

Západočeská univerzita v Plzni
Fakulta aplikovaných věd

Studijní program: N1101

Studijní obor: 1101T016

DIPLOMOVÁ PRÁCE
**Semidefinitní programování v kombinatorické
optimalizaci**

Autor: Ondřej Špaček

Vedoucí práce: Doc. Ing. Roman Čada, Ph.D.

Plzeň, 2020

Obsah

Úvod	2
I Optimalizace	3
1 Základní geometrické pojmy	4
1.1 Přímký a úsečky	4
1.2 Afinity prostory	4
1.3 Konvexní množiny	5
1.4 Kužely	5
1.5 Nadroviny a poloprostory	6
1.6 Polyedry a polytopy	8
2 Konvexní optimalizace	10
2.1 Obecná podmíněná úloha	10
2.2 Konvexní podmíněná úloha	11
2.3 Lagrangeova dualita	11
2.3.1 Dolní odhad na x^*	11
2.3.2 Duální úloha	12
2.3.3 Slabá dualita	12
2.3.4 Silná dualita a Slaterova podmínka	12
3 Lineární programování	14
3.1 Primární úloha	14
3.2 Dualita	15
3.3 Komplementární skluzovost	17
4 Semidefinitní programování	19
4.1 Vsuvka z lineární algebry	19
4.2 Primární úloha	22
4.3 Dualita	24

4.4	Vektorové programování	25
II	Kombinatorické úlohy	26
5	Shannonova kapacita	27
5.1	Formulace úlohy	27
5.2	$\Theta(C_5) = \sqrt{5}$	28
5.3	Semidefinitní programy pro $\vartheta(G)$	32
5.4	Určení hodnoty ϑ pro konkrétní grafy	34
6	Problém maximálního řezu	38
6.1	Formulace úlohy	38
6.2	Úloha MAX CUT	38
6.3	Úloha MAX k -CUT	42
6.3.1	Frieze-Jerrum a MAX- k -CUT	42
6.3.2	Goemans-Williamson a MAX-3-CUT	43
6.3.3	de Klerk-Pasechnik-Warners a MAX- k -CUT	43
6.3.4	Newman a MAX- k -CUT	43
6.3.5	Porovnání algoritmů	43
	Závěr	44

Úvod

Část I

Optimalizace

Kapitola 1

Základní geometrické pojmy

1.1 Přímký a úsečky

Mějme dva body $x_1, x_2 \in \mathbb{R}^n$ takové, že $x_1 \neq x_2$ a parametr $\theta \in \mathbb{R}$. Potom výraz

$$y = \theta x_1 + (1 - \theta)x_2 \quad (1.1)$$

popisuje **přímku** procházející body x_1 a x_2 . Pro $\theta = 0$ dostáváme bod x_2 a pro $\theta = 1$ bod x_1 . Omezíme-li θ na interval $\langle 0, 1 \rangle$, dostaneme **úsečku** s koncovými body x_1 a x_2 . Výraz 1.1 lze přepsat do tvaru

$$y = x_2 + \theta(x_1 - x_2),$$

který můžeme interpretovat jako součet počátečního bodu x_2 a nějakého násobku směrového vektoru $x_1 - x_2$.

1.2 Afinní prostory

Říkáme, že $C \subseteq \mathbb{R}^n$ je **afinní prostor**, jestliže přímka procházející libovolnými dvěma různými body z C leží v C . Tedy C obsahuje lineární kombinace libovolných dvou bodů z C , jestliže součet koeficientů lineární kombinace je roven jedné. To lze zobecnit i pro více než dva body. Lineární kombinace $\theta_1 x_1 + \dots + \theta_k x_k$ bodů x_1, \dots, x_k taková, že $\theta_1 + \dots + \theta_k = 1$, se nazývá **afinní kombinace** bodů x_1, \dots, x_k . Indukcí z definice afinního prostoru lze snadno ukázat, že pokud C je afinní množina, $x_1, \dots, x_k \in C$ a $\theta_1 + \dots + \theta_k = 1$, potom bod $\theta_1 x_1 + \dots + \theta_k x_k \in C$.

Nechť C je afinní prostor a $x_0 \in C$, potom množina

$$V = C - x_0 = \{x - x_0 \mid x \in C\}$$

je **vektorový prostor**, tj. množina, která je uzavřená na sčítání a násobení skalárem.

Afinní prostor C lze vyjádřit jako

$$C = V + x_0 = \{v + x_0 \mid v \in V\},$$

kde V je vektorový prostor a x_0 je počátek. Poznamenejme, že vektorový prostor V asociovaný s afinním prostorem C nezávisí na volbě počátku x_0 .

Dimenze afinního prostoru $C = V + x_0$ je definována jako dimenze vektorového prostoru $V = C - x_0$, kde x_0 je libovolný prvek z C . Množina všech afinních kombinací bodů množiny $C \subseteq \mathbb{R}^n$ se nazývá **afinní obal** množiny C . Afinní obal množiny C budeme značit

$$\mathbf{aff} C = \{\theta_1 x_1 + \cdots + \theta_k x_k \mid x_1, \dots, x_k \in C, \theta_1 + \cdots + \theta_k = 1\}.$$

Afinní obal je nejmenší afinní prostor, který obsahuje množinu C . Tedy, jestliže S je afinní prostor takový, že $C \subseteq S$, potom $\mathbf{aff} C \subseteq S$.

1.3 Konvexní množiny

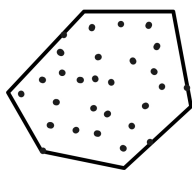
Říkáme, že množina C je **konvexní**, jestliže úsečka mezi libovolnými dvěma body z C leží také v C . Jinak řečeno, jestliže pro libovolné dva body $x_1, x_2 \in C$ a libovolné $\theta \in \langle 0, 1 \rangle$ platí, že $\theta x_1 + (1 - \theta)x_2 \in C$. Poznamenejme, že každý afinní prostor je zároveň konvexní množinou. Podobně jako afinní kombinaci definujeme **konvexní kombinaci** bodů x_1, \dots, x_k jako $\theta_1 x_1 + \cdots + \theta_k x_k$, kde $\theta_1 + \cdots + \theta_k = 1, \theta_i \geq 0$ pro $i = 1, \dots, k$. **Konvexní obal** množiny C je množina všech konvexních kombinací bodů z množiny C , značíme

$$\mathbf{conv} C = \{\theta_1 x_1 + \cdots + \theta_k x_k \mid x_i \in C, \theta_i \geq 0, i = 1, \dots, k, \theta_1 + \cdots + \theta_k = 1\}.$$

Analogicky, konvexní obal množiny C je nejmenší konvexní množina, která obsahuje množinu C . Pro představu viz obrázek 1.1.

1.4 Kužely

Množina C se nazývá **kužel**, jestliže pro každé $x \in C$ a $\theta \geq 0$ platí, že $\theta x \in C$. Je-li C navíc konvexní, pak se C nazývá **konvexní kužel**. Tedy C je konvexní kužel, jestliže pro libovolné $x_1, x_2 \in C$ a $\theta_1, \theta_2 \geq 0$ platí, že $\theta_1 x_1 + \theta_2 x_2 \in C$. Říkáme, že bod ve tvaru $\theta_1 x_1 + \cdots + \theta_k x_k$, kde $\theta_1, \dots, \theta_k \geq 0$ je **kuželovou kombinací** bodů x_1, \dots, x_k . Dále, pokud x_i leží v konvexním kuželu množiny C , potom libovolná kuželová kombinace bodu x_i leží rovněž

(a) Množina bodů C (b) $\text{conv } C$

Obrázek 1.1: Konvexní obal množiny

v konvexním kuželu množiny C . Platí, že množina C je konvexní kužel právě tehdy, když C obsahuje všechny kuželové kombinace svých bodů. **Kuželový obal** množiny C je množina, která obsahuje všechny kuželové kombinace množiny C , tj.

$$\text{cone } C = \{\theta_1 x_1 + \cdots + \theta_k x_k \mid x_i \in C, \theta_i \geq 0, i = 1, \dots, k\}.$$

Kuželový obal množiny C je zároveň nejmenší konvexní kužel, který obsahuje množinu C . Pro představu viz obrázek 1.2.

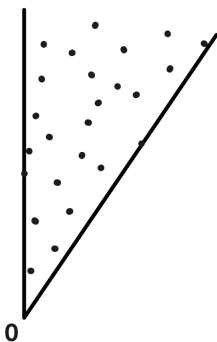
1.5 Nadroviny a poloprostory

Nadrovina je množina ve tvaru

$$\{x \mid a^T x = b\},$$

kde $a \in \mathbb{R}^n$, $a \neq 0$ a $b \in \mathbb{R}$. Analyticky na nadrovinu nahlížíme jako na množinu všech řešení netriviální lineární rovnice. Geometricky zase jako na množinu všech bodů takových, že mají konstantní skalární součin s normálovým vektorem a . Konstanta b značí posunutí nadroviny od počátku. Nadrovinu také můžeme vyjádřit jako

$$\{x \mid a^T(x - x_0) = 0\} = x_0 + \{v \mid a^T v = 0\},$$

(a) Množina bodů C (b) **cone** C

Obrázek 1.2: Kuželový obal množiny

kde x_0 je libovolný bod této nadroviny a $\{v \mid a^T v = 0\}$ je množina všech vektorů, které jsou kolmé k normálovému vektoru a . Nadrovina je tedy množina, která obsahuje bod x_0 a libovolný bod ve tvaru $x_0 + v$, kde v je vektor, který je kolmý k normálovému vektoru a . Pro ilustraci v \mathbb{R}^2 viz obrázek 1.3a.

