$.

Směrové derivace

Uvažujme $X \subseteq \mathbb{R}^n$, funkci $f : X \rightarrow \mathbb{R}$, bod $\mathbf{t} \in X$ a jednotkový vektor $\mathbf{s} \in \mathbb{R}^n$. Vytvoříme funkci jedné proměnné $\varphi(x) = f(\mathbf{t} + x \cdot \mathbf{s})$. Hodnotu $\varphi'(0)$ nazveme **derivací f ve směru \mathbf{s}** a značíme $f'_s(\mathbf{t})$. (pozn.: obdobně definujeme $f''_s(\mathbf{t})$ jako $\varphi''(0)$.) Dá se ukázat, že platí

$$f'_s(\mathbf{t}) = \mathbf{s} \cdot \nabla f(\mathbf{t}), \quad f''_s(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^\top.$$

Příklad : Určete první a druhou derivaci funkce $f(x, y) = x \cdot y^2$ ve směru $\mathbf{s} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$.

Řešení: Víme, že $\nabla f(x, y) = (y^2, 2xy)^\top$, tedy $f'_s(x, y) = \frac{y^2 + 2xy}{\sqrt{2}}$.

Hessova matice je: $H(x, y) = \begin{pmatrix} 0 & 2y \\ 2y & 2x \end{pmatrix}$, tedy

$$f''_s(x, y) = \frac{1}{2} \cdot (0 + 2y + 2y + 2x) = 2y + x.$$

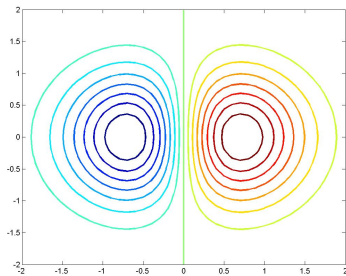
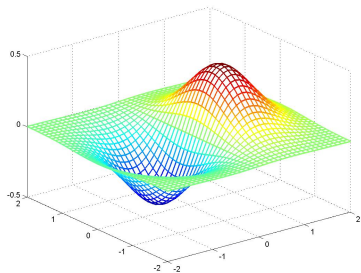
Grafické znázornění funkce dvou proměnných

V třírozměrném prostoru si můžeme graf funkce dvou proměnných představit jako zemský povrch. Pro znázornění povrchu ve 2D se používají většinou vrstevnice funkce.



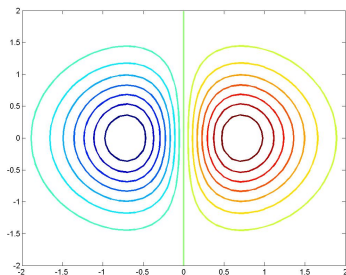
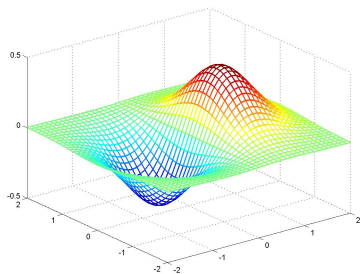
Grafické znázornění funkce dvou proměnných

Podobným způsobem si můžeme znázornit třeba funkci $f(x, y) = \frac{x}{e^{x^2+y^2}}$.



Grafické znázornění funkce dvou proměnných

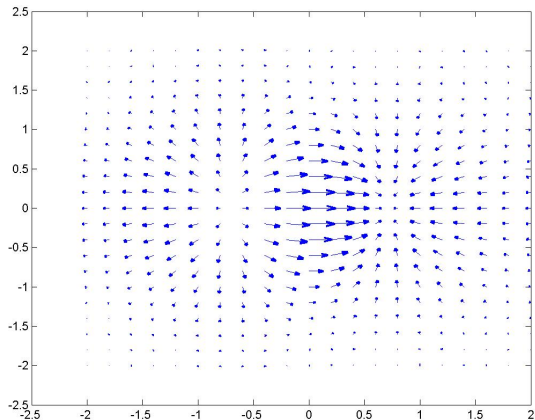
Podobným způsobem si můžeme znázornit třeba funkci $f(x, y) = \frac{x}{e^{x^2+y^2}}$.



Vrstevníci funkce $f(x, y)$ "o nadmořské výšce c " rozumíme množinu všech bodů $(x, y) \in \mathbb{R}^2$ takových, že platí $f(x, y) = c$. Například pro výše uvedenou funkci $f(x, y)$ určíme nulovou vrstevníci jako množinu všech řešení rovnice o dvou neznámých $\frac{x}{e^{x^2+y^2}} = 0$. Zřejmě musí být $x = 0$, ale y je libovolné, tedy dostaneme množinu $\{(0, y), y \in \mathbb{R}\}$.

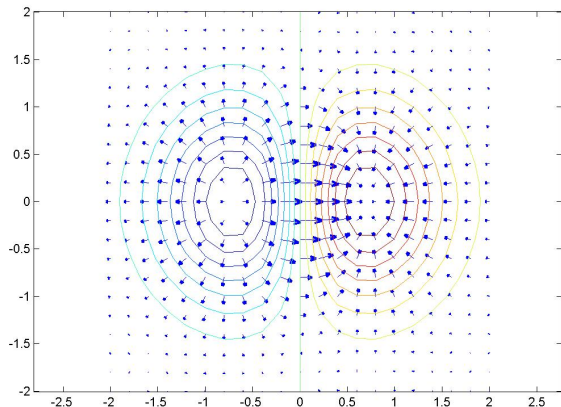
Grafické znázornění gradientů funkce dvou proměnných

Gradient $\nabla f(x, y)$ většinou znázorňujeme jako vektor vycházející z bodu (x, y) do bodu $(x + f'_x(x, y), y + f'_y(x, y))$. Na obrázku vidíme gradienty funkce $f(x, y) = \frac{x}{e^{x^2+y^2}}$ v bodech pravidelné sítě.



Gradienty a vrstevnice

Znázorníme-li gradienty funkce do stejného obrázku s vrstevnicemi, můžeme si všimnout, že se jeví jako normálové vektory vrstevnic. Dá se ukázat, že libovolná funkce f v zadaném bodě \mathbf{x} nejprudčeji roste ve směru gradientu $\nabla f(\mathbf{x})$ a nejstrměji klesá ve směru $-\nabla f(\mathbf{x})$.



Konvexní množina

Množinu $M \subset \mathbb{R}^n$ nazveme **konvexní**, jestliže pro každé dva její body A, B jsou všechny body úsečky AB také prvky množiny M . Tuto vlastnost můžeme analyticky vyjádřit symbolickým zápisem:

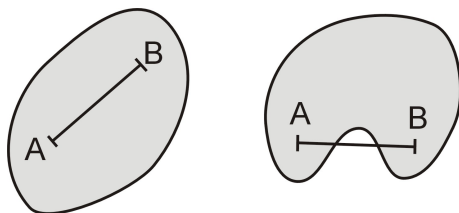
$$A, B \in M \Rightarrow \forall \lambda \in \langle 0, 1 \rangle : \lambda A + (1 - \lambda)B \in M$$

Konvexní množina

Množinu $M \subset \mathbb{R}^n$ nazveme **konvexní**, jestliže pro každé dva její body A, B jsou všechny body úsečky AB také prvky množiny M . Tuto vlastnost můžeme analyticky vyjádřit symbolickým zápisem:

$$A, B \in M \Rightarrow \forall \lambda \in \langle 0, 1 \rangle : \lambda A + (1 - \lambda)B \in M$$

Na obrázku je znázorněn příklad konvexní a nekonvexní množiny.



Poznámka: Průnik dvou konvexních množin je opět konvexní množinou (toto tvrzení lze tedy zřejmě rozšířit pro průnik více konvexních množin).

Konvexní funkce

Funkci f definovanou na konvexní množině $M \subseteq \mathbb{R}^n$ nazveme **konvexní** na M , jestliže pro každé dva body $A, B \in M$ platí:

$$\forall \lambda \in \langle 0, 1 \rangle : f(\lambda A + (1 - \lambda)B) \leq \lambda f(A) + (1 - \lambda)f(B).$$

Konvexní funkce

Funkci f definovanou na konvexní množině $M \subseteq \mathbb{R}^n$ nazveme **konvexní** na M , jestliže pro každé dva body $A, B \in M$ platí:

$$\forall \lambda \in \langle 0, 1 \rangle : f(\lambda A + (1 - \lambda)B) \leq \lambda f(A) + (1 - \lambda)f(B).$$

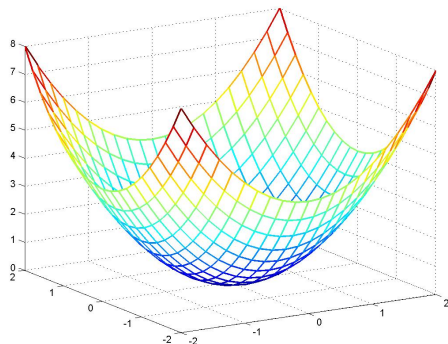
Pokud je pro všechna $A \neq B$ a $\lambda \in (0, 1)$ tato nerovnost ostrá, je funkce f na množině M **ostře konvexní**. Geometrický význam: "Spojnice každých dvou bodů grafu leží nad grafem."

Konvexní funkce

Funkci f definovanou na konvexní množině $M \subseteq \mathbb{R}^n$ nazveme **konvexní** na M , jestliže pro každé dva body $A, B \in M$ platí:

$$\forall \lambda \in \langle 0, 1 \rangle : f(\lambda A + (1 - \lambda)B) \leq \lambda f(A) + (1 - \lambda)f(B).$$

Pokud je pro všechna $A \neq B$ a $\lambda \in (0, 1)$ tato nerovnost ostrá, je funkce f na množině M **ostře konvexní**. Geometrický význam: "Spojnice každých dvou bodů grafu leží nad grafem." Na obrázku je znázorněn příklad funkce $f(x, y) = x^2 + y^2$, která je konvexní v \mathbb{R}^2 .



Konvexní funkce

Příklady konvexních funkcí v \mathbb{R}^n :

- Pro libovolný vektor $\mathbf{c} \in \mathbb{R}^n$ je lineární funkce $f(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x}$ konvexní na \mathbb{R}^n (není ale ostře konvexní).
- Euklidovská metrika $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ je konvexní na \mathbb{R}^n .

Konvexní funkce

Příklady konvexních funkcí v \mathbb{R}^n :

- Pro libovolný vektor $\mathbf{c} \in \mathbb{R}^n$ je lineární funkce $f(\mathbf{x}) = \mathbf{c}^T \cdot \mathbf{x}$ konvexní na \mathbb{R}^n (není ale ostře konvexní).
- Euklidovská metrika $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n x_i^2}$ je konvexní na \mathbb{R}^n .

Pro konvexní funkce platí řada tvrzení:

- Jsou-li $f(\mathbf{x})$ a $g(\mathbf{x})$ konvexní funkce, pak jejich součet $f(\mathbf{x}) + g(\mathbf{x})$ je též konvexní (totéž platí i pro součin $f(\mathbf{x}) \cdot g(\mathbf{x})$ v případě nezápornosti funkcí).
- Pro konvexní funkci $f(\mathbf{x})$ na \mathbb{R}^n a libovolnou konstantu c platí: Množina $X = \{\mathbf{x} \in \mathbb{R}^n : f(\mathbf{x}) \leq c\}$ je konvexní.
- Protože průnik konvexních množin je konvexní množinou, zřejmě je i přípustná množina lineárních úloh $X = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$ pro libovolnou matici \mathbf{A} a vektor pravých stran \mathbf{b} konvexní množinou. Říkáme, že X je konvexní polyedr.

Hessova matice konvexní funkce

Funkci f nazveme konvexní v bodě \mathbf{t} , jestliže existuje okolí tohoto bodu, na kterém je konvexní. U funkce jedné proměnné lze konvexitu rozpoznat podle znaménka druhé derivace.

Hessova matice konvexní funkce

Funkci f nazveme konvexní v bodě \mathbf{t} , jestliže existuje okolí tohoto bodu, na kterém je konvexní. U funkce jedné proměnné lze konvexitu rozpoznat podle znaménka druhé derivace. Zobecněním této úvahy pro funkci více proměnných dostaneme tvrzení: Dvakrát diferencovatelná funkce f je konvexní v bodě \mathbf{t} právě když pro každý směr \mathbf{s} platí: $f''_{\mathbf{s}}(\mathbf{t}) \geq 0$ (při platnosti ostré nerovnosti dostaneme ostrou konvexitu).

Hessova matice konvexní funkce

Funkci f nazveme konvexní v bodě \mathbf{t} , jestliže existuje okolí tohoto bodu, na kterém je konvexní. U funkce jedné proměnné lze konvexitu rozpoznat podle znaménka druhé derivace. Zobecněním této úvahy pro funkci více proměnných dostaneme tvrzení: Dvakrát diferencovatelná funkce f je konvexní v bodě \mathbf{t} právě když pro každý směr \mathbf{s} platí: $f''_{\mathbf{s}}(\mathbf{t}) \geq 0$ (při platnosti ostré nerovnosti dostaneme ostrou konvexitu). Tedy Hessova matice $H(\mathbf{t})$ má následující vlastnost:

$$\forall \mathbf{s} \in \mathbb{R}^n : f''_{\mathbf{s}}(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^T \geq 0$$

V lineární algebře se takové matice nazývají pozitivně semidefinitní (pokud je nerovnost pro každé nenulové \mathbf{s} ostrá, pak je matice pozitivně definitní).

Hessova matice konvexní funkce

Funkci f nazveme konvexní v bodě \mathbf{t} , jestliže existuje okolí tohoto bodu, na kterém je konvexní. U funkce jedné proměnné lze konvexitu rozpoznat podle znaménka druhé derivace. Zobecněním této úvahy pro funkci více proměnných dostaneme tvrzení: Dvakrát diferencovatelná funkce f je konvexní v bodě \mathbf{t} právě když pro každý směr \mathbf{s} platí: $f''_{\mathbf{s}}(\mathbf{t}) \geq 0$ (při platnosti ostré nerovnosti dostaneme ostrou konvexitu). Tedy Hessova matice $H(\mathbf{t})$ má následující vlastnost:

$$\forall \mathbf{s} \in \mathbb{R}^n : f''_{\mathbf{s}}(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^T \geq 0$$

V lineární algebře se takové matice nazývají pozitivně semidefinitní (pokud je nerovnost pro každé nenulové \mathbf{s} ostrá, pak je matice pozitivně definitní).

Příklad : Je funkce $f(x, y, z) = x^2 + z \cdot y^2$ konvexní v bodě $[1, 1, 1]$?

Hessova matice konvexní funkce

Funkci f nazveme konvexní v bodě \mathbf{t} , jestliže existuje okolí tohoto bodu, na kterém je konvexní. U funkce jedné proměnné lze konvexitu rozpoznat podle znaménka druhé derivace. Zobecněním této úvahy pro funkci více proměnných dostaneme tvrzení: Dvakrát diferencovatelná funkce f je konvexní v bodě \mathbf{t} právě když pro každý směr \mathbf{s} platí: $f''_{\mathbf{s}}(\mathbf{t}) \geq 0$ (při platnosti ostré nerovnosti dostaneme ostrou konvexitu). Tedy Hessova matice $H(\mathbf{t})$ má následující vlastnost:

$$\forall \mathbf{s} \in \mathbb{R}^n : f''_{\mathbf{s}}(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^T \geq 0$$

V lineární algebře se takové matice nazývají pozitivně semidefinitní (pokud je nerovnost pro každé nenulové \mathbf{s} ostrá, pak je matice pozitivně definitní).

Příklad : Je funkce $f(x, y, z) = x^2 + z \cdot y^2$ konvexní v bodě $[1, 1, 1]$?

Řešení: Spočítáme Hessovu matici:

Hessova matice konvexní funkce

Funkci f nazveme konvexní v bodě \mathbf{t} , jestliže existuje okolí tohoto bodu, na kterém je konvexní. U funkce jedné proměnné lze konvexitu rozpoznat podle znaménka druhé derivace. Zobecněním této úvahy pro funkci více proměnných dostaneme tvrzení: Dvakrát diferencovatelná funkce f je konvexní v bodě \mathbf{t} právě když **pro každý směr \mathbf{s} platí: $f''_{\mathbf{s}}(\mathbf{t}) \geq 0$** (při platnosti ostré nerovnosti dostaneme ostrou konvexitu). Tedy Hessova matice $H(\mathbf{t})$ má následující vlastnost:

$$\forall \mathbf{s} \in \mathbb{R}^n : f''_{\mathbf{s}}(\mathbf{t}) = \mathbf{s} \cdot H(\mathbf{t}) \cdot \mathbf{s}^T \geq 0$$

V lineární algebře se takové matice nazývají **pozitivně semidefinitní** (pokud je nerovnost pro každé nenulové \mathbf{s} ostrá, pak je matice pozitivně definitní).

Příklad : Je funkce $f(x, y, z) = x^2 + z \cdot y^2$ konvexní v bodě $[1, 1, 1]$?

Řešení: Spočítáme Hessovu matici: $H(1, 1, 1) = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 2 & 0 \end{pmatrix}$, například

pro vektor $\mathbf{s} = (-1, -1, 2)$ platí $\mathbf{s} \cdot H(1, 1, 1) \cdot \mathbf{s}^T = -4 < 0$. Funkce není v bodě $[1, 1, 1]$ konvexní.

Lokální extrémý

Řekneme, že funkce $f : \mathbb{R}^n \rightarrow \mathbb{R}$ má v bodě $\mathbf{a} \in Df$:

- 1 **lokální maximum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \leq f(\mathbf{a})$
- 2 **lokální minimum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \geq f(\mathbf{a})$

Lokální extrémy

Řekneme, že funkce $f : \mathbb{R}^n \rightarrow \mathbb{R}$ má v bodě $\mathbf{a} \in Df$:

- 1 **lokální maximum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \leq f(\mathbf{a})$
- 2 **lokální minimum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \geq f(\mathbf{a})$

Poznámka: jsou-li nerovnosti splněny na ryzím okolí $U_\delta(\mathbf{a}) \setminus \{\mathbf{a}\}$ ostře, pak extrémy nazýváme **ostré**.

Poznámka: Funkce f může mít lokální extrémy pouze ve **stacionárních bodech** (tedy bodech s nulovým gradientem), nebo v bodech, v nichž neexistuje aspoň jedna parciální derivace prvního řádu.

Lokální extrémy

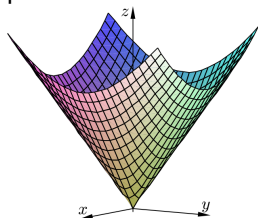
Řekneme, že funkce $f : \mathbb{R}^n \rightarrow \mathbb{R}$ má v bodě $\mathbf{a} \in Df$:

- 1 **lokální maximum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že
 $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \leq f(\mathbf{a})$
- 2 **lokální minimum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že
 $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \geq f(\mathbf{a})$

Poznámka: jsou-li nerovnosti splněny na ryzím okolí $U_\delta(\mathbf{a}) \setminus \{\mathbf{a}\}$ ostře, pak extrémy nazýváme **ostré**.

Poznámka: Funkce f může mít lokální extrémy pouze ve **stacionárních bodech** (tedy bodech s nulovým gradientem), nebo v bodech, v nichž neexistuje aspoň jedna parciální derivace prvního řádu.

Příklad: Funkce $f(x, y) = \sqrt{x^2 + y^2}$ má minimum v bodě $[0, 0]$, kde neexistují parciální derivace.



Lokální extrémy

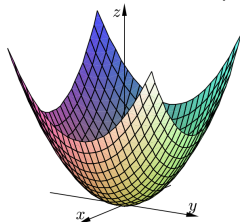
Řekneme, že funkce $f : \mathbb{R}^n \rightarrow \mathbb{R}$ má v bodě $\mathbf{a} \in Df$:

- 1 **lokální maximum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že
 $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \leq f(\mathbf{a})$
- 2 **lokální minimum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že
 $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \geq f(\mathbf{a})$

Poznámka: jsou-li nerovnosti splněny na ryzím okolí $U_\delta(\mathbf{a}) \setminus \{\mathbf{a}\}$ ostře, pak extrémy nazýváme **ostré**.

Poznámka: Funkce f může mít lokální extrémy pouze ve **stacionárních bodech** (tedy bodech s nulovým gradientem), nebo v bodech, v nichž neexistuje aspoň jedna parciální derivace prvního řádu.

Příklad: Funkce $f(x, y) = x^2 + y^2$ má minimum ve stacionárním bodě $[0, 0]$.



Lokální extrémy

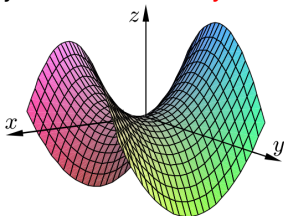
Řekneme, že funkce $f : \mathbb{R}^n \rightarrow \mathbb{R}$ má v bodě $\mathbf{a} \in Df$:

- 1 **lokální maximum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že
 $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \leq f(\mathbf{a})$
- 2 **lokální minimum**, když existuje jeho δ -okolí $U_\delta(\mathbf{a}) \subset Df$ takové, že
 $\forall x \in U_\delta(\mathbf{a})$ platí $f(x) \geq f(\mathbf{a})$

Poznámka: jsou-li nerovnosti splněny na ryzím okolí $U_\delta(\mathbf{a}) \setminus \{\mathbf{a}\}$ ostře, pak extrémy nazýváme **ostré**.

Poznámka: Funkce f může mít lokální extrémy pouze ve **stacionárních bodech** (tedy bodech s nulovým gradientem), nebo v bodech, v nichž neexistuje aspoň jedna parciální derivace prvního řádu.

Příklad: Funkce $f(x, y) = x^2 - y^2$ má stacionární bod $[0, 0]$, kde není extrém, jedná se o **sedlový bod**.



Lokální extrémy

Při rozhodování o tom, zda ve stacionárním bodě nastává lokální extrém, se řídíme pomocí Hessiany matice. Zřejmě je-li ve svém stacionárním bodě funkce f ostře konvexní, tj. má-li zde pozitivně definitní Hessovu matici, pak zde nabývá svého lokálního minima. Pozitivně definitní matici lze rozpoznat podle toho, že má všechny **hlavní minory**, tj. determinanty vycházející z levého horního rohu, kladné.

Lokální extrémy

Při rozhodování o tom, zda ve stacionárním bodě nastává lokální extrém, se řídíme pomocí Hessovy matice. Zřejmě je-li ve svém stacionárním bodě funkce f ostře konvexní, tj. má-li zde pozitivně definitní Hessovu matici, pak zde nabývá svého lokálního minima. Pozitivně definitní matici lze rozpoznat podle toho, že má všechny **hlavní minory**, tj. determinanty vycházející z levého horního rohu, kladné.

Podmínky pro existenci extrému shrnuje

Sylvestrovo kritérium Bud' $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a $\mathbf{a} \in Df$ její stacionární bod.

Označme $D_k(\mathbf{a})$, $k = 1, \dots, n$ determinant submatice vytvořené z prvních k řádků a sloupců Hessovy matice $H(\mathbf{a})$. Pak

- 1 Jestliže $D_1(\mathbf{a}) > 0$, $D_2(\mathbf{a}) > 0, \dots, D_n(\mathbf{a}) > 0$, má f v \mathbf{a} lok. minimum.
- 2 Jestliže $D_1(\mathbf{a}) < 0$, $D_2(\mathbf{a}) > 0, \dots, (-1)^n D_n(\mathbf{a}) > 0$, má f v \mathbf{a} lok. maximum.
- 3 Jestliže jsou všechny minory $D_k(\mathbf{a})$, $k = 1, \dots, n$ nenulové a přitom neplatí žádná z předchozích možností, pak v bodě \mathbf{a} není extrém.

Lokální extrémy

Při rozhodování o tom, zda ve stacionárním bodě nastává lokální extrém, se řídíme pomocí Hessovy matice. Zřejmě je-li ve svém stacionárním bodě funkce f ostře konvexní, tj. má-li zde pozitivně definitní Hessovu matici, pak zde nabývá svého lokálního minima. Pozitivně definitní matici lze rozpoznat podle toho, že má všechny **hlavní minory**, tj. determinanty vycházející z levého horního rohu, kladné.

Podmínky pro existenci extrému shrnuje

Sylvestrovo kritérium Bud' $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a $\mathbf{a} \in Df$ její stacionární bod.

Označme $D_k(\mathbf{a})$, $k = 1, \dots, n$ determinant submatice vytvořené z prvních k řádků a sloupců Hessovy matice $H(\mathbf{a})$. Pak

- 1 Jestliže $D_1(\mathbf{a}) > 0$, $D_2(\mathbf{a}) > 0, \dots, D_n(\mathbf{a}) > 0$, má f v \mathbf{a} lok. minimum.
- 2 Jestliže $D_1(\mathbf{a}) < 0$, $D_2(\mathbf{a}) > 0, \dots, (-1)^n D_n(\mathbf{a}) > 0$, má f v \mathbf{a} lok. maximum.
- 3 Jestliže jsou všechny minory $D_k(\mathbf{a})$, $k = 1, \dots, n$ nenulové a přitom neplatí žádná z předchozích možností, pak v bodě \mathbf{a} není extrém.

Příklad: Hessova matice funkce $f(x, y) = x^2 + y^2$ ve stacionárním bodě $[0, 0]$

je: $H(0, 0) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$, její hlavní minory jsou $D_1(0, 0) = 2$, $D_2(0, 0) = 4$,

tedy je pozitivně definitní a v bodě $[0, 0]$ je minimum.

Lokální extrémy

Při rozhodování o tom, zda ve stacionárním bodě nastává lokální extrém, se řídíme pomocí Hessovy matice. Zřejmě je-li ve svém stacionárním bodě funkce f ostře konvexní, tj. má-li zde pozitivně definitní Hessovu matici, pak zde nabývá svého lokálního minima. Pozitivně definitní matici lze rozpoznat podle toho, že má všechny **hlavní minory**, tj. determinanty vycházející z levého horního rohu, kladné.

Podmínky pro existenci extrému shrnuje

Sylvestrovo kritérium Bud' $f : \mathbb{R}^n \rightarrow \mathbb{R}$ a $\mathbf{a} \in Df$ její stacionární bod.

Označme $D_k(\mathbf{a})$, $k = 1, \dots, n$ determinant submatice vytvořené z prvních k řádků a sloupců Hessovy matice $H(\mathbf{a})$. Pak

- 1 Jestliže $D_1(\mathbf{a}) > 0$, $D_2(\mathbf{a}) > 0, \dots, D_n(\mathbf{a}) > 0$, má f v \mathbf{a} lok. minimum.
- 2 Jestliže $D_1(\mathbf{a}) < 0$, $D_2(\mathbf{a}) > 0, \dots, (-1)^n D_n(\mathbf{a}) > 0$, má f v \mathbf{a} lok. maximum.
- 3 Jestliže jsou všechny minory $D_k(\mathbf{a})$, $k = 1, \dots, n$ nenulové a přitom neplatí žádná z předchozích možností, pak v bodě \mathbf{a} není extrém.

Příklad: Hessova matice funkce $f(x, y) = x^2 - y^2$ ve stacionárním bodě $[0, 0]$ je: $H(0, 0) = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}$, její hlavní minory jsou $D_1(0, 0) = 2$, $D_2(0, 0) = -4$, tedy je indefinitní a v bodě $[0, 0]$ není extrém.

Lokální extrémy - příklad

Analytické řešení úlohy hledání volných extrémů funkce f v \mathbb{R}^n tedy spočívá v nalezení **stacionárních bodů** (resp. bodů, kde neexistuje gradient) a vyšetření **definitnosti** Hessiany matice v těchto bodech. Při hledání stacionárních bodů řešíme soustavu n rovnic o n neznámých.

Lokální extrémy - příklad

Analytické řešení úlohy hledání volných extrémů funkce f v \mathbb{R}^n tedy spočívá v nalezení **stacionárních bodů** (resp. bodů, kde neexistuje gradient) a vyšetření **definitnosti** Hessiany matice v těchto bodech. Při hledání stacionárních bodů řešíme soustavu n rovnic o n neznámých.

Příklad : Nalezněte extrémy funkce $f(x, y) = x^2y + y^2x - xy$.

Lokální extrémy - příklad

Analytické řešení úlohy hledání volných extrémů funkce f v \mathbb{R}^n tedy spočívá v nalezení **stacionárních bodů** (resp. bodů, kde neexistuje gradient) a vyšetření **definitnosti** Hessiany matice v těchto bodech. Při hledání stacionárních bodů řešíme soustavu n rovnic o n neznámých.

Příklad : Nalezněte extrémy funkce $f(x, y) = x^2y + y^2x - xy$.

Řešení: První derivace jsou $f'_x = 2xy + y^2 - y$, $f'_y = x^2 + 2xy - x$. Položíme-li obě parciální derivace současně nule, má soustava následující řešení:

$x = y = 0$; $x = 0, y = 1$; $x = 1, y = 0$; $x = 1/3, y = 1/3$, což dává čtyři stacionární body dané funkce.

