Univerzita Karlova v Praze

Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE

Kristýna Krejčová

Ortogonální kontraktor

Katedra aplikované matematiky

Vedoucí bakalářské práce:	Mgr. Milan Hladík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Obecná informatika

Praha 2013

Děkuji především svému školiteli Mgr. Milanu Hladíkovi Ph.D. za obětavý přístup, velkou trpělivost, cenné rady a konzultace, které mi vždy ochotně poskytl. Velký dík patří také mým přátelům za celkovou podporu.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V .... dne .... Podpis autora

Název práce: Ortogonální kontraktor

Autor: Kristýna Krejčová

Katedra: Katedra aplikované matematiky, MFF UK

Vedoucí bakalářské práce: Mgr. Milan Hladík, Ph.D., Katedra aplikované matematiky, MFF UK

Abstrakt: Práce se zabývá návrhem ortogonálního kontraktoru. Na začátku definujeme základní pojmy intervalové analýzy a intervalové lineární algebry. Dále navrhneme několik metod řešení. Na základě propagačních technik ukážeme kontrakci podle kvadratických podmínek a bilineárních podmínek, které dostaneme z vlastností ortogonálních matic. Dále jsme linearizovali bilineární podmínky a použili lineární programování. Kontraktoru podle vlastních čísel jsme využili ke konstrukci kontraktoru podle determinantu. Navržené metody řešení jsou implementovány v Matlabu s použitím toolboxu Intlab a knihovny Versoft. Na závěr navržené metody otestujeme a porovnáme. Na základě porovnání dílčích technik sestavíme výsledný kontraktor.

Klíčová slova: ortogonální matice, kontraktor, intervalová matice, intervalová analýza

Title: Orthogonal kontraktor

Author: Kristýna Krejčová

Department: Katedra aplikované matematiky, MFF UK

Supervisor: Mgr. Milan Hladík, Ph.D., Katedra aplikované matematiky, MFF UK

Abstract: The thesis proposes an orthogonal contractors. At the beginning we define the basic concepts of interval analysis and interval linear algebra. Then, we propose several contractor methods. Based on propagation techniques, we propose contraction according to bilinear and quadratic terms of the conditions that we get from the properties of orthogonal matrices. We also linearize bilinear terms and use linear programming. The known contractors by eigenvalues we employed to the construction contractor by the determinant. The proposed methods are implemented in Matlab with the use of the toolbox Intlab and the library Versoft. Finally, we test and compare the presented methods. Based on the comparison of particular techniques we build a final contractor.

Keywords: orthogonal matrix, contractor, interval matrix, interval analysis

Contents

Úvod

Od pradávna nacházela matematika velké uplatnění ve fyzikálních a technických výpočtech. Praktické využití matematiky v technice nastartovalo a značně přispělo k rychlému rozvoji elektroniky a počítačů v posledních letech. V dnešní době je využití počítače, jakožto silného výpočetního nástroje, při řešení těchto úloh naprosto stěžejní.

V matematice při numerických výpočtech fyzikálních a technických úloh pracujeme s čísly celými, racionálními a iracionálními. Počítač umí reprezentovat čísla a pracovat s nimi pouze s omezenou přesností. Často nastává situace, že bychom potřebovali do paměti uložit více cifer než je pro dané číslo vyhrazeno. Pak musíme toto číslo zaokrouhlit a dále pracovat s jeho aproximací. Pak výsledek celého výpočtu už nemůže být zcela přesný.

Obor intervalové analýzy, do něhož tematicky spadá i tato práce, přichází s možností řešení tohoto problému. Základní myšlenka spočívá v tom, že číslo, jehož všechny cifry se nevlezou do paměti, nahradíme intervalem. Interval tvoří uspořádaná dvojce čísel značící začátek a konec intervalu. Začátek intervalu představuje největší číslo, které je menší než nahrazované číslo a zároveň je reprezentovatelné v počítači. Podobně druhé číslo je nejmenší reprezentovatelné číslo, jenž je větší než nahrazované číslo.

S intervaly budeme pracovat podobně jako s původními čísly. Matematické operace budeme provádět tak, abychom měli jistotu, že když bychom vzali libovolná čísla z jednotlivých intervalů a provedli s nimi stejnou matematickou operaci, její výsledek bude vždy patřit do výsledného intervalu.

Počítání s intervaly nám však nabízí podstatě širší možnosti uplatnění. Při numerickém zpracování experimentů a prezentaci jejich výsledků jsme v mnohých případech více omezeni přesností měřících metod a přístrojů než přesností čísel v počítači. Přesnost měřícího přístroje není možné z fyzikálních důvodů libovolně zpřesnit. Naměřené hodnoty včetně jejich odchylek reprezentujeme jako intervaly a můžeme použít intervalovou aritmetiku další algoritmy pracující s intervaly.

0.0.1 Cíl práce

V tomto odstavci intuitivně nastíním pojmy, které jsou důležité pro tuto práci, a vysvětlím zadání práce.

Vezměme si matici, jejíž každý člen je interval. V podstatě pouze nahradíme čísla intervaly. Tím dostaneme intervalovou matici. Každý číselný interval měl prvky, což byly čísla, které do něj patřily. Podobně je to s intervalovou matici. Prvky intervalové matice jsou matice, jejíž členy jsou libovolná čísla z intervalů, které představují členy.

Úkolem této práce je navrhnout ortogonální kontraktor pro intervalovou matici. Kontraktor můžeme chápat jako algoritmus, který dostane na vstupu nějakou intervalovou matici. Jeho úkolem je co nejvíce zkrátit intervaly v matici, tak aby neodstranil žádnou ortogonální matici, která patřila do původní intervalové matice.

Ortogonální matice obecně mají determinant rovný 1 nebo -1. Kontraktor si ještě dourčíme tím, že budeme vybírat pouze matice s determinantem rovným

-1.

0.0.2 Motivace

Intervalová analýza sama nachází širokou škálu praktického využití v technických výpočtech, zvláště pak v robotice. Konkrétně náš problém se používá při kalibraci robotů.

Proces kalibrace robotů můžeme zjednodušeně chápat jako počáteční nastavení robota, které mu umožní lepší funkčnost. Potřebujeme, abychom mohli polohovat robota s velkou přesností. Jednotlivé díly nemohou být principiálně vyrobeny s touto přesností. Robot je navržen s jistými nominálními rozměry a vyroben s jistými skutečnými rozměry, které se od nominálních liší o všudypřítomné odchylky. Robot je ovládán nějakým počítačovým systémem. Pokud se nám podaří odchylky od nominálních rozměrů konkrétního robota zjistit a zadat do řídícího systému robota, je schopen řídící systém tyto odchylky kompenzovat a tím dosáhnout výrazně vyšší přesnosti.

Více se o dané problematice můžeme dozvědět například zde [17].

0.0.3 Reprezentace čísel

Ve většině výpočtů se neobejdeme bez racionálních a iracionálních čísel, které mohou mít ve svém zápisu nekonečně mnoho desetinných míst. Tyto čísla můžeme vyjádřit buď v matematickém tvaru, například jako $\sqrt{2}$ nebo použít jejich aproximaci pomocí čísla s konečným počtem desetinných míst. To nás staví před problém reprezentace čísel v počítači.