Nadrovina dělí \mathbb{R}^n na dva poloprostory. Množina

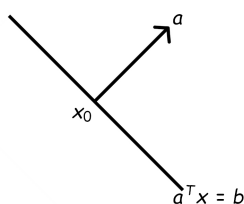
$$\{x \mid a^T x \leq b\}, \text{ resp. } \{x \mid a^T x < b\},$$

kde $a \neq 0$ se nazývá (uzavřený) **poloprostor**, resp. **otevřený poloprostor**. Je to tedy množina všech řešení netriviální lineární nerovnice. Podobně jako

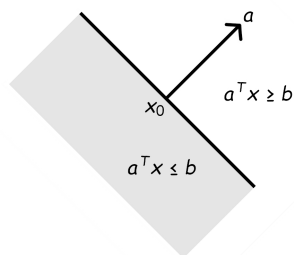
nadrovinu, můžeme poloprostor vyjádřit ve tvaru

$$\{x \mid a^T(x - x_0) \leq 0\}, \text{ resp. } \{x \mid a^T(x - x_0) < 0\},$$

kde $a \neq 0$ a x_0 je libovolný bod z nadroviny $\{x \mid a^T x = b\}$. Poloprostor tedy obsahuje bod x_0 a libovolný bod $x_0 + v$, kde v je vektor, který s vnějším normálovým vektorem svírá tupý nebo pravý úhel. Tato interpretace je v \mathbb{R}^2 ilustrována na obrázku 1.3b. Ještě poznamenejme, že poloprostory jsou konvexní množiny, ale samozřejmě nejsou afinní.



(a) Nadrovina



(b) Poloprostor

Obrázek 1.3: Nadrovina a poloprostor v \mathbb{R}^2 .

1.6 Polyedry a polytopy

Polytopy jsou zobecněním konvexních mnohoúhelníků v rovině do více dimenzí. Polytop v \mathbb{R}^3 je konvexní množina, která je ohraničena konečně mnoha konvexními mnohoúhelníky (příkladem polytopů v \mathbb{R}^3 jsou např. Platónská tělesa). Na takovou množinu je možné nahlížet dvěma způsoby. **H-polyedr** je průnik konečně mnoha uzavřených poloprostorů v \mathbb{R}^n , kde **H-polytop** je omezený H-polyedr. **V-polytop** je konvexní obal konečně mnoha bodů v \mathbb{R}^n . Následující věta říká, že H-polytop a V-polytop jsou matematicky ekvivalentní množiny.

Věta 1. *Každý V-polytop je H-polytop. Každý H-polytop je V-polytop.*

Poznamenejme, že V-polytop a H-polytop jsou sice ekvivalentní množiny, ale z algoritmického hlediska je velký rozdíl, zda pracujeme s bodovou množinou, nebo s uzavřenými poloprostory. Pro ilustraci: mějme lineární funkci, kterou chceme minimalizovat na daném polytopu. Pro V-polytop se jedná o triviální problém, protože stačí pro každý bod z množiny V určit hodnotu dané funkce a vybrat minimum. Na druhou stranu pro H-polytop se jedná o netriviální problém, kterým se zabývá lineární programování. Dále budeme mluvit jen o **polyedrech** a **polytopech**.

Důležitý fakt, že každý polyedr je konečně generovaný, říká Minkowského-Weylova věta.

Věta 2 (Minkowski-Weyl). $P \subseteq \mathbb{R}^n$. Potom $P = \mathbf{conv}(u_1, \dots, u_r) + \mathbf{cone}(v_1, \dots, v_s)$, kde u_i, v_i jsou extrémní vrcholy P právě tehdy, když $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

Kapitola 2

Konvexní optimalizace

2.1 Obecná podmíněná úloha

$$\begin{aligned} \min \quad & f(x) \\ g_i(x) & \leq 0, i = 1, \dots, m \\ h_i(x) & = 0, i = 1, \dots, p \end{aligned} \tag{2.1}$$

Hledáme $x \in \mathbb{R}^n$, které minimalizuje $f(x)$, vzhledem k omezením $g_i(x)$ a $h_i(x)$. Proměnné x říkáme **optimalizační proměnná**, funkci $f(x)$ říkáme **cenová** nebo **účelová funkce**. Výrazy $g_i(x) \leq 0$ jsou **omezení typu nerovnosti** a $h_i(x) = 0$ jsou **omezení typu rovnosti**. Říkáme, že problém 2.1 je **úloha nepodmíněné optimalizace**, jestliže $m = p = 0$. Jinak se jedná o **úlohu podmíněné optimalizace**.

Definiční obor \mathcal{D} úlohy 2.1 je

$$\mathcal{D} = \bigcap_{i=1}^m \text{dom } g_i \cap \bigcap_{i=1}^p \text{dom } h_i.$$

Říkáme, že bod $x \in \mathcal{D}$ je **přípustný**, jestliže splňuje všechna omezení $g_i(x) \leq 0$ a $h_i(x) = 0$. Úloha 2.1 je **přípustná**, jestliže existuje alespoň jeden bod $x \in \mathcal{D}$, který je přípustný. Množina všech přípustných bodů $x \in \mathcal{D}$ se nazývá **přípustná množina**.

Optimální hodnota x^* úlohy 2.1 je definována jako

$$x^* = \{f(x) \mid g_i(x) \leq 0, i = 1, \dots, m, h_i(x) = 0, i = 1, \dots, p\}.$$

2.2 Konvexní podmíněná úloha

$$\begin{aligned} \min \quad & f(x) \\ & g_i(x) \leq 0, i = 1, \dots, m \\ & a_i^T x = b_i, i = 1, \dots, p \end{aligned} \quad (2.2)$$

Oproti obecné úloze 2.1 jsou funkce $f(x)$, $g_i(x)$ konvexní a funkce $h_i(x) = a_i^T x - b_i$ jsou afinní. Přípustná množina takové úlohy je konvexní množinou.

2.3 Lagrangeova dualita

Mějme úlohu 2.1 s $\mathcal{D} \neq \emptyset$. Zobrazení $L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ takové, že

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \mu_i h_i(x) \quad (2.3)$$

se nazývá **Lagrangeova funkce**. Definiční obor $\text{dom } L = \mathcal{D} \times \mathbb{R}^m \times \mathbb{R}^p$. Vektory λ a μ nazýváme **duální proměnné** a prvkům těchto vektorů říkáme **Lagrangeovy multiplikátory**. Dále definujeme **duální funkci** $d : \mathbb{R}^m \times \mathbb{R}^p \rightarrow \mathbb{R}$ jako infimum Lagrangeovy funkce L přes všechna $x \in \mathcal{D}$. Tedy

$$d(\lambda, \mu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \mu) = \inf_{x \in \mathcal{D}} \left(f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^p \mu_i h_i(x) \right). \quad (2.4)$$

Poznamenejme, že duální funkce je konkávní bez ohledu na to, zda je úloha konvexní a je-li L zdola neomezená v proměnné x , potom duální funkce nabývá hodnoty $-\infty$.

2.3.1 Dolní odhad na x^*

Nechť \tilde{x} je přípustný bod. Pro $\lambda \geq 0$ je

$$\sum_{i=1}^m \lambda_i g_i(\tilde{x}) + \sum_{i=1}^p \mu_i h_i(\tilde{x}) \leq 0.$$

Potom pro Lagrangeovu funkci můžeme psát

$$L(\tilde{x}, \lambda, \mu) = f(\tilde{x}) + \sum_{i=1}^m \lambda_i g_i(\tilde{x}) + \sum_{i=1}^p \mu_i h_i(\tilde{x}) \leq f(\tilde{x}).$$

A tedy pro duální funkci platí

$$d(\lambda, \mu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \mu) \leq L(\tilde{x}, \lambda, \mu) \leq f(\tilde{x}).$$

2.3.2 Duální úloha

V části 2.3.1 jsme si ukázali, že duální funkce dává dolní odhad na optimální hodnotu x^* úlohy 2.1. Stále jsme si ale neřekli, jaký je nejlepší dolní odhad, který pomocí duální funkce jsme schopni dostat. To nás dostává k následující optimalizační úloze.

$$\begin{aligned} \max d(\lambda, \mu) \\ \lambda \geq 0 \end{aligned} \tag{2.5}$$

Úloze 2.5 se říká **Lagrangeova duální úloha** příslušná k úloze 2.1, kterou nazýváme **primární úlohou**.

2.3.3 Slabá dualita

Optimální řešení Lagrangeovy duální úlohy označíme d^* , které je už z definice nejlepší dolní odhad na optimální řešení primární úlohy p^* . Tato nerovnost platí i pokud primární úloha není konvexní. Této nerovnosti říkáme **slabá dualita**. Rozdíl optimálních řešení $p^* - d^*$ označujeme jako **optimální dualitní rozdíl** primární úlohy. Poznamenejme, že optimální dualitní rozdíl je vždy nezáporný.

2.3.4 Silná dualita a Slaterova podmínka

Pokud je optimální dualitní rozdíl $p^* - d^* = 0$, pak říkáme, že platí silná dualita. Silná dualita obecně neplatí, ale pro primární úlohu, která splňuje nějaké další podmínky to možné je. Těmto podmínkám se říká **podmínky kvalifikace omezení**. Jednou takovou je **Slaterova podmínka**:

$$\exists x \in \text{relint } \mathcal{D} : f_i(x) < 0, i = 1, \dots, m, Ax = b.$$

Bodu $x \in \mathcal{D}$, který splňuje Slaterovu podmínku, říkáme, že je **striktně přípustný**, protože omezení typu nerovnosti jsou ostré. Pokud jsou některé funkce f_i afinní, můžeme Slaterovu podmínku modifikovat. Nechť tedy $f_1, \dots, f_k, k \leq m$, jsou afinní funkce. Potom **modifikovaná Slaterova podmínka** má tvar:

$$\exists x \in \text{relint } \mathcal{D} : f_i(x) \leq 0, i = 1, \dots, k, f_i(x) < 0, i = k + 1, \dots, m, Ax = b.$$

Pro úlohu 2.2 platí následující věta.