Lokální extrémy - příklad

Analytické řešení úlohy hledání volných extrémů funkce f v \mathbb{R}^n tedy spočívá v nalezení **stacionárních bodů** (resp. bodů, kde neexistuje gradient) a vyšetření **definitnosti** Hessiany matice v těchto bodech. Při hledání stacionárních bodů řešíme soustavu n rovnic o n neznámých.

Příklad : Nalezněte extrémy funkce $f(x, y) = x^2y + y^2x - xy$.

Řešení: První derivace jsou $f'_x = 2xy + y^2 - y$, $f'_y = x^2 + 2xy - x$. Položíme-li obě parciální derivace současně nule, má soustava následující řešení:

$x = y = 0$; $x = 0, y = 1$; $x = 1, y = 0$; $x = 1/3, y = 1/3$, což dává čtyři stacionární body dané funkce. Hodnoty Hessiany matice ve stacionárních

bodech jsou postupně $H(0, 0) = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$, $H(0, 1) = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$,

$H(1, 0) = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$, $H(1/3, 1/3) = \begin{pmatrix} 2/3 & 1/3 \\ 1/3 & 2/3 \end{pmatrix}$. Pouze poslední matice

je pozitivně definitní, funkce má jen jedno lokální minimum, a to v bodě $[1/3, 1/3]$.

Globální extrémy

Řekneme, že funkce f dosahuje na množině X v bodě $\mathbf{a} \in X$ svého

1 globálního maxima jestliže $\forall \mathbf{x} \in X$ platí $f(\mathbf{x}) \leq f(\mathbf{a})$

2 globálního minima jestliže $\forall \mathbf{x} \in X$ platí $f(\mathbf{x}) \geq f(\mathbf{a})$

Poznámka: Místo pojmu globální též používáme pojem **absolutní**. Opět definujeme ostré extrémy, jestliže nerovnosti jsou ostré pro $\forall \mathbf{x} \neq \mathbf{a}$.

Globální extrémy

Řekneme, že funkce f dosahuje na množině X v bodě $\mathbf{a} \in X$ svého

1 globálního maxima jestliže $\forall \mathbf{x} \in X$ platí $f(\mathbf{x}) \leq f(\mathbf{a})$

2 globálního minima jestliže $\forall \mathbf{x} \in X$ platí $f(\mathbf{x}) \geq f(\mathbf{a})$

Poznámka: Místo pojmu globální též používáme pojem **absolutní**. Opět definujeme ostré extrémy, jestliže nerovnosti jsou ostré pro $\forall \mathbf{x} \neq \mathbf{a}$.

Weierstrassova věta:

Je-li $X \subset \mathbb{R}^n$ **ohraničená, uzavřená** množina a $f : \mathbb{R}^n \rightarrow \mathbb{R}$ **spojitá** funkce na X , pak má f na X globální extrémy, a to buď v bodech lokálních extrémů nebo na hranici množiny X .

Globální extrémy

Řekneme, že funkce f dosahuje na množině X v bodě $\mathbf{a} \in X$ svého

1 globálního maxima jestliže $\forall \mathbf{x} \in X$ platí $f(\mathbf{x}) \leq f(\mathbf{a})$

2 globálního minima jestliže $\forall \mathbf{x} \in X$ platí $f(\mathbf{x}) \geq f(\mathbf{a})$

Poznámka: Místo pojmu globální též používáme pojem **absolutní**. Opět definujeme ostré extrémy, jestliže nerovnosti jsou ostré pro $\forall \mathbf{x} \neq \mathbf{a}$.

Weierstrassova věta:

Je-li $X \subset \mathbb{R}^n$ **ohraničená, uzavřená** množina a $f : \mathbb{R}^n \rightarrow \mathbb{R}$ **spojitá** funkce na X , pak má f na X globální extrémy, a to buď v bodech lokálních extrémů nebo na hranici množiny X .

Poznámka: Není-li množina X uzavřená nebo ohraničená, pak globální extrémy nemusí existovat. Pokud extrémy existují, jsou jejich hodnoty určeny jednoznačně. Funkce však může nabývat těchto hodnot obecně ve více bodech. Hranici množiny lze většinou popsat pomocí rovnic. Vyšetřování hranice je úlohou s omezením.

Globální extrémy - příklad

Určete globální extrémy funkce $f(x,y) = (x - 1)^2 + (y - \frac{1}{2})^2$ na obdélníku, který je určen body $A = [0; 0]$; $B = [2; 0]$; $C = [2; 1]$; $D = [0; 1]$.

Globální extrémy - příklad

Určete globální extrémy funkce $f(x,y) = (x - 1)^2 + (y - \frac{1}{2})^2$ na obdélníku, který je určen body $A = [0; 0]$; $B = [2; 0]$; $C = [2; 1]$; $D = [0; 1]$.

Řešení: Nalezneme lokální extrémy funkce f . Spočteme parciální derivace $f'_x = 2x - 2$ a $f'_y = 2y - 1$ a nalezneme stacionární bod $s = [1, \frac{1}{2}]$. Matice druhých derivací je rovna $H(s) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$. Hlavní minory této matice jsou kladné a proto v bodě s nastává lokální minimum funkce f .

Globální extrémy - příklad

Určete globální extrémy funkce $f(x,y) = (x - 1)^2 + (y - \frac{1}{2})^2$ na obdélníku, který je určen body $A = [0; 0]$; $B = [2; 0]$; $C = [2; 1]$; $D = [0; 1]$.

Řešení: Nalezneme lokální extrémy funkce f . Spočteme parciální derivace $f'_x = 2x - 2$ a $f'_y = 2y - 1$ a nalezneme stacionární bod $s = [1, \frac{1}{2}]$. Matice druhých derivací je rovna $H(s) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$. Hlavní minory této matice jsou kladné a proto v bodě s nastává lokální minimum funkce f .

Hranice zadané množiny je tvořena čtyřmi úsečkami AB , BC , CD a DA . Je tedy třeba řešit čtyři optimalizační úlohy s funkcí f a postupně s podmínkami $V_1 : y = 0$, $V_2 : x = 2$, $V_3 : y = 1$ a $V_4 : x = 0$.

Pozor! Při této formulaci je zapotřebí zvlášť vyšetřit body A , B , C , D , protože nehledáme extrémy na celých hraničních přímkách, ale pouze na příslušných úsečkách.

Globální extrémy - příklad

Úlohy optimalizace f za podmínky V_i , kde $i = 1, 2, 3, 4$ převedeme na ekvivalentní úlohy nalezení lokálních extrémů funkcí F_i , kde

$$F_1(x) = f(x, 0) = (x - 1)^2 + \frac{1}{4},$$

$$F_2(y) = f(2, y) = (y - \frac{1}{2})^2 + 1,$$

$$F_3(x) = f(x, 1) = (x - 1)^2 + \frac{1}{4},$$

$$F_4(y) = f(0, y) = (y - \frac{1}{2})^2 + 1.$$

Globální extrémy - příklad

Úlohy optimalizace f za podmínky V_i , kde $i = 1, 2, 3, 4$ převedeme na ekvivalentní úlohy nalezení lokálních extrémů funkcí F_i , kde

$$F_1(x) = f(x, 0) = (x - 1)^2 + \frac{1}{4},$$

$$F_2(y) = f(2, y) = (y - \frac{1}{2})^2 + 1,$$

$$F_3(x) = f(x, 1) = (x - 1)^2 + \frac{1}{4},$$

$$F_4(y) = f(0, y) = (y - \frac{1}{2})^2 + 1.$$

Snadno se zjistí, že jednotlivé úlohy mají minimum v bodech postupně: $a = [1, 0]$, $b = [2, \frac{1}{2}]$, $c = [1, 1]$ a $d = [0, \frac{1}{2}]$. Hodnoty funkce f ve všech podezřelých bodech shrneme v tabulce:

\mathbf{x}	s	A	B	C	D	a	b	c	d
$f(\mathbf{x})$	0	$\frac{5}{4}$	$\frac{5}{4}$	$\frac{5}{4}$	$\frac{5}{4}$	$\frac{1}{4}$	1	$\frac{1}{4}$	1

Je zřejmé, že nejmenší hodnoty dosahuje funkce v bodě $s = [1, \frac{1}{2}]$ a největší v bodech A, B, C, D .

Úloha nelineárního programování (NLP)

Úlohu hledání extrémů funkce n proměnných $f(\mathbf{x})$ na množině M vymezené podmínkami

$$g_1(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0$$

$$g_2(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0$$

⋮

$$g_m(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0,$$

kde alespoň jedna z funkcí f, g_1, \dots, g_m je nelineární, nazveme úlohou **nelineárního programování (NLP)**.

Úloha nelineárního programování (NLP)

Úlohu hledání extrémů funkce n proměnných $f(\mathbf{x})$ na množině M vymezené podmínkami

$$g_1(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0$$

$$g_2(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0$$

⋮

$$g_m(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0,$$

kde alespoň jedna z funkcí f, g_1, \dots, g_m je nelineární, nazveme **úlohou nelineárního programování (NLP)**.

Je-li $m = 0$, tj. nejsou přítomna žádná omezení, hledáme tedy extrémy funkce f na celém \mathbb{R}^n a hovoříme o **volných extrémech**. V případě, kdy jsou na proměnné uvalena omezení, tj. $m > 0$ nebo když je definiční obor funkce f limitován na nějakou podmnožinu $X \subset \mathbb{R}^n$, hovoříme o **vázaných extrémech**.

Úloha nelineárního programování (NLP)

Úlohu hledání extrémů funkce n proměnných $f(\mathbf{x})$ na množině M vymezené podmínkami

$$g_1(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0$$

$$g_2(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0$$

⋮

$$g_m(\mathbf{x}) \leq (\text{resp. } =, \text{ resp. } \geq) 0,$$

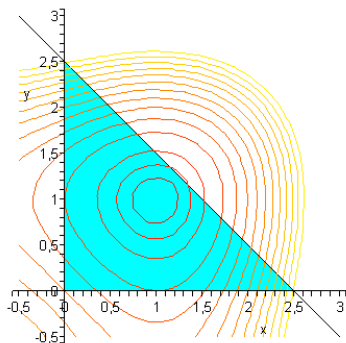
kde alespoň jedna z funkcí f, g_1, \dots, g_m je nelineární, nazveme **úlohou nelineárního programování (NLP)**.

Je-li $m = 0$, tj. nejsou přítomna žádná omezení, hledáme tedy extrémy funkce f na celém \mathbb{R}^n a hovoříme o **volných extrémech**. V případě, kdy jsou na proměnné uvalena omezení, tj. $m > 0$ nebo když je definiční obor funkce f limitován na nějakou podmnožinu $X \subset \mathbb{R}^n$, hovoříme o **vázaných extrémech**.

Poznámka: U úloh NLP nemusí existovat žádné řešení nebo naopak může existovat více extrémů, z nichž některé jsou pouze lokální. Optimum nemusí ležet pouze na hranici přípustné množiny!

Úloha nelineárního programování (NLP) - příklad

Uvažujme úlohu optimalizace funkce $f(x, y) = x^3 - 3x + y^3 - 3y$ na množině M vymezené nerovnostmi $x \geq 0$, $y \geq 0$, $2x + 2y \leq 5$. Problém si můžeme znázornit graficky. Modře je vyznačena přípustná množina, vrstevnice jsou odstupňovány od nejnižší červené po nejvyšší žlutou.



Zřejmě má funkce f na množině M minimum v bodě $[1, 1]$ a maximum v bodech $[0; 2, 5]$ a $[2, 5; 0]$.

Optimalizační úloha s omezením ve formě rovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m rovnic $g_i(\mathbf{x}) = 0, i = 1, \dots, m$.

Jsou - li funkce f i $g_i, i = 1, \dots, m$ spojitě diferencované a jsou - li gradienty omezení ∇g_i lineárně nezávislé vektory (tj. žádné omezení není nadbytečné), pak pro bod optima \mathbf{x}^* existují jednoznačné hodnoty $\lambda_1, \dots, \lambda_m$, takové, že:

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(\mathbf{x}^*) = 0.$$

Čísla $\lambda_1, \dots, \lambda_m$ se nazývají **Lagrangeovy multiplifikátory** a umožňují převedení optimalizační úlohy na řešení systému rovnic.

Optimalizační úloha s omezením ve formě rovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m rovnic $g_i(\mathbf{x}) = 0, i = 1, \dots, m$.

Jsou - li funkce f i $g_i, i = 1, \dots, m$ spojitě diferencované a jsou - li gradienty omezení ∇g_i lineárně nezávislé vektory (tj. žádné omezení není nadbytečné), pak pro bod optima \mathbf{x}^* existují jednoznačné hodnoty $\lambda_1, \dots, \lambda_m$, takové, že:

$$\nabla f(\mathbf{x}^*) + \sum_{i=1}^m \lambda_i \cdot \nabla g_i(\mathbf{x}^*) = 0.$$

Čísla $\lambda_1, \dots, \lambda_m$ se nazývají **Lagrangeovy multiplikátory** a umožňují převedení optimalizační úlohy na řešení systému rovnic. Jaký je formální postup? Vytvoří se tzv. **Lagrangeova funkce**

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot g_i(\mathbf{x})$$

a sestaví se podmínky pro její stacionární body, tzv. **podmínky 1. řádu**:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 0, \quad \nabla_{\lambda} L(\mathbf{x}, \lambda) = 0$$

Jde o systém $n + m$ rovnic pro $n + m$ neznámých (posledních m rovnic vyjadřuje vlastně vazební podmínky). Jestliže má tento systém řešení $(\mathbf{x}^*, \lambda^*)$ a jsou-li f i M konvexní, pak je bod \mathbf{x}^* globálním minimem funkce f na množině M .

Optimalizační úloha s omezením ve formě rovností

Ukažme si metodu Lagrangeových multiplikátorů na následující úloze:
Najděte patu kolmice spuštěné z bodu $[5, 1, 2]$ na rovinu $2x + 3y + z = 6$.

Optimalizační úloha s omezením ve formě rovnosti

Ukažme si metodu Lagrangeových multiplikátorů na následující úloze:
Najděte patu kolmice spuštěné z bodu $[5, 1, 2]$ na rovinu $2x + 3y + z = 6$.

Řešení: Hledáme tedy bod $[x, y, z]$ ležící v zadané rovině, pro nějž je vzdálenost od bodu $[5, 1, 2]$ minimální. Místo minimalizace funkce $f(x, y, z) = \sqrt{(x - 5)^2 + (y - 1)^2 + (z - 2)^2}$ můžeme pro zjednodušení výpočtu minimalizovat její druhou mocninu. Abychom mohli použít Lagrangeův multiplikátor, je třeba rovnici omezení anulovat:
 $2x + 3y + z - 6 = 0$.

Optimalizační úloha s omezením ve formě rovnosti

Ukažme si metodu Lagrangeových multiplikátorů na následující úloze:
Najděte patu kolmice spuštěné z bodu $[5, 1, 2]$ na rovinu $2x + 3y + z = 6$.

Řešení: Hledáme tedy bod $[x, y, z]$ ležící v zadané rovině, pro nějž je vzdálenost od bodu $[5, 1, 2]$ minimální. Místo minimalizace funkce $f(x, y, z) = \sqrt{(x-5)^2 + (y-1)^2 + (z-2)^2}$ můžeme pro zjednodušení výpočtu minimalizovat její druhou mocninu. Abychom mohli použít Lagrangeův multiplikátor, je třeba rovnici omezení anulovat:

$$2x + 3y + z - 6 = 0.$$

Sestavme Lagrangeovu funkci úlohy:

$L(x, y, z, \lambda) = (x-5)^2 + (y-1)^2 + (z-2)^2 + \lambda(2x + 3y + z - 6)$ a určíme její parciální derivace:

$$L_x(x, y, z, \lambda) = 2(x-5) + 2\lambda,$$

$$L_y(x, y, z, \lambda) = 2(y-1) + 3\lambda,$$

$$L_z(x, y, z, \lambda) = 2(z-2) + \lambda,$$

$$L_\lambda(x, y, z, \lambda) = 2x + 3y + z - 6.$$

Optimalizační úloha s omezením ve formě rovnosti

Ukažme si metodu Lagrangeových multiplikátorů na následující úloze:
Najděte patu kolmice spuštěné z bodu $[5, 1, 2]$ na rovinu $2x + 3y + z = 6$.

Řešení: Hledáme tedy bod $[x, y, z]$ ležící v zadané rovině, pro nějž je vzdálenost od bodu $[5, 1, 2]$ minimální. Místo minimalizace funkce $f(x, y, z) = \sqrt{(x-5)^2 + (y-1)^2 + (z-2)^2}$ můžeme pro zjednodušení výpočtu minimalizovat její druhou mocninu. Abychom mohli použít Lagrangeův multiplikátor, je třeba rovnici omezení anulovat:
 $2x + 3y + z - 6 = 0$.

Sestavme Lagrangeovu funkci úlohy:

$L(x, y, z, \lambda) = (x-5)^2 + (y-1)^2 + (z-2)^2 + \lambda(2x + 3y + z - 6)$ a určíme její parciální derivace:

$$L_x(x, y, z, \lambda) = 2(x-5) + 2\lambda,$$

$$L_y(x, y, z, \lambda) = 2(y-1) + 3\lambda,$$

$$L_z(x, y, z, \lambda) = 2(z-2) + \lambda,$$

$$L_\lambda(x, y, z, \lambda) = 2x + 3y + z - 6.$$

Položíme parciální rovnice rovny nule a dostaneme lineární systém, jehož vyřešením získáme bod optima $[\frac{52}{14}, \frac{-13}{14}, \frac{19}{14}]$. Protože minimalizovaná funkce i přípustná množina jsou konvexní, našli jsme bod minima.

Ekonomická interpretace

Příklad z M. W. Klein: Mathematical Methods for Economics:

Uvažujme úlohu hledání optima spotřebitele, který má 6 dolarů, které chce utratit za oběd v samoobslužné restauraci, kde se polévka i hlavní chod platí na váhu, přičemž cena za 1 unci polévky je \$0,25 a za 1 unci hlavního chodu je \$0,5. Označíme-li S a V zakoupené množství obou chodů (v uncích), pak rozpočtové omezení můžeme zapsat jako $6 = \frac{S}{4} + \frac{V}{2}$. Hledáme takovou kombinaci jídel splňující tuto podmínku, která maximalizuje užitkovou funkci $U(S, V) = \frac{\ln(S)}{2} + \frac{\ln(V)}{2}$.

Ekonomická interpretace

Příklad z M. W. Klein: Mathematical Methods for Economics:

Uvažujme úlohu hledání optima spotřebitele, který má 6 dolarů, které chce utratit za oběd v samoobslužné restauraci, kde se polévka i hlavní chod platí na váhu, přičemž cena za 1 unci polévky je \$0,25 a za 1 unci hlavního chodu je \$0,5. Označíme-li S a V zakoupené množství obou chodů (v uncích), pak rozpočtové omezení můžeme zapsat jako $6 = \frac{S}{4} + \frac{V}{2}$. Hledáme takovou kombinaci jídel splňující tuto podmínku, která maximalizuje užitkovou funkci $U(S, V) = \frac{\ln(S)}{2} + \frac{\ln(V)}{2}$.

Řešení: Zavedeme Lagrangeovu funkci

$L(S, V, \lambda) = \frac{\ln(S)}{2} + \frac{\ln(V)}{2} - \lambda\left(\frac{S}{4} + \frac{V}{2} - 6\right)$. Podmínky prvního řádu jsou:

$$L_S(S, V, \lambda) = \frac{1}{2S} - \frac{\lambda}{4} = 0$$

$$L_V(S, V, \lambda) = \frac{1}{2V} - \frac{\lambda}{2} = 0$$

$$L_\lambda(S, V, \lambda) = \frac{S}{4} + \frac{V}{2} - 6 = 0$$

Ekonomická interpretace

Příklad z M. W. Klein: Mathematical Methods for Economics:

Uvažujme úlohu hledání optima spotřebitele, který má 6 dolarů, které chce utratit za oběd v samoobslužné restauraci, kde se polévka i hlavní chod platí na váhu, přičemž cena za 1 unci polévky je \$0,25 a za 1 unci hlavního chodu je \$0,5. Označíme-li S a V zakoupené množství obou chodů (v uncích), pak rozpočtové omezení můžeme zapsat jako $6 = \frac{S}{4} + \frac{V}{2}$. Hledáme takovou kombinaci jídel splňující tuto podmínku, která maximalizuje užitkovou funkci $U(S, V) = \frac{\ln(S)}{2} + \frac{\ln(V)}{2}$.

Řešení: Zavedeme Lagrangeovu funkci

$L(S, V, \lambda) = \frac{\ln(S)}{2} + \frac{\ln(V)}{2} - \lambda\left(\frac{S}{4} + \frac{V}{2} - 6\right)$. Podmínky prvního řádu jsou:

$$L_S(S, V, \lambda) = \frac{1}{2S} - \frac{\lambda}{4} = 0$$

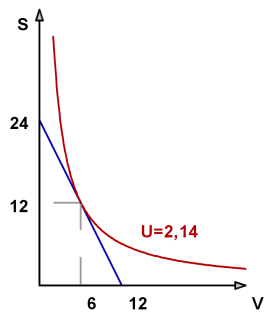
$$L_V(S, V, \lambda) = \frac{1}{2V} - \frac{\lambda}{2} = 0$$

$$L_\lambda(S, V, \lambda) = \frac{S}{4} + \frac{V}{2} - 6 = 0$$

Z prvních dvou rovnic lze vyjádřit, že $\lambda = \frac{2}{S} = \frac{1}{V}$, neboli $S = 2V$. Spolu s poslední podmínkou dostaneme řešení $S^* = 12$, $V^* = 6$ uncí, $\lambda^* = \frac{1}{6}$, což dává užitek $U(12, 6) = 2,14$.

Ekonomická interpretace

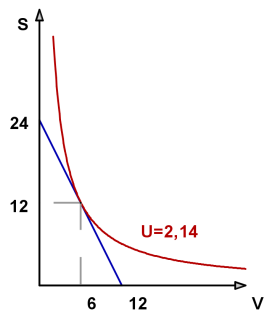
Znáznorněme si uvažovanou úlohu graficky.



Sklon rozpočtové linie je dán vztahem dán poměrem cen: $\frac{dS}{dV} = -2$.

Ekonomická interpretace

Znáznorněme si uvažovanou úlohu graficky.

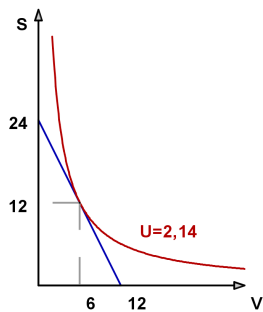


Sklon rozpočtové linie je dán vztahem dán poměrem cen: $\frac{dS}{dV} = -2$.

Sklon indiferenční křivky je dán poměrem mezních užiteků (mezní míra substituce): $\frac{dS}{dV} = -\frac{U_V}{U_S}$.

Ekonomická interpretace

Znáznorněme si uvažovanou úlohu graficky.



Sklon rozpočtové linie je dán vztahem dán poměrem cen: $\frac{dS}{dV} = -2$.

Sklon indifferenční křivky je dán poměrem mezních užitek (mezní míra substituce): $\frac{dS}{dV} = -\frac{U_V}{U_S}$.

Spočteme mezní užitky $U_S(S, V) = \frac{1}{2S}$, $U_V(S, V) = \frac{1}{2V}$, jejich podíl je $\frac{U_V}{U_S} = \frac{S}{V}$. Protože v bodě optima (S^*, V^*) je sklon rozpočtové linie a indifferenční křivky shodný, platí: $-2 = -\frac{S^*}{V^*}$, neboli $\frac{S^*}{4} = \frac{V^*}{2}$ (rovnováha spotřebitele).

Ekonomická interpretace

Získáme-li dodatečný dolar, budeme-li tedy chtít utratit \$7, u nového optimálního řešení bude zachován poměr spotřebovaných jídel:

$S^* = 14$, $V^* = 7$ a hodnota Lagrangeova multiplikátoru poklesne na $\frac{1}{7}$.

Získaný užitek bude $U(14, 7) = 2,29$. Přírůstek užitku je tedy

$\Delta U = U(14, 7) - U(12, 6) = 2,29 - 2,14 = 0,15 = \left(\frac{1}{6,6}\right) \cdot 1$. Zřejmě jeden dodatečný dolar vyvolal $\left(\frac{1}{6,6}\right)$ -násobný přírůstek užitku, což je něco mezi $\frac{1}{7}$ a

$\frac{1}{6}$. Můžeme říct, že optimální hodnota Lagrangeova multiplikátoru vyjadřuje **mezní užitek peněz** určených k útratě.

Ekonomická interpretace

Získáme-li dodatečný dolar, budeme-li tedy chtít utratit \$7, u nového optimálního řešení bude zachován poměr spotřebovaných jídel:

$S^* = 14$, $V^* = 7$ a hodnota Lagrangeova multiplikátoru poklesne na $\frac{1}{7}$.

Získaný užitek bude $U(14, 7) = 2,29$. Přírůstek užitku je tedy

$\Delta U = U(14, 7) - U(12, 6) = 2,29 - 2,14 = 0,15 = \left(\frac{1}{6,6}\right) \cdot 1$. Zřejmě jeden dodatečný dolar vyvolal $\left(\frac{1}{6,6}\right)$ -násobný přírůstek užitku, což je něco mezi $\frac{1}{7}$ a

$\frac{1}{6}$. Můžeme říct, že optimální hodnota Lagrangeova multiplikátoru vyjadřuje **mezní užitek peněz** určených k útratě.

Tento vztah lze ukázat, vyjádříme-li si optimální množství jídel při rozpočtovém omezení B : $S^* = 2B$, $V^* = B$ a $\lambda^* = \frac{1}{B}$. Optimální užitek při rozpočtu B je po úpravě: $f(B) = U(2B, B) = \frac{1}{2} \ln(2) + \ln(B)$. Derivováním podle B získáme jeho mezní užitek $f'(B) = \frac{1}{B}$, což je rovno λ^* .

Ekonomická interpretace

Získáme-li dodatečný dolar, budeme-li tedy chtít utratit \$7, u nového optimálního řešení bude zachován poměr spotřebovaných jídel:

$S^* = 14$, $V^* = 7$ a hodnota Lagrangeova multiplikátoru poklesne na $\frac{1}{7}$.

Získaný užitek bude $U(14, 7) = 2,29$. Přírůstek užitku je tedy

$\Delta U = U(14, 7) - U(12, 6) = 2,29 - 2,14 = 0,15 = \left(\frac{1}{6,6}\right) \cdot 1$. Zřejmě jeden dodatečný dolar vyvolal $\left(\frac{1}{6,6}\right)$ -násobný přírůstek užitku, což je něco mezi $\frac{1}{7}$ a

$\frac{1}{6}$. Můžeme říct, že optimální hodnota Lagrangeova multiplikátoru vyjadřuje **mezní užitek peněz** určených k útratě.

Tento vztah lze ukázat, vyjádříme-li si optimální množství jídel při rozpočtovém omezení B : $S^* = 2B$, $V^* = B$ a $\lambda^* = \frac{1}{B}$. Optimální užitek při rozpočtu B je po úpravě: $f(B) = U(2B, B) = \frac{1}{2} \ln(2) + \ln(B)$. Derivováním podle B získáme jeho mezní užitek $f'(B) = \frac{1}{B}$, což je rovno λ^* .

Podobně v úlohách optimalizace výrobního programu, budou Lagrangeovy multiplikátory u omezení pro jednotlivé vstupy vyjadřovat mezní užitek při zvýšení jejich kapacity. Tedy jsou rovny ceně, kterou je výrobce ochoten za dodatečnou jednotku vstupu zaplatit, proto se pro ně používá název **stínové ceny**.

Lagrangeova funkce - postačující podmínky pro optimum

Pro odvození podmínek 2. řádu uvažujme kvadratickou funkci $Q(x, y) = ax^2 + 2bxy + cy^2$, kterou lze též zapsat maticově jako

$$Q(x, y) = (x, y) \cdot \begin{pmatrix} a & b \\ b & c \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}. \text{ Předpokládejme, že hledáme optimum}$$

této funkce na množině dané lineární rovnicí $Ax + By = 0$.

Lagrangeova funkce - postačující podmínky pro optimum

Pro odvození podmínek 2. řádu uvažujme kvadratickou funkci $Q(x, y) = ax^2 + 2bxy + cy^2$, kterou lze též zapsat maticově jako

$$Q(x, y) = (x, y) \cdot \begin{pmatrix} a & b \\ b & c \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}. \text{ Předpokládejme, že hledáme optimum}$$

této funkce na množině dané lineární rovnicí $Ax + By = 0$.

Vyjádříme-li $y = -\frac{A}{B}x$ a dosadíme-li do Q , dostaneme

$$f(x) = Q(x, -\frac{A}{B}x) = ax^2 + 2bx(-\frac{A}{B}x) + c(-\frac{A}{B}x)^2 = -\frac{x^2}{B^2}[2bAB - aB^2 - cA^2].$$

Výraz uvnitř hranaté závorky lze také zapsat jako determinant matice

$$H = \begin{pmatrix} 0 & A & B \\ A & a & b \\ B & b & c \end{pmatrix}, \text{ což je vlastně Hessova matice Lagrangeovy funkce}$$

$L(\lambda, x, y) = \lambda(Ax + By) + Q(x, y)$. Pokud je **$\det(H)$ záporný**, je funkce $f(x)$ konvexní, což je postačující podmínka pro existenci **minima** ve stacionárním bodě (a je-li kladný, je $f(x)$ konkávní - podmínka pro maximum).