Pokud jsme schopni čísla reprezentovat přesně včetně mezivýsledků, výpočty s nimi jsou pak přesné. Přitom nezáleží, zda je číslo celé nebo desetinné s malým počtem cifer. Obdobného stavu můžeme dosáhnout při výpočtech s iracionálními čísly a racionálními čísly s velkým počtem cifer, vyjádříme-li je pomocí algebraického tvaru a ním pak pracujeme. Tímto způsobem umí s čísly zacházet některé algebraické programy, například Matematica. Pro značnou část výpočtů však tento způsob je nevhodný nebo příliš časově náročný. Proto jsme nuceni pracovat pouze s konečnou reprezentací čísel a zaokrouhlovat. Během výpočtu pak dochází k další akumulaci chyb kvůli dalšímu zaokrouhlování.

Předpokládejme, že máme nějaký výpočet s malým množstvím matematických operací. Poprvé jej provedeme s nějakou přesností. Reprezentace čísel bude mít určité množství cifer. Podruhé jej spočítáme s větší přesností. Čísla budeme reprezentovat větším počtem cifer. Nyní nás bude zajímat, jestli cifry výsledků obou výpočtů, které jsou stejné, můžeme považovat za správné.

Mějme rekurzivní formuli[1]

$$x_{n+1} = x_n^2 \qquad n = 0, 1, 2, \dots$$

Vezměme první hodnotu posloupnosti $x_0 = 1-10^{-21}$ a spočtěme x_{75}. Provedeme-li tento výpočet v desetimístné aritmetice dopočítáme se k výsledku 1. Požitím dvacetimístné aritmetiky dostaneme výsledek opět 1. Ve skutečnosti však hledaná hodnota $x_{75} < 10^{-10}$.

[1] Příklad byl převzat z [4].

Tento příklad jasně ukazuje, že výše zmíněná domněnka o potvrzení správnosti prvních cifer výsledku pomocí jeho zpřesněním je špatná. V našem příkladu problém vzniká kvůli tomu, že neumíme přesně vyjádřit x_0 v deseti- ani dvacetimístné reprezentaci.

Další příklady bychom mohli nalézt v [4] nebo v [16].

V mnoha případech platí, že pokud zpřesníme výpočet, použijeme více cifer, dostaneme přesnější výsledek. Musíme si však uvědomit, ze toto není univerzální pravidlo a nejde na něj spolehnout.

Často můžeme dosáhnout libovolné přesnosti výpočtu pokud používáme dostatek cifer a začneme s čísly, které je možné přesně reprezentovat. Obecně je však těžké říci, kolik cifer musíme použít, abychom této požadované přesnosti dosáhli.

Obecně platí fakt, že čím více desetinných míst jsme ponechali, tím bude výpočet přesnější. Přidání desetinných míst nám výpočet nepokazí. Problém však zůstává v tom, že neumíme určit, jak moc je nutné zpřesnit výpočet abychom dostali správný výsledek nebo alespoň požadovaný počet správných cifer. Pro různé výpočty se závislost přesnosti čísel na počátku a přesnost výsledku značně liší.

1. Základní pojmy

1.0.4 Jednoduchý interval a intervalová aritmetika

Interval

Definice 1. *Interval nazveme množinu*

$$\boldsymbol{x} := [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}$$

Body $\underline{x}$ a $\overline{x}$ nazýváme koncové body intervalu.

V řeči matematické analýzy budeme pojmem interval vždy chápat uzavřený reálný interval.

Dva intervaly jsou si rovny, pokud jsou si rovny jejich koncové body.

Definice 2. *Máme interval x pak, středem intervalu nazveme bod*

$$x_c = \frac{\underline{x} + \overline{x}}{2},$$

poloměrem nazveme bod

$$x_\Delta = \frac{\overline{x} - \underline{x}}{2}$$

a šířkou intervalu nazveme

$$w(\boldsymbol{x}) = \overline{x} - \underline{x}$$

Definice 3 (Degenerovaný interval)**.** *Interval nazveme degenerovaný, pokud pro něj platí, že $\underline{x} = \overline{x}$.*

Do degenerovaného intervalu patří právě jedno reálné číslo, které je rovno jeho koncovým bodům. V intervalové aritmetice jej používáme k vyjádření reálného čísla pomocí intervalu. To je užitečné zejména při výpočtech na počítači. Máme všechny vstupní čísla ve stejném formátu a můžeme na ně použít pravidla intervalové aritmetiky.

Sjednocení a průnik intervalů

Průnik intervalů odpovídá množinovému průniku, protože průnik dvou libovolných intervalů je zase interval. Pomocí horních a dolních mezí jej můžeme vyjádřit takto:

$$\boldsymbol{x} \cap \boldsymbol{y} = \{z \mid z \in \boldsymbol{x} \wedge z \in \boldsymbol{y}\}$$

Pokud je průnik neprázdný můžeme jej zapsat jako interval

$$[\,\max(\underline{x}, \underline{y}),\ \min(\overline{x}, \overline{y})\,]$$

Průnik je v intervalové analýze hojně využíván. Je to často v situaci, kdy máme dva intervaly a hledaná hodnota patří do obou z nich. Pak musí patřit do jejich průniku.

Množinové sjednocení dvou intervalů nemusí být nutně souvislý interval. Jasně to vidíme například na sjednocení intervalů $[1, 2]$ a $[3, 4]$.

Pro potřeby počítání s intervaly definujeme sjednocení intervalů jako nejmenší nadmnožinu množinového sjednocení intervalů. Tato množina odpovídá vztahu:

$$\boldsymbol{x} \cup \boldsymbol{y} = [\,\min(\underline{x}, \underline{y}),\ \max(\overline{x}, \overline{y})\,]$$

Nevýhodou tohoto pojetí je ztráta informace. Do výsledku přidáváme navíc hodnoty, které nepatří ani do jednoho z intervalů.

Intervalová aritmetika

Na intervalech definujeme základní aritmetické operace, které běžně používáme s reálnými čísly. Operace s intervaly jsou definovány tak, abychom měli zaručeno, že pokud vezmeme libovolné číslo z intervalů, které vstupují do operace, výsledek vždy bude ležet ve výsledném intervalu. Pomocí množin bychom to mohli zapsat takto:

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \wedge y \in \boldsymbol{y}\}$$

Rozepíšeme jednotlivé operace s intervaly:

$$\boldsymbol{x} + \boldsymbol{y} = [\,\underline{x} + \underline{y},\ \overline{x} + \overline{y}\,]$$

$$\boldsymbol{x} - \boldsymbol{y} = [\,\underline{x} - \overline{y},\ \overline{x} - \underline{y}\,]$$

Násobení už je trochu komplikovanější.

$$\boldsymbol{x} \cdot \boldsymbol{y} = [\,\min S,\ \max S\,] \qquad S := \{\underline{x}\,\underline{y},\ \underline{x}\,\overline{y},\ \overline{x}\,\underline{y},\ \overline{x}\,\overline{y}\}$$

Na základě porovnávání koncových bodů intervalů s nulou můžeme určit výsledný interval efektivněji.