Věta 3 (Slaterova). *Nechť primární úloha je konvexní a platí (modifikovaná) Slaterova podmínka, potom $p^* = d^*$.*

TODO

Použití duální úlohy: obecnou primární úlohu je těžké vyřešit, ale duální úloha je vždy konvexní, tak vyřeším tu a mám alespoň dolní odhad na primární úlohu

Kapitola 3

Lineární programování

3.1 Primární úloha

Úlohou lineárního programování rozumíme minimalizaci nebo maximalizaci lineární **účelové funkce** vzhledem k lineárním **omezením**, kde tato omezení jsou dána soustavou lineární rovnic a nerovnic. Úlohu lineárního programování lze formulovat v několika ekvivalentních tvarech, které se liší zadáním omezení. Úloha v **kanonickém tvaru** má svá omezení dána soustavou lineárních nerovnic $Ax \leq b$. Tedy:

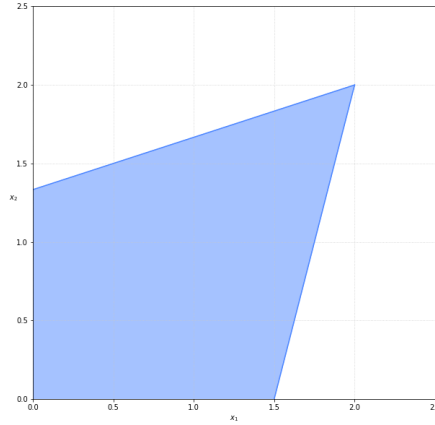
$$\max \{c^T x \mid Ax \leq b, x \geq 0\}, \quad (\text{LP-P})$$

kde $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, $x \in \mathbb{R}^n$ a $c \in \mathbb{R}^n$. **Přípustná množina řešení** je průnikem poloprostorů, které jsou definovány soustavou nerovnic $Ax \leq b$ a **nezáporného ortantu**, tj. množiny $\{x \in \mathbb{R}^n \mid x_i \geq 0, i = 1, \dots, n\}$. Obě tyto množiny jsou konvexní a tedy i jejich průnik je rovněž konvexní množina. Dále, protože přípustnou množinu máme popsanou soustavou konečně mnoha lineárních nerovnic, geometricky se na úlohu LP-P můžeme koukat jako na maximalizaci lineární funkce přes polyedr, který je definován touto soustavou.

Příklad. Mějme následující úlohu:

$$\begin{aligned} \max x_1 + x_2 \\ -x_1 + 3x_2 &\leq 4 \\ 4x_1 - x_2 &\leq 6 \\ x &\geq 0. \end{aligned} \quad (\text{P1})$$

Přípustná množina řešení je zobrazena na obrázku 3.1. Řešením úlohy je vektor $x^* = (2, 2)$ s cenou 4. Implementace v softwaru MOSEK: <https://github.com/c0n73x7/D1PL0MK4/blob/master/mosek/ex1.py>.



Obrázek 3.1: Přípustná množina řešení k úloze P1.

3.2 Dualita

Pomocí Lagrangeovy duality odvodíme duální úlohu k primární úloze LP-P. Máme tedy optimalizační úlohu

$$\min \{ -c^T x \mid Ax \leq b, x \geq 0 \}.$$

Pro ní vytvoříme Lagrangeovu funkci

$$\begin{aligned} L(x, \lambda) &= -c^T x + \lambda^T (Ax - b) - \lambda^T x \\ &= -b^T \lambda + (A^T \lambda - c - \lambda)^T x. \end{aligned}$$

Z Lagrangeovy funkce přejdeme k duální funkci

$$\begin{aligned} d(\lambda) &= \inf_x L(x, \lambda) \\ &= \inf_x -b^T \lambda + (A^T \lambda - c - \lambda)^T x \\ &= \begin{cases} -b^T \lambda & \text{pokud } A^T \lambda - c - \lambda = 0, \\ -\infty & \text{jinak.} \end{cases} \end{aligned}$$

Tu nakonec použijeme v duální úloze:

$$\max \{ -b^T \lambda \mid A^T \lambda - c - \lambda = 0 \}$$

$$\max \{ -b^T \lambda \mid A^T \lambda \geq c, \lambda \geq 0 \}$$

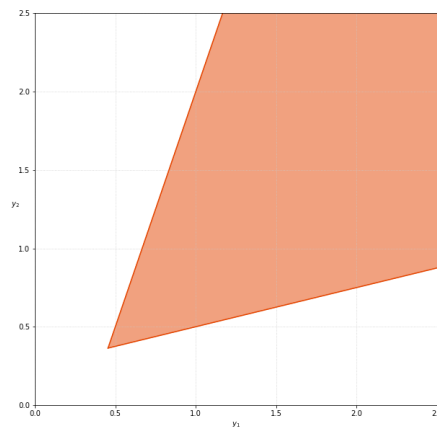
$$\min \{b^T \lambda \mid A^T \lambda \geq c, \lambda \geq 0\} \quad (\text{LP-D})$$

Dostáváme tedy duální úlohu LP-D k primární úloze LP-P.

Příklad. Duální úloha k úloze P1 je ve tvaru:

$$\begin{aligned} \min & 4y_1 + 6y_2 \\ & -y_1 + 4y_2 \geq 1 \\ & 3y_1 - y_2 \geq 1 \\ & y \geq 0. \end{aligned} \quad (\text{P2})$$

Přípustná množina řešení je zobrazena na obrázku 3.2. Řešením úlohy je vektor $y^* \approx (0.4546, 0.3636)$ s cenou 4. Implementace v softwaru MOSEK: <https://github.com/c0n73x7/D1PL0MK4/blob/master/mosek/ex2.py>.



Obrázek 3.2: Přípustná množina řešení k úloze P2.

Všimněme si, že v příkladech P1 a P2 mají řešení x^* i y^* stejnou cenu. To není náhoda a tento fakt je obsahem silné věty o dualitě lineárního programování, kterou dokázala skupina kolem Alberta W. Tuckera v roce 1948. Začneme slabou větou o dualitě lineárního programování.

Věta 4 (Slabá o dualitě). *Nechť \tilde{x} je přípustné řešení LP-P a \tilde{y} je přípustné řešení LP-D. Potom $c^T \tilde{x} \leq b^T \tilde{y}$.*

Tedy každé přípustné řešení \tilde{y} duální úlohy LP-D nám dává horní odhad na maximum účelové funkce primární úlohy LP-P. Graficky můžeme slabou

větu o dualitě interpretovat jako na obrázku 3.3. Zatím tedy nevíme, zda vždy existují přípustná (optimální) řešení x^* pro úlohu LP-P a y^* pro úlohu LP-D, pro která platí $c^T x^* = b^T y^*$. Kladnou odpověď dostaneme z již zmíněné silné věty od dualitě.



Obrázek 3.3: Slabá věta o dualitě.

Věta 5 (Silná o dualitě). *Jestliže úlohy LP-P a LP-D mají přípustná řešení. Potom*

$$\max \{c^T x \mid Ax \leq b, x \geq 0\} = \min \{b^T y \mid A^T y \geq c, y \geq 0\}.$$

Se znalostí silné věty o dualitě můžeme obrázek 3.3 upravit na obrázek 3.4.



Obrázek 3.4: Ceny přípustných řešení primární a příslušné duální úlohy.

3.3 Komplementární skluzovost

Pro odvození tzv. podmínky komplementární skluzovosti nejprve převedeme úlohy LP-P a LP-D do jiných tvarů. V primární úloze povolíme $x \in \mathbb{R}^n$. Tedy primární úloha je ve tvaru:

$$\max \{c^T x \mid Ax \leq b\}. \quad (\text{LP-P2})$$

A příslušná duální úloha je ve tvaru:

$$\min \{b^T y \mid A^T y = c, y \geq 0\}. \quad (\text{LP-D2})$$

Nechť \tilde{x} je přípustné řešení a x^* je optimální řešení úlohy LP-P2, \tilde{y} je přípustné řešení a y^* je optimální řešení úlohy LP-D2. **Dualitní rozdíl** \tilde{x} a \tilde{y} je číslo $b^T \tilde{y} - c^T \tilde{x} \geq 0$. Ze silné věty o dualitě samozřejmě plyne, že pro optimální řešení x^* a y^* je dualitní rozdíl roven 0. Vyjdeme z dualitního rozdílu optimálních řešení:

$$b^T y^* - c^T x^* = y^{*T} b - y^{*T} A x^* = y^{*T} (b - A x^*) = 0.$$

Poslední rovnost přepíšeme maticově:

$$[y_1^*, \dots, y_m^*] \left(\begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} - \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1^* \\ \vdots \\ x_n^* \end{bmatrix} \right) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Dostáváme tedy soustavu rovnic $y_i^* (b_i - a_{i\cdot} x^*) = 0$, kde $i = 1, \dots, m$. Tedy buď $y_i^* = 0$ nebo $b_i - a_{i\cdot} x^* = 0$. **Podmínka komplementární skluzovosti** je splněna, jestliže pro přípustná řešení \tilde{x}, \tilde{y} platí buď $\tilde{y}_i = 0$ nebo $b_i - a_{i\cdot} \tilde{x} = 0$, $i = 1, \dots, m$. Pokud nastane $b_i - a_{i\cdot} \tilde{x} = 0$, potom říkáme, že **vazba** $a_{i\cdot} \tilde{x} \leq b_i$ **je aktivní**.

Věta 6. *Nechť \tilde{x} je přípustné řešení LP-P2 a \tilde{y} je přípustné řešení LP-D2. Potom \tilde{x}, \tilde{y} jsou optimální právě tehdy, když platí podmínka komplementární skluzovosti.*

Kapitola 4

Semidefinitní programování

Na semidefinitní programování se můžeme koukat jako na zobecnění lineárního programování, kde proměnné jsou symetrické matice. Jedná se tedy o optimalizaci lineární funkce vzhledem k tzv. lineárním maticovým nerovnostem.

4.1 Vsuvka z lineární algebry

Pozitivně definitní matice

Pracujeme s reálnými symetrickými maticemi $S = S^T$. Ty mají všechna vlastní čísla reálná a některé z nich mají zajímavou vlastnost, že všechna jejich vlastní čísla jsou kladná. Takovým maticím říkáme, že jsou pozitivně definitní. Alternativní definicí je, že matice S je pozitivně definitní, jestliže $x^T S x > 0$ pro všechny nenulové vektory x .

Příklad.

$$x^T S x = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 4 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2x_1^2 + 8x_1x_2 + 9x_2^2$$

Je pro všechny x nenulové $x^T S x > 0$? Ano, protože můžeme výraz přepsat na součet čtverců:

$$x^T S x = 2x_1^2 + 8x_1x_2 + 9x_2^2 = 2(x_1 + 2x_2)^2 + x_2^2.$$

TODO: obrázek kyblíčku

Ukážeme si několik kritérií, jak otestovat pozitivní definitnost dané matice.