Lagrangeova funkce - postačující podmínky pro optimum

Zobecníme-li odvození z předchozího slajdu pro funkci dvou proměnných $f(x, y)$ optimalizovanou na množině dané rovnicí $g(x, y) = c$, dostaneme **podmínky 2. řádu**: Označme $H(\lambda, x, y)$ Hessovu matici Lagrangeovy funkce $L(\lambda, x, y) = \lambda(g(x, y) - c) + f(x, y)$. Pak je-li ve stacionárním bodě determinant $\det(H(\lambda, x, y))$

- **kladný**, je stacionární bod bodem **maxima**
- **záporný**, je stacionární bod bodem **minima**

Lagrangeova funkce - postačující podmínky pro optimum

Zobecníme-li odvození z předchozího slajdu pro funkci dvou proměnných $f(x, y)$ optimalizovanou na množině dané rovnicí $g(x, y) = c$, dostaneme **podmínky 2. řádu**: Označme $H(\lambda, x, y)$ Hessovu matici Lagrangeovy funkce $L(\lambda, x, y) = \lambda(g(x, y) - c) + f(x, y)$. Pak je-li ve stacionárním bodě determinant $\det(H(\lambda, x, y))$

- **kladný**, je stacionární bod bodem **maxima**
- **záporný**, je stacionární bod bodem **minima**

Příklad: Pro úlohu zákazníka restaurace má Lagrangeova funkce

$$L(\lambda, S, V) = \frac{\ln(S)}{2} + \frac{\ln(V)}{2} + \lambda\left(\frac{S}{4} + \frac{V}{2} - 6\right)$$
 Hessovu matici

$$H = \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{4} & -\frac{1}{2S^2} & 0 \\ \frac{1}{2} & 0 & -\frac{1}{2V^2} \end{pmatrix}. \text{ Její determinant je } \det(H(\lambda, S, V)) = \frac{1}{8S^2} + \frac{1}{32V^2},$$

což je kladné číslo pro jakékoliv S, V , takže ve stacionárním bodě nastává maximum.

Lagrangeova funkce - postačující podmínky pro optimum

Zobecněme **podmínky 2. řádu** dále pro funkci n proměnných $f(x_1, \dots, x_n)$ optimalizovanou na množině dané soustavou rovnic

$$g_i(x_1, \dots, x_n) = c_i, \quad i = 1, \dots, m:$$

Označme $H(\lambda_1, \dots, \lambda_m, x_1, \dots, x_n)$ Hessovu matici Lagrangeovy funkce $L(\lambda_1, \dots, \lambda_m, x_1, \dots, x_n)$, pak pro její hodnotu v případném stacionárním bodě platí:

- 1 má - li determinant $H(\lambda_1, \dots, \lambda_m, x_1, \dots, x_n)$ znaménko $(-1)^n$ a **všech $n - m$ jejích největších hlavních minorů střídá znaménka**, pak ve stacionárním bodě nastává **maximum**.
- 2 má - li **všech $n - m$ největších hlavních minorů** včetně determinantu $H(\lambda_1, \dots, \lambda_m, x_1, \dots, x_n)$ **znaménko $(-1)^m$** , pak ve stacionárním bodě nastává **minimum**.
- 3 Je-li $n - m$ největších hlavních minorů nenulových a přitom neplatí ani jedna z výše uvedených podmínek, pak ve stacionárním bodě není extrém.

Lagrangeova funkce - postačující podmínky pro optimum, příklad

Uvažujme modifikaci úlohy zákazníka restaurace, kdy je možné konzumovat ještě džus v ceně 1 dolar za 12 uncí, přitom novou funkci užitku vyjádříme jako $U(S, V, J) = \frac{1}{3}\ln(S) + \frac{1}{3}\ln(V) + \frac{1}{3}\ln(J)$ a rozpočet zůstává \$6.

Lagrangeova funkce - postačující podmínky pro optimum, příklad

Uvažujme modifikaci úlohy zákazníka restaurace, kdy je možné konzumovat ještě džus v ceně 1 dolar za 12 uncí, přitom novou funkci užítka vyjádříme jako $U(S, V, J) = \frac{1}{3}\ln(S) + \frac{1}{3}\ln(V) + \frac{1}{3}\ln(J)$ a rozpočet zůstává \$6.

Lagrangeova funkce úlohy bude

$L(\lambda, S, V, J) = \frac{1}{3}\ln(S) + \frac{1}{3}\ln(V) + \frac{1}{3}\ln(J) + \lambda(\frac{S}{4} + \frac{V}{2} + \frac{J}{12} - 6)$. Z podmínek **prvního řádu** pro proměnné odvodíme $S = 2V$, $J = 6V$ a po dosazení do rozpočtového omezení vyjde $S^* = 8$, $V^* = 4$ a $J^* = 24$ uncí.

Lagrangeova funkce - postačující podmínky pro optimum, příklad

Uvažujme modifikaci úlohy zákazníka restaurace, kdy je možné konzumovat ještě džus v ceně 1 dolar za 12 uncí, přitom novou funkci užítka vyjádříme jako $U(S, V, J) = \frac{1}{3} \ln(S) + \frac{1}{3} \ln(V) + \frac{1}{3} \ln(J)$ a rozpočet zůstává \$6.

Lagrangeova funkce úlohy bude

$L(\lambda, S, V, J) = \frac{1}{3} \ln(S) + \frac{1}{3} \ln(V) + \frac{1}{3} \ln(J) + \lambda(\frac{S}{4} + \frac{V}{2} + \frac{J}{12} - 6)$. Z podmínek **prvního řádu** pro proměnné odvodíme $S = 2V$, $J = 6V$ a po dosazení do rozpočtového omezení vyjde $S^* = 8$, $V^* = 4$ a $J^* = 24$ uncí. Hessova matice

Lagrangeovy funkce je: $H(\lambda, S, V, J) = \begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{12} \\ \frac{1}{4} & -\frac{1}{3S^2} & 0 & 0 \\ \frac{1}{2} & 0 & -\frac{1}{3V^2} & 0 \\ \frac{1}{12} & 0 & 0 & -\frac{1}{3J^2} \end{pmatrix}$.

Podmínka **druhého řádu** pro případ $n = 3$, $m = 1$ nabývá podoby: "je-li znaménko $\det(H)$ rovno $(-1)^3$ a znaménko druhého hlavního minoru $-(-1)^3$, pak ve stacionárním bodě nastává maximum" Pro naši funkci je $\det(H) = -(\frac{1}{1296S^2V^2} + \frac{1}{144J^2V^2} + \frac{1}{36S^2J^2})$, což je záporné v každém bodě a přitom determinant 2.hlavního minoru je $\frac{1}{12S^2} + \frac{1}{48V^2}$, což je kladné v každém bodě, takže ve stacionárním bodě nastává maximum.

Lagrangeova funkce - postačující podmínky pro optimum, příklad

Uvažujme další modifikaci, kdy kromě rozpočtového omezení uvažujeme ještě podmínku na celkový objem tekutin $S + J = 24$.

Lagrangeova funkce - postačující podmínky pro optimum, příklad

Uvažujme další modifikaci, kdy kromě rozpočtového omezení uvažujeme ještě podmínku na celkový objem tekutin $S + J = 24$.

Lagrangeova funkce úlohy bude

$$L(\lambda, \mu, S, V, J) = \frac{1}{3} \ln(S) + \frac{1}{3} \ln(V) + \frac{1}{3} \ln(J) + \lambda \left(\frac{S}{4} + \frac{V}{2} + \frac{J}{12} - 6 \right) + \mu(S + J - 24).$$

Z podmínek **prvního řádu** pro proměnné odvodíme $V = \frac{SJ}{3(J-S)}$ a po dosazení do soustavy omezení vyjde $S^* = 8$, $V^* = \frac{16}{3}$ a $J^* = 16$ uncí.

Lagrangeova funkce - postačující podmínky pro optimum, příklad

Uvažujme další modifikaci, kdy kromě rozpočtového omezení uvažujeme ještě podmínku na celkový objem tekutin $S + J = 24$.

Lagrangeova funkce úlohy bude

$$L(\lambda, \mu, S, V, J) = \frac{1}{3} \ln(S) + \frac{1}{3} \ln(V) + \frac{1}{3} \ln(J) + \lambda \left(\frac{S}{4} + \frac{V}{2} + \frac{J}{12} - 6 \right) + \mu (S + J - 24).$$

Z podmínek **prvního řádu** pro proměnné odvodíme $V = \frac{SJ}{3(J-S)}$ a po dosazení do soustavy omezení vyjde $S^* = 8$, $V^* = \frac{16}{3}$ a $J^* = 16$ uncí. Hessova matice

$$\text{Lagrangeovy funkce je: } H(\lambda, \mu, S, V, J) = \begin{pmatrix} 0 & 0 & \frac{1}{4} & \frac{1}{2} & \frac{1}{12} \\ 0 & 0 & 1 & 0 & 1 \\ \frac{1}{4} & 1 & -\frac{1}{3S^2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 & -\frac{1}{3V^2} & 0 \\ \frac{1}{12} & 1 & 0 & 0 & -\frac{1}{3J^2} \end{pmatrix}.$$

Podmínka **druhého řádu** pro případ $n = 3$, $m = 2$ nabývá podoby: "je-li znaménko $\det(H)$ rovno $(-1)^3$, pak ve stacionárním bodě nastává maximum a je-li jeho znaménko rovno $(-1)^2$, pak se jedná o minimum." Lze vypočítat, že pro naši funkci je $\det(H) = -\left(\frac{1}{12S^2} + \frac{1}{12J^2} + \frac{1}{108V^2}\right)$, což je záporné v každém bodě, takže ve stacionárním bodě nastává maximum.

Optimalizační úloha s omezením ve formě nerovnosti - motivační příklad

Uvažujme jednoduchou optimalizační úlohu

$$f(x, y) = x^2 - 2x + 2y^2 - y + 3 \rightarrow \min \text{ za podmínky } x + y \leq 1.$$

Můžeme problém převést na úlohu s omezením ve tvaru rovnosti?

Optimalizační úloha s omezením ve formě nerovnosti - motivační příklad

Uvažujme jednoduchou optimalizační úlohu

$$f(x, y) = x^2 - 2x + 2y^2 - y + 3 \rightarrow \min \text{ za podmínky } x + y \leq 1.$$

Můžeme problém převést na úlohu s omezením ve tvaru rovnosti?

Přidáme na levou stranu omezující podmínky doplňkovou proměnnou:

Optimalizační úloha s omezením ve formě nerovnosti - motivační příklad

Uvažujme jednoduchou optimalizační úlohu

$$f(x, y) = x^2 - 2x + 2y^2 - y + 3 \rightarrow \min \text{ za podmínky } x + y \leq 1.$$

Můžeme problém převést na úlohu s omezením ve tvaru rovnosti?

Přidáme na levou stranu omezující podmínky doplňkovou proměnnou:

$x + y + w^2 = 1$ a vytvoříme Lagrangeovu funkci

$$L(x, y, w, \lambda) = x^2 - 2x + y^2 - 2y + 3 + \lambda \cdot (x + y + w^2 - 1)$$

Optimalizační úloha s omezením ve formě nerovnosti - motivační příklad

Uvažujme jednoduchou optimalizační úlohu

$$f(x, y) = x^2 - 2x + 2y^2 - y + 3 \rightarrow \min \text{ za podmínky } x + y \leq 1.$$

Můžeme problém převést na úlohu s omezením ve tvaru rovnosti?

Přidáme na levou stranu omezující podmínky doplňkovou proměnnou:

$x + y + w^2 = 1$ a vytvoříme Lagrangeovu funkci

$$L(x, y, w, \lambda) = x^2 - 2x + y^2 - 2y + 3 + \lambda \cdot (x + y + w^2 - 1)$$

Kromě podmínek $2x - 2 + \lambda = 0$, $2y - 2 + \lambda = 0$ a $x + y + w^2 - 1 = 0$

dostaneme derivováním podle w ještě $2w\lambda = 0$.

Je zřejmé, že možnost $\lambda = 0$ nedává žádné řešení, protože z prvních dvou rovnic dostaneme $x = 1$, $y = 1$ a pak třetí podmínka nemá řešení pro w . Musí tedy být $w = 0$, tedy $x + y = 1$, pak sečtením prvních dvou rovnic

$$2(x + y) - 4 + 2\lambda = 0, \text{ takže } \lambda = 2 - (x + y) = 1 \text{ a } x = y = 1 - \frac{\lambda}{2} = \frac{1}{2}.$$

Optimalizační úloha s omezením ve formě nerovností - motivační příklad

Uvažujme jednoduchou optimalizační úlohu

$$f(x, y) = x^2 - 2x + 2y^2 - y + 3 \rightarrow \min \text{ za podmínky } x + y \leq 1.$$

Můžeme problém převést na úlohu s omezením ve tvaru rovností?

Přidáme na levou stranu omezující podmínky doplňkovou proměnnou:

$x + y + w^2 = 1$ a vytvoříme Lagrangeovu funkci

$$L(x, y, w, \lambda) = x^2 - 2x + y^2 - 2y + 3 + \lambda \cdot (x + y + w^2 - 1)$$

Kromě podmínek $2x - 2 + \lambda = 0$, $2y - 2 + \lambda = 0$ a $x + y + w^2 - 1 = 0$

dostaneme derivováním podle w ještě $2w\lambda = 0$.

Je zřejmé, že možnost $\lambda = 0$ nedává žádné řešení, protože z prvních dvou rovnic dostaneme $x = 1$, $y = 1$ a pak třetí podmínka nemá řešení pro w . Musí tedy být $w = 0$, tedy $x + y = 1$, pak sečtením prvních dvou rovnic

$$2(x + y) - 4 + 2\lambda = 0, \text{ takže } \lambda = 2 - (x + y) = 1 \text{ a } x = y = 1 - \frac{\lambda}{2} = \frac{1}{2}.$$

Rovnice $2w\lambda = 0$ říká, že extrém leží buď na hranici ($w = 0$) nebo ve stacionárním bodě ($\lambda = 0$). Kdybychom zvětšili omezující konstantu o δ , dostaneme $x + y \leq 1 + \delta$. Je-li omezení aktivní (tj. $w = 0$), musí být $\delta \leq 1$ a tedy zřejmě $\lambda = 2 - (x + y) = 1 - \delta \geq 0$. Shrňme podmínky pro obecnou úlohu.

Optimalizační úloha s omezením ve formě nerovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i, i = 1, \dots, m.$

Formální postup je obdobný případu s omezujícími rovnostmi: vytvoří se opět **Lagrangeova funkce** $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a sestaví se podmínky 1. řádu:

Optimalizační úloha s omezením ve formě nerovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i, i = 1, \dots, m.$

Formální postup je obdobný případu s omezujícími rovnostmi: vytvoří se opět **Lagrangeova funkce** $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a sestaví se podmínky 1. řádu:

$$\nabla_x L(\mathbf{x}, \lambda) = 0, \quad (\text{stacionarita})$$

Optimalizační úloha s omezením ve formě nerovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i, i = 1, \dots, m$.

Formální postup je obdobný případu s omezujícími rovnostmi: vytvoří se opět **Lagrangeova funkce** $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a sestaví se podmínky 1. řádu:

$$\nabla_x L(\mathbf{x}, \lambda) = 0, \quad (\text{stacionarita})$$

$$g_i(\mathbf{x}) \leq c_i, \quad i = 1, \dots, m \quad (\text{primární přípustnost})$$

Optimalizační úloha s omezením ve formě nerovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i, i = 1, \dots, m$.

Formální postup je obdobný případu s omezujícími rovnostmi: vytvoří se opět **Lagrangeova funkce** $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a sestaví se podmínky 1. řádu:

$$\nabla_x L(\mathbf{x}, \lambda) = 0, \quad (\text{stacionarita})$$

$$g_i(\mathbf{x}) \leq c_i, \quad i = 1, \dots, m \quad (\text{primární přípustnost})$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m \quad (\text{duální přípustnost})$$

Optimalizační úloha s omezením ve formě nerovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i, i = 1, \dots, m$.

Formální postup je obdobný případu s omezujícími rovnostmi: vytvoří se opět **Lagrangeova funkce** $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a sestaví se podmínky 1. řádu:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 0, \quad (\text{stacionarita})$$

$$g_i(\mathbf{x}) \leq c_i, \quad i = 1, \dots, m \quad (\text{primární přípustnost})$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m \quad (\text{duální přípustnost})$$

$$\lambda_i \cdot (g_i(\mathbf{x}) - c_i) = 0, \quad i = 1, \dots, m \quad (\text{komplementarita})$$

Optimalizační úloha s omezením ve formě nerovností

Uvažujme úlohu na vázaný extrém $f(\mathbf{x}) \rightarrow \min$,

na množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i, i = 1, \dots, m$.

Formální postup je obdobný případu s omezujícími rovnostmi: vytvoří se opět **Lagrangeova funkce** $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a sestaví se podmínky 1. řádu:

$$\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 0, \quad (\text{stacionarita})$$

$$g_i(\mathbf{x}) \leq c_i, \quad i = 1, \dots, m \quad (\text{primární přípustnost})$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m \quad (\text{duální přípustnost})$$

$$\lambda_i \cdot (g_i(\mathbf{x}) - c_i) = 0, \quad i = 1, \dots, m \quad (\text{komplementarita})$$

Tyto vztahy se nazývají **Kuhn-Tuckerovy podmínky** úlohy. Za určitých předpokladů **regularity** (nezávislost gradientů omezení) se jedná o **nutné** podmínky pro to, aby v bodě \mathbf{x}^* měla úloha **lokální minimum**. V tomto kontextu se též používá označení **Karush-Kuhn-Tuckerovy podmínky**, zkráceně KKT-podmínky.

Optimalizační úloha s omezením ve formě nerovností

Interpretace KKT podmínek vychází z ekonomického významu Lagrangeových multiplikátorů.

- **Stacionarita a primární přípustnost** jsou přirozené požadavky na optimalitu řešení úlohy s omezením.

Optimalizační úloha s omezením ve formě nerovností

Interpretace KKT podmínek vychází z ekonomického významu Lagrangeových multiplikátorů.

- **Stacionarita a primární přípustnost** jsou přirozené požadavky na optimalitu řešení úlohy s omezením.
- Z čeho plyne požadavek **duální přípustnosti**? Jestliže Lagrangeův multiplikátor λ_i reprezentuje mezní užitek při uvolnění i - tého omezení, pak nerovnost $\lambda_i \geq 0$ znamená, že optimální hodnota účelové funkce se nemůže uvolněním tohoto omezení zhoršit.

Optimalizační úloha s omezením ve formě nerovnosti

Interpretace KKT podmínek vychází z ekonomického významu Lagrangeových multiplikátorů.

- **Stacionarita a primární přípustnost** jsou přirozené požadavky na optimalitu řešení úlohy s omezením.
- Z čeho plyne požadavek **duální přípustnosti**? Jestliže Lagrangeův multiplikátor λ_i reprezentuje mezní užitek při uvolnění i - tého omezení, pak nerovnost $\lambda_i \geq 0$ znamená, že optimální hodnota účelové funkce se nemůže uvolněním tohoto omezení zhoršit.
- Poslední požadavek **komplementarity** lze rozepsat jako:
 $\lambda_i = 0$ nebo $g_i(\mathbf{x}) = c_i$ (případně mohou platit obě rovnosti)

To znamená, že je-li příslušné omezení v bodě optima **neaktivní** (tj. není splněné jako rovnost), pak jeho multiplikátor musí být **nulový**, jinak by šla hodnota účelové funkce zlepšit uvolněním omezení. U aktivních omezení může být obecně i kladná hodnota multiplikátoru.

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené
nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Zapišme KKT podmínky pro minimum:

$$L'_x = 1 - x + \lambda x = 0, L'_y = 2y + 2\lambda y - \mu \quad (\text{stacionarita})$$

$$\frac{x^2}{2} + y^2 \leq \frac{9}{8}, y \geq 0 \quad (\text{primární přípustnost})$$

$$\lambda \geq 0, \mu \geq 0 \quad (\text{duální přípustnost})$$

$$\lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) = 0, \mu y = 0 \quad (\text{komplementarita})$$

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Zapišme KKT podmínky pro minimum:

$$L'_x = 1 - x + \lambda x = 0, L'_y = 2y + 2\lambda y - \mu \quad (\text{stacionarita})$$

$$\frac{x^2}{2} + y^2 \leq \frac{9}{8}, y \geq 0 \quad (\text{primární přípustnost})$$

$$\lambda \geq 0, \mu \geq 0 \quad (\text{duální přípustnost})$$

$$\lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) = 0, \mu y = 0 \quad (\text{komplementarita})$$

Výpočet provedeme zvlášť pro všechny kombinace (ne)nulovosti multiplikátorů λ a μ :

$\lambda = \mu = 0$, pak z podmínek stacionarity dostaneme $x = 1$, $y = 0$, účelová funkce zde má hodnotu $f(1, 0) = \frac{1}{2}$.

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Zapišme KKT podmínky pro minimum:

$$L'_x = 1 - x + \lambda x = 0, L'_y = 2y + 2\lambda y - \mu \quad (\text{stacionarita})$$

$$\frac{x^2}{2} + y^2 \leq \frac{9}{8}, y \geq 0 \quad (\text{primární přípustnost})$$

$$\lambda \geq 0, \mu \geq 0 \quad (\text{duální přípustnost})$$

$$\lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) = 0, \mu y = 0 \quad (\text{komplementarita})$$

Výpočet provedeme zvlášť pro všechny kombinace (ne)nulovosti multiplikátorů λ a μ :

$\lambda = 0, \mu > 0$, pak opět stacionarita dává $x = 1$ a z podmínek komplementarity musí být $y = 0$. Dostali jsme opět bod $[1, 0]$.

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Zapišme KKT podmínky pro minimum:

$$L'_x = 1 - x + \lambda x = 0, \quad L'_y = 2y + 2\lambda y - \mu \quad (\text{stacionarita})$$

$$\frac{x^2}{2} + y^2 \leq \frac{9}{8}, \quad y \geq 0 \quad (\text{primární přípustnost})$$

$$\lambda \geq 0, \quad \mu \geq 0 \quad (\text{duální přípustnost})$$

$$\lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) = 0, \quad \mu y = 0 \quad (\text{komplementarita})$$

Výpočet provedeme zvlášť pro všechny kombinace (ne)nulovosti multiplikátorů λ a μ :

$\lambda > 0, \mu = 0$, pak z podmínek stacionarity dostaneme $2y(1 + \lambda) = 0$, tedy buď $y = 0$ nebo $\lambda = -1$, což nelze. Druhé souřadnice dopočítáme ze vztahu $\frac{x^2}{2} + 0^2 = \frac{9}{8}$, získáme tedy body $[-\frac{3}{2}, 0]$, $[\frac{3}{2}, 0]$, hodnoty λ jsou $\frac{5}{3}$, resp. $\frac{1}{3}$.

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Zapišme KKT podmínky pro minimum:

$$L'_x = 1 - x + \lambda x = 0, L'_y = 2y + 2\lambda y - \mu \quad (\text{stacionarita})$$

$$\frac{x^2}{2} + y^2 \leq \frac{9}{8}, y \geq 0 \quad (\text{primární přípustnost})$$

$$\lambda \geq 0, \mu \geq 0 \quad (\text{duální přípustnost})$$

$$\lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) = 0, \mu y = 0 \quad (\text{komplementarita})$$

Výpočet provedeme zvlášť pro všechny kombinace (ne)nulovosti multiplikátorů λ a μ :

$\lambda > 0, \mu > 0$ nedává žádný nový bod, protože kvůli komplementaritě je $y = 0$ a současně $\frac{x^2}{2} + y^2 = \frac{9}{8}$, což se shoduje s předchozím případem.

Optimalizační úloha s omezením ve formě nerovností - příklad

Najděte minimum funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ na "půlelipse" vymezené nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$.

Řešení: druhé omezení přepíšeme do tvaru $-y \leq 0$ a vytvoříme Lagrangeovu funkci $L = x - \frac{x^2}{2} + y^2 + \lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) + \mu(-y)$

Zapišme KKT podmínky pro minimum:

$$L'_x = 1 - x + \lambda x = 0, L'_y = 2y + 2\lambda y - \mu \quad (\text{stacionarita})$$

$$\frac{x^2}{2} + y^2 \leq \frac{9}{8}, y \geq 0 \quad (\text{primární přípustnost})$$

$$\lambda \geq 0, \mu \geq 0 \quad (\text{duální přípustnost})$$

$$\lambda \left(\frac{x^2}{2} + y^2 - \frac{9}{8} \right) = 0, \mu y = 0 \quad (\text{komplementarita})$$

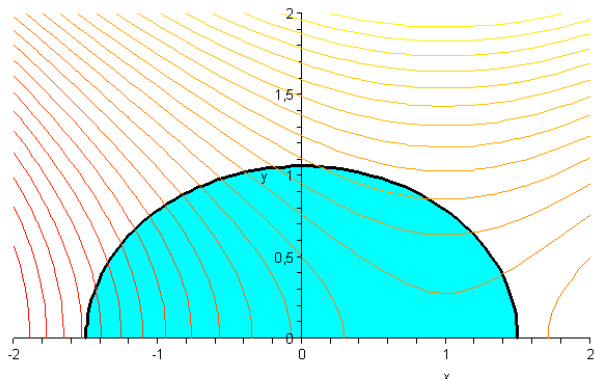
Výpočet provedeme zvlášť pro všechny kombinace (ne)nulovosti multiplikátorů λ a μ :

Porovnáním hodnot ve všech podezřelých bodech

$f(1, 0) = \frac{1}{2}$, $f(-\frac{3}{2}, 0) = -\frac{21}{8}$, $f(\frac{3}{2}, 0) = \frac{3}{8}$ zjistíme, že minimum nastává v bodě $[-\frac{3}{2}, 0]$

Optimalizační úloha s omezením ve formě nerovnosti - příklad

Znáznorněme si vrstevnice funkce $f(x, y) = x - \frac{x^2}{2} + y^2$ a přípustnou množinu vymezenou nerovnostmi $\frac{x^2}{2} + y^2 \leq \frac{9}{8}$, $y \geq 0$, minimum nastává v bodě $[-\frac{3}{2}, 0]$, maximum v bodě $[\frac{1}{2}, 1]$.



Konvexní programování

Úlohu $f(\mathbf{x}) \rightarrow \min$ na přípustné množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i$, $i = 1, \dots, m$ nazveme úlohou **konvexního programování**, jestliže účelová funkce $f(\mathbf{x})$ i levé strany omezení $g_i(\mathbf{x})$, $i = 1, \dots, m$ jsou **konvexní** funkce.

Konvexní programování

Úlohu $f(\mathbf{x}) \rightarrow \min$ na přípustné množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i$, $i = 1, \dots, m$ nazveme úlohou **konvexního programování**, jestliže účelová funkce $f(\mathbf{x})$ i levé strany omezení $g_i(\mathbf{x})$, $i = 1, \dots, m$ jsou **konvexní** funkce.

Věta: Pro úlohu konvexního programování za předpokladu regularity platí:

Vyhovuje-li bod \mathbf{x}^* KKT podmínkám, pak je bodem minima funkce $f(\mathbf{x})$ na M .

V případě konvexního programování je tedy splnění KKT vztahů **postačující podmínkou** pro existenci minima.

Konvexní programování

Úlohu $f(\mathbf{x}) \rightarrow \min$ na přípustné množině M vymezené soustavou m nerovnic $g_i(\mathbf{x}) \leq c_i$, $i = 1, \dots, m$ nazveme úlohou **konvexního programování**, jestliže účelová funkce $f(\mathbf{x})$ i levé strany omezení $g_i(\mathbf{x})$, $i = 1, \dots, m$ jsou **konvexní** funkce.

Věta: Pro úlohu konvexního programování za předpokladu regularity platí:

Vyhovuje-li bod \mathbf{x}^* KKT podmínkám, pak je bodem minima funkce $f(\mathbf{x})$ na M .

V případě konvexního programování je tedy splnění KKT vztahů **postačující podmínkou** pro existenci minima.

Poznámka: Uvažujeme-li v dané úloze konvexního programování při vymezení přípustné množiny M také **obligátní** podmínky nezápornosti $\mathbf{x} \geq \mathbf{0}$, pak lze KKT podmínky přeformulovat následovně:

Funkce $f(\mathbf{x})$ nabývá svého minima na M v bodě $\mathbf{x}^* \geq \mathbf{0}$ právě tehdy když existuje vektor $\lambda^* \geq \mathbf{0}$, takový že: $L(\mathbf{x}, \lambda^*) \geq L(\mathbf{x}^*, \lambda^*) \geq L(\mathbf{x}^*, \lambda)$ pro všechny nezáporné vektory \mathbf{x}, λ .