Dělení intervalů definujeme pomocí převrácené hodnoty a násobení.

$$\boldsymbol{x}/\boldsymbol{y} = \boldsymbol{x} \cdot 1/\boldsymbol{y} \qquad 1/\boldsymbol{y} = [1/\overline{y}, 1/\underline{y}] \qquad 0 \notin \boldsymbol{y}$$

Z definic 1 a 2 použitím intervalové aritmetiky plyne vztah

$$\boldsymbol{x} = x_c + [-x_\Delta, x_\Delta] = x_c + \left[-\frac{1}{2}w(\boldsymbol{x}), \frac{1}{2}w(\boldsymbol{x})\right]$$

1.0.5 Intervalové matice a počítání s nimi

Intervalová matice

Definice 4. *Matici $\boldsymbol{A}$ nazveme intervalovou, pokud*

$$\boldsymbol{A} := [\underline{A}, \overline{A}] = \{A \in \mathbb{R}^{m \times n} \mid \underline{A} \leq A \leq \overline{A}\}$$

kde $\underline{A}, \overline{A} \in \mathbb{R}^{m \times n}$ a pro každé $1 \leq i \leq m$ a $1 \leq j \leq n$ platí že $\underline{A}_{ij} \leq \overline{A}_{ij}$, což zapisujeme stručněji jako $\underline{A} \leq \overline{A}$.

Intervalovou matici si lze představit jako matici, která má místo členů intervaly. Koncové matice $\underline{A}$ resp. $\overline{A}$ jsou matice, jenž vzniknou z horních resp. dolních mezí intervalů. Prvek intervalové matice je reálná matice $A \in \mathbb{R}$, jejíž všechny členy jsou čísla z příslušných intervalů.

Pro obyčejný interval jsme definovali pojmy střed, poloměr a šířka intervalu. Provedeme jejich zobecnění pro intervalové matice.

Definice 5. *Střed A_c a poloměr A_Δ intervalové matice $\boldsymbol{A}$ matice definujeme těmito vztahy:*

$$A_c := \frac{1}{2}(\underline{A} + \overline{A})$$
$$A_\Delta := \frac{1}{2}(\overline{A} - \underline{A})$$
$$w(\boldsymbol{A}) := \overline{A} - \underline{A}$$

Intervalová aritmetika matic

Práce s reálnými maticemi má svoje pravidla. Pokud máme intervalovou matici a intervalovou aritmetiku, můžeme tyto pravidla jednoduchým způsobem zobecnit pro intervalové matice. Tímto způsobem dostaneme operace sčítaní a násobené matic, násobení konstantou a další, které využíváme při složitějších výpočtech.

Díky nejistotě dané intervalem vznikají nové problémy. Použití algoritmů pro reálné matice s intervalovou aritmetikou není vždy nejlepším řešením. Příklad můžeme najít třeba v [11], kde se hovoří o problému přeurčených lineárních systémů rovnic. Na řešení je použita Gaussova eliminace upravená pro počítání s intervaly. Dále jsou demonstrovány jiné metody, které dosahují lepších výsledků.

1.0.6 Ortogonalita

Definice 6. *Reálnou čtvercovou matici Q nazveme ortogonální (nebo také ortonormální) jestliže platí*

$$Q^T = Q^{-1}$$

tj.: její transponovaná matice je současně maticí inverzní.

Vlastnosti ortogonálních matic

1. Označme q_j, $j = 1 \dots n$ sloupce ortogonální matice Q. Pak z rovnosti

   $$QQ^T = I$$

   plyne:

   - $\langle q_i, q_j \rangle = 1$ pokud $i = j$ normované vektory délky 1
   - $\langle q_i, q_j \rangle = 0$ pokud $i \neq j$ vektory jsou vzájemně na sebe kolmé

   tato vlastnost říká, že sloupce ortogonální matice jsou normované vektory délky jedna, které jsou na sebe vzájemně kolmé. Totéž platí i pro řádky matice.

2. $\det(Q) = \pm 1$

Těchto dvou vlastností budeme využijeme při konstrukci kontraktoru.

1.0.7 Kontraktor

V [1] je formálně definován kontraktor pro symetrické intervalové matice a jejich vlastní čísla. Formální definice našeho kontraktoru bude podobná.

Definice 7 (Problém splňování podmínek, CSP[1]). *Problém splňování podmínek se skládá z*

- *konečné množiny proměnných*
- *domén proměnných. Doména proměnné je množina hodnot, kterých může proměnná nabývat.*
- *konečné množiny podmínek. Podmínka je libovolná relace nad množinou proměnných.*

Úkolem CSP je najít všechny ohodnocení proměnných, tak aby byly splněny všechny podmínky.

Náš problém můžeme jako problém splňování podmínek vyjádřit takto:

Definice 8. *Náš CSP*

(i) matice $Q \in \boldsymbol{Q}$,

(ii) Q je ortogonální

V našem problému máme jednu proměnnou, matici Q a jednu podmínku na ortogonalitu této matice. Doménou proměnné je vstupní intervalová matice $\boldsymbol{Q}$.

Při řešení CSP zkracujeme domény proměnných, dokud nebudou platit všechny podmínky pro všechny hodnoty proměnných. V našem případě je doménou intervalová matice, takže budeme zkracovat jednotlivé intervaly v matici.

CSP řešíme metodou propagace podmínek. Každou podmínku c_j nahradíme operátorem C_j, které bude zkracovat domény proměnných o hodnoty, jenž nejsou konzistentní s příslušnou podmínkou c_j. Tyto operátory nazýváme kontraktory.

Pro podmínku (ii) z definice 8 definujeme kontraktor takto:

Definice 9 (Kontraktor). *Kontraktor příslušný k podmínce ortogonality je*

$$C(\boldsymbol{Q}) = \boldsymbol{Q}'$$

tak, že

1. $\boldsymbol{Q} \subseteq \boldsymbol{Q}'$
2. *Q je ortogonální matice, pak $Q \in \boldsymbol{Q} \Rightarrow Q \in \boldsymbol{Q}'$*

[1]zkratka pochází z anglického názvu Constraint Satisfaction Problem

2. Návrh kontraktoru

2.0.8 Obecné vlastnosti

Vlastnost 1 ortogonálních matic popisuje reálnou ortogonální matici Q soustavou rovnic. Pro řádky matice dostáváme

$$\sum_{j=1}^{n} q_{ij}^2 = 1 \quad \text{pro } i = 1 \dots n \qquad\qquad \sum_{j=1}^{n} q_{ij} q_{kj} = 0 \quad \text{pro všechny } i \neq k \tag{2.1}$$

a pro sloupce

$$\sum_{i=1}^{n} q_{ij}^2 = 1 \quad \text{pro } j = 1 \dots n \qquad\qquad \sum_{i=1}^{n} q_{ij} q_{ik} = 0 \quad \text{pro všechny } j \neq k \tag{2.2}$$

Z této vlastnosti také plyne, že každý člen ortogonální matice musí být v absolutní hodnotě menší než jedna. To nám může posloužit jako malá indicie, že členy intervalové ortogonální matice $\boldsymbol{Q}_{i,j}$ nejsou dlouhé intervaly s velkými dírami. Dírou chápeme podinterval $I = [\underline{\boldsymbol{x}}, \overline{\boldsymbol{x}}]$ intervalu $\boldsymbol{Q}_{ij}$, jehož žádný prvek není součást ortogonální matice, ale zároveň v $\boldsymbol{Q}_{ij}$ jsou hodnoty menší než $\underline{\boldsymbol{x}}$ a větší než $\overline{\boldsymbol{x}}$, pro něž ortogonální matice existuje. Protože požadujeme výsledek ve tvaru intervalové matice jsme nuceni tyto díry zahrnout do výsledného intervalu, což způsobuje nafouknutí o nadbytečné hodnoty.