Věta 7. $S = S^T$ je pozitivně definitní, jestliže lze napsat jako $S = A^T A$ pro nějakou matici A , která má lineárně nezávislé sloupcečky.

Důkaz.

$$x^T S x = x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0 \quad (4.1)$$

$\|Ax\|^2 > 0$, jestliže sloupčky matice A jsou lineárně nezávislé \square

Příklad.

$$S = \begin{bmatrix} 2 & 3 & 4 \\ 3 & 5 & 7 \\ 4 & 7 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} = A A^T$$

A má lineárně závislé sloupčky, tj. S není pozitivně definitní

Dalším testem je tzv. Sylvesterovo kritérium.

Věta 8. $S = S^T$ je pozitivně definitní, jestliže všechny hlavní minory S jsou kladné.

Příklad.

$$S = \begin{bmatrix} 3 & 4 \\ 4 & 6 \end{bmatrix}, D_1 = 3, D_2 = 3 \cdot 6 - 4 \cdot 4 = 2$$

hlavní minory $D_1, D_2 > 0$; matice S je pozitivně definitní

A poslední, které si uvedeme souvisí s Gaussovou eliminací.

Věta 9. $S = S^T$ je pozitivně definitní, jestliže jsou všechny pivoty při Gaussově eliminaci kladné.

Příklad.

$$S = \begin{bmatrix} 3 & 4 \\ 4 & 6 \end{bmatrix} \sim \begin{bmatrix} 3 & 4 \\ 0 & \frac{2}{3} \end{bmatrix}, p_1 = 3, p_2 = \frac{2}{3}$$

pivoty $p_1, p_2 > 0$; matice S je pozitivně definitní

Pozitivně semidefinitní matice

Pro pozitivní semidefinitnost modifikujeme předcházejí definice a tvrzení pro pozitivně definitní matice následovně:

1. $S = S^T$ je pozitivně semidefinitní, jestliže všechna její čísla jsou nezáporná.
2. $S = S^T$ je pozitivně semidefinitní, jestliže $x^T S x \geq 0$ pro všechny nenulové vektory x .
3. $S = S^T$ je pozitivně semidefinitní, jestliže lze napsat jako $S = A^T A$ pro nějakou matici A .
4. $S = S^T$ je pozitivně semidefinitní, jestliže všechny hlavní minory S jsou nezáporné.
5. $S = S^T$ je pozitivně semidefinitní, jestliže jsou všechny pivoty při eliminaci nezáporné.

Pozitivně semidefinitní kužel

Množinu všech symetrických matic značíme S^n , množinu všech pozitivně semidefinitních matic značíme S_+^n a množinu všech pozitivně definitních matic značíme S_{++}^n .

Lemma 1. *Množina S_+^n je uzavřená*

Důkaz. TODO □

Lemma 2. *Množina S_+^n tvoří konvexní kužel.*

Důkaz. $\Theta_1, \Theta_2 \geq 0, A, B \in S_+^n$

$$x^T (\Theta_1 A + \Theta_2 B) x = x^T \Theta_1 A x + x^T \Theta_2 B x \geq 0.$$

□

Lemma 3. *Kužel S_+^n je pointed.*

Důkaz. TODO □

Lemma 4. *Kužel S_+^n je samoduální.*

Důkaz. TODO □

Shrneme předchozí lemmata do následující věty.

Věta 10. *Množina S_+^n tvoří konvexní, pointed a uzavřený kužel, který je samoduální.*

Říkáme, že S_+^n je **pozitivně semidefinitní kužel**.

Spektraedry

Definujeme tzv. **Löwnerovo částečné uspořádání**:

$$A \succeq B \iff A - B \in S_+^n,$$

tj. matice $A - B$ je pozitivně semidefinitní. **Lineární maticová nerovnost** (LMI) je ve tvaru

$$A_0 + \sum_{i=1}^n A_i x_i \succeq 0,$$

kde $A_i \in S^n$.

Množina $S \subset \mathbb{R}^n$, která je definována pomocí konečně mnoha LMI, se nazývá **spektraedr**. Tedy:

$$S = \left\{ (x_1, \dots, x_m) \in \mathbb{R}^m \mid A_0 + \sum_{i=1}^m A_i x_i \succeq 0 \right\}$$

pro nějaké symetrické matice $A_0, \dots, A_m \in S^n$.

Můžeme si všimnout analogie s definicí polyedru, který je přípustnou množinou pro lineární program. Podobně spektraedr je přípustnou množinou pro semidefinitní program.

Geometricky je spektraedr definován jako průnik pozitivně semidefinitního kuželu S_+^n a afinního podprostoru $\text{span}\{A_1, \dots, A_m\}$ posunutého do A_0 .

Spektraedry jsou uzavřené množiny, neboť LMI je ekvivalentní nekonečně mnoha skalárním nerovnostem ve tvaru $v^T(A_0 + \sum_{i=1}^m A_i x_i)v \geq 0$, jednu pro každou hodnotu $v \in \mathbb{R}^n$.

Vždy můžeme několik LMI „scucnout“ do jedné. Stačí zvolit matice A_i blokově diagonální. Odtud snadno víme, že polyedr je speciálním případem spektraedru. Polyedr bude mít všechny matice A_i diagonální.

Příklad.

$$\left\{ (x, y) \in \mathbb{R}^2 \mid A(x, y) = \begin{bmatrix} x+1 & 0 & y \\ 0 & 2 & -x-1 \\ y & -x-1 & 2 \end{bmatrix} \succeq 0 \right\}$$

4.2 Primární úloha

Semidefinitní program je lineární optimalizační problém přes spektraedr. Primární úlohu ve standardním tvaru můžeme napsat jako:

$$\inf \{ \langle C, X \rangle \mid \langle A_i, X \rangle = b_i, i = 1, \dots, m; X \succeq 0 \}, \quad (\text{SDP-P})$$

kde $C, A_i \in S^n$, $\langle X, Y \rangle = \text{Tr}(X^T Y) = \sum_{ij} X_{ij} Y_{ij}$ a $X \in S^n$ je proměnná, nad kterou provádíme minimalizaci.

Příklad.

$$\inf \left\{ \left\langle \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \right\rangle \mid \left\langle \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \right\rangle = 1, \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \succeq 0 \right\} \quad (\text{P3})$$

Po úpravě:

$$\inf \left\{ 2x_{11} + 2x_{12} \mid x_{11} + x_{22} = 1, \begin{bmatrix} x_{11} & x_{12} \\ x_{12} & x_{22} \end{bmatrix} \succeq 0 \right\}.$$

Jak vypadá přípustná množina? Použijeme Sylvesterovo kritérium, tj.

$$x_{11} \geq 0, x_{11}x_{22} - x_{12}^2 \geq 0.$$

Z LMI vyjádříme x_{22} , tj.

$$x_{22} = 1 - x_{11}$$

Dosadíme do přechozího a dostaneme

$$x_{11} \geq 0, x_{11}(1 - x_{11}) - x_{12}^2 \geq 0$$

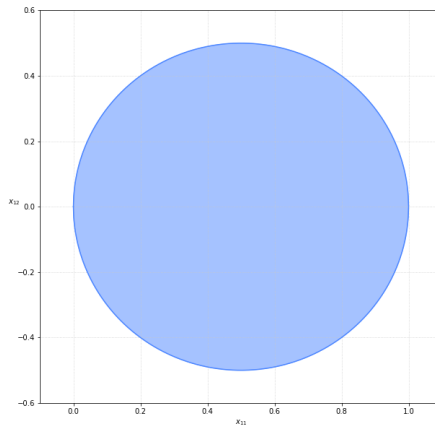
Po úpravě

$$x_{11} \geq 0, \left(x_{11} - \frac{1}{2}\right)^2 + x_{12}^2 \leq \frac{1}{4}$$

Vidíme tedy, že přípustná množina (zobrazena na obrázku 4.1) je kruh s poloměrem $\frac{1}{2}$ a se středem v bodě $(x_{11}, x_{12}) = (\frac{1}{2}, 0)$. Řešením úlohy je matice

$$X^* \approx \begin{bmatrix} 0.1464 & -0.3536 \\ -0.3536 & 0.8536 \end{bmatrix}$$

s cenou ≈ -0.4142 . Implementace v softwaru MOSEK: <https://github.com/c0n73x7/D1PL0MK4/blob/master/mosek/ex3.py>.



Obrázek 4.1: Přípustná množina řešení k úloze P3.

4.3 Dualita

Duální úloha

Podobně jako u lineárního programování použijeme Lagrangeovu dualitu k odvození duální úlohy k úloze SDP-P. Lagrangeova funkce je ve tvaru:

$$L(X, \lambda, Z) = \langle C, X \rangle - \sum_{i=1}^m \lambda_i (\langle A_i, X \rangle - b_i) - \langle Z, X \rangle.$$

K ní duální funkce:

$$d(\lambda, Z) = \inf_{X \in S^n} L(X, \lambda, Z) = \begin{cases} \lambda^T b & \dots \ C - \sum_{i=1}^m \lambda_i A_i - Z = 0, \\ -\infty & \dots \text{jinak.} \end{cases}$$

Duální funkci použijeme v duální úloze:

$$\sup \left\{ \lambda^T b \mid C - \sum_{i=1}^m \lambda_i A_i \succeq 0 \right\}, \quad (\text{SDP-D})$$

kde $\lambda = (\lambda_1, \dots, \lambda_m)$ je duální proměnná.

Dostali jsme duální úlohu SDP-D k úloze SDP-P.