Toto tvrzení se nazývá **Kuhn-Tuckerova věta o sedlovém bodě**.

Konvexní programování

Poznámka: Vlastnosti sedlového bodu: Je-li $(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$ sedlovým bodem Lagrangeovy funkce na M , pak pro její gradienty zde platí:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \cdot \mathbf{x} = \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \leq \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \cdot \lambda = \mathbf{0},$$

kde symbolem " \cdot " rozumíme násobení vektorů po složkách.

Konvexní programování

Poznámka: Vlastnosti sedlového bodu: Je-li $(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$ sedlovým bodem Lagrangeovy funkce na M , pak pro její gradienty zde platí:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \cdot \mathbf{x} = \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \leq \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \cdot \lambda = \mathbf{0},$$

kde symbolem " \cdot " rozumíme násobení vektorů po složkách.

Funkce $L(\mathbf{x}, \lambda)$ nabývá na M v bodě $(\mathbf{x}^*, \lambda^*)$ svého minima vzhledem k \mathbf{x} a maxima vzhledem k λ . První dvě podmínky jsou zřejmě zobecněním požadavku stacionarity ($\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \mathbf{0}$) pro případ s obligátními podmínkami $x \geq 0$.

Konvexní programování

Poznámka: Vlastnosti sedlového bodu: Je-li $(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$ sedlovým bodem Lagrangeovy funkce na M , pak pro její gradienty zde platí:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \cdot \mathbf{x} = \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \leq \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \cdot \lambda = \mathbf{0},$$

kde symbolem " \cdot " rozumíme násobení vektorů po složkách.

Funkce $L(\mathbf{x}, \lambda)$ nabývá na M v bodě $(\mathbf{x}^*, \lambda^*)$ svého minima vzhledem k \mathbf{x} a maxima vzhledem k λ . První dvě podmínky jsou zřejmě zobecněním požadavku stacionarity ($\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \mathbf{0}$) pro případ s obligátními podmínkami $x \geq 0$. Pověsimně si ještě, že $\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) = (g_i(\mathbf{x}) - c_i)_{i=1, \dots, m}$, poslední dva vztahy vyjadřují tedy podmínky přípustnosti a komplementarity vzhledem k omezujícím podmínkám.

Konvexní programování

Poznámka: Vlastnosti sedlového bodu: Je-li $(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$ sedlovým bodem Lagrangeovy funkce na M , pak pro její gradienty zde platí:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \geq \mathbf{0}$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \cdot \mathbf{x} = \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \leq \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \cdot \lambda = \mathbf{0},$$

kde symbolem " \cdot " rozumíme násobení vektorů po složkách.

Funkce $L(\mathbf{x}, \lambda)$ nabývá na M v bodě $(\mathbf{x}^*, \lambda^*)$ svého minima vzhledem k \mathbf{x} a maxima vzhledem k λ . První dvě podmínky jsou zřejmě zobecněním požadavku stacionarity ($\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \mathbf{0}$) pro případ s obligátními podmínkami $x \geq 0$. Pověsimně si ještě, že $\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) = (g_i(\mathbf{x}) - c_i)_{i=1, \dots, m}$, poslední dva vztahy vyjadřují tedy podmínky přípustnosti a komplementarity vzhledem k omezujícím podmínkám.

Poznámka: Pro maximalizační úlohy se sestaví Lagrangeova funkce ve tvaru $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_{i=1}^m \lambda_i \cdot (g_i(\mathbf{x}) - c_i)$ a v podmínkách pro sedlový bod se uvažují opačné nerovnosti.

Konvexní programování - příklad

Poznámka: Lineární problém $\mathbf{c}^\top \cdot \mathbf{x} \rightarrow \max$, $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ je vlastně speciálním případem konvexního programování. Přepíšeme-li si účelovou funkci jako $-\mathbf{c}^\top \cdot \mathbf{x} \rightarrow \min$, budou podmínky pro sedlový bod $(\mathbf{x}, \lambda) \geq \mathbf{0}$ Lagrangeovy funkce následující:

$$-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A} \geq \mathbf{0}$$

$$(-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A}) \cdot \mathbf{x} = 0$$

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} \leq \mathbf{0}$$

$$(\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) \cdot \lambda = 0$$

Konvexní programování - příklad

Poznámka: Lineární problém $\mathbf{c}^\top \cdot \mathbf{x} \rightarrow \max$, $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ je vlastně speciálním případem konvexního programování. Přepíšeme-li si účelovou funkci jako $-\mathbf{c}^\top \cdot \mathbf{x} \rightarrow \min$, budou podmínky pro sedlový bod $(\mathbf{x}, \lambda) \geq \mathbf{0}$ Lagrangeovy funkce následující:

$$-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A} \geq \mathbf{0}$$

$$(-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A}) \cdot \mathbf{x} = 0$$

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} \leq \mathbf{0}$$

$$(\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) \cdot \lambda = 0$$

S těmito tvrzeními jsme se již setkali dříve, vyjadřují **dualitu v úlohách LP**.

Optimální λ je řešením tzv. **duální úlohy** $\mathbf{b}^\top \cdot \lambda \rightarrow \min$, $\mathbf{A}^\top \cdot \lambda \leq \mathbf{c}$, $\lambda \geq \mathbf{0}$,

protože dosazením vztahu $(-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A}) \cdot \mathbf{x} = 0$ do Lagrangeovy funkce dostaneme tvar $L(\mathbf{x}, \lambda) = -\mathbf{c}^\top \cdot \mathbf{x} + \lambda^\top \cdot \mathbf{A} \cdot \mathbf{x} - \lambda^\top \cdot \mathbf{b} = -\lambda^\top \cdot \mathbf{b}$.

Konvexní programování - příklad

Poznámka: Lineární problém $\mathbf{c}^\top \cdot \mathbf{x} \rightarrow \max$, $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$ je vlastně speciálním případem konvexního programování. Přepíšeme-li si účelovou funkci jako $-\mathbf{c}^\top \cdot \mathbf{x} \rightarrow \min$, budou podmínky pro sedlový bod $(\mathbf{x}, \lambda) \geq \mathbf{0}$ Lagrangeovy funkce následující:

$$-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A} \geq \mathbf{0}$$

$$(-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A}) \cdot \mathbf{x} = 0$$

$$\mathbf{A} \cdot \mathbf{x} - \mathbf{b} \leq \mathbf{0}$$

$$(\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) \cdot \lambda = 0$$

S těmito tvrzeními jsme se již setkali dříve, vyjadřují **dualitu v úlohách LP**.

Optimální λ je řešením tzv. **duální úlohy** $\mathbf{b}^\top \cdot \lambda \rightarrow \min$, $\mathbf{A}^\top \cdot \lambda \leq \mathbf{c}$, $\lambda \geq \mathbf{0}$,

protože dosazením vztahu $(-\mathbf{c}^\top + \lambda^\top \cdot \mathbf{A}) \cdot \mathbf{x} = 0$ do Lagrangeovy funkce dostaneme tvar $L(\mathbf{x}, \lambda) = -\mathbf{c}^\top \cdot \mathbf{x} + \lambda^\top \cdot \mathbf{A} \cdot \mathbf{x} - \lambda^\top \cdot \mathbf{b} = -\lambda^\top \cdot \mathbf{b}$. Řešení primární i duální úlohy jsou totožná. Při velkém počtu omezení ($m \gg n$) může být duální úloha mnohem jednodušší, proto se v praxi často při řešení primární úlohy používá tzv. **duální algoritmus**.

Kvadratické programování

Problém minimalizace funkce $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \cdot \mathbf{C} \cdot \mathbf{x} + \mathbf{d} \cdot \mathbf{x}$

na množině $M = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq \mathbf{0}, \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$

nazveme **úlohou kvadratického programování**.

Kvadratické programování

Problém minimalizace funkce $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \cdot \mathbf{C} \cdot \mathbf{x} + \mathbf{d} \cdot \mathbf{x}$

na množině $M = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq \mathbf{0}, \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$

nazveme **úlohou kvadratického programování**.

Jedná o úlohu konvexního programování? Přípustná množina M je jistě konvexní, stačí ověřit konvexitu účelové funkce. Hessova matice účelové funkce je $\mathbf{H}(\mathbf{x}) = \mathbf{C}$, je-li tedy tato matice pozitivně definitní, jedná se o konvexní problém.

Kvadratické programování

Problém minimalizace funkce $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \cdot \mathbf{C} \cdot \mathbf{x} + \mathbf{d} \cdot \mathbf{x}$

na množině $M = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq \mathbf{0}, \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$

nazveme **úlohou kvadratického programování**.

Jedná o úlohu konvexního programování? Přípustná množina M je jistě konvexní, stačí ověřit konvexitu účelové funkce. Hessova matice účelové funkce je $\mathbf{H}(\mathbf{x}) = \mathbf{C}$, je-li tedy tato matice pozitivně definitní, jedná se o konvexní problém. Zapišme podmínky pro sedlový bod Lagrangeovy funkce:

$$\nabla_{\mathbf{x}}L(\mathbf{x}^*, \lambda^*) = \mathbf{C} \cdot \mathbf{x} + \mathbf{d} + \mathbf{A}^\top \cdot \lambda \geq \mathbf{0}$$

$$\nabla_{\mathbf{x}}L(\mathbf{x}^*, \lambda^*) \cdot \mathbf{x} = (\mathbf{C} \cdot \mathbf{x} + \mathbf{d} + \mathbf{A}^\top \cdot \lambda) \cdot \mathbf{x} = \mathbf{0}$$

$$\nabla_{\lambda}L(\mathbf{x}^*, \lambda^*) = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \leq \mathbf{0}$$

$$\nabla_{\lambda}L(\mathbf{x}^*, \lambda^*) \cdot \lambda = (\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) \cdot \lambda = \mathbf{0}$$

Kvadratické programování

Problém minimalizace funkce $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \cdot \mathbf{C} \cdot \mathbf{x} + \mathbf{d} \cdot \mathbf{x}$

na množině $M = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq \mathbf{0}, \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}\}$

nazveme **úlohou kvadratického programování**.

Jedná o úlohu konvexního programování? Přípustná množina M je jistě konvexní, stačí ověřit konvexitu účelové funkce. Hessova matice účelové funkce je $\mathbf{H}(\mathbf{x}) = \mathbf{C}$, je-li tedy tato matice pozitivně definitní, jedná se o konvexní problém. Zapišme podmínky pro sedlový bod Lagrangeovy funkce:

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) = \mathbf{C} \cdot \mathbf{x} + \mathbf{d} + \mathbf{A}^\top \cdot \lambda \geq \mathbf{0}$$

$$\nabla_{\mathbf{x}} L(\mathbf{x}^*, \lambda^*) \cdot \mathbf{x} = (\mathbf{C} \cdot \mathbf{x} + \mathbf{d} + \mathbf{A}^\top \cdot \lambda) \cdot \mathbf{x} = \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) = \mathbf{A} \cdot \mathbf{x} - \mathbf{b} \leq \mathbf{0}$$

$$\nabla_{\lambda} L(\mathbf{x}^*, \lambda^*) \cdot \lambda = (\mathbf{A} \cdot \mathbf{x} - \mathbf{b}) \cdot \lambda = \mathbf{0}$$

Tato soustava se zpravidla upraví pomocí zavedení doplňkových proměnných \mathbf{w} , kterými se první nerovnost převede na rovnici $\mathbf{C} \cdot \mathbf{x} + \mathbf{d} + \mathbf{A}^\top \cdot \lambda - \mathbf{w} = \mathbf{0}$. Dále se úloha řeší jako lineární pomocí upravené simplexové metody, přičemž dodržení podmínky $\mathbf{w} \cdot \mathbf{x} = \mathbf{0}$ se zajistí tak, že hlídáme, aby se do báze nedostaly proměnné x_i a w_i nikdy současně ($i = 1, \dots, n$). Na této myšlence je založena tzv. **Wolfeho metoda** řešení úloh kvadratického programování.

Kvadratické programování, příklad

Uvažujme příklad minimalizace funkce

$$z = x_1^2 + 2x_1x_2 + 3x_2^2 + 10x_1 - 2x_2$$

na množině dané omezeními

$$2x_1 + x_2 \geq 5$$

$$x_1 + 3x_2 \leq 10, \quad x_1, x_2 \geq 0.$$

Kvadratické programování, příklad

Uvažujme příklad minimalizace funkce

$$z = x_1^2 + 2x_1x_2 + 3x_2^2 + 10x_1 - 2x_2$$

na množině dané omezeními

$$2x_1 + x_2 \geq 5$$

$$x_1 + 3x_2 \leq 10, \quad x_1, x_2 \geq 0.$$

Jde o úlohu kvadratického programování, navíc lze snadno ověřit, že Hessova matice je pozitivně definitní, tudíž je funkce konvexní. Sestavme Lagrangeovu funkci

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1^2 + 2x_1x_2 + 3x_2^2 + 10x_1 - 2x_2 + \lambda_1(x_1 + 3x_2 - 10) + \lambda_2(5 - 2x_1 - x_2)$$

a formulujme KKT podmínky:

Kvadratické programování, příklad

Uvažujme příklad minimalizace funkce

$$z = x_1^2 + 2x_1x_2 + 3x_2^2 + 10x_1 - 2x_2$$

na množině dané omezeními

$$2x_1 + x_2 \geq 5$$

$$x_1 + 3x_2 \leq 10, \quad x_1, x_2 \geq 0.$$

Jde o úlohu kvadratického programování, navíc lze snadno ověřit, že Hessova matice je pozitivně definitní, tudíž je funkce konvexní. Sestavme Lagrangeovu funkci

$$L(x_1, x_2, \lambda_1, \lambda_2) = x_1^2 + 2x_1x_2 + 3x_2^2 + 10x_1 - 2x_2 + \lambda_1(x_1 + 3x_2 - 10) + \lambda_2(5 - 2x_1 - x_2)$$

a formulujme KKT podmínky:

$$L_{x_1} = 2x_1 + 2x_2 - 2\lambda_1 + \lambda_2 + 10 \geq 0; \quad x_1 \cdot L_{x_1} = 0$$

$$L_{x_2} = 2x_1 + 6x_2 - \lambda_1 + 3\lambda_2 - 2 \geq 0; \quad x_2 \cdot L_{x_2} = 0$$

$$L_{\lambda_1} = -2x_1 - x_2 + 5 \leq 0; \quad \lambda_1 \cdot L_{\lambda_1} = 0$$

$$L_{\lambda_2} = x_1 + 3x_2 - 10 \leq 0; \quad \lambda_2 \cdot L_{\lambda_2} = 0$$

Kvadratické programování, příklad

Použijeme **Wolfeho metodu**: nejprve přepíšeme nerovnice na rovnice zavedením nezáporných doplňkových proměnných:

$w_1 = L_{x_1}, w_2 = L_{x_2}, u_1 = -L_{\lambda_1}, u_2 = -L_{\lambda_2}$. Sestavíme simplexovou tabulku úlohy, přičemž kvůli přípustnosti je nutné držet pravé strany kladné.

Kvadratické programování, příklad

Použijeme **Wolfeho metodu**: nejprve přepíšeme nerovnice na rovnice zavedením nezáporných doplňkových proměnných:

$w_1 = L_{x_1}$, $w_2 = L_{x_2}$, $u_1 = -L_{\lambda_1}$, $u_2 = -L_{\lambda_2}$. Sestavíme simplexovou tabulku úlohy, přičemž kvůli přípustnosti je nutné držet pravé strany kladné.

x_1	x_2	λ_1	λ_2	w_1	w_2	u_1	u_2	b_i
-2	-2	2	-1	1				10
2	6	-1	3		-1			2
2	1					-1		5
1	3						1	10

Kvadratické programování, příklad

Použijeme **Wolfeho metodu**: nejprve přepíšeme nerovnice na rovnice zavedením nezáporných doplňkových proměnných:

$w_1 = L_{x_1}$, $w_2 = L_{x_2}$, $u_1 = -L_{\lambda_1}$, $u_2 = -L_{\lambda_2}$. Sestavíme simplexovou tabulku úlohy, přičemž kvůli přípustnosti je nutné držet pravé strany kladné.

x_1	x_2	λ_1	λ_2	w_1	w_2	u_1	u_2	b_i
-2	-2	2	-1	1				10
2	6	-1	3		-1			2
2	1					-1		5
1	3						1	10

Do báze nesmíme zahrnout současně oba prvky z následujících dvojic: w_1 a x_1 ; w_2 a x_2 ; u_1 a λ_1 ; u_2 a λ_2 . Bohužel sloupce u doplňkových proměnných mají na 2. a 3. řádku záporné koeficienty a tedy netvoří jednotkovou matici, takže je třeba ještě přidat umělé proměnné y_1 , y_2 . Zatím jsme hledali libovolné řešení vyhovující KKT podmínkám. Díky umělým proměnným získáme vlastně účelovou funkci: chceme dostat $y_1 = y_2 = 0$, minimalizujeme tedy $z' = y_1 + y_2$.

Kvadratické programování, příklad

Rozšířená simplexová tabulka:

báze	x_1	x_2	λ_1	λ_2	w_1	w_2	u_1	u_2	y_1	y_2	b_i
w_1	-2	-2	2	-1	1						10
y_1	2	6	-1	3		-1			1		2
y_2	2	1					-1			1	5
u_2	1	3						1			10
Z_j									-1	-1	0

Je třeba anulovat redukované ceny u bazických proměnných, takže k poslednímu řádku přičteme druhý a třetí.

Kvadratické programování, příklad

Rozšířená simplexová tabulka:

báze	x_1	x_2	λ_1	λ_2	w_1	w_2	u_1	u_2	y_1	y_2	b_i
w_1	-2	-2	2	-1	1						10
y_1	2	6	-1	3		-1			1		2
y_2	2	1					-1			1	5
u_2	1	3						1			10
z_j	4	7	-1	3		-1	-1				7

Jde o minimalizační úlohu, volíme tedy největší kladnou redukovanou cenu, tj. 7 a x_2 vstoupí do báze místo y_1 .

Kvadratické programování, příklad

Rozšířená simplexová tabulka:

báze	x_1	x_2	λ_1	λ_2	w_1	w_2	u_1	u_2	y_1	y_2	b_i
w_1	-4/3		5/3		1	-1/3			1/3		32/3
x_2	2/6	1	-1/6	3/6		-1/6			1/6		2/6
y_2	10/6		1/6	-3/6		1/6	-1		-1/6	1	28/6
u_2			1/2	-3/2		1/2		1	-1/2		9
z_j	5/3		1/6	-1/2		1/6	-1		-7/6		14/3

Největší redukováná cena je 5/3, ale x_1 do báze vstoupit nemůže, protože už tam je w_1 . Do báze tedy vstoupí λ_1 .

Kvadratické programování, příklad

Rozšířená simplexová tabulka:

báze	x_1	x_2	λ_1	λ_2	w_1	w_2	u_1	u_2	y_1	y_2	b_i
λ_1	-4/5		1		3/5	-1/5			1/5		32/5
x_2	1/5	1		1/2	1/10	-1/5			1/5		7/5
y_2	9/5			-1/2	-1/10	-1/5			1/5		18/5
u_2	2/5			-3/2	-3/10	3/5		1	-3/5		29/5
z_j	9/5			-1/2	-1/10	1/5	-1		-6/5		18/5

Nyní již do báze může vstoupit x_1 , které má redukovanou cenu 9/5. Nahradí tam proměnnou y_2 . Podaří se nám vynulovat účelovou funkci, výpočet končí. Dostaneme:

$$x_1 = (18/5)/(9/5) = 2, \quad x_2 = 7/5 - 2 \cdot 1/5 = 1, \quad \lambda_1 = 32/5 + 2 \cdot 4/5 = 8, \quad \lambda_2 = 0.$$

Kvadratické programování, příklad: optimalizace portfolia

Předpokládejme, že chceme sestavit portfolio z cenných papírů, jejichž výnosy jsou náhodné veličiny, které označíme X_1, \dots, X_n . Tyto náhodné veličiny můžeme charakterizovat **očekávaným výnosem** $E(X_1), \dots, E(X_n)$ a také variabilitou vyjádřenou rozptyly $D(X_1), \dots, D(X_n)$. Navíc mezi jednotlivými dvojicemi cenných papírů může existovat nějaký vztah, kdy se jejich ceny mohou vyvíjet souhlasně či naopak protichůdně - tyto závislosti jsou vyjádřeny pomocí kovariancí $C(X_i, X_j)$. Kovariance a rozptyly lze zapsat souhrnně pomocí **variační matice** $\mathbf{V}(\mathbf{X})$. Přístupem, založeným na těchto charakteristikách, se zabývá **Markowitzův model portfolia**.

Kvadratické programování, příklad: optimalizace portfolia

Předpokládejme, že chceme sestavit portfolio z cenných papírů, jejichž výnosy jsou náhodné veličiny, které označíme X_1, \dots, X_n . Tyto náhodné veličiny můžeme charakterizovat **očekávaným výnosem** $E(X_1), \dots, E(X_n)$ a také variabilitou vyjádřenou rozptyly $D(X_1), \dots, D(X_n)$. Navíc mezi jednotlivými dvojicemi cenných papírů může existovat nějaký vztah, kdy se jejich ceny mohou vyvíjet souhlasně či naopak protichůdně - tyto závislosti jsou vyjádřeny pomocí kovariancí $C(X_i, X_j)$. Kovariance a rozptyly lze zapsat souhrnně pomocí **variační matice** $\mathbf{V}(\mathbf{X})$. Přístupem, založeným na těchto charakteristikách, se zabývá **Markowitzův model portfolia**.

Výnos portfolia, ve kterém jsou jednotlivé cenné papíry zastoupeny v podílech p_1, \dots, p_n je náhodná veličina $Y = \sum_{i=1}^n p_i X_i$. Očekávaný výnos bude roven

$E(Y) = \sum_{i=1}^n p_i E(X_i)$. Variabilitu výnosu portfolia je možné vyjádřit pomocí

jeho rozptylu, $D(Y) = (p_1, \dots, p_n) \cdot \mathbf{V}(\mathbf{X}) \cdot (p_1, \dots, p_n)^T$.

Kvadratické programování, příklad: optimalizace portfolia

Investor zpravidla požaduje co největší očekávaný výnos $E(Y)$ za co nejmenšího rizika (riziko, tedy nejistotu můžeme vyjádřit právě rozptylem $D(Y)$) Jde o úlohu vícekriteriálního programování

$E(Y) \rightarrow \max, D(Y) \rightarrow \min$ za omezující podmínky $\sum_{i=1}^n p_i = 1$.

Kvadratické programování, příklad: optimalizace portfolia

Investor zpravidla požaduje co největší očekávaný výnos $E(Y)$ za co nejmenšího rizika (riziko, tedy nejistotu můžeme vyjádřit právě rozptylem $D(Y)$) Jde o úlohu vícekriteriálního programování

$E(Y) \rightarrow \max, D(Y) \rightarrow \min$ za omezující podmínky $\sum_{i=1}^n p_i = 1$.

Jeden z možných přístupů k řešení této vícekriteriální úlohy je stanovení minimálního požadovaného výnosu R_{min} a minimalizace rizika mezi všemi portfolii s výnosem alespoň R_{min} . Dostaneme úlohu kvadratického programování $f(p_1, \dots, p_n) = (p_1, \dots, p_n) \cdot \mathbf{V}(\mathbf{X}) \cdot (p_1, \dots, p_n)^T \rightarrow \min$, za omezení $p_i \geq 0, \sum_{i=1}^n p_i = 1, \sum_{i=1}^n p_i E(X_i) \geq R_{min}$. Pro nedegenerovaná portfolia je matice $\mathbf{V}(\mathbf{X})$ pozitivně definitní, takže problém je **konvexní**.

Kvadratické programování, příklad: optimalizace portfolia

Investor zpravidla požaduje co největší očekávaný výnos $E(Y)$ za co nejmenšího rizika (riziko, tedy nejistotu můžeme vyjádřit právě rozptylem $D(Y)$) Jde o úlohu vícekriteriálního programování

$E(Y) \rightarrow \max, D(Y) \rightarrow \min$ za omezující podmínky $\sum_{i=1}^n p_i = 1$.

Jeden z možných přístupů k řešení této vícekriteriální úlohy je stanovení minimálního požadovaného výnosu R_{min} a minimalizace rizika mezi všemi portfolii s výnosem alespoň R_{min} . Dostaneme úlohu kvadratického

programování $f(p_1, \dots, p_n) = (p_1, \dots, p_n) \cdot \mathbf{V}(\mathbf{X}) \cdot (p_1, \dots, p_n)^T \rightarrow \min$, za

omezení $p_i \geq 0, \sum_{i=1}^n p_i = 1, \sum_{i=1}^n p_i E(X_i) \geq R_{min}$. Pro nedegenerovaná portfolia je matice $\mathbf{V}(\mathbf{X})$ pozitivně definitní, takže problém je **konvexní**.

Příklad: Navrhněte strukturu portfolia z dvou cenných papírů P_1, P_2 , tak aby jeho očekávaný výnos byl alespoň 0,04 a riziko minimální. Sledováním časových řad cenového vývoje cenných papírů jsme odhadli očekávané výnosy $E(X_1) = 0,03, E(X_2) = 0,05$, rozptyly $D(X_1) = 3, D(X_2) = 4$ a kovarianci $C(X_1, X_2) = 2$.

Kvadratické programování, příklad: optimalizace portfolia

Zapíšeme matematický model úlohy:

$$f(p_1, p_2) = (p_1, p_2) \cdot \begin{pmatrix} 3 & 2 \\ 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \rightarrow \min \text{ za podmínky}$$

$$p_1, p_2 \geq 0, 3p_1 + 5p_2 \geq 4, p_1 + p_2 \leq 1.$$

Poznámka: Očekávané výnosy jsme pro snadnější výpočty vynásobili 100 a poslední podmínku jsme zapsali jako nerovnici (připouštíme tedy i možnost nevyčerpání celé částky na investici!)

Kvadratické programování, příklad: optimalizace portfolia

Zapíšeme matematický model úlohy:

$$f(p_1, p_2) = (p_1, p_2) \cdot \begin{pmatrix} 3 & 2 \\ 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \rightarrow \min \text{ za podmínky}$$

$$p_1, p_2 \geq 0, 3p_1 + 5p_2 \geq 4, p_1 + p_2 \leq 1.$$

Poznámka: Očekávané výnosy jsme pro snadnější výpočty vynásobili 100 a poslední podmínku jsme zapsali jako nerovnici (připouštíme tedy i možnost nevyčerpání celé částky na investici!)

Lagrangeova funkce úlohy má tvar

$$L(p_1, p_2, \lambda, \mu) = 3p_1^2 + 4p_1p_2 + 4p_2^2 + \lambda(p_1 + p_2 - 1) + \mu(-3p_1 - 5p_2 + 4)$$

Kvadratické programování, příklad: optimalizace portfolia

Zapišeme matematický model úlohy:

$$f(p_1, p_2) = (p_1, p_2) \cdot \begin{pmatrix} 3 & 2 \\ 2 & 4 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \rightarrow \min \text{ za podmínky}$$

$$p_1, p_2 \geq 0, 3p_1 + 5p_2 \geq 4, p_1 + p_2 \leq 1.$$

Poznámka: Očekávané výnosy jsme pro snadnější výpočty vynásobili 100 a poslední podmínku jsme zapsali jako nerovnici (připouštíme tedy i možnost nevyčerpání celé částky na investici!)

Lagrangeova funkce úlohy má tvar

$$L(p_1, p_2, \lambda, \mu) = 3p_1^2 + 4p_1p_2 + 4p_2^2 + \lambda(p_1 + p_2 - 1) + \mu(-3p_1 - 5p_2 + 4)$$

Kuhn-Tuckerovy podmínky pro $p_1, p_2, \lambda, \mu \geq 0$ jsou:

$$L'_{p_1} = 6p_1 + 4p_2 + \lambda - 3\mu \geq 0, (6p_1 + 4p_2 + \lambda - 3\mu)p_1 = 0$$

$$L'_{p_2} = 4p_1 + 8p_2 + \lambda - 5\mu \geq 0, (4p_1 + 8p_2 + \lambda - 5\mu)p_2 = 0$$

$$L'_{\lambda} = p_1 + p_2 - 1 \leq 0, (p_1 + p_2 - 1)\lambda = 0$$

$$L'_{\mu} = -3p_1 - 5p_2 + 4 \leq 0, (-3p_1 - 5p_2 + 4)\mu = 0$$

Kvadratické programování, příklad: optimalizace portfolia

Při hledání bodu vyhovujícího KT podmínkám uvažujme opět všechny možnosti (ne)nulovosti multiplikátorů λ a μ :

$\lambda > 0, \mu > 0$: Komplementarita pro oba multiplikátory dává $3p_1 + 5p_2 = 4$, $p_1 + p_2 = 1$, což nám dá řešení $p_1 = p_2 = \frac{1}{2}$, po dosazení do prvních dvou rovnic dostaneme $5 + \lambda - 3\mu = 0$, $6 + \lambda - 5\mu = 0$, tato soustava ale dává záporné λ , takže nezískáme žádné řešení.