2.0.9 Algoritmus kontraktoru

Na základě rovnic (2.1) a (2.2) navrhneme algoritmus kontraktoru, který bude ořezávat matice $Q \in \boldsymbol{Q}$, jejichž členy nevyhovují těmto rovnicím. Rovnice (2.1) a (2.2) určují reálnou ortogonální matici, jejíž determinant podle vlastnosti 2 je 1 nebo -1. Proto determinant intervalové matice $\boldsymbol{Q}_{ij}$ bude často nadintervalem intervalu $[-1, 1]$. Pokud by -1 nepatřila do intervalu determinantu $\det(\boldsymbol{Q}_{ij})$, pak by v $\boldsymbol{Q}_{ij}$ neexistovala ani jedna ortogonální matice s determinantem rovným -1. V zadání požadujeme, aby výsledná intervalová matice obsahovala pouze matice s determinantem -1. Za tímto účelem navrhneme kontraktor podle determinantu rovnému -1, který bude ořezávat ortogonální matice s determinantem 1.

Kontraktor bude mít dvě části, kontraktor podle rovnic (2.1) a (2.2) (KONTRAKTORPODLEROVNIC()) a kontraktor podle determinantu rovnému -1 (KONTRAKTORPODLEDETERMINANTU()). Kontraktor podle rovnic bude ořezávat neortogonální matice a kontraktor podle determinantu ořeže matice, které mají determinant různý od -1. Průnikem výsledků těchto kontraktorů dostaneme intervalovou ortogonální matici s determinantem -1.

Pro oba kontraktory uděláme proceduru, která se pokusí každou proměnou matice $\boldsymbol{Q}_{ij}$ jednou zkontrahovat. Procedury budeme pouštět střídavě. Celý algoritmus necháme běžet dokud v jednom kroku ani jeden z dílčích kontraktorů neořízne matici.

Další možností je pustit algoritmus na předem nastavený počet kroků.

V pseudokódu bychom to zapsali takto:

KONTRAKTOR($\boldsymbol{Q}$)
  $\forall i, j \quad \boldsymbol{Q}_{ij} \leftarrow \boldsymbol{Q}_{ij} \cap [-1, 1]$
  while (oříznutí ¿ 0) **do**
    oříznutí ← KONTRAKTORPODLEROVNIC(& $\boldsymbol{Q}$)
    oříznutí ← KONTRAKTORPODLEDETERMINANTU(& $\boldsymbol{Q}$)
  end
  return $\boldsymbol{Q}$

2.1 Propagace

2.1.1 Formulace problému jako CCSP

Vlastnost 1 popisuje ortogonální matici množinou rovnic, které pracují s jednotlivými členy matice, což nám umožní vyjádřit kontrakci ortogonální matice jako problém splňování spojitých podmínek a použít techniky pro řešení těchto problémů.

Definice CCSP

Definice 10 (Problém splňování spojitých podmínek, CCSP[1])**.** *Spojitý problém splňování podmínek je dán:*

- *konečnou množinou proměnných a jejich doménami, které jsou tvořeny reálnými intervaly*
- *konečnou množinou podmínek. Podmínka může být například rovnice nebo nerovnice nad proměnnými.*

Úkolem obecného CCSP je najít jednu nebo všechny kombinace hodnot proměnných vybraných z jejich domén, tak aby byly splněny všechny podmínky. My budeme požadovat najití všech hodnot, které vyhovují podmínkám. Tyto hodnoty uzavřeme do intervalů.

Rešení CCSP

CCSP se v praxi řeší vhodnou kombinací různých metod. Klíčová část je tvořena propagací podmínek (constraint propagation), která je obvykle kombinovaná s náhodným prohledáváním nebo prořezáváním větví, pokud chceme prohledat celý prostor možných řešení. Časté je také doplnění různými filtrovacími metodami, které fungují na principech optimalizace.

Všechny možné hodnoty proměnných CCSP si můžeme představit jako n-rozměrný interval, který dostaneme jako skalární součin jednotlivých intervalů, které tvoří domény proměnných.

Filtrovací techniku nazveme propagace podmínek (constraint propagation), jestliže je složená z posloupnosti kroku a v každém kroku používá právě jednu podmínku, podle níž ořezává množinu všech možných bodů řešení.

V praxi tato metoda často funguje tak, že vybírá vhodné podmínky a provádí podle nich redukci všech možných řešení, dokud to jde.

[1]zkratka pochází z anglického názvu Continuous Constraint Satisfaction Problem

Dopředná propagace (Forward propagation) používá meze intervalů jednotlivých proměnných ke zlepšení (globálních) podmínek.

Zpětná propagace (backward propagation) naopak používá (globální) podmínky ke zkrácení intervalů (zlepšení odhadu) jednotlivých proměnných.

Definice našeho CCSP

- Proměnné budou členy reálných ortogonálních matic Q. Označíme je x_{ij}. Jejich domény $\boldsymbol{x}_{ij}$ jsou příslušné intervaly $\boldsymbol{Q}_{ij}$ z intervalové matice $\boldsymbol{Q}$. Všechny domény na začátku omezíme intervalem $[-1, 1]$.

- Podmínky dostaneme z vlastnosti 1 ortogonální matice Q:

  Kvadratické podmínky pro řádky a sloupce:

$$\sum_{j=1}^{n} x_{ij}^2 = 1 \quad \text{pro } i = 1 \dots n \qquad\qquad \sum_{i=1}^{n} x_{ij}^2 = 1 \quad \text{pro } j = 1 \dots n \tag{2.3}$$

  Bilineární podmínky pro řádky a sloupce:

$$\sum_{j=1}^{n} x_{ij} x_{kj} = 0 \quad \text{pro všechny } i \neq k \qquad \sum_{i=1}^{n} x_{ij} x_{ik} = 0 \quad \text{pro všechny } i \neq k \tag{2.4}$$

3. Propagace kvadratických podmínek

V článku [3] je popsaná metoda propagace kvadratických podmínek, kterou upravíme pro naši úlohu.

Kvadratické podmínky máme ve tvaru součtu polynomů, který je roven konstantě. Dopředná propagace by pracovala tak, že by se snažila na základě mezí jednotlivých proměnných zlepšit odhad celého polynomu. Polynom je konstantní, takže v tomto případě dopředná propagace nemá smysl. Proto se dále zaměříme pouze na zpětnou propagaci.

Vybereme si proměnou x_{ij} a nastíníme postup, jakým bude probíhat kontrakce intervalu $\boldsymbol{Q}_{ij}$ příslušného této proměnné. Pro další výklad si zafixujeme index j na pevnou hodnotu a z důvodu vyšší přehlednosti jej budeme vynechávat.

Podmínku (2.3) do zjednodušeného tvaru, aby se s ní lépe pracovalo:

$$\sum_{k=1}^{n} x_k^2 = 1\,, \quad \text{kde } x_{ij} \equiv x_k$$

Interval všech funkčních hodnot funkce x_k^2 pro x_k z intervalu $\boldsymbol{x}_k$ označíme $\boldsymbol{p}_k$. Platí

$$\boldsymbol{p}_k = \boldsymbol{x}_k^2$$

a $\underline{p_k}$ bude dolní mez intervalu $\boldsymbol{p}_k$ a $\overline{p_k}$ je horní mez.