Slabá dualita semidefinitního programování

Vztah mezi primární a duální úlohou je stejně jako u lineárního programování takový, že řešení jedné úlohy lze použít jako odhad na úlohu druhou. Necht' X je libovolné přípustné řešení primární úlohy a y je libovolné přípustné řešení duální úlohy. Potom

$$\langle C, X \rangle - b^T y = \langle C, X \rangle - \sum_{i=1}^m y_i \langle A_i, X \rangle = \left\langle C - \sum_{i=1}^m A_i y_i, X \right\rangle \geq 0. \quad (4.2)$$

Za pozornost stojí poslední nerovnost, která plyne z toho, že skalární součin dvou pozitivně semidefinitních matic je nezáporný. Odvození je následující: mějme dvě matice $S, T \succeq 0$. Matici S můžeme napsat jako součet „rank one“ matic. Označme $r_S = \text{rank}(S)$, tj.

$$S = \sum_{i=1}^{r_S} \lambda_i s_i s_i^T,$$

kde s_i je i -tý sloupeček matice S . Dále se podíváme na součin $T \cdot S_i$. Tedy pro $i = 1, \dots, r_S$ platí

$$T \cdot S_i = \lambda_i s_i^T T s_i \geq 0,$$

kde nerovnost plyne z toho, že matice T je pozitivně semidefinitní.

O nerovnosti 4.2 se mluví jako o slabé dualitě semidefinitního programování.

Silná dualita semidefinitního programování

Věta 11 (podmínka optimality). *Nechť X je přípustné řešení úlohy SDP-P a y je přípustné řešení úlohy SDP-D taková, že splňují podmínku (komplementární skluzovosti)*

$$\left(C - \sum_{i=1}^m A_i y_i \right) X = 0.$$

Potom X je optimální řešení úlohy SDP-P a y je optimální řešení úlohy SDP-D.

Obracená implikace sama o sobě neplatí, což znamená, že obecně dualitní rozdíl u semidefinitního programování není nulový. Musíme přidat podmínku kvalifikace omezení, kterou je například již zmíněná Slaterova podmínka. Ta je pro úlohu SDP-P ve tvaru $X \succ 0$ a pro úlohu SDP-D ve tvaru $C - \sum_i A_i y_i \succ 0$.

Věta 12 (silná dualita semidefinitního programování). *Nechť úloha SDP-P a úloha SDP-D jsou striktně přípustné. Potom dualitní rozdíl jejich optimálních řešení je nulový.*

4.4 Vektorové programování

Mějme n vektorových proměnných $\mathbf{v}_1, \dots, \mathbf{v}_n$ v \mathbb{R}^n . **Vektorový program** je problém optimalizace lineární funkce skalárních součinů $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$, vzhledem k lineárním omezením na tyto skalární součiny.

Nyní ukážeme, že vektorové programy jsou ekvivalentní semidefinitním programům. Nechť tedy \mathcal{V} je vektorový program s vektorovými proměnnými $\mathbf{v}_1, \dots, \mathbf{v}_n$ v \mathbb{R}^n a \mathcal{S} je příslušný semidefinitní program s n^2 proměnnými y_{ij} , kde hodnota y_{ij} odpovídá skalárnímu součinu $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$. Matice \mathbf{Y} je navíc pozitivně semidefinitní. Potom platí následující věta.

Věta 13. *Vektorový program \mathcal{V} je ekvivalentní semidefinitnímu programu \mathcal{S} .*

Důkaz. TODO

□

Část II

Kombinatorické úlohy

Kapitola 5

Shannonova kapacita

5.1 Formulace úlohy

Představme si komunikační kanál, kterým posíláme zprávy. Tyto zprávy jsou složeny ze symbolů nějaké konečné abecedy. Vlivem šumu mohou být některé symboly druhou stranou špatně interpretovány a naším cílem je vybrat co největší množinu slov délky k tak, aby žádná dvě odeslaná slova nebyla vlivem tohoto šumu zaměnitelná.

Problém si formalizujeme v řeči teorie grafů. Mějme neorientovaný graf $G = (V, E)$, kde množina vrcholů představuje symboly z konečné abecedy a dva vrcholy x, y jsou spojeny hranou, pokud vrchol x může být vlivem šumu zaměněn za y .

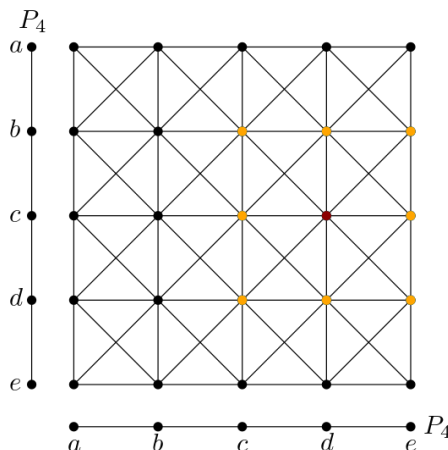
Maximální počet nezaměnitelných zpráv délky 1 je roven $\alpha(G)$, kde $\alpha(G)$ značí velikost největší nezávislé množiny v grafu G . Pro popis delších zpráv definujeme **silný součin** $G \cdot H$ grafů G a H následovně

$$\begin{aligned} V(G \cdot H) &= V(G) \times V(H), \\ E(G \cdot H) &= \{(i, u)(j, v) \mid ij \in E(G) \wedge uv \in E(H)\} \cup \\ &\quad \{(i, u)(j, v) \mid ij \in E(G) \wedge u = v\} \cup \\ &\quad \{(i, u)(j, v) \mid i = j \wedge uv \in E(H)\}. \end{aligned}$$

Příklad. Pro graf $P_4 = a - b - c - d - e$ je silný součin $P_4 \cdot P_4$ zobrazen na obrázku 5.1, ze kterého je hezky vidět, že např. zpráva cd (na obrázku červeně) může být zaměněna s bc , bd , be , cc , ce , dc , dd a de (na obrázku oranžově). Podobně pro další zprávy.

Pro jednoduchost budeme silný součin k kopií grafu G značit G^k . Tedy $\alpha(G^k)$ je maximální počet nezaměnitelných zpráv délky k . **Shannonova kapacita** grafu G je definována jako:

$$\Theta(G) = \sup \{ \alpha(G^k)^{1/k} \mid k = 1, 2, \dots \}.$$

Obrázek 5.1: $P_4 \cdot P_4$

Neví se, zda pro libovolný graf G existuje vůbec nějaký algoritmus, kterým bychom určili hodnotu $\Theta(G)$. Přesto je alespoň něco známo. Pro perfektní grafy Claude E. Shannon ukázal, že $\Theta(G) = \alpha(G)$. To také znamená, že pro perfektní grafy lze $\Theta(G)$ určit v polynomiálním čase. Dalším kdo se problémem zabýval byl László Lovász, který velmi hezkým způsobem ukázal, že kružnice délky 5 má kapacitu $\sqrt{5}$. Na Lovászův postup se dále podíváme, protože vede k obecnému hornímu odhadu na $\Theta(G)$.

5.2 $\Theta(C_5) = \sqrt{5}$

Nejprve potřebujeme zavést několik pojmů. **Tenzorový součin** vektorů $\mathbf{u} = (u_1, \dots, u_n)$ a $\mathbf{v} = (v_1, \dots, v_m)$ je

$$\mathbf{u} \circ \mathbf{v} = (u_1 v_1, \dots, u_1 v_m, u_2 v_1, \dots, u_n v_m).$$

Užitečné bude následující pozorování, které dává do souvislosti skalární a tenzorový součin.

Pozorování. *Nechť \mathbf{x}, \mathbf{u} jsou vektory délky n a \mathbf{y}, \mathbf{v} jsou vektory délky m . Potom platí*

$$(\mathbf{x} \circ \mathbf{y})^T (\mathbf{u} \circ \mathbf{v}) = (\mathbf{x}^T \mathbf{u}) (\mathbf{y}^T \mathbf{v}). \quad (5.1)$$

Důkaz. Levá strana:

$$\begin{aligned} & (x_1 y_1, x_1 y_2, \dots, x_1 y_m, \dots, x_n y_m)^T (u_1 v_1, u_1 v_2, \dots, u_1 v_m, \dots, u_n v_m) = \\ & x_1 y_1 u_1 v_1 + x_1 y_2 u_1 v_2 + \dots + x_1 y_m u_1 v_m + \dots + x_n y_m u_n v_m \end{aligned}$$

Pravá strana:

$$(x_1 u_1 + \cdots + x_n u_n) \cdot (y_1 v_1 + \cdots + y_m v_m) = \\ x_1 y_1 u_1 v_1 + x_1 y_2 u_1 v_2 + \cdots + x_1 y_m u_1 v_m + \cdots + x_m y_m u_n v_m$$

□

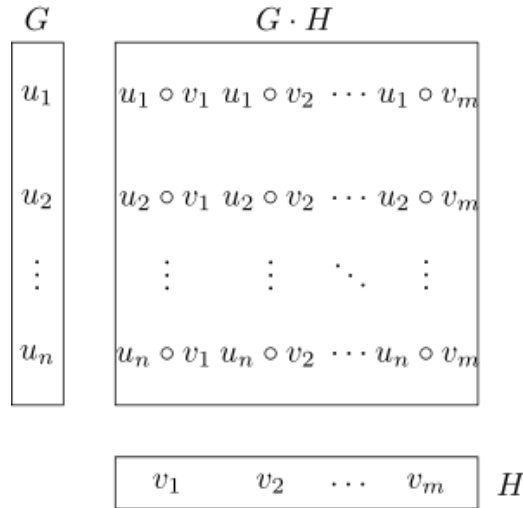
Mějme graf $G = (V, E)$, kde $V = \{1, \dots, n\}$. Systém $(\mathbf{v}_1, \dots, \mathbf{v}_n)$ jednotkových vektorů v Euklidovském prostoru takový, že

$$\forall ij \notin E \implies \mathbf{v}_i \perp \mathbf{v}_j$$

nazýváme **ortonormální reprezentace** grafu G . Poznamenejme, že každý graf má nějakou ortonormální reprezentaci, např. $1 \mapsto \mathbf{e}_1, \dots, n \mapsto \mathbf{e}_n$.

Lemma 5. *Nechť $(\mathbf{u}_1, \dots, \mathbf{u}_n)$ je ortonormální reprezentace grafu G a $(\mathbf{v}_1, \dots, \mathbf{v}_m)$ je ortonormální reprezentace grafu H . Potom $\mathbf{u}_i \circ \mathbf{v}_j$ je ortonormální reprezentace grafu $G \cdot H$.*

Důkaz. Použijeme vztah 5.1. $(u_i \circ v_j)^T (u_k \circ v_l) = (u_i^T u_k) (v_j^T v_l) = 0 \iff ik \notin E(G) \vee jl \notin E(H)$. □



Obrázek 5.2: Lemma 5

Hodnotu ortonormální reprezentace (u_1, \dots, u_n) definujeme jako

$$\min_c \max_{i=1, \dots, n} \frac{1}{(c^T u_i)^2}.$$

Vektoru c , pro který nastává minimum říkáme **handle** dané ortonormální reprezentace.