Kvadratické programování, příklad: optimalizace portfolia

Při hledání bodu vyhovujícího KT podmínkám uvažujme opět všechny možnosti (ne)nulovosti multiplikátorů λ a μ :

$\lambda > 0, \mu > 0$: Komplementarita pro oba multiplikátory dává $3p_1 + 5p_2 = 4$, $p_1 + p_2 = 1$, což nám dá řešení $p_1 = p_2 = \frac{1}{2}$, po dosazení do prvních dvou rovnic dostaneme $5 + \lambda - 3\mu = 0$, $6 + \lambda - 5\mu = 0$, tato soustava ale dává záporné λ , takže nezískáme žádné řešení.

$\lambda = 0, \mu > 0$: Z komplementarity pro μ plyne $3p_1 + 5p_2 = 4$. Zřejmě tedy $p_2 \neq 0$ a druhou podmínku lze vydělit p_2 : $4p_1 + 8p_2 - 5\mu = 0$, po vyloučení možnosti $p_1 = 0$ (pokazila by se nezápornost L'_{p_1}) musí být $6p_1 + 4p_2 + \lambda - 3\mu = 0$, takže dostaneme lineární soustavu, jejímž řešením je $p_1 = 0, 16, p_2 = 0, 7, \mu = 1, 25$ a očekávaný výnos 4 při riziku 2, 5.

Kvadratické programování, příklad: optimalizace portfolia

Při hledání bodu vyhovujícího KT podmínkám uvažujme opět všechny možnosti (ne)nulovosti multiplikátorů λ a μ :

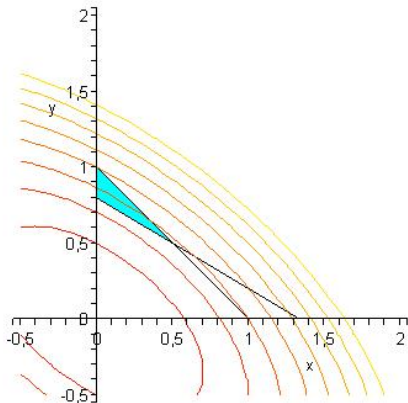
$\lambda > 0, \mu > 0$: Komplementarita pro oba multiplikátory dává $3p_1 + 5p_2 = 4$, $p_1 + p_2 = 1$, což nám dá řešení $p_1 = p_2 = \frac{1}{2}$, po dosazení do prvních dvou rovnic dostaneme $5 + \lambda - 3\mu = 0$, $6 + \lambda - 5\mu = 0$, tato soustava ale dává záporné λ , takže nezískáme žádné řešení.

$\lambda = 0, \mu > 0$: Z komplementarity pro μ plyne $3p_1 + 5p_2 = 4$. Zřejmě tedy $p_2 \neq 0$ a druhou podmínku lze vydělit p_2 : $4p_1 + 8p_2 - 5\mu = 0$, po vyloučení možnosti $p_1 = 0$ (pokazila by se nezápornost L'_{p_1}) musí být $6p_1 + 4p_2 + \lambda - 3\mu = 0$, takže dostaneme lineární soustavu, jejímž řešením je $p_1 = 0, p_2 = 0,7, \mu = 1,25$ a očekávaný výnos 4 při riziku 2,5.

$\lambda > 0, \mu = 0$: Z komplementarity pro λ plyne $p_1 + p_2 = 1$. Pro extrémní případ $p_1 = 0, p_2 = 1$ nebo naopak se pokazí nezápornost derivace L'_{p_1} nebo L'_{p_2} . Tedy $p_1, p_2 \neq 0$ a první dvě podmínky lze vydělit p_1 a p_2 : $6p_1 + 4p_2 + \lambda = 0$, $4p_1 + 8p_2 + \lambda = 0$ a dostaneme soustavu, pro niž však vyjde $\lambda < 0$, takže nezískáme žádné řešení.

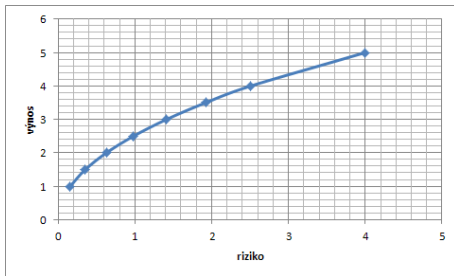
Kvadratické programování, příklad: optimalizace portfolia

$\lambda = \mu = 0$: Z prvních dvou rovnic dostaneme $(6p_1 + 4p_2)p_1 = 0$, $(4p_1 + 8p_2)p_2 = 0$, což má jediné nezáporné řešení $p_1 = p_2 = 0$, což však nevyhovuje podmínce $3p_1 + 5p_2 \geq 4$.



Kvadratické programování, příklad: optimalizace portfolia

Úlohu bychom mohli řešit i pro jiné aspirační úrovně výnosu R_{min} , například $R_{min} \in \{1; 1, 5; 2; 2, 5; 3; 3, 5; 4; 5\}$. Vynesme si tyto hodnoty spolu s optimálními hodnotami účelové funkce (tj. minimálními riziky při daném výnosu) graficky v tzv. **kriteriálním prostoru**:



Spojnice mezi body je odhadem tzv. **efektivní hranice**, tvořené portfolii, jejichž výnos lze zlepšit jen za cenu zhoršení rizika a naopak. Efektivní hranici také najít pomocí optimalizace funkcí $f(Y) = c \cdot E(Y) - (1 - c) \cdot D(Y)$ na množině všech portfolií, kde konstanta c probíhá interval $\langle 0, 1 \rangle$.

Numerická optimalizace

Řadu problémů není možné řešit analyticky (neznalost analytického vyjádření, mnoho proměnných, složité funkce - nelineární rovnice při určování stacionárních bodů) nebo by bylo analytické řešení příliš výpočetně náročné a tudíž neefektivní. Často nám stačí nalézt pouze přibližné řešení (v reálném světě stejně zaokrouhlujeme). Proto se v praxi používají **metody iterační**. Jejich charakteristikou je konstrukce posloupnosti bodů definičního oboru x^0, x^1, x^2, \dots , které se blíží k optimálnímu řešení x^* . Nasazení těchto metod pro vyhledání extrému je velice výhodné v kombinaci s výpočetní technikou.

Numerická optimalizace

Řadu problémů není možné řešit analyticky (neznalost analytického vyjádření, mnoho proměnných, složité funkce - nelineární rovnice při určování stacionárních bodů) nebo by bylo analytické řešení příliš výpočetně náročné a tudíž neefektivní. Často nám stačí nalézt pouze přibližné řešení (v reálném světě stejně zaokrouhlujeme). Proto se v praxi používají **metody iterační**. Jejich charakteristikou je konstrukce posloupnosti bodů definičního oboru x^0, x^1, x^2, \dots , které se blíží k optimálnímu řešení x^* . Nasazení těchto metod pro vyhledání extrému je velice výhodné v kombinaci s výpočetní technikou.

Existuje řada metod, a to jak pro funkce jedné proměnné, tak pro více proměnných. Zřejmě nejznámější jsou **pro funkce jedné proměnné**:

- metoda bisekce intervalu
- metoda zlatého řezu
- metoda kvadratické interpolace
- Newtonova metoda
- metoda Regula falsi

Jednorozměrná optimalizace - komparativní metody

Nejprve se zaměříme na metody **jednorozměrné numerické optimalizace**, ty jsou totiž součástí některých vícerozměrných metod. Metody nevyžadující výpočet derivace funkce f nazýváme **komparativní**. Uvedeme tři z nich, a to trisekci, metodu zlatého řezu a metodu kvadratické interpolace. Většinou je třeba předem přibližně určit polohu bodu optima a stanovit počáteční interval $I = \langle a, b \rangle$ tak, aby na změněném intervalu účelová funkce f měla jen jedno minimum, tj. byla **unimodální**. V jednotlivých iteracích nahrazujeme krajní body intervalu obsahujícího minimum novými body, tak abychom jej postupně zužovali. Jako odhad bodu optima se vezme střed posledního intervalu

$$I^n = \langle a^n, b^n \rangle.$$

Jednorozměrná optimalizace - komparativní metody

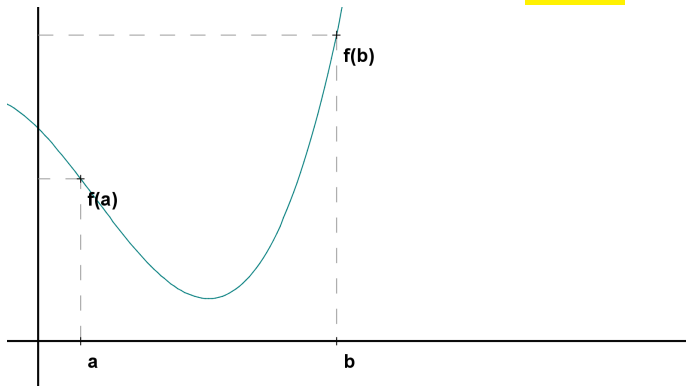
Nejprve se zaměříme na metody **jednorozměrné numerické optimalizace**, ty jsou totiž součástí některých vícerozměrných metod. Metody nevyžadující výpočet derivace funkce f nazýváme **komparativní**. Uvedeme tři z nich, a to trisekci, metodu zlatého řezu a metodu kvadratické interpolace. Většinou je třeba předem přibližně určit polohu bodu optima a stanovit počáteční interval $I = \langle a, b \rangle$ tak, aby na změněném intervalu účelová funkce f měla jen jedno minimum, tj. byla **unimodální**. V jednotlivých iteracích nahrazujeme krajní body intervalu obsahujícího minimum novými body, tak abychom jej postupně zužovali. Jako odhad bodu optima se vezme střed posledního intervalu $I^n = \langle a^n, b^n \rangle$.

U všech iteračních metod musíme předem nastavit nějaké pravidlo **ukončení výpočtu**. Metody většinou vyžadují splnění jistých podmínek, za kterých platí, že čím více iteračních kroků provedeme, tím přesnější odhad hledaného extrému získáme. Možné přístupy, kdy výpočet ukončit jsou například:

- po provedení zadaného počtu kroků
- dosažením stanovené přesnosti ε (tedy když šířka intervalu I^n splňuje: $|b^n - a^n| < 2\varepsilon$)

Jednorozměrná optimalizace - metoda trisekce

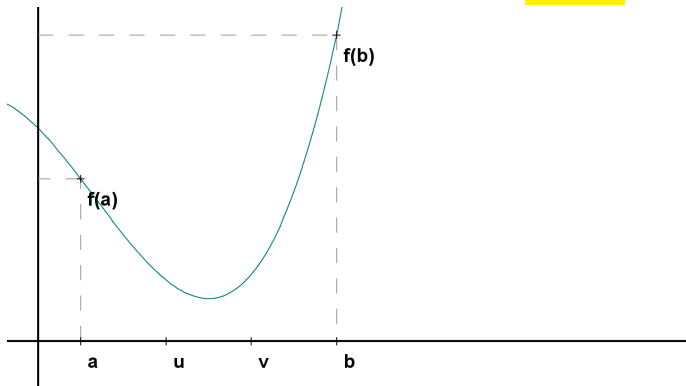
Jako úvodní motivaci ukážeme intervalovou **trisekci**.



Označme $d = (b - a)$.

Jednorozměrná optimalizace - metoda trisekce

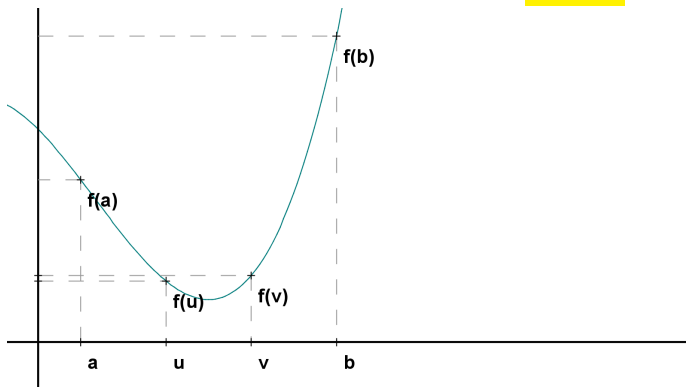
Jako úvodní motivaci ukážeme intervalovou **trisekci**.



Body $u = a + d/3$ a $v = b - d/3$ dělí interval $\langle a, b \rangle$ na tři stejné části.

Jednorozměrná optimalizace - metoda trisekce

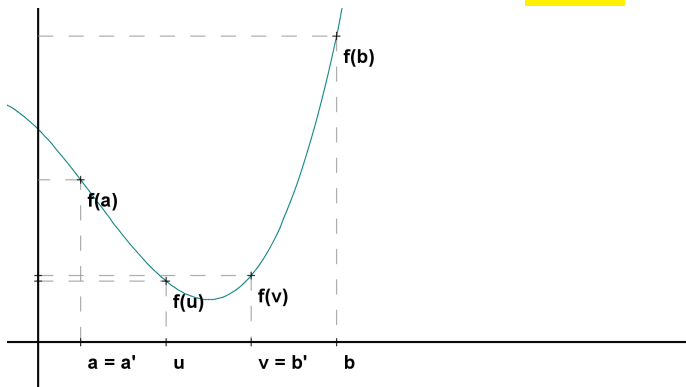
Jako úvodní motivaci ukážeme intervalovou **trisekci**.



Nejprve vyhodnotíme $f(u)$ a $f(v)$. Předpokládejme, že $f(u) < f(v)$.

Jednorozměrná optimalizace - metoda trisekce

Jako úvodní motivaci ukážeme intervalovou **trisekci**.



Minimum jistě leží vlevo od v , takže b nahradíme pomocí $b^1 = v$, levá hranice zůstává $a^1 = a$. Tím se délka intervalu (obsahujícího minimum) zkrátí na dvě třetiny své původní délky. Dále pokračujeme s intervalem $I^1 = \langle a^1, b^1 \rangle$, atd. Bod u však už nelze v následujícím kroku využít. Funkci musíme vyhodnotit v každém kroku dvakrát, což je neefektivní.

Jednorozměrná optimalizace - metoda zlatého řezu

Metoda zlatého řezu je založena na šikovnějším výběru dělicích bodů u a v . Označme ρ číslo o něco větší než $1/3$, jehož přesnou hodnotu teprve určíme. Označme body $u = a + \rho d$ a $v = b - \rho d$.

Jednorozměrná optimalizace - metoda zlatého řezu

Metoda zlatého řezu je založena na šikovnějším výběru dělicích bodů u a v . Označme ρ číslo o něco větší než $1/3$, jehož přesnou hodnotu teprve určíme. Označme body $u = a + \rho d$ a $v = b - \rho d$. Předpokládejme opět, že $f(u) < f(v)$, minimum je tedy mezi a a v . Nahradíme b pomocí v a proces opakujeme. Zvolíme-li správnou hodnotu ρ , bod u bude v pozici použitelné i v příštím kroku. Po prvním kroku se tak funkce f bude vyhodnocovat pokaždé už jen jednou.

Jednorozměrná optimalizace - metoda zlatého řezu

Metoda zlatého řezu je založena na šikovnějším výběru dělicích bodů u a v . Označme ρ číslo o něco větší než $1/3$, jehož přesnou hodnotu teprve určíme. Označme body $u = a + \rho d$ a $v = b - \rho d$. Předpokládejme opět, že $f(u) < f(v)$, minimum je tedy mezi a a v . Nahradíme b pomocí v a proces opakujeme. Zvolíme-li správnou hodnotu ρ , bod u bude v pozici použitelné i v příštím kroku. Po prvním kroku se tak funkce f bude vyhodnocovat pokaždé už jen jednou.

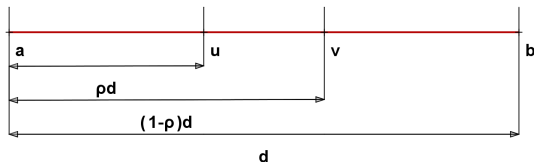
Jak tedy zvolit ρ ? Tak, aby bod u hrál v redukovaném intervalu $\langle a, v \rangle$ stejnou roli jako bod v v původním intervalu, tj. aby poměr délky intervalu $\langle a, u \rangle$ k délce intervalu $\langle a, v \rangle$ byl stejný jako poměr délky intervalu $\langle a, v \rangle$ k délce intervalu $\langle a, b \rangle$, tj. $\rho : (1 - \rho) = (1 - \rho) : 1$.

Jednorozměrná optimalizace - metoda zlatého řezu

Metoda zlatého řezu je založena na šikovnějším výběru dělicích bodů u a v . Označme ρ číslo o něco větší než $1/3$, jehož přesnou hodnotu teprve určíme. Označme body $u = a + \rho d$ a $v = b - \rho d$. Předpokládejme opět, že $f(u) < f(v)$, minimum je tedy mezi a a v . Nahradíme b pomocí v a proces opakujeme. Zvolíme-li správnou hodnotu ρ , bod u bude v pozici použitelné i v příštím kroku. Po prvním kroku se tak funkce f bude vyhodnocovat pokaždé už jen jednou.

Jak tedy zvolit ρ ? Tak, aby bod u hrál v redukovaném intervalu $\langle a, v \rangle$ stejnou roli jako bod v v původním intervalu, tj. aby poměr délky intervalu $\langle a, u \rangle$ k délce intervalu $\langle a, v \rangle$ byl stejný jako poměr délky intervalu $\langle a, v \rangle$ k délce intervalu $\langle a, b \rangle$, tj. $\rho : (1 - \rho) = (1 - \rho) : 1$. Po úpravě tohoto vztahu obdržíme kvadratickou rovnici $\rho^2 - 3\rho + 1 = 0$, jejímž řešením je

$\rho = \frac{3 - \sqrt{5}}{2} \approx 0,382$, kde číslo $1 - \rho \approx 0,618$ je tzv. **poměr zlatého řezu**.



Jednorozměrná optimalizace - metoda zlatého řezu, příklad

Lze ukázat, že požadujeme-li při metodě zlatého řezu ukončení při dosažení stanovené přesnosti ε , stačí provést N iterací, kde

$$N \geq \frac{\log\left(\frac{b-a}{2\varepsilon}\right)}{-\log(1-\rho)}.$$

Výsledný odhad bodu optima zapíšeme ve tvaru $x^* = \frac{b^N + a^N}{2} \pm \varepsilon$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad

Lze ukázat, že požadujeme-li při metodě zlatého řezu ukončení při dosažení stanovené přesnosti ε , stačí provést N iterací, kde

$$N \geq \frac{\log\left(\frac{b-a}{2\varepsilon}\right)}{-\log(1-\rho)}.$$

Výsledný odhad bodu optima zapíšeme ve tvaru $x^* = \frac{b^N + a^N}{2} \pm \varepsilon$.

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0, 5; 3 \rangle$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad

Lze ukázat, že požadujeme-li při metodě zlatého řezu ukončení při dosažení stanovené přesnosti ε , stačí provést N iterací, kde

$$N \geq \frac{\log\left(\frac{b-a}{2\varepsilon}\right)}{-\log(1-\rho)}.$$

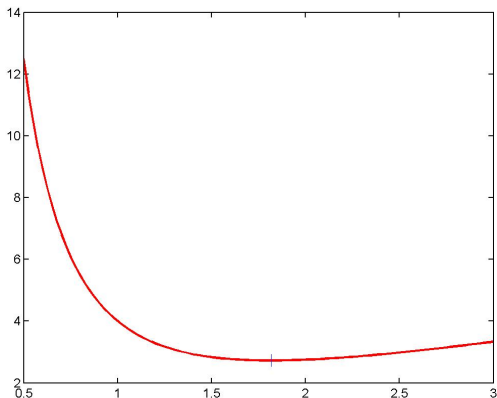
Výsledný odhad bodu optima zapíšeme ve tvaru $x^* = \frac{b^N + a^N}{2} \pm \varepsilon$.

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0, 5; 3 \rangle$.

Spočteme nejprve, kolik bude třeba provést iterací:

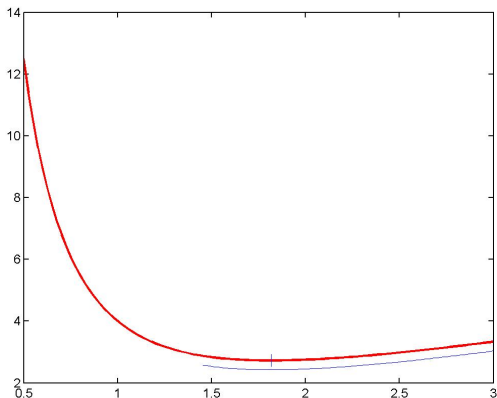
$$N \geq \frac{\log\left(\frac{b-a}{2\varepsilon}\right)}{-\log(1-\rho)} = \frac{\log\left(\frac{2,5}{0,1}\right)}{-\log(0,618)} \approx 6,69, \text{ stačí tedy } 7 \text{ iterací.}$$

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



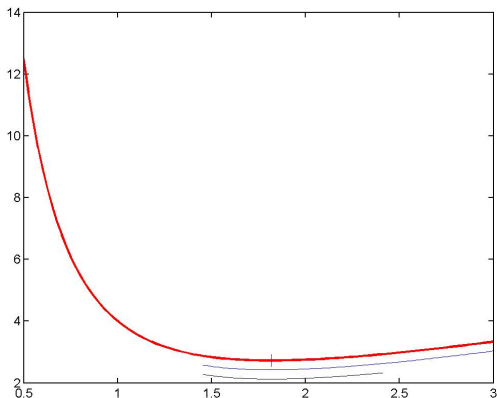
Graf funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $a = 0,5$; $b = 3$, svistou čarou je znázorněno skutečné minimum $x^* = \sqrt[3]{6} \approx 1,817120$

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



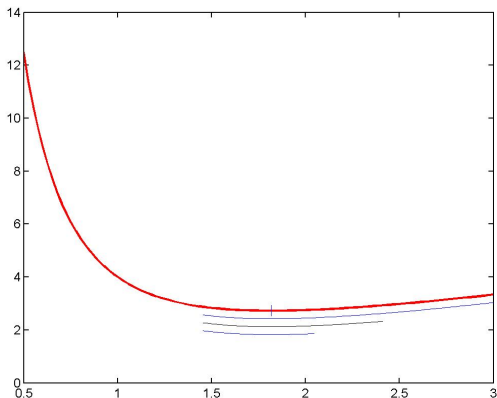
1. krok: $u = 1.454915$, $v = 2.045085$, $f(u) = 2.872163$, $f(v) = 2.762381$.
Nový interval je $I^1 = \langle 1.454915, 3.000000 \rangle$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



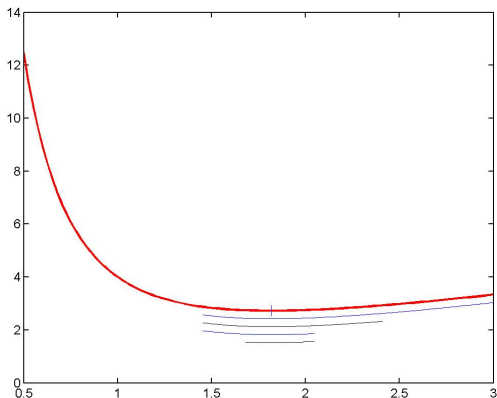
2. krok: $u = 2.045085$, $v = 2.409830$, $f(u) = 2.762381$, $f(v) = 2.926423$.
Nový interval je $I^2 = \langle 1.454915, 2.409830 \rangle$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



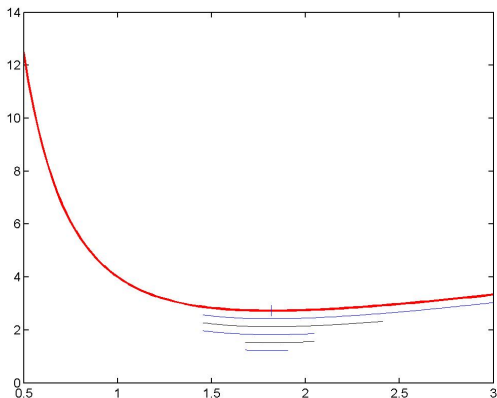
3. krok: $u = 1.819660$, $v = 2.045085$, $f(u) = 2.725686$, $f(v) = 2.762381$.
Nový interval je $I^3 = \langle 1.454915, 2.045085 \rangle$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



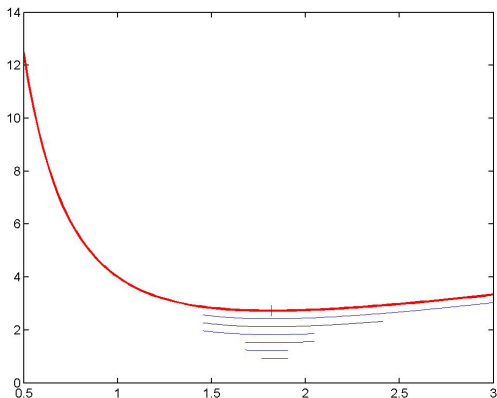
4. krok: $u = 1.680340$, $v = 1.819660$, $f(u) = 2.742835$, $f(v) = 2.725686$.
Nový interval je $I^4 = \langle 1.680340, 2.045085 \rangle$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



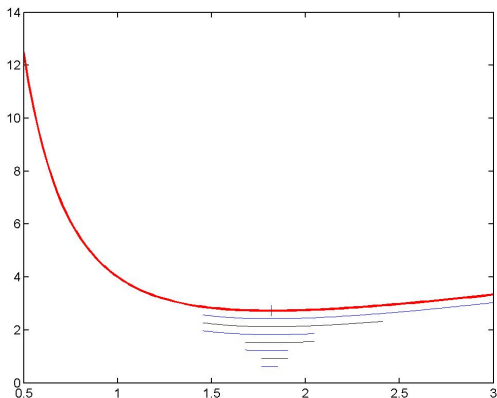
5. krok: $u = 1.819660$, $v = 1.905765$, $f(u) = 2.725686$, $f(v) = 2.731770$.
Nový interval je $I^5 = \langle 1.680340, 1.905765 \rangle$.

Jednorozměrná optimalizace - metoda zlatého řezu, příklad



6. krok: $u = 1.766445$, $v = 1.819660$, $f(u) = 2.727882$, $f(v) = 2.725686$.
Nový interval je $I^6 = \langle 1.766445, 1.905765 \rangle$.

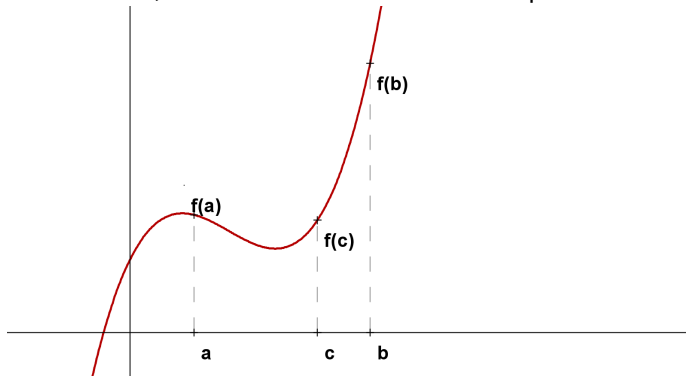
Jednorozměrná optimalizace - metoda zlatého řezu, příklad



7. krok: $u = 1.819660$, $v = 1.852549$, $f(u) = 2.725686$, $f(v) = 2.726691$.
Nový interval je $I^7 = \langle 1.766445, 1.852549 \rangle$. Šířka intervalu je menší než 0, 1,
tedy odhadneme bod optima jeho středem: $x^* = 1,81 \pm 0,05$

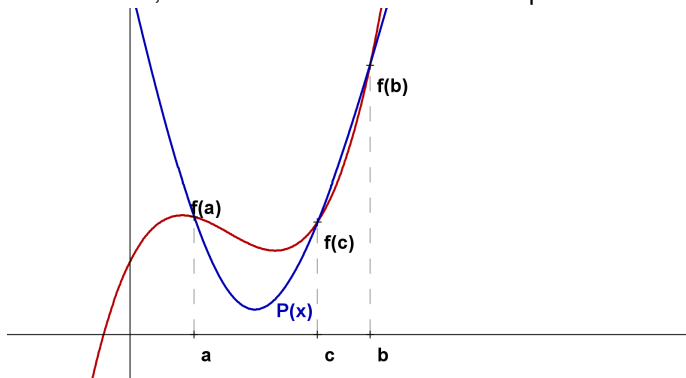
Jednorozměrná optimalizace - metoda kvadratické interpolace

Předpokládejme, že minimum leží v intervalu $\langle a, b \rangle$, a že je znám nějaký jeho vnitřní bod c , ve kterém hodnota funkce f nepřesáhne hodnoty $f(a)$, $f(b)$.