Proměnou p_i budu chápat nějakou hodnotu z intervalu $\boldsymbol{p}_i$. Pak z podmínky (2.3) plyne nerovnost

$$\sum_{k=1,k\neq i}^{n} \overline{p_k} + p_i \geq \sum_{k=1,k\neq i}^{n} p_k + p_i = 1$$

vyjádříme p_i

$$p_i \geq 1 - \sum_{k=1,k\neq i}^{n} \overline{p_k}$$

Analogicky můžeme postupovat pro horní odhad p_i.

$$\sum_{k=1,k\neq i}^{n} \underline{p_k} + p_i \leq \sum_{k=1,k\neq i}^{n} p_k + p_i = 1$$

$$p_i \leq 1 - \sum_{k=1,k\neq i}^{n} \underline{p_k}$$

Propagaci provádíme pro všechny proměnné v podmínce. Abychom ušetřili výpočetní čas, tak si součet všech horních (ale i dolních) mezí spočítáme předem.

Definujeme si interval $\boldsymbol{e} = [\underline{e}, \overline{e}]$ jako:

$$\underline{e} := \sum_{k=1}^{n} \underline{p_k} \qquad\qquad \overline{e} := \sum_{k=1}^{n} \overline{p_k}$$

Pak platí

$$\sum_{k=1}^{n} \boldsymbol{p}_k = [\underline{e}, \overline{e}] \tag{3.1}$$

Dolní a horní odhad pro p_i potom můžeme upravit do tvaru

$$p_i \geq 1 - (\overline{e} - \overline{p_i})$$

$$p_i \leq 1 - (\underline{e} - \underline{p_i})$$

Zbývá vypočítat hranice $\overline{p_k}$ a $\underline{p_k}$. Funkce $f(x) = x^2$ má extrém v bodě $x = 0$ a tento extrém je roven nule. Graf funkce x^2 je orientovaný v kladném směru osy y, takže v bodě $x = 0$ je globální minimum. Pokud bod 0 patří do intervalu $\boldsymbol{x}_k$ je globální minimum funkce x_k^2 na intervalu $\boldsymbol{x}_k$ rovno nule. V opačném případě, $0 \notin x_k$, je funkce x_k^2 monotonní na intervalu $\boldsymbol{x}_k$ a funkční hodnota jedné z mezí je globální minimum na $\boldsymbol{x}_k$ a funkční hodnota druhé meze je globální maximum. Z toho vyplývá, že

$$\overline{p_k} = \max(\underline{x_k}^2, \overline{x_k}^2)$$

a

$$\begin{array}{ll} \underline{p_k} = 0 & \text{pro } 0 \in [\underline{x_k}, \overline{x_k}] \\ \underline{p_k} = \min(\underline{x_k}^2, \overline{x_k}^2) & \text{pro } 0 \notin [\underline{x_k}, \overline{x_k}] \end{array} \tag{3.2}$$

Pro interval $p_i \in [\underline{c}, \overline{c}]$ jsme získali nové meze

$$\underline{c} = 1 - \overline{e} + \overline{p_i}$$

$$\overline{c} = 1 - \underline{e} + \underline{p_i} \tag{3.3}$$

Víme, že $\boldsymbol{x}_i$ je podinterval $[-1, 1]$, proto $\boldsymbol{p}_i = x_i^2$ musí být podinterval $[0, 1]$. U výsledného intervalu $[\underline{c}, \overline{c}]$ provedeme průnik s intervalem $[0, 1]$ a upravíme hranice

$$\underline{c} := \max(\underline{c}, 0)$$

$$\overline{c} := \min(\overline{c}, 1) \tag{3.4}$$

Nyní můžeme ověřit konzistenci podmínky. Pokud platí, že $\overline{c} < \underline{c}$, je interval $[\underline{c}, \overline{c}]$ prázdný a ve výchozí intervalové matici $\boldsymbol{Q}$ není žádná ortogonální matice.

Na základě $p_i \in [\underline{c}, \overline{c}]$ dostaneme dvě nerovnice (pro dolní a horní mez).

$$x_i^2 \geq \underline{c}\,,$$

$$x_i^2 \leq \overline{c}$$

Jejich řešením jsou množiny X_D pro dolní mez a X_H pro horní mez.

$$X_D = \{x \in [-1, 1] \mid x^2 - \underline{c} \geq 0\} = (-1, -\sqrt{\underline{c}}\rangle \cup \langle\sqrt{\underline{c}}, 1)$$

$$X_H = \{x \in [-1, 1] \mid x^2 - \overline{c} \leq 0\} = \langle -\sqrt{\overline{c}}, \sqrt{\overline{c}}\rangle$$

Rovnice platí obě zároveň, takže řešením je průnik.

$$X := X_D \cap X_H = \langle -\sqrt{\overline{c}}, -\sqrt{\underline{c}}\rangle \cup \langle\sqrt{\underline{c}}, \sqrt{\overline{c}}\rangle$$

Hodnoty proměnné x_i musí patřit do původního intervalu x_i a zároveň do množiny řešení konjunkce nerovnic X. Nový interval $\boldsymbol{x}_i'$ dostaneme uzavřením množiny $x_i \cap X$ do nejmenšího možného intervalu.

Při hledání mezí intervalu $\boldsymbol{x}_i'$ mohou nastat následující tři možnosti:

(i) $\underline{x_i} > -\sqrt{\underline{c}}$ pak $\boldsymbol{x}'_i := \boldsymbol{x}_i \cap \langle\sqrt{\underline{c}}, \sqrt{\overline{c}}\rangle$

Interval $\boldsymbol{x}_i$ nemá žádné společné body se zápornou částí množiny X (intervalem v záporné části osy), takže průnik počítáme pouze s kladnou částí X.

(ii) $\overline{x_i} < \sqrt{\underline{c}}$ pak $\boldsymbol{x}'_i := \boldsymbol{x}_i \cap \langle-\sqrt{\overline{c}}, -\sqrt{\underline{c}}\rangle$

Analogická situace pro zápornou část X.

(iii) jinak $\boldsymbol{x}'_i := \boldsymbol{x}_i \cap \langle-\sqrt{\overline{c}}, \sqrt{\overline{c}}\rangle$

Původní interval $\boldsymbol{x}_i$ vyplňuje mezeru mezi intervalem v kladné a záporné části X.

Pseudokód kontrakce podle kvadratických rovnic

Na základě předchozího odstavce zapíšeme algoritmus kontrakce podle kvadratických rovnic pseudokódu.

KvadratickáPropagace($\boldsymbol{Q}$)
  for $i = 1$ **to** n **do begin**
    Spočti $[\underline{e}, \overline{e}]$ podle vzorce (3.1)
    for $j = 1$ **to** n **do begin**
      Spočti $[\underline{c}, \overline{c}]$ dle vzorců (3.3) a (3.4)
      if $\underline{x}_{ij} > -\sqrt{\underline{c}}$
        $\boldsymbol{Q}_{ij} \leftarrow \boldsymbol{Q}_{ij} \cap \langle\sqrt{\underline{c}}, \sqrt{\overline{c}}\rangle$
      else if $\overline{x}_{ij} < \sqrt{\underline{c}}$
        $\boldsymbol{Q}_{ij} \leftarrow \boldsymbol{Q}_{ij} \cap \langle-\sqrt{\overline{c}}, -\sqrt{\underline{c}}\rangle$
      else
        $\boldsymbol{Q}_{ij} \leftarrow \boldsymbol{Q}_{ij} \cap \langle-\sqrt{\overline{c}}, \sqrt{\overline{c}}\rangle$
    end
  end

Pro sloupce použijeme tentýž algoritmus, který pustíme na transponovanou matici $\boldsymbol{Q}$.

4. Propagace bilineárních podmínek

Pro propagaci bilineárních podmínek jsme vyzkoušeli tři různé metody.