Dále definujeme funkci $\vartheta(G)$ jako minimální hodnotu přes všechny ortonormální reprezentace grafu G . Ortonormální reprezentaci, pro kterou nastává minimum nazýváme **optimální**. Funkci $\vartheta(G)$ se říká **Lovászova theta funkce** a ona je právě již zmíněným horním odhadem na $\Theta(G)$. Podívejme se na některé její vlastnosti.

Lemma 6. $\vartheta(G \cdot H) \leq \vartheta(G)\vartheta(H)$

Důkaz. Nechtě (u_1, \dots, u_n) je optimální ortonormální reprezentace grafu G s handle c a (v_1, \dots, v_m) je optimální ortonormální reprezentace grafu H s handle d . Pak $c \circ d$ je jednotkový vektor a platí

$$\vartheta(G \cdot H) \leq \max_{i,j} \frac{1}{\left((c \circ d)^T (u_i \circ v_j)\right)^2} = \max_i \frac{1}{(c^T u_i)^2} \cdot \max_j \frac{1}{(d^T v_j)^2} = \vartheta(G)\vartheta(H).$$

□

Lemma 7. $\alpha(G) \leq \vartheta(G)$

Důkaz. Mějme maximální nezávislou množinu $I \subseteq V(G)$ v grafu G a optimální ortonormální reprezentaci $\mathcal{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$ grafu G s handle \mathbf{c} . Platí

$$\forall i, j \in I : i \neq j \implies \mathbf{u}_i \perp \mathbf{u}_j.$$

Máme tedy systém ortonormálních vektorů $\{\mathbf{u}_i \in \mathcal{U} \mid i \in I\}$ v \mathbb{R}^n . Ten rozšíříme na ortonormální bázi \mathcal{B} . Potom i -tá souřadnice vektoru \mathbf{c} v bázi \mathcal{B} je $\mathbf{c}^T \mathbf{u}_i$. Tedy

$$1 = \|\mathbf{c}\|^2 = \sum_{i=1}^n (\mathbf{c}^T \mathbf{u}_i)^2.$$

Dále vynecháme přidáné vektory do ortonormální báze \mathcal{B}

$$\sum_{i=1}^n (\mathbf{c}^T \mathbf{u}_i)^2 \geq \sum_{i \in I} (\mathbf{c}^T \mathbf{u}_i)^2.$$

Poslední výraz přepíšeme

$$\sum_{i \in I} (\mathbf{c}^T \mathbf{u}_i)^2 \geq |I| \cdot \min_{i \in I} (\mathbf{c}^T \mathbf{u}_i)^2 = \alpha(G) \cdot \min_{i \in I} (\mathbf{c}^T \mathbf{u}_i)^2.$$

Předchozí výrazy dáme dohromady

$$1 \geq \alpha(G) \cdot \min_{i \in I} (\mathbf{c}^T \mathbf{u}_i)^2,$$

a dostáváme

$$\alpha(G) \leq \frac{1}{\min_{i \in I} (\mathbf{c}^T \mathbf{u}_i)^2} = \max_{i \in I} \frac{1}{(\mathbf{c}^T \mathbf{u}_i)^2} \leq \max_{i \in V(G)} \frac{1}{(\mathbf{c}^T \mathbf{u}_i)^2} = \vartheta(G).$$

□

Lemma 8. $\Theta(G) \leq \vartheta(G)$

Důkaz. Pro každé k platí, že

$$\alpha(G^k) \leq \vartheta(G^k) \leq \vartheta(G)^k.$$

Odtud

$$\sqrt[k]{\alpha(G^k)} \leq \vartheta(G),$$

a limitním přechodem dostáváme požadovanou nerovnost

$$\Theta(G) = \lim_{k \rightarrow \infty} \sqrt[k]{\alpha(G^k)} \leq \vartheta(G).$$

□

Věta 14. $\Theta(C_5) = \sqrt{5}$

Důkaz. Ukážeme konstrukci ortonormální reprezentace grafu C_5 , ze které dostaneme horní odhad na $\Theta(C_5)$. Nechť $V(C_5) = \{v_1, \dots, v_5\}$ a $E(C_5) = \{v_1v_2, v_2v_3, v_3v_4, v_4v_5, v_1v_5\}$. Mějme vektory $\bar{\mathbf{u}}_i$ ve tvaru

$$\bar{\mathbf{u}}_i = \left(\cos \frac{2\pi i}{5}, \sin \frac{2\pi i}{5}, z \right), i = 1, \dots, 5.$$

Každý vektor $\bar{\mathbf{u}}_i$ je svázán s vrcholem v_i . Chceme, aby dva vektory, které jsou příslušné sousedním vrcholům, byly ortogonální. Tedy například $\langle \bar{\mathbf{u}}_2, \bar{\mathbf{u}}_5 \rangle = 0$. Dosadíme

$$\langle \bar{\mathbf{u}}_2, \bar{\mathbf{u}}_5 \rangle = \left\langle \left(\cos \frac{4\pi}{5}, \sin \frac{4\pi}{5}, z \right), (1, 0, z) \right\rangle = \cos \frac{4\pi}{5} + z^2 = 0.$$

Dostáváme tedy

$$z = \sqrt{-\cos \frac{4\pi}{5}}.$$

Definujeme ortonormální reprezentaci \mathcal{U} grafu C_5 (projekce do první a druhé souřadnice, viz Obrázek 5.3) tak, že

$$\mathbf{u}_i = \frac{\bar{\mathbf{u}}_i}{\|\bar{\mathbf{u}}\|}, i = 1, \dots, 5,$$

s handle $\mathbf{c} = (0, 0, 1)$. Dostáváme

$$\vartheta(C_5) \leq \vartheta(\mathcal{U}) = \max_{i=1,\dots,5} \frac{1}{(\mathbf{c}^T \mathbf{u}_i)^2} = \frac{1}{(\mathbf{c}^T \mathbf{u}_5)^2} = \frac{1 - \cos \frac{4\pi}{5}}{-\cos \frac{4\pi}{5}}.$$

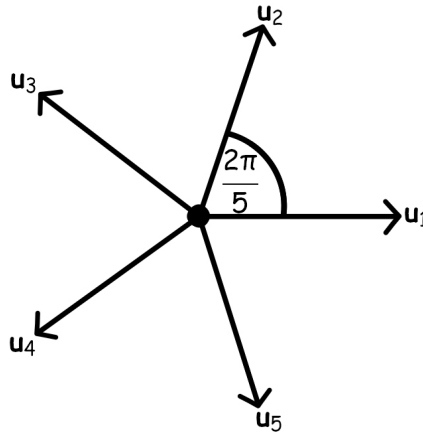
Do posledního výrazu dosadíme známou hodnotu pro $\cos 36^\circ$.

$$\frac{1 - \cos \frac{4\pi}{5}}{-\cos \frac{4\pi}{5}} = \frac{1 + \frac{1+\sqrt{5}}{4}}{\frac{1+\sqrt{5}}{4}} = \frac{5 + \sqrt{5}}{1 + \sqrt{5}} = \sqrt{5}.$$

Dostáváme

$$\vartheta(C_5) \leq \sqrt{5}.$$

Této ortonormální reprezentaci se říká **Lovászův deštník**, viz Obrázek 5.4. Druhou nerovnost $\vartheta(C_5) \geq \sqrt{5}$ dostaneme snadno. Sice $\alpha(C_5) = 2$, ale $\alpha(C_5^2) = 5$. Z čehož plyne druhá nerovnost. \square



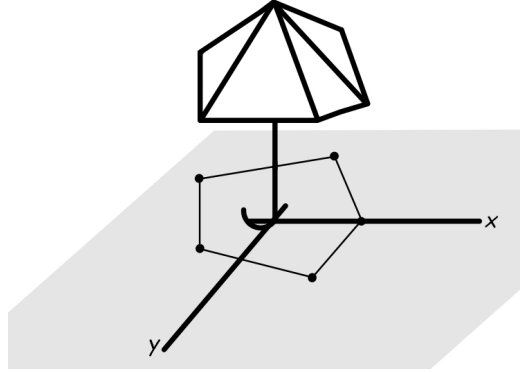
Obrázek 5.3: Projekce \mathbf{u}_i do první a druhé souřadnice.