Jednorozměrná optimalizace - metoda kvadratické interpolace

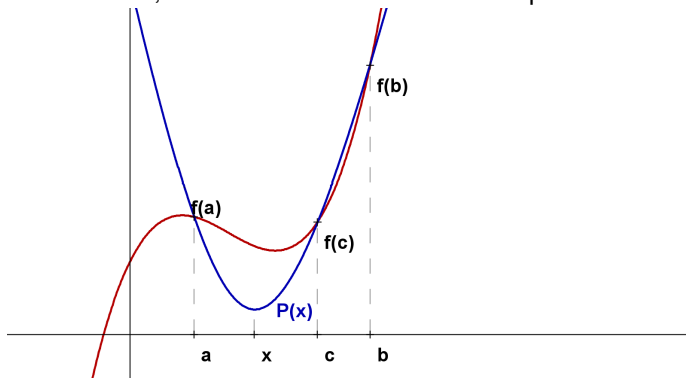
Předpokládejme, že minimum leží v intervalu $\langle a, b \rangle$, a že je znám nějaký jeho vnitřní bod c , ve kterém hodnota funkce f nepřesáhne hodnoty $f(a)$, $f(b)$.



Body $[a, f(a)]$, $[c, f(c)]$ a $[b, f(b)]$ proložíme parabolou $P(x)$

Jednorozměrná optimalizace - metoda kvadratické interpolace

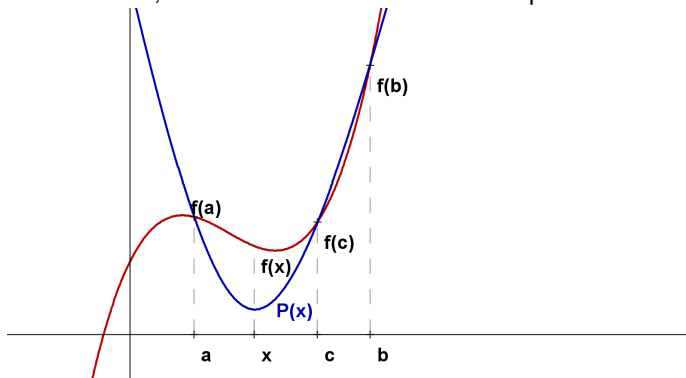
Předpokládejme, že minimum leží v intervalu $\langle a, b \rangle$, a že je znám nějaký jeho vnitřní bod c , ve kterém hodnota funkce f nepřesáhne hodnoty $f(a)$, $f(b)$.



Bod jejího minima x považujeme za aproximaci bodu optima x^* .

Jednorozměrná optimalizace - metoda kvadratické interpolace

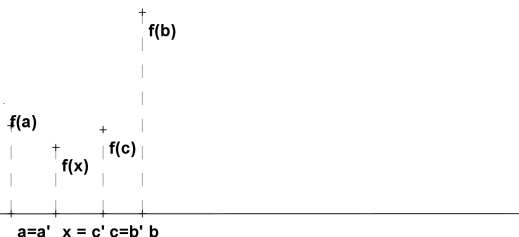
Předpokládejme, že minimum leží v intervalu $\langle a, b \rangle$, a že je znám nějaký jeho vnitřní bod c , ve kterém hodnota funkce f nepřesáhne hodnoty $f(a)$, $f(b)$.



Dá se ukázat, že $x = \frac{1}{2} \frac{f(a)(b^2 - c^2) + f(c)(a^2 - b^2) + f(b)(c^2 - a^2)}{f(a)(b - c) + f(c)(a - b) + f(b)(c - a)}$. Další postup se volí podle hodnoty $f(x)$.

Jednorozměrná optimalizace - metoda kvadratické interpolace

Víme, že $x \in (a, b)$, vyšlo nám $x < c$. Je-li $f(x) < f(c)$, do další iterace použijeme vypočtené x místo bodu c , který se stane novým krajním bodem (pokud $x < c$, tak jím nahradíme horní hranici b , viz obr., v opačném případě dolní hranici a). Je-li naopak $f(x) > f(c)$, pak c ponecháme a novým krajním bodem se stane x . Příklad $x = c$ znamená, že buď jsme se strefili do optima nebo je nutná jiná volba c . Tento a další nedostatky překonává kombinovaný přístup, tzv. **Brentova metoda**.



Výpočet se zastaví, až rozdíl dvou po sobě jdoucích odhadů $|x^n - x^{n-1}|$ klesne pod hodnotu ε .

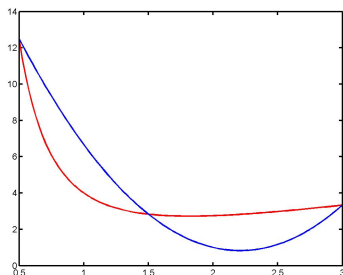
Jednorozměrná optimalizace - metoda kvadratické interpolace, příklad

Příklad: Minimalizujte funkci $f(x) = x + \frac{3}{x^2}$ na intervalu $\langle 0, 5; 3 \rangle$ s přesností $\varepsilon = 0,05$, jako první aproximaci volte $x^0 = 1,5$.

Jednorozměrná optimalizace - metoda kvadratické interpolace, příklad

Příklad: Minimalizujte funkci $f(x) = x + \frac{3}{x^2}$ na intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$, jako první aproximaci volte $x^0 = 1,5$.

První interpolaci provedeme pro body $a = 0,5$, $c = 1,5$ a $b = 3$. Znázorníme červeně graf $f(x)$ na $\langle a, b \rangle$ a modře parabolou, protínající jej v bodech a, b, c :



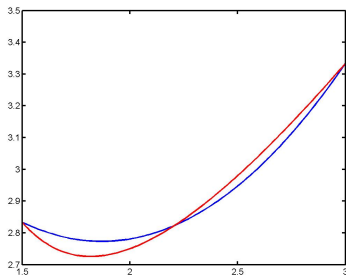
1. krok: $f(a) = 12.500000$, $f(c) = 2.833333$, $f(b) = 3.333333 \Rightarrow$

$x = 2.208333$, $f(x) = 2.823499 \Rightarrow$ Nově:

$a = 1.500000$, $c = 2.208333$, $b = 3.000000$, chyba $|x^1 - x^0| = 0.708333$

Jednorozměrná optimalizace - metoda kvadratické interpolace, příklad

Příklad: Minimalizujte funkci $f(x) = x + \frac{3}{x^2}$ na intervalu $\langle 0, 5; 3 \rangle$ s přesností $\varepsilon = 0, 05$, jako první aproximaci volte $x^0 = 1, 5$.



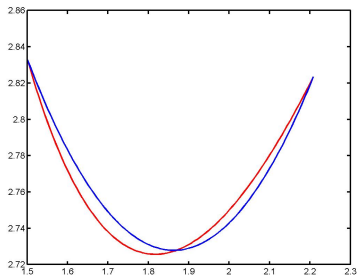
2. krok: $f(a) = 2.833333$, $f(c) = 2.823499$, $f(b) = 3.333333 \Rightarrow$

$x = 1.869995$, $f(x) = 2.727902 \Rightarrow$ Nově:

$a = 1.500000$, $c = 1.869995$, $b = 2.208333$, chyba $|x^2 - x^1| = 0.338339$.

Jednorozměrná optimalizace - metoda kvadratické interpolace, příklad

Příklad: Minimalizujte funkci $f(x) = x + \frac{3}{x^2}$ na intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$, jako první aproximaci volte $x^0 = 1,5$.



3. krok: $f(a) = 2.833333$, $f(c) = 2.727902$, $f(b) = 2.823499 \Rightarrow$
 $x = 1.862831$, $f(x) = 2.727350 \Rightarrow$ Nově:

$a = 1.500000$, $c = 1.862831$, $b = 1.869995$, chyba

$|x^3 - x^2| = 0.007163 < 0,05$, výpočet se zastaví: $x^* \approx 1,862831 \pm 0,05$.

Jednorozměrná optimalizace - metody využívající derivace

Kromě standardního předpokladu unimodálnosti funkce f na intervalu $\langle a, b \rangle$ předpokládejme dále, že f je zde diferencovatelná. Nejjednodušší iterační optimalizační metodou využívající derivaci funkce f je **metoda bisekce**, čili půlení intervalů. Jestliže označíme bod, v němž funkce f nabývá svého minima na intervalu $\langle a, b \rangle$ jako p , pak je f klesající na $\langle a, p \rangle$ (tudíž zde platí $f'(x) < 0$) a rostoucí na intervalu $\langle p, b \rangle$ (tudíž zde platí $f'(x) > 0$).

Jednorozměrná optimalizace - metody využívající derivace

Kromě standartního předpokladu unimodálnosti funkce f na intervalu $\langle a, b \rangle$ předpokládejme dále, že f je zde diferencovatelná. Nejjednodušší iterační optimalizační metodou využívající derivaci funkce f je **metoda bisekce**, čili půlení intervalů. Jestliže označíme bod, v němž funkce f nabývá svého minima na intervalu $\langle a, b \rangle$ jako p , pak je f klesající na $\langle a, p \rangle$ (tudíž zde platí $f'(x) < 0$) a rostoucí na intervalu $\langle p, b \rangle$ (tudíž zde platí $f'(x) > 0$).

Vezmeme-li jako odhad bodu minima střed intervalu $s = \frac{a+b}{2}$, pak pro $f'(s) < 0$ leží minimum vpravo, klademe tedy $a^1 = s$ a pro $f'(s) > 0$ leží minimum vlevo a klademe $b^1 = s$, druhý krajní bod zůstává (pokud $f'(s) = 0$, našli jsme přímo bod optima x^*).

Jednorozměrná optimalizace - metody využívající derivace

Kromě standartního předpokladu unimodálnosti funkce f na intervalu $\langle a, b \rangle$ předpokládejme dále, že f je zde diferencovatelná. Nejjednodušší iterační optimalizační metodou využívající derivaci funkce f je **metoda bisekce**, čili půlení intervalů. Jestliže označíme bod, v němž funkce f nabývá svého minima na intervalu $\langle a, b \rangle$ jako p , pak je f klesající na $\langle a, p \rangle$ (tudíž zde platí $f'(x) < 0$) a rostoucí na intervalu $\langle p, b \rangle$ (tudíž zde platí $f'(x) > 0$).

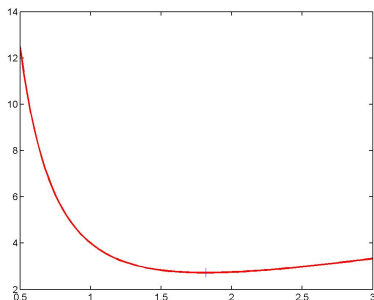
Vezmeme-li jako odhad bodu minima střed intervalu $s = \frac{a+b}{2}$, pak pro $f'(s) < 0$ leží minimum vpravo, klademe tedy $a^1 = s$ a pro $f'(s) > 0$ leží minimum vlevo a klademe $b^1 = s$, druhý krajní bod zůstává (pokud $f'(s) = 0$, našli jsme přímo bod optima x^*).

Výpočet se zastaví po stanoveném počtu kroků nebo klesne-li šířka intervalu pod 2ε , kde ε je požadovaná přesnost. Lze ukázat, že pak stačí provést

$$N \geq \frac{\log\left(\frac{b-a}{2\varepsilon}\right)}{\log 2} \text{ iterací.}$$

Jednorozměrná optimalizace - metoda bisekce, příklad

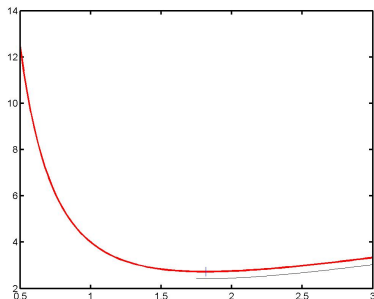
Ukažme si metodu opět pro minimalizaci funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$. Funkce má derivaci $f'(x) = 1 - \frac{6}{x^3}$.



Graf funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $a = 0,5$; $b = 3$, svislou čarou je znázorněno skutečné minimum $x^* = \sqrt[3]{6} \approx 1,817120$

Jednorozměrná optimalizace - metoda bisekce, příklad

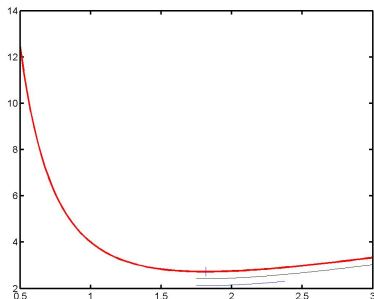
Ukažme si metodu opět pro minimalizaci funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$. Funkce má derivaci $f'(x) = 1 - \frac{6}{x^3}$.



1. krok: $s = 1.750000$, $f'(s) = -0.119534 \Rightarrow$ nový interval bude:
 $I^1 = \langle 1.750000, 3.000000 \rangle$

Jednorozměrná optimalizace - metoda bisekce, příklad

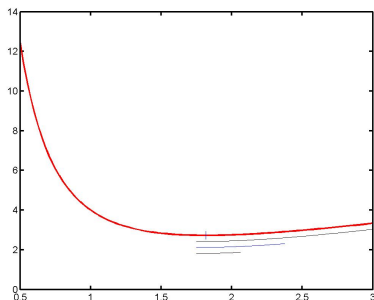
Ukažme si metodu opět pro minimalizaci funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$. Funkce má derivaci $f'(x) = 1 - \frac{6}{x^3}$.



2. krok: $s = 2.375000$, $f'(s) = 0.552121 \Rightarrow$ nový interval bude:
 $I^1 = \langle 1.750000, 2.375000 \rangle$

Jednorozměrná optimalizace - metoda bisekce, příklad

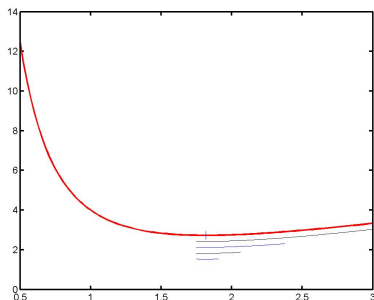
Ukažme si metodu opět pro minimalizaci funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$. Funkce má derivaci $f'(x) = 1 - \frac{6}{x^3}$.



3. krok: $s = 2.062500$, $f'(s) = 0.316137 \Rightarrow$ nový interval bude:
 $I^1 = \langle 1.750000, 2.062500 \rangle$

Jednorozměrná optimalizace - metoda bisekce, příklad

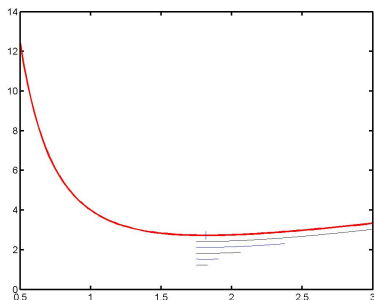
Ukažme si metodu opět pro minimalizaci funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$. Funkce má derivaci $f'(x) = 1 - \frac{6}{x^3}$.



4. krok: $s = 1.906250$, $f'(s) = 0.133813 \Rightarrow$ nový interval bude:
 $I^1 = \langle 1.750000, 1.906250 \rangle$

Jednorozměrná optimalizace - metoda bisekce, příklad

Ukažme si metodu opět pro minimalizaci funkce $f(x) = x + \frac{3}{x^2}$ na počátečním intervalu $\langle 0,5; 3 \rangle$ s přesností $\varepsilon = 0,05$. Funkce má derivaci $f'(x) = 1 - \frac{6}{x^3}$.



5. krok: $s = 1.828125$, $f'(s) = 0.017950 \Rightarrow$ nový interval bude:
 $I^1 = \langle 1.750000, 1.828125 \rangle$. protože šířka intervalu $< \varepsilon$, výpočet končí:
 $x^* \approx 1.789063 \pm 0,05$

Jednorozměrná optimalizace - metody využívající derivace

Další metoda hledá minimum jako kořen derivace. Nazývá se **Newtonova metoda** (též metoda tečen). Metoda využívá navíc i druhé derivace, předpokládejme tedy její existenci v každém bodě zadaného intervalu.

Jednorozměrná optimalizace - metody využívající derivace

Další metoda hledá minimum jako kořen derivace. Nazývá se **Newtonova metoda** (též metoda tečen). Metoda využívá navíc i druhé derivace, předpokládejme tedy její existenci v každém bodě zadaného intervalu. Jak název napovídá, vedeme v bodě x_0 tečnu ke grafu funkce $f'(x)$ a jako následující odhad vezmeme průsečík této tečny s osou x . Rovnici tečny lze zapsat jako: $y = f'(x_0) + f''(x_0) \cdot (x - x_0)$. Položíme-li pravou stranu rovnu

nule, spočteme odtud $x = x_0 - \frac{f'(x_0)}{f''(x_0)}$

Jednorozměrná optimalizace - metody využívající derivace

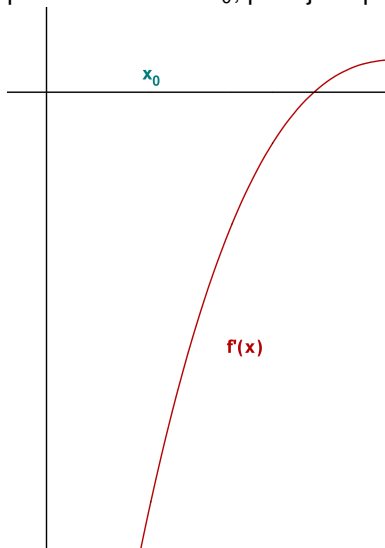
Další metoda hledá minimum jako kořen derivace. Nazývá se **Newtonova metoda** (též metoda tečen). Metoda využívá navíc i druhé derivace, předpokládejme tedy její existenci v každém bodě zadaného intervalu. Jak název napovídá, vedeme v bodě x_0 tečnu ke grafu funkce $f'(x)$ a jako následující odhad vezmeme průsečík této tečny s osou x . Rovnici tečny lze zapsat jako: $y = f'(x_0) + f''(x_0) \cdot (x - x_0)$. Položíme-li pravou stranu rovny

nule, spočteme odtud $x = x_0 - \frac{f'(x_0)}{f''(x_0)}$

Tuto hodnotu označíme x_1 a pokračujeme ve výpočtu až dokud není splněno $|x_n - x_{n-1}| < \varepsilon$. Předností metody je její rychlost, značnou nevýhodou je ale fakt, že nekonverguje vždy. Ke konvergenci postačuje, aby na výchozím intervalu $\langle a, b \rangle$ neměnila $f''(x)$ ani $f'''(x)$ znaménko a aby platilo $f'(a) \cdot f'(b) < 0$. Důležité je též dobrá volba bodu x_0 , doporučuje se volit tak, aby $f'(x_0) \cdot f'''(x_0) > 0$, jinak by x_1 neležel v intervalu $\langle a, b \rangle$.

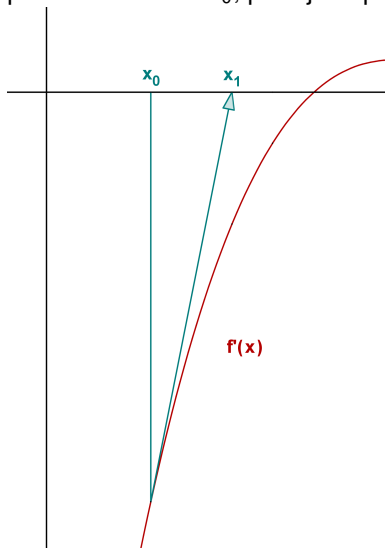
Jednorozměrná optimalizace - Newtonova metoda

Geometrická interpretace je naznačena na obrázku, kde vidíme funkci $f'(x)$ a počáteční iteraci x_0 , poté jsou provedeny tři další iterace.



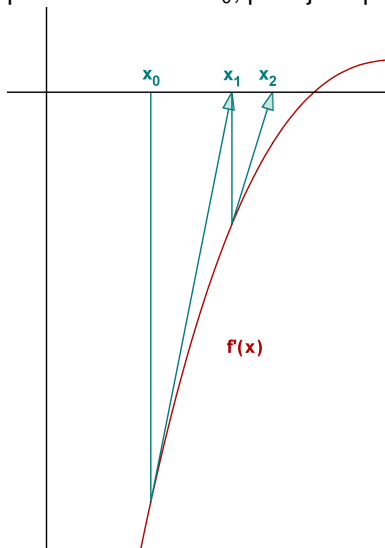
Jednorozměrná optimalizace - Newtonova metoda

Geometrická interpretace je naznačena na obrázku, kde vidíme funkci $f'(x)$ a počáteční iteraci x_0 , poté jsou provedeny tři další iterace.



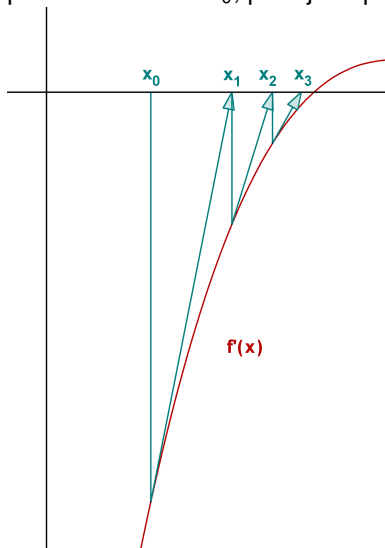
Jednorozměrná optimalizace - Newtonova metoda

Geometrická interpretace je naznačena na obrázku, kde vidíme funkci $f'(x)$ a počáteční iteraci x_0 , poté jsou provedeny tři další iterace.



Jednorozměrná optimalizace - Newtonova metoda

Geometrická interpretace je naznačena na obrázku, kde vidíme funkci $f'(x)$ a počáteční iteraci x_0 , poté jsou provedeny tři další iterace.



Jednorozměrná optimalizace, Newtonova metoda, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0, 5; 3 \rangle$.

Jednorozměrná optimalizace, Newtonova metoda, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Řešení: $f'(x) = 1 - \frac{6}{x^3}$, $f''(x) = \frac{18}{x^4}$.

Zvolme $x_0 = 1.750000$ (1. aproximace z metody bisekce)

Jednorozměrná optimalizace, Newtonova metoda, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Řešení: $f'(x) = 1 - \frac{6}{x^3}$, $f''(x) = \frac{18}{x^4}$.

Zvolme $x_0 = 1,750000$ (1. aproximace z metody bisekce)

1.krok: $x_0 = 1,750000$, $f'(x_0) = -0,119534$, $f''(x_0) = -1,919200 \Rightarrow$
 $x_1 = 1,812283$, $|x_1 - x_0| = 0,062283$

Jednorozměrná optimalizace, Newtonova metoda, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Řešení: $f'(x) = 1 - \frac{6}{x^3}$, $f''(x) = \frac{18}{x^4}$.

Zvolme $x_0 = 1,750000$ (1. aproximace z metody bisekce)

1.krok: $x_0 = 1,750000$, $f'(x_0) = -0,119534$, $f''(x_0) = -1,919200 \Rightarrow$
 $x_1 = 1,812283$, $|x_1 - x_0| = 0,062283$

2.krok: $x_1 = 1,812283$, $f'(x_1) = -0,008029$, $f''(x_1) = 1,668662 \Rightarrow$
 $x_2 = 1,817095$, $|x_2 - x_1| = 0,004812$, výpočet končí: $x^* \approx 1,817095 \pm 0,05$

Jednorozměrná optimalizace - metody využívající derivace

V předchozí metodě potřebujeme v každém kroku spočítat jak první, tak druhou derivaci funkce v daném bodě. Jelikož výpočet derivace funkce nemusí být vždy snadný, nahrazuje se tzv. **poměrnou diferencí**:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

My tuto aproximaci provedeme pro druhou derivaci:

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

Jednorozměrná optimalizace - metody využívající derivace

V předchozí metodě potřebujeme v každém kroku spočítat jak první, tak druhou derivaci funkce v daném bodě. Jelikož výpočet derivace funkce nemusí být vždy snadný, nahrazuje se tzv. **poměrnou diferencí**:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

My tuto aproximaci provedeme pro druhou derivaci:

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

Nahradíme-li v iteračním vzorci Newtonovy metody druhou derivaci uvedenou aproximací, dostáváme:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k)$$

Právě uvedený vzorec reprezentuje tzv. **metodu sečen**. K výpočtu jsou třeba dva počáteční body x_0, x_1 .

Jednorozměrná optimalizace - metody využívající derivace

V předchozí metodě potřebujeme v každém kroku spočítat jak první, tak druhou derivaci funkce v daném bodě. Jelikož výpočet derivace funkce nemusí být vždy snadný, nahrazuje se tzv. **poměrnou diferencí**:

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

My tuto aproximaci provedeme pro druhou derivaci:

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

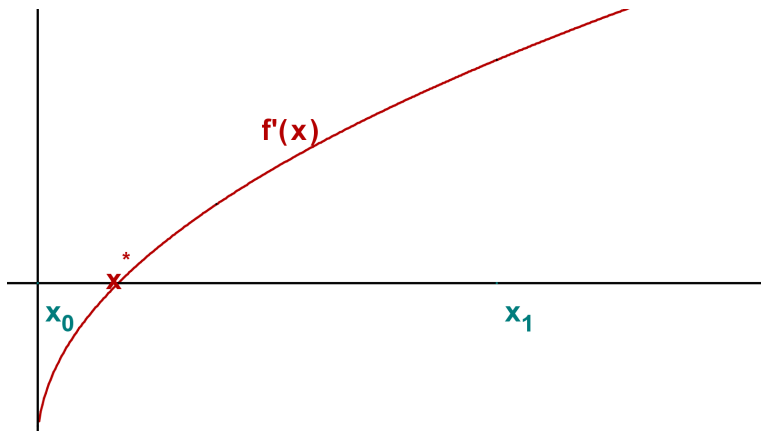
Nahradíme-li v iteračním vzorci Newtonovy metody druhou derivaci uvedenou aproximací, dostáváme:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})} f'(x_k)$$

Právě uvedený vzorec reprezentuje tzv. **metodu sečen**. K výpočtu jsou třeba dva počáteční body x_0, x_1 . Tato metoda také není vždy konvergentní, lze ji však modifikovat tak, že do vzorce použijeme místo dvou po sobě jdoucích iterací body x_m, x_k s co největšími indexy, tak, aby platilo $f'(x_m) \cdot f'(x_k) < 0$. Tuto modifikaci nazýváme **metoda regula falsi**.

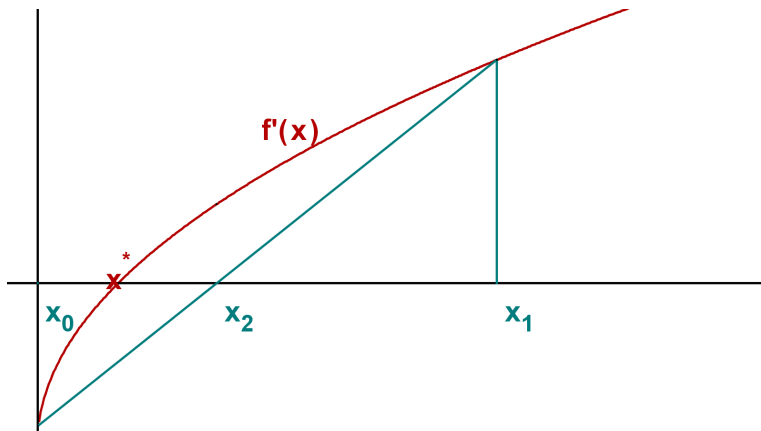
Jednorozměrná optimalizace - metoda regula falsi

Zázorněme si metodu graficky, bod optima je x^* a počáteční iterace jsou x_0 , x_1 , provedeme ještě další tři:



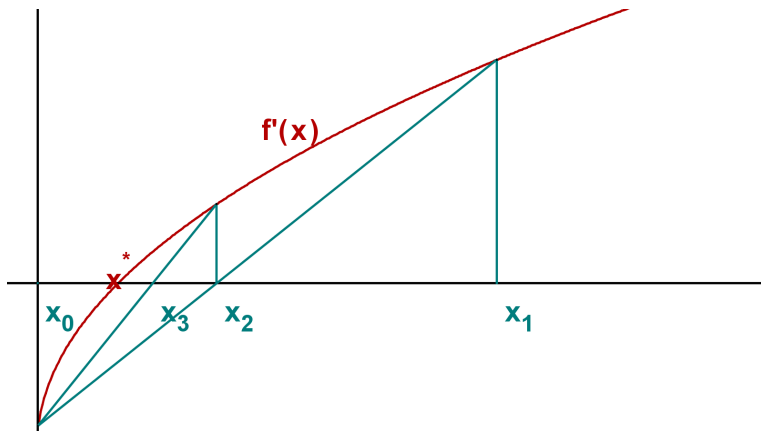
Jednorozměrná optimalizace - metoda regula falsi

Zázorněme si metodu graficky, bod optima je x^* a počáteční iterace jsou x_0 , x_1 , provedeme ještě další tři:



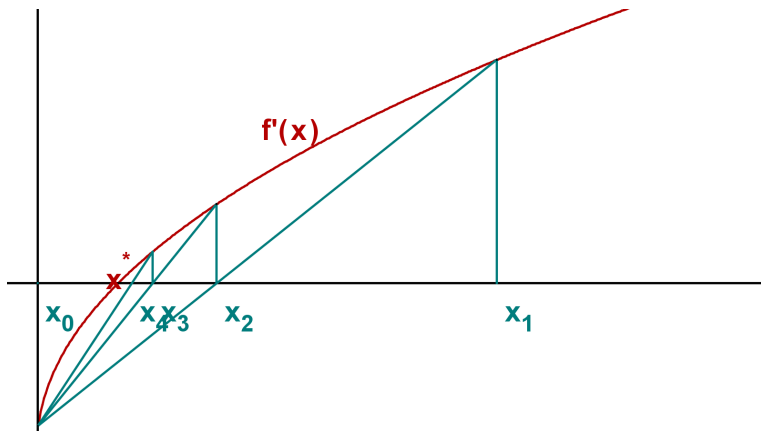
Jednorozměrná optimalizace - metoda regula falsi

Zázorněme si metodu graficky, bod optima je x^* a počáteční iterace jsou x_0 , x_1 , provedeme ještě další tři:



Jednorozměrná optimalizace - metoda regula falsi

Zázorněme si metodu graficky, bod optima je x^* a počáteční iterace jsou x_0 , x_1 , provedeme ještě další tři:



Jednorozměrná optimalizace - metoda regula falsi, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Jednorozměrná optimalizace - metoda regula falsi, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Řešení: Pro nalezení prvních dvou iterací můžeme použít např. metodu bisekce. Tedy: $x_0 = 1,750000$, $x_1 = 2,375000$.