4.0.2 Kontrakce pomocí intervalové aritmetiky

Nejjednodušší metoda propagace bilineárních podmínek spočítá ve vyjádření kontrahované proměnné, dosazení za ostatní proměnné a vypočítaní nového intervalu pomocí intervalové aritmetiky.

Pro další výklad si opět zafixujeme indexy i a j.

Vyjádříme proměnou x_{ij} ze všech rovnic (2.4), kde se vyskytuje.

$$x_{ij} = -\frac{1}{x_{kj}} \sum_{l=1, l \neq j}^{n} x_{il} x_{kl} \quad \text{pro všechny } k = 1 \dots n, k \neq i \tag{4.1}$$

$$x_{ij} = -\frac{1}{x_{ik}} \sum_{l=1}^{n} x_{lj} x_{lk} \quad \text{pro všechny } k = 1 \dots n, k \neq j \tag{4.2}$$

Pokud $0 \notin \boldsymbol{Q}_{kj}$ v prvním případě nebo $0 \notin \boldsymbol{Q}_{ik}$ v druhém případě, dosadíme do rovnic (4.1) a (4.2) za všechny proměnné na pravé straně příslušné intervaly z matice $\boldsymbol{Q}$ a intervalovou aritmetikou vypočítáme interval $\boldsymbol{x}'_{ij}$. Nový interval $\boldsymbol{Q}_{ij}$ dostaneme jako průnik původního intervalu s nově vypočteným intervalem $\boldsymbol{x}'_{ij}$.

Tento výpočet opakujeme pro všechny rovnice a všechny proměnné.

Abychom ušetřili výpočetní čas, předpočítáme si interval $[\underline{e}, \overline{e}]$, podobně jako jsme to udělali u kvadratických podmínek.

$$[\underline{e}, \overline{e}] = \sum_{l=1}^{n} \boldsymbol{x}_{il} \boldsymbol{x}_{kl} \tag{4.3}$$

Pro výpočet intervalů $\boldsymbol{x}'_{ij}$ a $\boldsymbol{x}_{kj}$ si předpočítáme

$$\boldsymbol{u}_j = -[\underline{e} - \underline{(\boldsymbol{x}_{ij} \boldsymbol{x}_{kj})}, \overline{e} - \overline{(\boldsymbol{x}_{ij} \boldsymbol{x}_{kj})}] \tag{4.4}$$

a pak jej použijeme pro výpočet

$$\boldsymbol{x}'_{ij} = \frac{\boldsymbol{u}_j}{\boldsymbol{x}_{kj}} \qquad \boldsymbol{x}'_{kj} = \frac{\boldsymbol{u}_j}{\boldsymbol{x}_{ij}} \tag{4.5}$$

Pro vzorec 4.2 bude výpočet analogický.

Preudokód

Nyní zapíšeme algoritmus v pseudokódu.

BilinPropagaceIntAritm()
  for $i = 1$ **to** n **do**
    for $k = i + 1$ **to** n **do**

Spočítej $\boldsymbol{e}$ dle vzorce (4.3)
for $j = 1$ **to** n **do**
  Spočítej $\boldsymbol{u}_j$ dle vzorce (4.4)
  if $(0 \notin \boldsymbol{Q}_{kj}\ )$
    $\boldsymbol{x}'_{ij} \leftarrow \frac{\boldsymbol{u}_j}{\boldsymbol{x}_{kj}}$ (vzorec (4.5))
    $\boldsymbol{Q}_{ij} \leftarrow \boldsymbol{Q}_{ij} \cap \boldsymbol{x}'_{ij}$
  if $(0 \notin \boldsymbol{Q}_{ij})$
    $\boldsymbol{x}'_{kj} \leftarrow \frac{\boldsymbol{u}_j}{\boldsymbol{x}_{ij}}$ (vzorec (4.5))
    $\boldsymbol{Q}_{kj} \leftarrow \boldsymbol{Q}_{kj} \cap \boldsymbol{x}'_{kj}$
end

Kontrakci pro rovnice, které vznikly jako skalární součiny sloupců, opět provedeme stejný algoritmem, který bude mít na vstupu transpozici $\boldsymbol{Q}^T$.

Při propagaci kvadratických podmínek můžeme také použít intervalovou aritmetiku podobně jako u bilineárních podmínek. Kontraktor z minulé kapitoly dělá v případě (iii) to stejné jako by udělala intervalová aritmetika. V ostatních případech je kontrahuje více, protože intervalová aritmetika rozšířením na intervaly při matematických operacích přidává do výsledku nepotřebné hodnoty navíc.

4.0.3 Lineární programování

Úloha Lineárního programování

Lineární programování je optimalizační technika, která pracuje se sadou proměnných a lineárních podmínek definovaných na proměnných. Jejím cílem je maximalizovat hodnotící funkci, která je také lineární vůči všem proměnným.

Při lineárním programování používáme schematického zápisu pomocí matic a vektorů. Všechny proměnné naskládáme do jednoho vektoru a označíme jej x. Podmínky zapíšeme do matice. Z každé podmínky $\sum_{i=1}^{n} a_i x_i \leq c$, kde a_i jsou reálná čísla, vznikne jeden řádek matice $(a_1\ a_2\ \ldots\ a_n)$, která se značí A. Z konstantních členů v podmínkách vytvoříme vektor b. Lineární podmínky pak můžeme zapsat v maticovém tvaru $Ax \leq b$.

Pokud bude podmínka jiného tvaru, například s opačnou nerovností nebo rovností, převedeme ji prvně do vhodného tvaru. Otočení nerovnosti provedeme násobením -1 a rovnost nahradíme dvěma podmínkami s opačnými nerovnostmi (z nichž jednu musíme opět násobit -1).

Definice 11 (Úloha lineárního programování)**.** *Úlohou lineárního programování je minimalizovat funkci $c^T x$, kde $c \in \mathbb{R}^n$ a x je vektor proměnných za podmínek $A \cdot x \leq b$, $A \in \mathbb{R}^{m \times n}$ a $b \in \mathbb{R}^m$.*

Pro řešení úloh lineárního programování je známo několik algoritmů. Nejjednodušší je simplexový algoritmus popsaný například zde [13]. Jako další metody řešení můžeme jmenovat metodu vnitřního bodu a elipsoidovou metodu, které jsou asymptoticky lepší než simplexová metoda. Více o nich najdeme třeba zde [14].

Linearizace podmínek

V našem problému máme bilineární podmínky, takže nemůžeme použít lineární programování přímo. Podmínky je nutné prvně linearizovat. K tomu použijeme následující vztahy uvedené v [15].

Pro všechny $x \in \boldsymbol{x}$ a $y \in \boldsymbol{y}$ platí

$$xy \geq \underline{x}y + \underline{y}x - \underline{x}\underline{y} \tag{4.6}$$
$$xy \geq \overline{x}y + \overline{y}x - \overline{x}\overline{y} \tag{4.7}$$

$$xy \leq \overline{x}y + \underline{y}x - \overline{x}\underline{y} \tag{4.8}$$
$$xy \leq \underline{x}y + \overline{y}x - \underline{x}\overline{y} \tag{4.9}$$

Jedna z možností jak provézt linearizaci podmínek (2.4) je odhadnout každý term $x_{ij}x_{kj}$ jedním ze vztahů (4.6) - (4.9).