5.3 Semidefinitní programy pro $\vartheta(G)$

Program pro $1/\sqrt{\vartheta}$

První formulací je semidefinitní program, jehož řešením je hodnota $\frac{1}{\sqrt{\vartheta}}$. Mějme graf $G = (V, E)$. Hodnota Lovászovy Theta funkce $\vartheta(G)$ je z definice

$$\vartheta(G) = \min_{\mathcal{U}} \vartheta(\mathcal{U}) = \min_{\mathcal{U}} \min_{\|\mathbf{c}\|=1} \max_{i \in V(G)} \frac{1}{(\mathbf{c}^T \mathbf{u}_i)^2},$$



Obrázek 5.4: Lovászův deštník.

kde \mathcal{U} probíhá přes všechny ortonormální reprezentace grafu G . Pokud se stane, že $(\mathbf{c}^T \mathbf{u}_i)^2 \leq 0$, potom místo \mathbf{u}_i budeme dále uvažovat vektor $-\mathbf{u}_i$. Můžeme tedy předpokládat, že $\forall i \in V(G) : (\mathbf{c}^T \mathbf{u}_i)^2 \geq 0$. Potom hodnotu $1/\sqrt{\vartheta(G)}$ můžeme vyjádřit jako

$$\frac{1}{\sqrt{\vartheta(G)}} = \max_{\mathcal{U}} \frac{1}{\sqrt{\vartheta(\mathcal{U})}} = \max_{\mathcal{U}} \max_{\|c\|=1} \min_{i \in V(G)} \mathbf{c}^T \mathbf{u}_i.$$

Z čehož dostaneme následující vektorový program

$$\begin{aligned} & \max t \\ & \forall i, j \in E(\tilde{G}) : \langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0 \\ & \forall i \in V(G) : \langle \mathbf{c}, \mathbf{u}_i \rangle \geq t \\ & \forall i \in V(G) : \|\mathbf{u}_i\| = 1 \\ & \|c\| = 1. \end{aligned} \tag{VP1}$$

Z vektorového programu VP1 dále odvodíme semidefinitní program. Budeme uvažovat matici

$$X = \begin{bmatrix} - & \mathbf{c}^T & - \\ - & \mathbf{u}_1^T & - \\ & \vdots & \\ - & \mathbf{u}_n^T & - \end{bmatrix} \begin{bmatrix} | & | & & | \\ \mathbf{c} & \mathbf{u}_1 & \dots & \mathbf{u}_n \\ | & | & & | \end{bmatrix},$$

která je samozřejmě pozitivně semidefinitní. Podmínkám $\forall i \in V(G) : \|\mathbf{u}_i\| = 1$ a $\|c\| = 1$ odpovídá podmínka $x_{ii} = 1$ pro $i = 0, 1, \dots, n$ (pro teď budeme

indexovat matici X od 0). Podmínku $\forall ij \in E(\bar{G}) : \langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$ přepíšeme na

$$\forall ij \in E(\bar{G}) : x_{ij} = 0.$$

A konečně poslední podmínku $\forall i \in V(G) : \langle \mathbf{c}, \mathbf{u}_i \rangle \geq t$ přepíšeme takto

$$\forall i \in V(G) : x_{0i} \geq t.$$

Dostáváme tedy semidefinitní program

$$\begin{aligned} & \max t \\ & x_{ii} = 1, i = 0, 1, \dots, n \\ & \forall ij \in E(\bar{G}) : x_{ij} = 0 \\ & \forall i \in V(G) : x_{0i} \geq t \\ & X \succeq 0. \end{aligned} \tag{SDP1}$$

Program pro ϑ

V původním článku od L. Lovásze [CITACE] je další semidefinitní program, jehož řešením je přímo hodnota $\vartheta(G)$.

$$\begin{aligned} & \max \mathbf{Tr} \, XJ \\ & \forall ij \in E(G) : x_{ij} = 0 \\ & \mathbf{Tr} \, X = 1 \\ & X \succeq 0, \end{aligned} \tag{SDP2}$$

kde J je matice samých jedniček.

5.4 Určení hodnoty ϑ pro konkrétní grafy

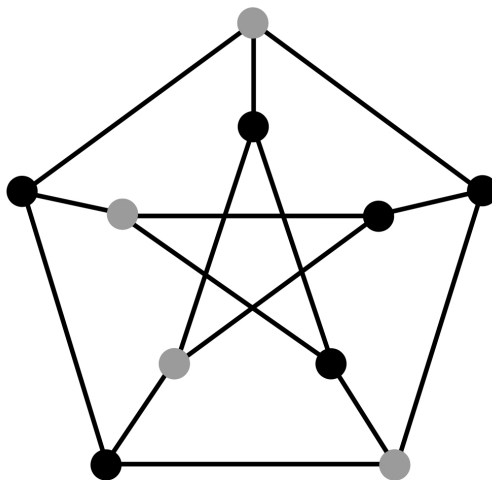
Pomocí formulace SDP2 určíme hodnotu ϑ pro C_5 , C_7 , Petersenův graf, K_5 a S_5 . Nejdříve však uvedeme tzv. Sendvičovou větu.

Věta 15. *Mějme graf G a jeho doplněk \bar{G} . Potom*

$$\alpha(G) \leq \vartheta(G) \leq \chi(\bar{G}).$$

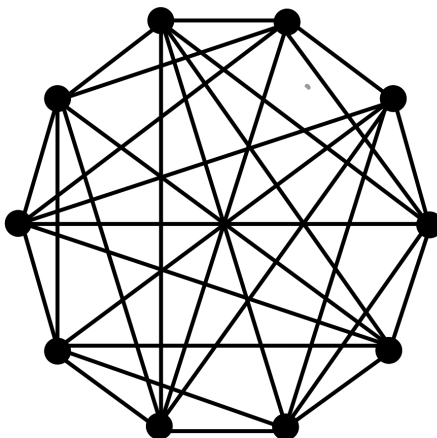
$\alpha(G)$ pro vybrané grafy

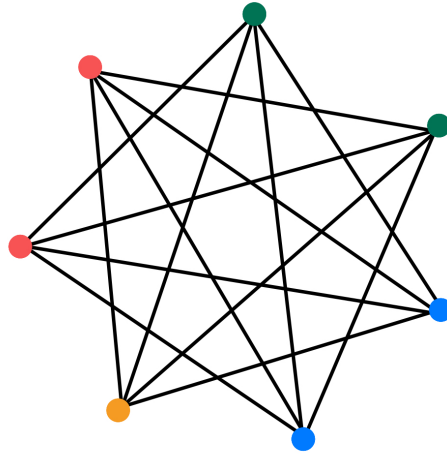
Je zřejmé, že pro $\alpha(C_5) = 2$, $\alpha(C_7) = 3$, $\alpha(K_5) = 1$ a $\alpha(S_5) = 5$. Na obrázku 5.5 je, pro Petersenův graf, nezávislá množina velikosti 4. Probírkou všech možností zjistíme, že větší nezávislou množinu se nám najít nepodaří.

Obrázek 5.5: Největší nezávislá množina v $GP_{5,2}$.

$\chi(\bar{G})$ pro vybrané grafy

Doplněk C_5 je opět C_5 a lichá kružnice má chromatické číslo 3. K_5 má jako svůj doplněk diskrétní graf, který má chromatické číslo 1. U hvězdy S_5 dostaneme jako doplněk graf s šesti vrcholy, kde pět z nich tvoří úplný graf a jeden vrchol nemá žádného souseda. Takový graf má chromatické číslo 5. Pro Petersenův graf je jeho doplněk T_5 , který má $\chi(T_5) = 5$ (T_5 viz Obrázek 5.6). Nakonec doplněk k C_7 je na obrázku 5.7 a opět probírkou všech možností zjistíme, že chromatické číslo je 4.

Obrázek 5.6: Triangular graf T_5 .

Obrázek 5.7: Obarvení \bar{C}_7 .

$\vartheta(G)$ pro vybrané grafy

Dříve jsme ukázali, že $\vartheta(C_5) = \sqrt{5} \approx 2.2361$. Navíc pro liché n platí, že

$$\vartheta(C_n) = \frac{n \cdot \cos(\frac{\pi}{n})}{1 + \cos(\frac{\pi}{n})}.$$

Dostáváme tedy, že $\vartheta(C_7) \approx 3.3177$. Hodnoty ϑ pro zbylé grafy byly spočítány na počítači v softwaru Mosek v programovacím jazyku Python (viz <https://github.com/c0n73x7/D1PL0MK4/blob/master/mosek/theta.py>). Byla využita formulace SDP2. V tabulce 5.1 jsou shrnuty všechny zmíněné hodnoty.

G	$\alpha(G)$	$\vartheta(G)$	$\chi(\bar{G})$
C_5	2	2.2361	3
C_7	3	3.3177	4
$GP_{5,2}$	4	4	5
K_5	1	1	1
S_5	5	5	5

Tabulka 5.1: $\alpha(G)$, $\vartheta(G)$, $\chi(\bar{G})$ pro vybrané grafy.

Co víme o $\Theta(G)$

Shannonova kapacita je známá jen pro několik málo grafů. V [REF] Shannon dal dolní odhad na $\Theta(C_5)$ a až za 23 let Lovász ukázal pomocí konstrukce,

kterou jsme si ukázali výše, že $\Theta(C_5) = \sqrt{5}$. Ve stejném článku [REF] je důkaz, že kapacita Petersenova grafu je $\Theta(GP_{5,2}) = 4$. Triviální případy S_5 a K_5 dostaneme ze sendvičové věty, tj. $\Theta(S_5) = 5$ a $\Theta(K_5) = 1$. Naopak pro C_7 hodnotu Θ neznáme. Máme dolní odhad $\alpha(C_7) = 3$ a horní odhad $\vartheta(C_7) \approx 3.3177$. Lepší dolní odhad, než dává $\alpha(C_7)$, ukázali Polak a Schriever [REF] tak, že pomocí počítače našli nezávislou množinu v grafu C_7^5 velikosti 367. Z čehož dostaneme dolní odhad $\sqrt[5]{367} \approx 3.2579$. Hodnota $\Theta(C_7)$ je tedy někde mezi

$$3.2578 < \Theta(C_7) \leq 3.3177.$$

Kapitola 6

Problém maximálního řezu

6.1 Formulace úlohy

Mějme neorientovaný graf $G = (V, E)$ s nezáporným ohodnocením hran w . Cílem je rozložit množinu vrcholů V na nejvýše $k \in \mathbf{N} \setminus \{1\}$ disjunktních množin tak, aby součet hran spojující různé množiny byl maximální. Pokud $k = 2$ hovoříme o úloze **MAX CUT** a pro $k \geq 3$ o úloze **MAX k -CUT**.

6.2 Úloha MAX CUT

Nejprve se podíváme na aproximační algoritmus z článku [REF] pro úlohu MAX CUT.

Striktní kvadratický program pro MAX CUT

Kvadratický program je problém maximalizace / minimalizace kvadratické funkce celočíselných proměnných, vzhledem ke kvadratickým omezením těchto proměnných. Je-li navíc každý monom (jednočlen) cenové funkce i daných omezení stupně 0 nebo 2, potom mluvíme o **strikním kvadratickém programu**.