Ověříme, že $f'(x_0)f'(x_1) = -0,119534 \cdot 0,552121 < 0$.

Jednorozměrná optimalizace - metoda regula falsi, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Řešení: Pro nalezení prvních dvou iterací můžeme použít např. metodu bisekce. Tedy: $x_0 = 1,750000$, $x_1 = 2,375000$.

Ověříme, že $f'(x_0)f'(x_1) = -0,119534 \cdot 0,552121 < 0$.

1.krok: $x_2 = x_1 - \frac{x_1 - x_0}{f'(x_1) - f'(x_0)} f'(x_1) = 1,861230$, $|x_2 - x_1| = 0,062283$

Protože $f'(x_2) = 0,069426 > 0$, použijeme dále x_0 a x_2 .

Jednorozměrná optimalizace - metoda regula falsi, příklad

Příklad: Nalezněte minimum funkce $f(x) = x + \frac{3}{x^2}$ s přesností $\varepsilon = 0,05$, víte-li, že se nalézá v intervalu $\langle 0,5; 3 \rangle$.

Řešení: Pro nalezení prvních dvou iterací můžeme použít např. metodu bisekce. Tedy: $x_0 = 1,750000$, $x_1 = 2,375000$.

Ověříme, že $f'(x_0)f'(x_1) = -0,119534 \cdot 0,552121 < 0$.

1.krok: $x_2 = x_1 - \frac{x_1 - x_0}{f'(x_1) - f'(x_0)} f'(x_1) = 1,861230$, $|x_2 - x_1| = 0,062283$

Protože $f'(x_2) = 0,069426 > 0$, použijeme dále x_0 a x_2 .

2.krok: $x_3 = x_2 - \frac{x_2 - x_0}{f'(x_2) - f'(x_0)} f'(x_2) = 1,820363$, $|x_3 - x_2| = 0,040867$,

výpočet končí: $x^* \approx 1,820363 \pm 0,05$.

Numerické metody bez omezení

- Komparativní metody
- Gradientní metody
- Newtonova metoda a její modifikace
- Gaussova-Newtonova metoda
- Metody konjugovaných směrů
- Metoda konjugovaných gradientů
- Kvazi-newtonovské metody
- Simulované žíhání, genetické algoritmy, tabu search, atd.

Numerické metody bez omezení

- Komparativní metody
- Gradientní metody
- Newtonova metoda a její modifikace
- Gaussova-Newtonova metoda
- Metody konjugovaných směrů
- Metoda konjugovaných gradientů
- Kvazi-newtonovské metody
- Simulované žíhání, genetické algoritmy, tabu search, atd.

Numerické metody s omezením

- Metody přípustných směrů
- Metody aktivních množin
- Metoda projekce gradientu
- Metoda redukovaného gradientu
- Metody pokutových a bariérových funkcí
- Metody vnitřního bodu
- Sekvenční kvadratické programování

Vícerozměrná optimalizace - Komparativní metody

Metoda cyklické záměny proměnných převádí vícerozměrnou optimalizaci na posloupnost jednorozměrných optimalizačních úloh ve směru jednotlivých souřadnic: z výchozího bodu provádíme minimalizaci ve směru první souřadnice. Po nalezení lokálního extrému funkce $f(\mathbf{x})$ ve směru

$\mathbf{s}_1 = (1, 0, \dots, 0)$, pokračujeme ve směru druhé souřadnice

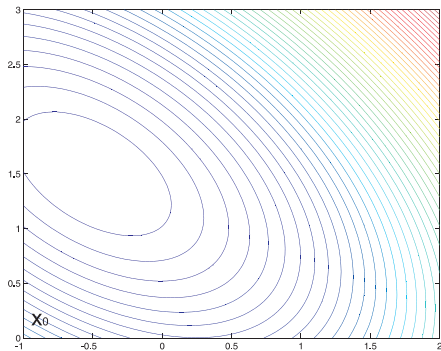
$\mathbf{s}_2 = (0, 1, \dots, 0)$ atd., až dostaneme první iteraci. Nevýhodou metody je pomalý výpočet, navíc není ani zaručena konvergence metody.

Vícerozměrná optimalizace - Komparativní metody

Metoda cyklické záměny proměnných převádí vícerozměrnou optimalizaci na posloupnost jednorozměrných optimalizačních úloh ve směru jednotlivých souřadnic: z výchozího bodu provádíme minimalizaci ve směru první souřadnice. Po nalezení lokálního extrému funkce $f(\mathbf{x})$ ve směru

$\mathbf{s}_1 = (1, 0, \dots, 0)$, pokračujeme ve směru druhé souřadnice

$\mathbf{s}_2 = (0, 1, \dots, 0)$ atd., až dostaneme první iteraci. Nevýhodou metody je pomalý výpočet, navíc není ani zaručena konvergence metody.

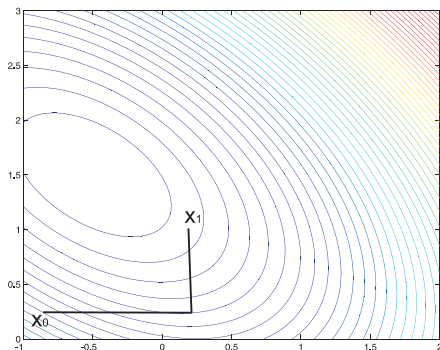


Vícerozměrná optimalizace - Komparativní metody

Metoda cyklické záměny proměnných převádí vícerozměrnou optimalizaci na posloupnost jednorozměrných optimalizačních úloh ve směru jednotlivých souřadnic: z výchozího bodu provádíme minimalizaci ve směru první souřadnice. Po nalezení lokálního extrému funkce $f(\mathbf{x})$ ve směru

$\mathbf{s}_1 = (1, 0, \dots, 0)$, pokračujeme ve směru druhé souřadnice

$\mathbf{s}_2 = (0, 1, \dots, 0)$ atd., až dostaneme první iteraci. Nevýhodou metody je pomalý výpočet, navíc není ani zaručena konvergence metody.

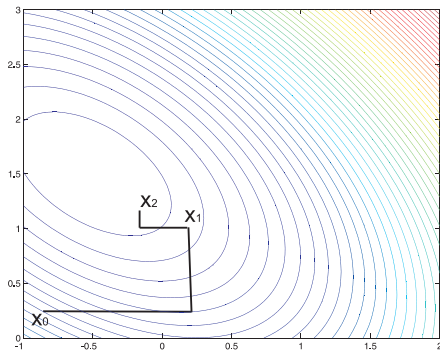


Vícerozměrná optimalizace - Komparativní metody

Metoda cyklické záměny proměnných převádí vícerozměrnou optimalizaci na posloupnost jednorozměrných optimalizačních úloh ve směru jednotlivých souřadnic: z výchozího bodu provádíme minimalizaci ve směru první souřadnice. Po nalezení lokálního extrému funkce $f(\mathbf{x})$ ve směru

$\mathbf{s}_1 = (1, 0, \dots, 0)$, pokračujeme ve směru druhé souřadnice

$\mathbf{s}_2 = (0, 1, \dots, 0)$ atd., až dostaneme první iteraci. Nevýhodou metody je pomalý výpočet, navíc není ani zaručena konvergence metody.

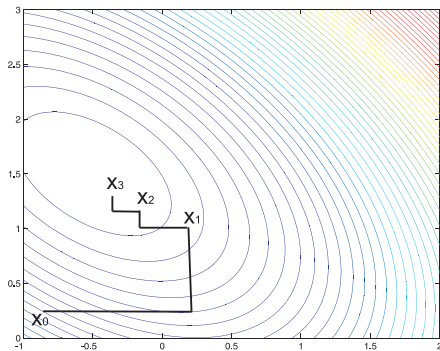


Vícerozměrná optimalizace - Komparativní metody

Metoda cyklické záměny proměnných převádí vícerozměrnou optimalizaci na posloupnost jednorozměrných optimalizačních úloh ve směru jednotlivých souřadnic: z výchozího bodu provádíme minimalizaci ve směru první souřadnice. Po nalezení lokálního extrému funkce $f(\mathbf{x})$ ve směru

$\mathbf{s}_1 = (1, 0, \dots, 0)$, pokračujeme ve směru druhé souřadnice

$\mathbf{s}_2 = (0, 1, \dots, 0)$ atd., až dostaneme první iteraci. Nevýhodou metody je pomalý výpočet, navíc není ani zaručena konvergence metody.



Metoda cyklické záměny proměnných - příklad

Řešme následující úlohu z knihy V. Pánková, Nelineární optimalizace pro ekonomy: Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody cyklické záměny proměnných. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$.

Metoda cyklické záměny proměnných - příklad

Řešme následující úlohu z knihy V. Pánková, Nelineární optimalizace pro ekonomy: Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody cyklické záměny proměnných. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$.

Řešení: Při postupu ve směru první proměnné optimalizujeme funkci $f(x_1, 1) = -12 + 4x_1^2 + 4 + 4x_1$, dostaneme $x_1 = -\frac{1}{2}$. Následně ve směru druhé proměnné minimalizujeme $f(-\frac{1}{2}, x_2) = -12x_2 + 1 + 4x_2^2 - 2x_2$, dostaneme $x_2 = \frac{7}{4}$.

Metoda cyklické záměny proměnných - příklad

Řešme následující úlohu z knihy V. Pánková, Nelineární optimalizace pro ekonomy: Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí **metody cyklické záměny proměnných**. Proveďte první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$.

Řešení: Při postupu ve směru první proměnné optimalizujeme funkci $f(x_1, 1) = -12 + 4x_1^2 + 4 + 4x_1$, dostaneme $x_1 = -\frac{1}{2}$. Následně ve směru druhé proměnné minimalizujeme $f(-\frac{1}{2}, x_2) = -12x_2 + 1 + 4x_2^2 - 2x_2$, dostaneme $x_2 = \frac{7}{4}$.

Dále postupujeme z bodu $\mathbf{x}^1 = (-\frac{1}{2}, \frac{7}{4})^\top$ ve směru první proměnné: minimalizujeme $f(x_1, \frac{7}{4}) = -21 + 4x_1^2 + \frac{49}{4} + 7x_1$, dostaneme $x_1 = -\frac{7}{8}$. Poté ve směru druhé proměnné: funkce $f(-\frac{7}{8}, x_2) = -12x_2 + \frac{196}{64} + 4x_2^2 - \frac{7}{2}x_2$ nabývá minima pro $x_2 = \frac{31}{16}$.

Metoda cyklické záměny proměnných - příklad

Řešme následující úlohu z knihy V. Pánková, Nelineární optimalizace pro ekonomy: Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí **metody cyklické záměny proměnných**. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$.

Řešení: Při postupu ve směru první proměnné optimalizujeme funkci $f(x_1, 1) = -12 + 4x_1^2 + 4 + 4x_1$, dostaneme $x_1 = -\frac{1}{2}$. Následně ve směru druhé proměnné minimalizujeme $f(-\frac{1}{2}, x_2) = -12x_2 + 1 + 4x_2^2 - 2x_2$, dostaneme $x_2 = \frac{7}{4}$.

Dále postupujeme z bodu $\mathbf{x}^1 = (-\frac{1}{2}, \frac{7}{4})^\top$ ve směru první proměnné: minimalizujeme $f(x_1, \frac{7}{4}) = -21 + 4x_1^2 + \frac{49}{4} + 7x_1$, dostaneme $x_1 = -\frac{7}{8}$. Poté ve směru druhé proměnné: funkce $f(-\frac{7}{8}, x_2) = -12x_2 + \frac{196}{64} + 4x_2^2 - \frac{7}{2}x_2$ nabývá minima pro $x_2 = \frac{31}{16}$.

Dostali jsme odhad $\mathbf{x}^2 = (-\frac{7}{8}, \frac{31}{16})^\top$, skutečné minimum nastává v bodě $\mathbf{x}^* = (-1, 2)^\top$.

Vícerozměrná optimalizace - Komparativní metody

Metoda pravidelného a flexibilního simplexu: Startuje vytvořením výchozího **simplexu** v zadaném prostoru (Simplexem v n - rozměrném prostoru rozumíme konvexní obal $n + 1$ vrcholů v obecné poloze, tj. v rovině je to trojúhelník, v třírozměrném prostoru je to čtyřstěn atd.). Z vrcholů nahradíme ten s nejhorší hodnotou účelové funkce pomocí **reflexe** vzhledem k těžišti ostatních vrcholů novým vrcholem, čímž vznikne nový simplex, tj. další iterace. Metoda je heuristická, ale názorná a snadno implementovatelná.

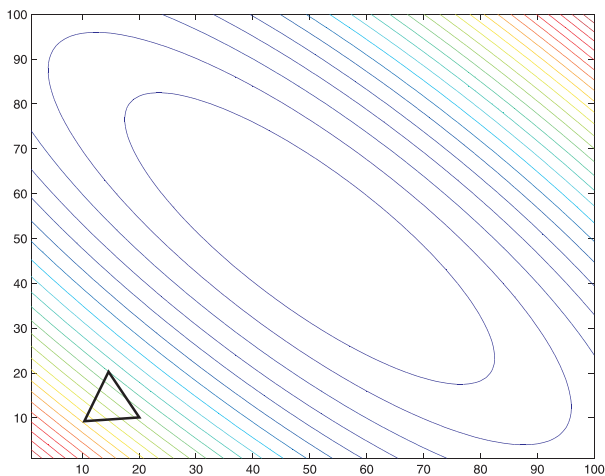
Vícerozměrná optimalizace - Komparativní metody

Metoda pravidelného a flexibilního simplexu: Startuje vytvořením výchozího **simplexu** v zadaném prostoru (Simplexem v n - rozměrném prostoru rozumíme konvexní obal $n + 1$ vrcholů v obecné poloze, tj. v rovině je to trojúhelník, v třírozměrném prostoru je to čtyřstěn atd.). Z vrcholů nahradíme ten s nejhorší hodnotou účelové funkce pomocí **reflexe** vzhledem k těžišti ostatních vrcholů novým vrcholem, čímž vznikne nový simplex, tj. další iterace. Metoda je heuristická, ale názorná a snadno implementovatelná.

Rychlost a přesnost nalezení optimálního bodu záleží na velikosti simplexu. Čím větší je simplex, tím se rychleji blížíme k optimu. Přesnost výpočtu naopak vyžaduje malou velikost simplexu. Proto je při výpočtu třeba měnit délku hrany simplexu. V **Nelder-Meadově metodě** se toto zajistí pomocí operací **expanze, kontrakce a redukce** základního simplexu.

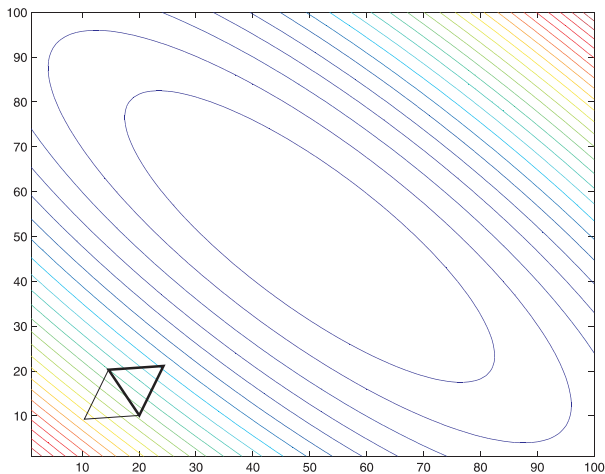
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



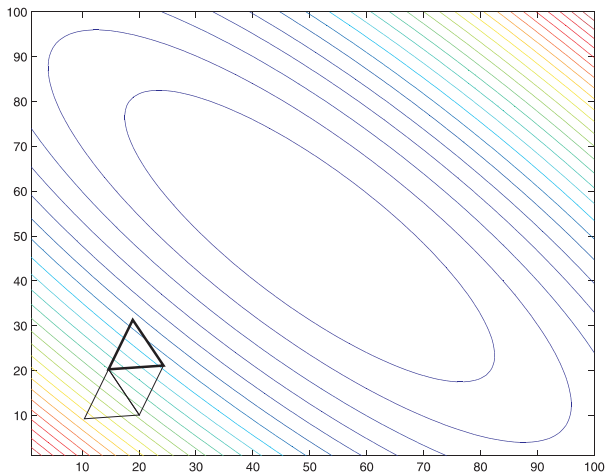
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



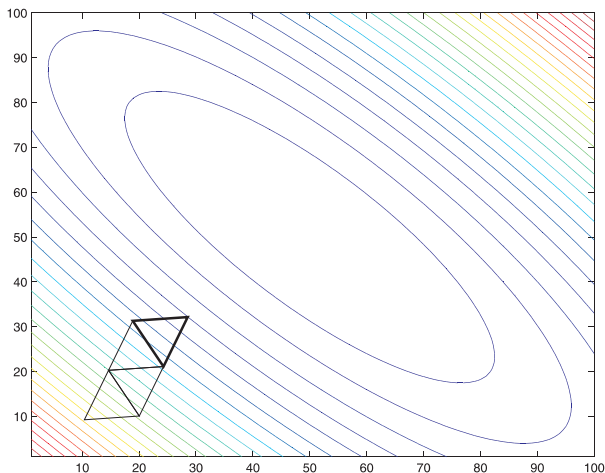
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



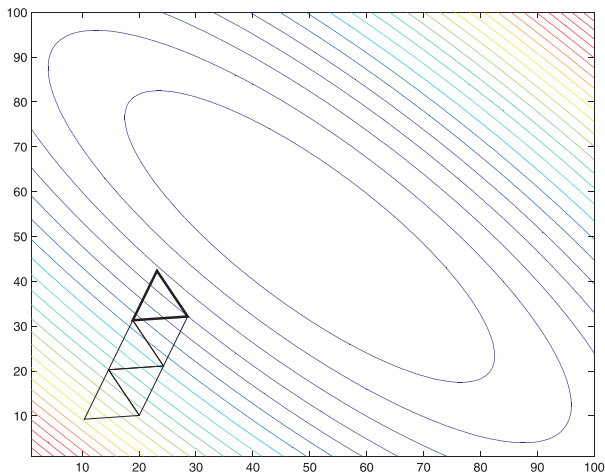
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



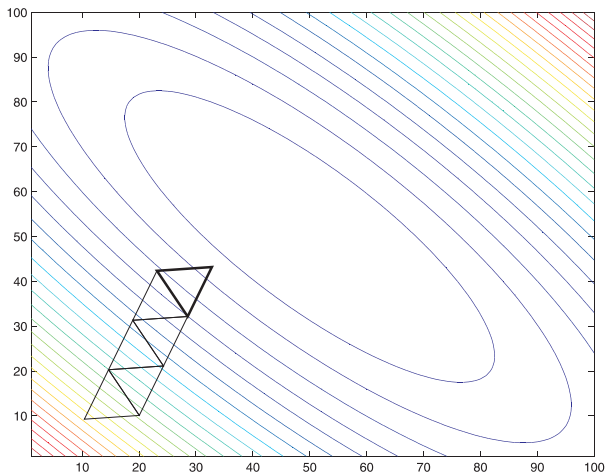
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



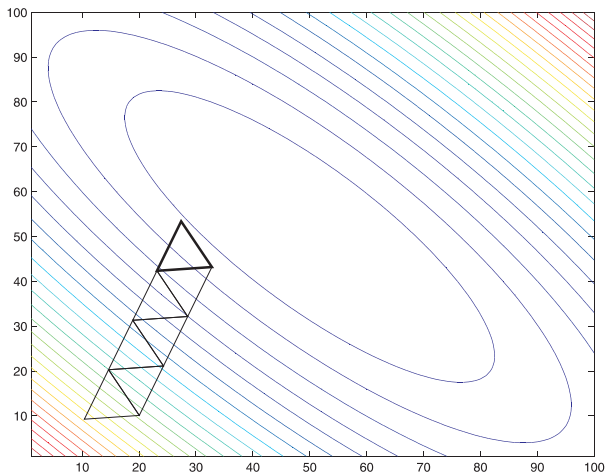
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



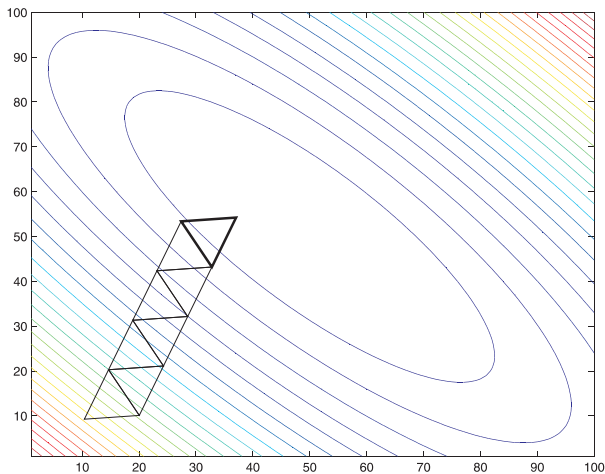
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



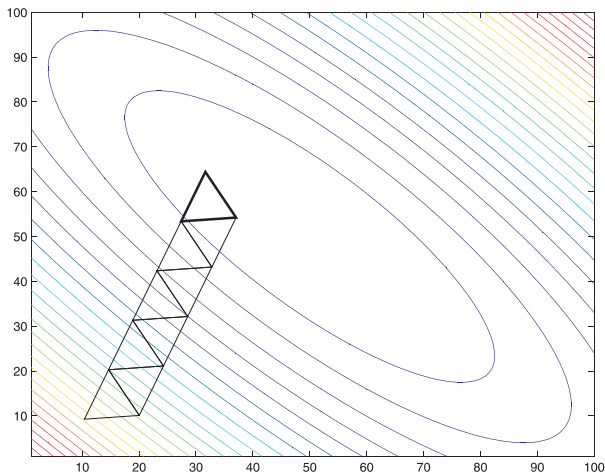
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



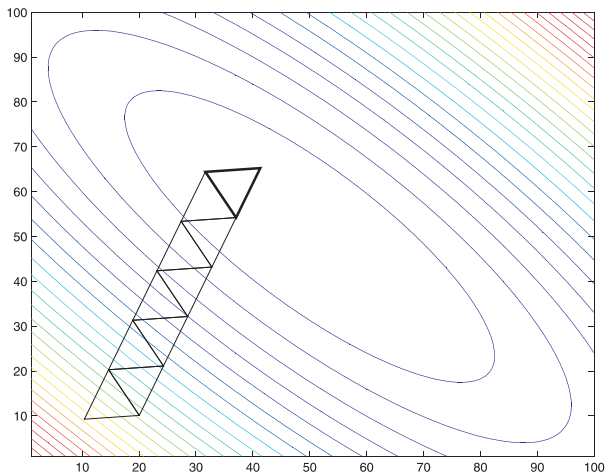
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



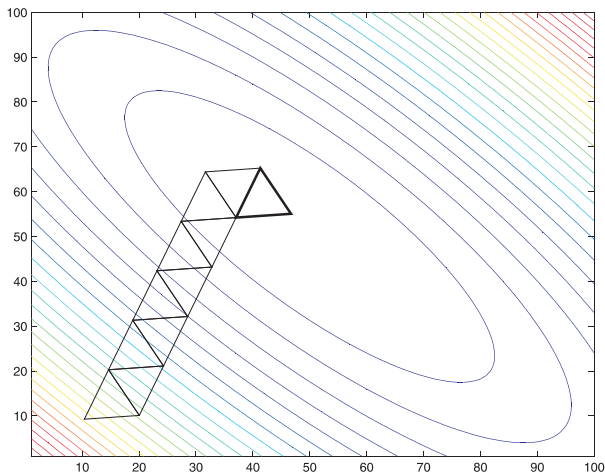
Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



Vícerozměrná optimalizace - metoda pravidelného simplexu

Pro volbu pevné velikosti simplexu dosáhneme pouze omezené přesnosti.



Vícerozměrná optimalizace - gradientní metody

K nejstarším a zároveň nejpoužívanějším přístupům vícerozměrné numerické optimalizace patří **gradientové metody**. Ukažme si obecný princip gradientové metody pro minimalizaci diferencovatelné funkce $f(\mathbf{x})$ pro $\mathbf{x} \in \mathbb{R}^n$:

- 1 urči výchozí bod \mathbf{x}^0
- 2 urči gradient funkce f v tomto bodě: $\nabla f(\mathbf{x}^0)$
- 3 přejdi z bodu \mathbf{x}^0 do bodu \mathbf{x}^1 ve směru "antigradientu" $-\nabla f(\mathbf{x}^0)$ tak, aby $f(\mathbf{x}^0) > f(\mathbf{x}^1)$.
- 4 celý postup opakuj z bodu \mathbf{x}^1 , atd. dokud není splněno pravidlo ukončení výpočtu

Vícerozměrná optimalizace - gradientní metody

K nejstarším a zároveň nejpoužívanějším přístupům vícerozměrné numerické optimalizace patří **gradientové metody**. Ukažme si obecný princip gradientové metody pro minimalizaci diferencovatelné funkce $f(\mathbf{x})$ pro $\mathbf{x} \in \mathbb{R}^n$:

- 1 urči výchozí bod \mathbf{x}^0
- 2 urči gradient funkce f v tomto bodě: $\nabla f(\mathbf{x}^0)$
- 3 přejdi z bodu \mathbf{x}^0 do bodu \mathbf{x}^1 ve směru "antigradientu" $-\nabla f(\mathbf{x}^0)$ tak, aby $f(\mathbf{x}^0) > f(\mathbf{x}^1)$.
- 4 celý postup opakuj z bodu \mathbf{x}^1 , atd. dokud není splněno pravidlo ukončení výpočtu

Výpočet je možné ukončit:

- po provedení předem stanoveného počtu iterací
- pokud hodnota účelové funkce klesne o méně než předem stanovené ε , tedy $f(\mathbf{x}^i) - f(\mathbf{x}^{i+1}) < \varepsilon$
- pokud vzdálenost dvou po sobě jdoucích iterací je menší než předem stanovené ε , tedy $|\mathbf{x}^i - \mathbf{x}^{i+1}| < \varepsilon$, kde symbolem $|\cdot|$ rozumíme vzdálenost vycházející z euklidovské metriky dané vztahem $|\mathbf{x}| = \sqrt{\sum_{i=1}^n x_i^2}$.

Vícerozměrná optimalizace - gradientová metoda s pevným krokem

V obecném principu gradientových metod je třeba specifikovat, jak přecházet z bodu \mathbf{x}^i do bodu \mathbf{x}^{i+1} . Víme, že máme postupovat ve směru opačném ke gradientu, ale je třeba stanovit délku kroku. U gradientové **metody s pevným krokem** je stanovena počáteční délka iteračního kroku, označme ji α . Dále se postupuje následovně:

Spočítáme $\mathbf{x}^{i+1} = \mathbf{x}^i - \alpha \frac{\nabla f(\mathbf{x}^i)}{|\nabla f(\mathbf{x}^i)|}$. Potom:

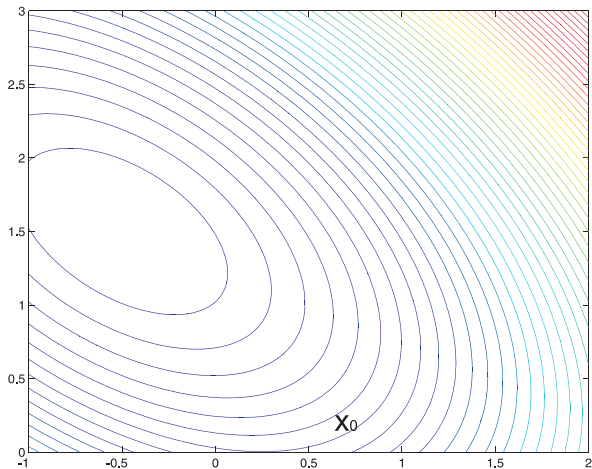
- je-li $f(\mathbf{x}^{i+1}) < f(\mathbf{x}^i)$, pak pokračujeme s další iterací
- jestliže nedošlo k poklesu účelové funkce, pak zmenšíme α (zpravidla na polovinu), iteraci \mathbf{x}^{i+1} přepočítáme a dále už pokračujeme s novou hodnotou α . Při tomto přístupu se nabízí možnost ukončit výpočet při splnění podmínky $\alpha < \varepsilon$

Vícerozměrná optimalizace - gradientová metoda s pevným krokem

Znázorněme si postup metody graficky. V každé iteraci vycházíme ve směru kolmém k vrstevnicím.