Pokud použijeme k odhadu podmínek (2.4), varianty pro řádky, vztah (4.6) dostaneme následující nerovnost

$$0 = \sum_{j=1}^{n} x_{ij}x_{kj} \geq \sum_{j=1}^{n}(\underline{x}_{ij}x_{kj} + x_{ij}\underline{x}_{kj} - \underline{x}_{ij}\underline{x}_{kj})$$

kterou upravíme do podmínky ve vhodném tvaru pro lineární programování

$$\sum_{j=1}^{n}(\underline{x}_{ij}x_{kj} + x_{ij}\underline{x}_{kj}) \leq -\sum_{j=1}^{n}\underline{x}_{ij}\underline{x}_{kj} \quad \text{pro všechny } i \neq k$$

K odhadu jednotlivých termů $x_{ij}x_{kj}$ v jedné podmínce můžeme samozřejmě použít i vztah (4.7). Díky tomu můžu vytvořit exponenciálně mnoho podmínek vůči počtu bilineárních termů v podmínce, což by už nebylo příliš efektivní. Proto zvolíme jiný postup, abychom odstranili problém s exponenciálním počtem podmínek, ale zároveň nemuseli žádnou podmínku vynechat. Zavedeme nové proměnné $x_{ij,kj}$ pro podmínky pro řádky a $x_{ij,ik}$ pro sloupce, kterými nahradíme bilineární termy $x_{ij}x_{kl}$ v podmínkách (2.4). Tím dostaneme nové podmínky

$$\sum_{j=1}^{n} x_{ij,kj} = 0 \quad \text{pro všechny } i \neq k \qquad \sum_{i=1}^{n} x_{ij,ik} = 0 \quad \text{pro všechny } i \neq k \tag{4.10}$$

Další podmínky vytvoříme ze vztahu (4.6) - (4.9). Bilineární term opět nahradíme novou proměnou.

$$x_{ij,kj} \geq \underline{x}_{ij}x_{kj} + \underline{x}_{kj}x_{ij} - \underline{x}_{ij}\underline{x}_{kj} \tag{4.11}$$

Podobně získáme ze vztahů (4.7) - (4.9) pro každý bilineární člen (novou proměnou $x_{ij,kj}$ nebo $x_{ij,ik}$) čtyři podmínky. Dvě jej budou odhadovat shor a a dvě zespodu.

Uvedli jsme variantu pro řádky. Varianta pro sloupce by vypadala podobně.

Naše úloha lineárního programování

Proměnné budou členy q_{ij} matice $\boldsymbol{Q}$. Další proměnné pro odhady bilineárních členu budou x_{ijk} pro odhad bilineárních členů v podmínkách vzniklých násobením řádků a y_{jik} pro násobení sloupců.

$$\begin{array}{ll} q_{ij} & i,j = 1\dots n \\ x_{ijk} & i,j = 1\dots n, k = i+1\dots n \\ y_{jik} & i,j = 1\dots n, k = j+1\dots n \end{array}$$

Proměnné q_{ij} jsou z matice Q, takže jich je n^2. Počet proměnných x_{ijk} a y_{jik} je stejný a odpovídá počtu podmínek (2.4) krát rozměr matice Q, což je $(n^2-n)n$. Celkový počet všech proměnných naší lineární úlohy je n^3.

Podmínky vzniklé nahrazením bilineárních členu novou proměnou v rovnicích (2.4)

$$\sum_{j=1}^{n} x_{ijk} = 0 \qquad \begin{array}{rcl} i & = & 1\dots n \\ k & = & i+1\dots n \end{array}$$

$$\sum_{i=1}^{n} y_{jik} = 0 \qquad \begin{array}{rcl} j & = & 1\dots n \\ k & = & j+1\dots n \end{array}$$

a podmínky, které dostaneme z nerovnosti (4.6) - (4.9).

$$\begin{array}{ll} x_{ijk} \geq \underline{x}_{ij} x_{kj} + \underline{x}_{kj} x_{ij} - \underline{x}_{ij}\underline{x}_{kj} & \\ x_{ijk} \geq \overline{x}_{ij} x_{kj} + \overline{x}_{kj} x_{ij} - \overline{x}_{ij}\overline{x}_{kj} & j = 1\dots n \\ x_{ijk} \leq \overline{x}_{ij} x_{kj} + \underline{x}_{kj} x_{ij} - \overline{x}_{ij}\underline{x}_{kj} & k = j+1\dots n \\ x_{ijk} \leq \underline{x}_{ij} x_{kj} + \overline{x}_{kj} x_{ij} - \underline{x}_{ij}\overline{x}_{kj} & \end{array}$$

Tyto nerovnice jsou pouze pro bilineární členy, které se vyskytují v bilineárních podmínkách vzniklých násobením řádků. Bilineární členy z podmínky vytvořené násobením sloupců dostaneme nahrazením písmene x za y a prohozením indexů na pravých stranách nerovnic.

Bilineárních podmínek 2.4 je celkem $(n^2 - n)n$. Pro každou novou proměnou x_{ijk} nebo y_{jik} jsme přidali čtyři podmínky, takže celkem $4n^3$ nerovnic.

Účelová funkce bude $+q_{ij}$ pokud budeme počítat dolní hranici $\boldsymbol{Q}_{ij}$ a $-q_{ij}$ pokud budeme počítat horní hranici $\boldsymbol{Q}_{ij}$.

Kvůli zadání hodnotící funkce budeme pro obě hranice každého intervalu $\boldsymbol{Q}_{ij}$ pouštět lineární program zvlášť.

Pseudokód

LINEÁRNÍPROGRAMOVÁNÍ()
  Urči matici podmínek A a vektor b
  for $i,j = 1$ **to** n **do**
    $x = 0\dots0\, x_{ij}\, 0\dots0$
    LinProg(x, A, b)
    Aktualizuj A, b
end

5. Propagace podle součtu podmínek

V článku [3] je popsaná propagace kvadratických podmínek, kterou nejprve vysvětlíme a pak ji aplikujeme na náš problém.

5.0.4 Vysvětlení kvadratické propagace

Jednoduchá kvadratická podmínka

Předpokládejme, že máme jednoduchou kvadratickou podmínku

$$ax^2 + 2bx \geq c \tag{5.1}$$

a chceme najít množinu X takovou, že v X jsou všechna kladná reálná čísla, která splňují tuto podmínku.

$$X = \{x \geq 0 \mid ax^2 + 2bx \geq c\}$$

V nerovnici (5.1) jsme použili před koeficientem b dvojku, což se zdá na první pohled zbytečné, ale později se nám to bude hodit.