Dále odvodíme strikní kvadratický program pro úlohu MAX CUT. Necht' $y_i \in \{1, -1\}$ je proměnná příslušná vrcholu i . Množiny S a \bar{S} definujeme následovně

$$S = \{i \in V \mid y_i = 1\}, \bar{S} = \{i \in V \mid y_i = -1\}.$$

Pokud $i \in S$ a $j \in \bar{S}$, potom je součin $y_i y_j = -1$ a chceme, aby tato hrana přispívala hodnotou w_{ij} k cenové funkci. Ve zbylých dvou možnostech je

$y_i y_j = 1$ a chceme, aby se hodnota cenové funkce nezměnila. Použitím těchto podmínek definujeme striktní kvadratický program.

$$\begin{aligned} OPT = \max & \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} (1 - y_i y_j) \\ & \forall i \in V : y_i^2 = 1, \\ & \forall i \in V : y_i \in \mathbf{Z}. \end{aligned} \quad (\text{SQ-MAX-CUT})$$

Vektorový program pro MAX CUT

Poznamenejme jen, že úloha celočíselného programování je NP-těžká. Proto se dále budeme zabývat relaxací úlohy SQ-MAX-CUT, což znamená, že upustíme od podmínek celočíselnosti a původní úlohu relaxujeme vektorovým programem. Modifikujeme tedy program SQ-MAX-CUT tak, že každý součin $y_i y_j$ nahradíme skalárním součinem vektorů $\langle \mathbf{v}_i, \mathbf{v}_j \rangle$ v \mathbf{R}^n . Dostáváme následující vektorový program.

$$\begin{aligned} RELAX = \max & \frac{1}{2} \sum_{1 \leq i < j \leq n} w_{ij} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \\ & \forall i \in V : \langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1 \\ & \forall i \in V : \mathbf{v}_i \in \mathbf{R}^n \end{aligned} \quad (\text{V-MAX-CUT})$$

Semidefinitní program pro MAX CUT

Uvedeme si ještě semidefinitní formulaci předchozí vektorové relaxace. Nechť W je matice vah jednotlivých hran taková, že nenulové prvky má pouze nad diagonálou a w_{ij} je váha hrany ij . Matice J je matice samých jedniček.

$$\begin{aligned} RELAX = \max & \frac{1}{2} W \bullet (J - Y) \\ & \forall i \in V : y_{ii} = 1 \\ & Y \succeq 0 \end{aligned} \quad (\text{SDP-MAX-CUT})$$

Randomizovaný zaokrouhlovací algoritmus

Mějme dva vektory $\mathbf{a}_i, \mathbf{a}_j$ optimálního řešení programu V-MAX-CUT. Označme Θ_{ij} úhel, který tyto vektory svírají. Z podmínky

$$\forall i \in V : \langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1$$

dostáváme, že $\langle \mathbf{a}_i, \mathbf{a}_j \rangle = \cos \Theta_{ij}$ a příspěvek těchto vektorů k optimálnímu řešení je

$$\frac{w_{ij}}{2}(1 - \cos \Theta_{ij}).$$

Tedy čím „blíží“ je úhel Θ_{ij} hodnotě π , tím větší příspěvek mají tyto vektory k hodnotě optimálního řešení. Dále si uvedeme algoritmus pro řešení úlohy MAX-CUT a jeho analýzu.

Algoritmus (MAX-CUT).

1. Najdi řešení $\mathbf{a}_1, \dots, \mathbf{a}_n$ programu V-MAX-CUT.
2. Zvol náhodně vektor \mathbf{r} na jednotkové sféře S_{n-1} .
3. $S = \{i \in V \mid \langle \mathbf{a}_i, \mathbf{r} \rangle \geq 0\}$.

Začneme dvěma lemmaty.

Lemma 9. *Nechť X_{ij} je jev takový, že vrcholy i a j jsou od sebe separovány, tj. jsou v různých množinách. Potom*

$$P[X_{ij}] = \frac{\Theta_{ij}}{\pi}.$$

Důkaz. Vektory $\mathbf{a}_i, \mathbf{a}_j$ generují rovinu. Uvažme projekci náhodného vektoru \mathbf{r} na jednotkové sféře S_{n-1} do této roviny. Potom vrcholy i a j jsou separovány, jestliže

$$\langle \mathbf{a}_i, \mathbf{a}_j \rangle = \langle \mathbf{r}, \mathbf{a}_i \rangle + \langle \mathbf{r}, \mathbf{a}_j \rangle$$

nebo

$$\langle \mathbf{a}_i, \mathbf{a}_j \rangle = \langle -\mathbf{r}, \mathbf{a}_i \rangle + \langle -\mathbf{r}, \mathbf{a}_j \rangle.$$

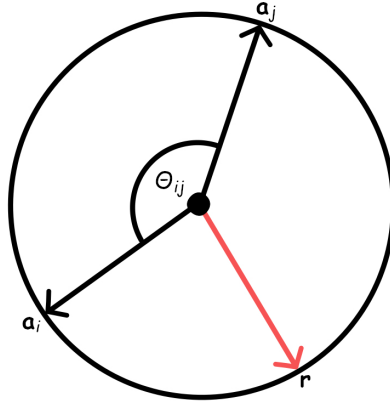
Předchozí podmínku separace vrcholů ilustruje obrázek 6.1. □

Lemma 10 (KNUTH 2, 135). *Nechť $\mathbf{x} = (x_1, \dots, x_n)$ je vektor, jehož prvky jsou zvoleny nezávisle z normovaného normálního rozdělení $N(0, 1)$. Potom $\mathbf{r} = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ je náhodný vektor, který leží na jednotkové sféře S_{n-1} .*

Dále ukážeme, jak „dobrou“ aproximaci algoritmem **REF** dostaneme. Označme

$$\alpha = \min_{0 \leq \Theta \leq \pi} \frac{2\Theta}{\pi(1 - \cos \Theta)}.$$

Snadno se ukáže, použitím derivace, že $\alpha \approx 0.87856$.

Obrázek 6.1: Separace vrcholů i, j náhodným vektorem \mathbf{r} .

Lemma 11. *Nechť Y je náhodná veličina, která označuje součet vah hran, které vedou z S do \bar{S} , nalezeny algoritmem **REF**. Potom*

$$E[Y] \geq \alpha \cdot RELAX.$$

Důkaz. Z definice čísla α , pro $0 \leq \Theta \leq \pi$, dostáváme

$$\frac{\Theta}{\pi} = \frac{2\Theta}{\pi(1 - \cos \Theta)} \cdot \frac{1 - \cos \Theta}{2} \geq \frac{\alpha}{2}(1 - \cos \Theta). \quad (6.1)$$

Použitím lemmatu 9 a nerovnosti 6.1 dostáváme

$$\begin{aligned} E[Y] &= \sum w_{ij} P[X_{ij}] \\ &= \sum w_{ij} \frac{\Theta_{ij}}{\pi} \\ &\geq \frac{\alpha}{2} \sum w_{ij} (1 - \cos \Theta_{ij}) \\ &= \alpha \cdot RELAX. \end{aligned}$$

□

Poznamenejme, že samozřejmě platí

$$OPT \geq E[Y] \geq \alpha \cdot RELAX.$$

Definujeme **mezeru celočíselnosti** relaxace (pro maximalizační problém) jako

$$\inf_I \frac{OPT(I)}{RELAX(I)},$$

kde infimum probíhá přes všechny instance I daného programu. Tedy mezera celočíselnosti relaxace V-MAX-CUT je alespoň $\alpha \approx 0.87856$.

Předchozí odvození je založeno na střední hodnotě náhodné veličiny Y . Proto kroky **2** a **3**, v algoritmu, opakujeme vícekrát a jako výsledek zvolíme množinu S , která dává největší součet hran z S do \bar{S} . Dále jen specifikujeme kolikrát musíme tyto kroky opakovat. Kompletní důkaz je v **CITE**. Zvolíme tedy $\varepsilon > 0$ (malé), nechť

$$c = \frac{\varepsilon \alpha}{2 + 2\varepsilon - \alpha},$$

a kroky **2**, **3** opakujeme $\lceil \frac{1}{c} \rceil$ -krát.

6.3 Úloha MAX k -CUT

V následující části si uvedeme jen formulace s aproximačními schématy z několika článků pro úlohu MAX- k -CUT a provedeme experiment, ve kterém je porovnáme na náhodných grafech.

6.3.1 Frieze-Jerrum a MAX- k -CUT

Formulace

$$\begin{aligned} \max \quad & \frac{k}{k-1} \sum_{1 \leq i < j \leq n} w_{ij} (1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle) \\ \forall i \in V : \quad & \langle \mathbf{v}_i, \mathbf{v}_i \rangle = 1, \\ \forall i, j \in V : \quad & \langle \mathbf{v}_i, \mathbf{v}_j \rangle \geq -\frac{1}{k-1} \\ \forall i \in V : \quad & \mathbf{v}_i \in \mathbf{R}^n. \end{aligned} \tag{FJ-MAX- k -CUT}$$

Aproximační schéma

6.3.2 Goemans-Williamson a MAX-3-CUT

Formulace

$$\max \frac{2}{3} \sum_{1 \leq i < j \leq n} w_{ij} (1 - \langle \mathbf{v}_i^1, \mathbf{v}_j^1 \rangle)$$

$$\forall i \in V \forall a, b \in \{1, 2, 3\} : a \neq b \implies \langle v_i^a, v_i^b \rangle = -\frac{1}{2}$$

$$\forall i, j \in V \forall a, b, c \in \{1, 2, 3\} : \langle v_i^a, v_i^b \rangle = \langle v_i^{a+c}, v_i^{b+c} \rangle \quad (\text{GW-MAX-3-CUT})$$

$$\forall i, j \in V \forall a, b \in \{1, 2, 3\} : \langle v_i^a, v_j^b \rangle \geq -\frac{1}{2}$$

$$\forall i \in V \forall a \in \{1, 2, 3\} : \langle v_i^a, v_i^a \rangle = 1$$

$$\forall i \in V \forall a \in \{1, 2, 3\} : v_i^a \in \mathbf{R}^{3n}$$

Aproximační schéma

6.3.3 de Klerk-Pasechnik-Warners a MAX- k -CUT

Formulace

Aproximační schéma

6.3.4 Newman a MAX- k -CUT

Formulace

Používá formulaci FJ-MAX- k -CUT.

Aproximační schéma

6.3.5 Porovnání algoritmů

- tabulka s aproximačními poměry (vyznačit nejlepší)
- tabulky s jednotlivými výsledky běhů programů

Závěr