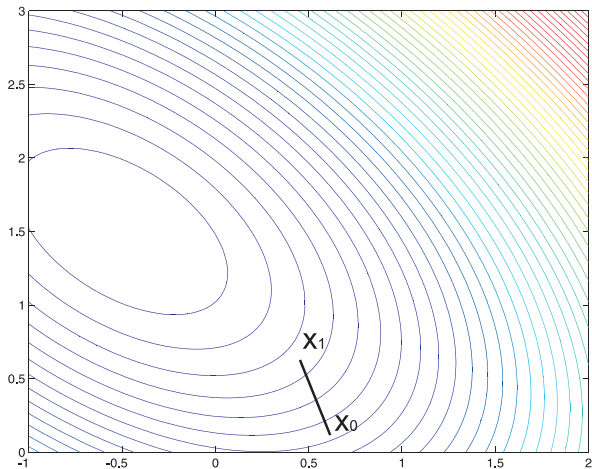
Vícerozměrná optimalizace - gradientová metoda s pevným krokem

Znáznorněme si postup metody graficky. V každé iteraci vycházíme ve směru kolmém k vrstevnicím.



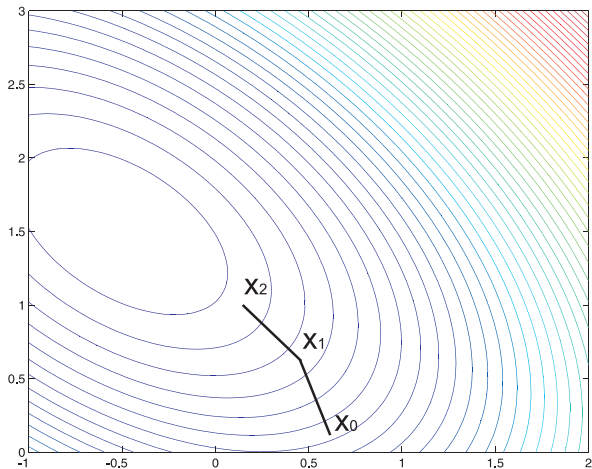
Vícerozměrná optimalizace - gradientová metoda s pevným krokem

Znáznorněme si postup metody graficky. V každé iteraci vycházíme ve směru kolmém k vrstevnicím.



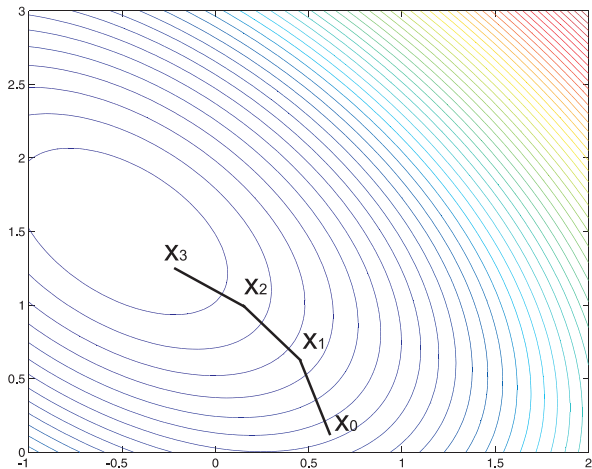
Vícerozměrná optimalizace - gradientová metoda s pevným krokem

Znáznorněme si postup metody graficky. V každé iteraci vycházíme ve směru kolmém k vrstevnicím.



Vícerozměrná optimalizace - gradientová metoda s pevným krokem

Znáznorněme si postup metody graficky. V každé iteraci vycházíme ve směru kolmém k vrstevnicím.



Gradientová metoda s pevným krokem-příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x^2 + 4x_1x_2$ pomocí metody s pevným krokem $\alpha = 0, 1$. Proved'te první tři iterace z bodu $\mathbf{x}^0 = (0, 1)^\top$.

Gradientová metoda s pevným krokem-příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody s pevným krokem $\alpha = 0,1$. Proved'te první tři iterace z bodu $\mathbf{x}^0 = (0, 1)^\top$.

Řešení: Spočteme gradient $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$ a vyčíslíme

$$\mathbf{x}^1 = \mathbf{x}^0 - \alpha \cdot \frac{\nabla f(\mathbf{x}^0)}{|\nabla f(\mathbf{x}^0)|} = (0, 1)^\top - 0,1 \cdot (4, -4)^\top \cdot \frac{1}{\sqrt{32}} = (-0.0707; 1.0707)^\top.$$

Gradientová metoda s pevným krokem-příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody s pevným krokem $\alpha = 0,1$. Proved'te první tři iterace z bodu $\mathbf{x}^0 = (0, 1)^\top$.

Řešení: Spočteme gradient $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$ a vyčíslíme

$$\mathbf{x}^1 = \mathbf{x}^0 - \alpha \cdot \frac{\nabla f(\mathbf{x}^0)}{|\nabla f(\mathbf{x}^0)|} = (0, 1)^\top - 0,1 \cdot (4, -4)^\top \cdot \frac{1}{\sqrt{32}} = (-0.0707; 1.0707)^\top.$$

$$\mathbf{x}^2 = \mathbf{x}^1 - \alpha \cdot \frac{\nabla f(\mathbf{x}^1)}{|\nabla f(\mathbf{x}^1)|} = (-0.0707; 1.0707)^\top - 0,1 \cdot (3.7172; -3.7172)^\top \cdot \frac{1}{5.2569} = (-0.1414; 1.1414)^\top.$$

Gradientová metoda s pevným krokem-příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody s pevným krokem $\alpha = 0,1$. Proved'te první tři iterace z bodu $\mathbf{x}^0 = (0, 1)^\top$.

Řešení: Spočteme gradient $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$ a vyčíslíme

$$\mathbf{x}^1 = \mathbf{x}^0 - \alpha \cdot \frac{\nabla f(\mathbf{x}^0)}{|\nabla f(\mathbf{x}^0)|} = (0, 1)^\top - 0,1 \cdot (4, -4)^\top \cdot \frac{1}{\sqrt{32}} = (-0.0707; 1.0707)^\top.$$

$$\mathbf{x}^2 = \mathbf{x}^1 - \alpha \cdot \frac{\nabla f(\mathbf{x}^1)}{|\nabla f(\mathbf{x}^1)|} = (-0.0707; 1.0707)^\top - 0,1 \cdot (3.7172; -3.7172)^\top \cdot \frac{1}{5.2569} = (-0.1414; 1.1414)^\top.$$

$$\mathbf{x}^3 = \mathbf{x}^2 - \alpha \cdot \frac{\nabla f(\mathbf{x}^2)}{|\nabla f(\mathbf{x}^2)|} = (-0.1414; 1.1414)^\top - 0,1 \cdot (3.4343; -3.4343)^\top \cdot \frac{1}{4.8569} = (-0.2121; 1.2121)^\top.$$

Vícerozměrná optimalizace - metoda největšího spádu

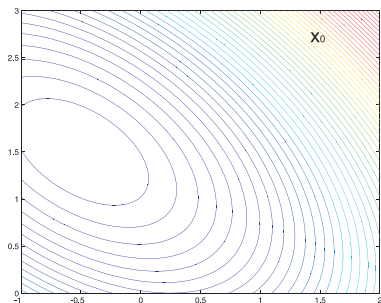
Narozdíl od předchozí metody je velikost iteračního kroku proměnlivá. Novou iteraci \mathbf{x}^{i+1} hledáme tak, že postupujeme z \mathbf{x}^i ve směru antigradientu tak dlouho, dokud účelová funkce klesá. Jinými slovy, zvolíme délku α_i tak, aby hodnota $f(\mathbf{x}^i - \alpha \nabla f(\mathbf{x}^i))$ byla **minimální**. Tuto jednorozměrnou minimalizační úlohu můžeme řešit analyticky nebo numericky.

Vícerozměrná optimalizace - metoda největšího spádu

Narozdíl od předchozí metody je velikost iteračního kroku proměnlivá. Novou iteraci \mathbf{x}^{i+1} hledáme tak, že postupujeme z \mathbf{x}^i ve směru antigradientu tak dlouho, dokud účelová funkce klesá. Jinými slovy, zvolíme délku α_i tak, aby hodnota $f(\mathbf{x}^i - \alpha \nabla f(\mathbf{x}^i))$ byla **minimální**. Tuto jednorozměrnou minimalizační úlohu můžeme řešit analyticky nebo numericky. Metoda největšího spádu rychleji konverguje než metoda s pevným krokem, která navíc může být silně ovlivněna počáteční volbou α . To se samozřejmě odráží i v časové výpočetní náročnosti výpočtu. Na druhou stranu metoda s pevným krokem může být snadnější na implementaci.

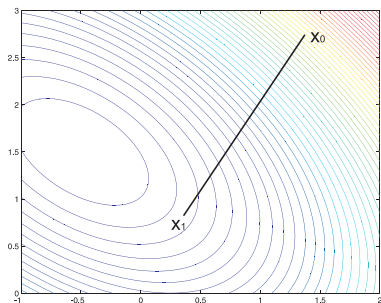
Vícerozměrná optimalizace - metoda největšího spádu

Narozdíl od předchozí metody je velikost iteračního kroku proměnlivá. Novou iteraci \mathbf{x}^{i+1} hledáme tak, že postupujeme z \mathbf{x}^i ve směru antigradientu tak dlouho, dokud účelová funkce klesá. Jinými slovy, zvolíme délku α_i tak, aby hodnota $f(\mathbf{x}^i - \alpha \nabla f(\mathbf{x}^i))$ byla **minimální**. Tuto jednorozměrnou minimalizační úlohu můžeme řešit analyticky nebo numericky. Metoda největšího spádu rychleji konverguje než metoda s pevným krokem, která navíc může být silně ovlivněna počáteční volbou α . To se samozřejmě odráží i v časové výpočetní náročnosti výpočtu. Na druhou stranu metoda s pevným krokem může být snadnější na implementaci.



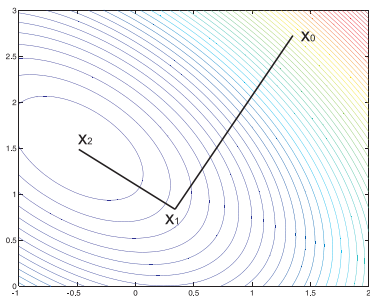
Vícerozměrná optimalizace - metoda největšího spádu

Narozdíl od předchozí metody je velikost iteračního kroku proměnlivá. Novou iteraci \mathbf{x}^{i+1} hledáme tak, že postupujeme z \mathbf{x}^i ve směru antigradientu tak dlouho, dokud účelová funkce klesá. Jinými slovy, zvolíme délku α_i tak, aby hodnota $f(\mathbf{x}^i - \alpha \nabla f(\mathbf{x}^i))$ byla **minimální**. Tuto jednorozměrnou minimalizační úlohu můžeme řešit analyticky nebo numericky. Metoda největšího spádu rychleji konverguje než metoda s pevným krokem, která navíc může být silně ovlivněna počáteční volbou α . To se samozřejmě odráží i v časové výpočetní náročnosti výpočtu. Na druhou stranu metoda s pevným krokem může být snadnější na implementaci.



Vícerozměrná optimalizace - metoda největšího spádu

Narozdíl od předchozí metody je velikost iteračního kroku proměnlivá. Novou iteraci \mathbf{x}^{i+1} hledáme tak, že postupujeme z \mathbf{x}^i ve směru antigradientu tak dlouho, dokud účelová funkce klesá. Jinými slovy, zvolíme délku α_i tak, aby hodnota $f(\mathbf{x}^i - \alpha \nabla f(\mathbf{x}^i))$ byla **minimální**. Tuto jednorozměrnou minimalizační úlohu můžeme řešit analyticky nebo numericky. Metoda největšího spádu rychleji konverguje než metoda s pevným krokem, která navíc může být silně ovlivněna počáteční volbou α . To se samozřejmě odráží i v časové výpočetní náročnosti výpočtu. Na druhou stranu metoda s pevným krokem může být snadnější na implementaci.



Metoda největšího spádu- příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x^2 + 4x_1x_2$ pomocí metody největšího spádu. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (0, 0)^\top$.

Metoda největšího spádu- příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody největšího spádu. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (0, 0)^\top$.

Řešení: Víme, že $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$. Minimalizujeme tedy funkci $g_1(\alpha) = f(\mathbf{x}^0 - \alpha \cdot \nabla f(\mathbf{x}^0)) = f(0, 12\alpha) = -144\alpha + 576\alpha^2$, dostaneme optimum pro $\alpha = 0,125$.

Metoda největšího spádu- příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody největšího spádu. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (0, 0)^\top$.

Řešení: Víme, že $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$. Minimalizujeme tedy funkci $g_1(\alpha) = f(\mathbf{x}^0 - \alpha \cdot \nabla f(\mathbf{x}^0)) = f(0, 12\alpha) = -144\alpha + 576\alpha^2$, dostaneme optimum pro $\alpha = 0,125$.

Tedy $\mathbf{x}_1 = \mathbf{x}^0 - 0,125 \cdot \nabla f(\mathbf{x}^0) = (0; 1,5)^\top$. Dále minimalizujeme funkci $g_2(\alpha) = f(\mathbf{x}^1 - \alpha \cdot \nabla f(\mathbf{x}^1)) = f(-6\alpha; 1,5) = -9 - 36\alpha + 144\alpha^2$, minimum opět nastává pro $\alpha = 0,125$.

Metoda největšího spádu- příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x^2 + 4x_1x_2$ pomocí metody největšího spádu. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (0, 0)^\top$.

Řešení: Víme, že $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$. Minimalizujeme tedy funkci $g_1(\alpha) = f(\mathbf{x}^0 - \alpha \cdot \nabla f(\mathbf{x}^0)) = f(0, 12\alpha) = -144\alpha + 576\alpha^2$, dostaneme optimum pro $\alpha = 0,125$.

Tedy $\mathbf{x}_1 = \mathbf{x}^0 - 0,125 \cdot \nabla f(\mathbf{x}^0) = (0; 1,5)^\top$. Dále minimalizujeme funkci $g_2(\alpha) = f(\mathbf{x}^1 - \alpha \cdot \nabla f(\mathbf{x}^1)) = f(-6\alpha; 1,5) = -9 - 36\alpha + 144\alpha^2$, minimum opět nastává pro $\alpha = 0,125$.

Tedy $\mathbf{x}_2 = \mathbf{x}^1 - 0,125 \cdot \nabla f(\mathbf{x}^1) = (-0,75; 1,5)^\top$. Dále bychom minimalizovali funkci $g_3(\alpha) = f(\mathbf{x}^2 - \alpha \cdot \nabla f(\mathbf{x}^2)) = f(-0,75; 1,5 + 0,375\alpha)$, atd.

Vícerozměrná optimalizace - newtonovské metody

Podobně jako v jednorozměrném případě v **newtonovských metodách** funkci $f(\mathbf{x})$ aproximujeme při přechodu z bodu \mathbf{x}_k kvadratickou funkcí, tedy Taylorovým polynomem

$$T_2(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top \cdot (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^\top \cdot \mathbf{H}(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k)$$

a zapíšeme podmínku pro stacionární bod $\nabla T_2(\mathbf{x}) = \mathbf{0}$, neboli

$$\nabla f(\mathbf{x}_k)^\top + \mathbf{H}(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k) = \mathbf{0}.$$

Vícerozměrná optimalizace - newtonovské metody

Podobně jako v jednorozměrném případě v **newtonovských metodách** funkci $f(\mathbf{x})$ aproximujeme při přechodu z bodu \mathbf{x}_k kvadratickou funkcí, tedy Taylorovým polynomem

$$T_2(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top \cdot (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^\top \cdot \mathbf{H}(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k)$$

a zapíšeme podmínku pro stacionární bod $\nabla T_2(\mathbf{x}) = \mathbf{0}$, neboli

$$\nabla f(\mathbf{x}_k)^\top + \mathbf{H}(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k) = \mathbf{0}.$$

Další iteraci \mathbf{x}_{k+1} dostaneme jako řešení této soustavy.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k)$$

Vícerozměrná optimalizace - newtonovské metody

Podobně jako v jednorozměrném případě v **newtonovských metodách** funkci $f(\mathbf{x})$ aproximujeme při přechodu z bodu \mathbf{x}_k kvadratickou funkcí, tedy Taylorovým polynomem

$$T_2(\mathbf{x}) = f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^\top \cdot (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_k)^\top \cdot \mathbf{H}(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k)$$

a zapíšeme podmínku pro stacionární bod $\nabla T_2(\mathbf{x}) = \mathbf{0}$, neboli

$$\nabla f(\mathbf{x}_k)^\top + H(\mathbf{x}_k) \cdot (\mathbf{x} - \mathbf{x}_k) = \mathbf{0}.$$

Další iteraci \mathbf{x}_{k+1} dostaneme jako řešení této soustavy.

$$\mathbf{x}_{k+1} = \mathbf{x}_k - H(\mathbf{x}_k)^{-1} \cdot \nabla f(\mathbf{x}_k)$$

Pokud startujeme daleko od minima, je kvadratická aproximace nepřesná a Hessova matice může být singulární nebo negativně definitní. Newtonova metoda pak nefunguje nebo vede k maximu. Proto se používají **modifikace Newtonovy metody**, například **Levenberg - Marquardtova metoda** nebo **trust region algoritmus**.

Newtonovské metody - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí

Newtonovy metody . Proveďte první iteraci z bodu $\mathbf{x}^0 = (0, 0)^T$.

Newtonovské metody - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí **Newtonovy metody**. Proveďte první iteraci z bodu $\mathbf{x}^0 = (0, 0)^\top$.

Řešení: Již známe gradient $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$.

Spočteme Hessovu matici: $\mathbf{H} = \begin{pmatrix} 8 & 4 \\ 4 & 8 \end{pmatrix}$, její inverze je

$$\mathbf{H}^{-1} = \begin{pmatrix} \frac{2}{12} & \frac{-1}{12} \\ \frac{-1}{12} & \frac{2}{12} \end{pmatrix}.$$

Newtonovské metody - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí **Newtonovy metody**. Proveďte první iteraci z bodu $\mathbf{x}^0 = (0, 0)^\top$.

Řešení: Již známe gradient $\nabla f(x_1, x_2) = (8x_1 + 4x_2, -12 + 4x_1 + 8x_2)^\top$.

Spočteme Hessovu matici: $\mathbf{H} = \begin{pmatrix} 8 & 4 \\ 4 & 8 \end{pmatrix}$, její inverze je

$$\mathbf{H}^{-1} = \begin{pmatrix} \frac{2}{12} & \frac{-1}{12} \\ \frac{-1}{12} & \frac{2}{12} \end{pmatrix}.$$

Tedy můžeme spočítat první iteraci

$$\mathbf{x}^1 = \mathbf{x}^0 - \mathbf{H}^{-1} \cdot \nabla f(\mathbf{x}^0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} \frac{2}{12} & \frac{-1}{12} \\ \frac{-1}{12} & \frac{2}{12} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -12 \end{pmatrix} = (-1, 2)^\top.$$

Nalezli jsme přesně bod optima.

Vícerozměrná optimalizace - metoda sdružených směrů

Metody **sdružených (konjugovaných) směrů** byly vyvinuty proto, aby urychlily konvergenci gradientních metod a vyhnuly se potížím spojeným s modifikací Newtonovy metody. Nejprve uveďme definici:

Definice: Vektory $\mathbf{s}_1, \dots, \mathbf{s}_p$ jsou konjugované vzhledem k symetrické matici \mathbf{Q} tehdy, když platí $\mathbf{s}_i^T \cdot \mathbf{Q} \cdot \mathbf{s}_j = 0, \forall 1 \leq i, j \leq p, i \neq j$

Vícerozměrná optimalizace - metoda sdružených směrů

Metody **sdružených (konjugovaných) směrů** byly vyvinuty proto, aby urychlily konvergenci gradientních metod a vyhnuly se potížím spojeným s modifikací Newtonovy metody. Nejprve uveďme definici:

Definice: Vektory $\mathbf{s}_1, \dots, \mathbf{s}_p$ jsou konjugované vzhledem k symetrické matici \mathbf{Q} tehdy, když platí $\mathbf{s}_i^T \cdot \mathbf{Q} \cdot \mathbf{s}_j = 0, \forall 1 \leq i, j \leq p, i \neq j$

Pokud funkci $f(\mathbf{x})$ optimalizujeme postupně ve směrech $\mathbf{s}_1, \dots, \mathbf{s}_p$ konjugovaných vzhledem k Hessově matici \mathbf{H} , pro kvadratické funkce máme zajištěnu konvergenci pro $p = n$. Pro nekvadratické funkce je nutné metodu nejpozději po $n + 1$ krocích nastartovat znovu. Metoda bohužel není konstruktivní, neříká, jak sdružené směry určit. V případě, že se směry generují z gradientů a dosavadního směru pohybu, dostaneme modifikaci známou jako **metoda sdružených gradientů**. Další modifikací je metoda paralelních tečen, **PARTAN**.

Metoda sdružených směrů - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody konjugovaných směrů. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$ a jako výchozí směr vezměte $\mathbf{s}_1 = (1, 0)$.

Metoda sdružených směrů - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody konjugovaných směrů. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$ a jako výchozí směr vezměte $\mathbf{s}_1 = (1, 0)$.

Řešení: Hessova matice je $\mathbf{H} = \begin{pmatrix} 8 & 4 \\ 4 & 8 \end{pmatrix}$. Směr $\mathbf{s}_2 = (a, b)^\top$ je sdružený s \mathbf{s}_1 vzhledem k matici \mathbf{H} pokud $\mathbf{s}_1 \cdot \mathbf{H} \cdot \mathbf{s}_2 = 8a + 4b = 0$. To je splněno například pro $a = 1$, $b = -2$.

Metoda sdružených směrů - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x_2^2 + 4x_1x_2$ pomocí metody konjugovaných směrů. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$ a jako výchozí směr vezměte $\mathbf{s}_1 = (1, 0)$.

Řešení: Hessova matice je $\mathbf{H} = \begin{pmatrix} 8 & 4 \\ 4 & 8 \end{pmatrix}$. Směr $\mathbf{s}_2 = (a, b)^\top$ je sdružený s \mathbf{s}_1 vzhledem k matici \mathbf{H} pokud $\mathbf{s}_1 \cdot \mathbf{H} \cdot \mathbf{s}_2 = 8a + 4b = 0$. To je splněno například pro $a = 1$, $b = -2$.

Začneme minimalizaci z \mathbf{x}^0 ve směru \mathbf{s}_1 : $g_1(\alpha) = f(1 + \alpha, 1) = 12\alpha + 4\alpha^2$, což je minimální pro $\alpha = -\frac{3}{2}$, takže $\mathbf{x}^1 = (-\frac{1}{2}, 1)^\top$.

Metoda sdružených směrů - příklad

Hledejte minimum funkce $f(x_1, x_2) = -12x_2 + 4x_1^2 + 4x^2 + 4x_1x_2$ pomocí metody konjugovaných směrů. Proved'te první dvě iterace z bodu $\mathbf{x}^0 = (1, 1)^\top$ a jako výchozí směr vezměte $\mathbf{s}_1 = (1, 0)$.

Řešení: Hessova matice je $\mathbf{H} = \begin{pmatrix} 8 & 4 \\ 4 & 8 \end{pmatrix}$. Směr $\mathbf{s}_2 = (a, b)^\top$ je sdružený s \mathbf{s}_1 vzhledem k matici \mathbf{H} pokud $\mathbf{s}_1 \cdot \mathbf{H} \cdot \mathbf{s}_2 = 8a + 4b = 0$. To je splněno například pro $a = 1$, $b = -2$.

Začneme minimalizaci z \mathbf{x}^0 ve směru \mathbf{s}_1 : $g_1(\alpha) = f(1 + \alpha, 1) = 12\alpha + 4\alpha^2$, což je minimální pro $\alpha = -\frac{3}{2}$, takže $\mathbf{x}^1 = (-\frac{1}{2}, 1)^\top$.

Pokračujeme minimalizací z \mathbf{x}^1 ve směru \mathbf{s}_2 :
 $g_2(\alpha) = f(-\frac{1}{2} + \alpha, 1 - 2\alpha) = 9 + 12\alpha + 12\alpha^2$, což je minimální pro $\alpha = -\frac{1}{2}$, takže $\mathbf{x}^2 = (-1, 2)^\top$. Nalezli jsme přesně optimum funkce.

Vícerozměrná optimalizace - kvazi-newtonovské metody a Gauss-Newtonova metoda

Kvazi-newtonovské metody jsou gradientní metody, které leží někde mezi metodou nejrychlejšího spádu a Newtonovou metodou a snaží se využít předností obou metod. Gradientní metody mají zaručenou konvergenci a Newtonova metoda v okolí optima konverguje rychle. Newtonova metoda ale vyžaduje výpočet Hessianovy matice, respektive její inverze. Aproximujeme-li tyto matice na základě dat z jednotlivých kroků iteračního algoritmu, dostaneme metodu **Broydena, Fletchera, Goldfarba a Shannoa (BFGS)** nebo metodu **Davidona, Fletchera a Powella (DFP)**.

Vícerozměrná optimalizace - kvazi-newtonovské metody a Gauss-Newtonova metoda

Kvazi-newtonovské metody jsou gradientní metody, které leží někde mezi metodou nejrychlejšího spádu a Newtonovou metodou a snaží se využít předností obou metod. Gradientní metody mají zaručenou konvergenci a Newtonova metoda v okolí optima konverguje rychle. Newtonova metoda ale vyžaduje výpočet Hessianovy matice, respektive její inverze. Aproximujeme-li tyto matice na základě dat z jednotlivých kroků iteračního algoritmu, dostaneme metodu **Broydena, Fletchera, Goldfarba a Shannoa (BFGS)** nebo metodu **Davidona, Fletchera a Powella (DFP)**.

Gaussova-Newtonova metoda řeší problém minimalizace kritéria ve tvaru nejmenších čtverců nelineární funkce.

Vícerozměrná optimalizace - kvazi-newtonovské metody a Gauss-Newtonova metoda

Kvazi-newtonovské metody jsou gradientní metody, které leží někde mezi metodou nejrychlejšího spádu a Newtonovou metodou a snaží se využít předností obou metod. Gradientní metody mají zaručenou konvergenci a Newtonova metoda v okolí optima konverguje rychle. Newtonova metoda ale vyžaduje výpočet Hessiany matice, respektive její inverze. Aproximujeme-li tyto matice na základě dat z jednotlivých kroků iteračního algoritmu, dostaneme metodu **Broydena, Fletchera, Goldfarba a Shannoa (BFGS)** nebo metodu **Davidona, Fletchera a Powella (DFP)**.

Gaussova-Newtonova metoda řeší problém minimalizace kritéria ve tvaru nejmenších čtverců nelineární funkce.

Na závěr ještě podotkněme, že metody obecně nejsou globálně konvergentní, doporučuje se tedy optimalizaci spustit vícekrát z různých počátečních bodů. Stále jsou vyvíjeny nové algoritmy, z modernějších metod můžeme zmínit **simulované žíhání, genetické algoritmy, tabu search**, atd.

- Fletcher, Roger: Practical methods of optimization, 1st ed., Chichester : John Wiley and Sons, 1987.
- Šubrt, Tomáš a kol.: Ekonomicko-matematické metody, Plzeň : Vydavatelství a nakladatelství Aleš Čeněk, 2011
- Pánková, Václava: Nelineární optimalizace pro ekonomy, 1. vyd., Praha : Professional Publishing, 2003
- Intriligator, Michael D.: Mathematical optimization and economic theory, Philadelphia : Siam, 2002
- Jablonský, Josef, Petr Fiala, Miroslav Maňas: Vícekriteriální rozhodování, 1. vyd., Praha : Vysoká škola ekonomická v Praze, 1994.
- Klapka, Jindřich, Jiří Dvořák, Pavel Popela Metody operačního výzkumu, Vyd. 2., Brno : VUTIUM, 2001
- Klein, Michael W.: Mathematical methods for economics, 2nd ed., Boston : Addison-Wesley, c2002
- JABLONSKÝ, Josef: Operační výzkum : kvantitativní modely pro ekonomické rozhodování, 1. vyd. Praha: Professional Publishing, 2002
- PLEVNÝ, Miroslav a Miroslav ŽIŽKA: Modelování a optimalizace v manažerském rozhodování, Vyd. 2. Plzeň: Západočeská univerzita, 2010
- GROS, Ivan: Kvantitativní metody v manažerském rozhodování, 1. vyd. Praha: Grada, 2003