Množinu X určíme řešením nerovnice $ax^2 + 2bx - c \geq 0$.

$$x^2 - 2bx - c = 0 \qquad D = 4b^2 + 4ac$$

$$t_{1,2} = \frac{-2b \pm \sqrt{4(b^2+ac)}}{2a} = \frac{-b \pm \sqrt{b^2+ac}}{a}$$

Umístění t_1,t_2 na reálné ose závisí na velikostech a,b,c a jejich znaménkách. Obecně bychom to mohli zapsat takto.

$$X = \begin{cases} \langle 0,\infty) \setminus [t_1, t_2] & \text{pokud } a > 0 \\ \langle 0,\infty) \cap [t_2, t_1]\rangle & \text{pokud } a < 0 \end{cases} \tag{5.2}$$

Porovnáním jednotlivých proměnných a,b,c můžeme určit hranice X přesněji. Zavedeme následující proměnné

$$\Delta = b^2 + ac \qquad z = |b| + \sqrt{\Delta}$$

pak t_1 a t_2 zapíšeme jako

$$t_1 = \frac{-b - \sqrt{\Delta}}{a} = \frac{c}{b - \sqrt{\Delta}} \qquad t_2 = \frac{-b + \sqrt{\Delta}}{a} = \frac{c}{b + \sqrt{\Delta}}$$

Pokud je $D > 0$, může rozlišit osm možných případů, jak bude výsledná X vypadat v závislosti na proměnných a,b,c.

$$X = \begin{cases} \emptyset & a < 0, b \leq 0, c > 0 \\ \left[\frac{z}{a}, \infty\right) & a > 0, b \leq 0, c > 0 \\ \left[0, -\frac{c}{z}\right] & a < 0, b \leq 0, c \leq 0 \\ \left[0, -\frac{c}{z}\right] \cup \left[\frac{z}{a}, \infty\right) & a > 0, b \leq 0, c \leq 0 \\ \left[-\frac{-c}{z}, \frac{z}{-a}\right] & a < 0, b \geq 0, c > 0 \\ \left[-\frac{-c}{z}, \infty\right) & a > 0, b \geq 0, c > 0 \\ \left[0, \frac{z}{-a}\right] & a < 0, b \geq 0, c \leq 0 \\ [0, \infty] & a > 0, b \geq 0, c \leq 0 \end{cases} \tag{5.3}$$

Pokud je $D < 0$, vypadá množina X takto

$$X = \begin{cases} [0, \infty) & c < 0 \\ \emptyset & c \geq 0 \end{cases}$$

.

Při řešení jsme předpokládali, že $a \neq 0$. V opačném případě by rovnice nebyla kvadratická ale lineární a množina X by vypadala takto

$$X = \begin{cases} \emptyset & b \leq 0, c > 0 \\ \left[0, \frac{c}{2b}\right] & b > 0, c \leq 0 \\ \left[\frac{c}{2b}, \infty\right) & b > 0, c \geq 0 \\ [0, \infty) & b \geq 0, c \leq 0 \end{cases}$$

Příklad 1.

$$x^2 - 4x \geq -5$$

Nerovnost upravíme na $x^2 - 4x + 5 \geq 0$*, pak* $a = 1$*,* $b = -2$*,* $c = -5$ *takže množina* X *je tvaru*

$$X = \left[0, -\frac{c}{z}\right) \cup \left[\frac{z}{a}, \infty\right) = [0, 1) \cup (5, \infty)$$

Popsali jsme postup zpětné propagace podmínky. V článku [3] je popsána také dopředná propagace, kterou nebudeme na náš problém aplikovat, proto ji zde neuvádíme.

Složená kvadratická podmínka

Myšlenku propagace ukázanou výše lze použít i v případě, že máme jednu podmínku s větším množstvím separovaných kvadratických proměnných. V článku [3] je popsána i propagace neseparovaných podmínek. My ji však nebudeme potřebovat.

Označme

$$p_k(x_k) = a_k x_k^2 + b_k x_k \tag{5.4}$$

kde x_k je proměnná a a_k, b_k jsou libovolná reálná čísla. Podmínku se separovanými proměnnými pak můžeme zapsat jako

$$\sum_{k=1}^{n} p_k(x_k) \geq c$$

Hodnoty proměnné x_k jsou z nějakého intervalu $\boldsymbol{x}_k$, proto můžeme použít pro p_k také intervalového vyjádření. Potom $\boldsymbol{p}_k$ odpovídá intervalu vypočtenému intervalovou aritmetikou dle definičního vztahu (5.4).

Platí následující vztah

$$\sum_{k=i,k\neq i}^{n} \overline{p}_k + \boldsymbol{p}_i(\boldsymbol{x}_i) \geq \sum_{k=1,k\neq i}^{n} \boldsymbol{p}_k(\boldsymbol{x}_k) + \boldsymbol{p}_i(\boldsymbol{x}_i) \geq c$$

Když vyjádříme $\boldsymbol{p}_i(\boldsymbol{x}_i)$, dostaneme

$$\boldsymbol{p}_i(\boldsymbol{x}_i) \geq c - \sum_{k=i,k\neq i}^{n} \overline{p}_k$$

Nyní jsme získali podmínku v požadovaném tvaru nerovnice (5.1). Pro $x_i \in \boldsymbol{x}_i$, $x_i \geq 0$ můžeme použít postup popsaný výše. Pokud má interval $\boldsymbol{x}_i$ zápornou část nebo je celý záporný musíme podmínku trochu upravit. Do vztahu $p_i(x_i) = a_i x_i^2 + b_i x_i \geq d_i$ dosadíme $-x_i$ za x_i a dostaneme

$$p_i(x_i) = a_i x_i^2 - b_i x_i \geq d_i$$

Pak platí, že množina opačných prvků k množině

$$\{x_i \geq 0 \mid a_i x_i^2 - b_i x_i \geq d_i\} \tag{5.5}$$

je rovna množině

$$\{x_i \leq 0 \mid a_i x_i^2 + b_i x_i \geq d_i\} \tag{5.6}$$

kde $d_i = c - \sum_{k=i,k\neq i}^{n} \overline{p}_k$. Stačí hledat množinu (5.5) a pak její hodnoty vynásobit -1, čímž dostaneme hledanou množinu (5.6). Tím jsme převedli problém se záporným intervalem $\boldsymbol{x}_i$ na předchozí případ.

Pokud 0 je uvnitř intervalu $\boldsymbol{x}_i$ (a není krajním bodem), pak interval $\boldsymbol{x}_i$ rozdělíme na dva intervaly $[\underline{x}_i, 0]$ a $[0, \overline{x}_i]$ a každý řešíme zvlášť. Výsledkem je sjednocení výsledných intervalů z obou výpočtů.

Intervalové koeficienty

Koeficienty kvadratické podmínky (5.1) mohou být čísla z nějakých intervalů.

$$X = \{x \geq 0 \mid \boldsymbol{a}x^2 + \boldsymbol{b}x \subseteq \boldsymbol{c}\}$$

Množinu X nahradíme průnikem dvou množin, které jsou zapsané nerovnostmi s dolními a horními mezemi intervalů v původní nerovnici.

$$X' = \{x \geq 0 \mid \underline{a}x^2 + \underline{b}x \leq \overline{c}\} \cap \{x \geq 0 \mid \overline{a}x^2 + \overline{b}x \geq \underline{c}\} \tag{5.7}$$

Pro $\forall x \in X$, existuje $a \in \boldsymbol{a}$, $b \in \boldsymbol{b}$ a $c \in \boldsymbol{c}$ tak, že platí

$$\underline{a}x^2 + \underline{b}x \leq ax^2 + bx \leq c \leq \overline{c}$$

podobně pro druhou množinu

$$\overline{a}x^2 + \overline{b}x \geq ax^2 + bx \geq c \geq \underline{c}$$

Proto pro každé $x \in X$ platí, že $x \in X'$ a naopak, takže $X = X'$.

Na množinu určenou nerovností $\overline{a}x^2 + \overline{b}x \geq \underline{c}$ z průniku (5.7) můžeme aplikovat postup popsaný výše. Nerovnost určující první množinu vynásobíme -1, čímž dostaneme požadovaný tvar odpovídající (5.1).

$$\{x \geq 0 \mid -\underline{a}x^2 - \underline{b}x \geq -\overline{c}\}$$

Nyní máme problém ve takovém tvaru, že můžeme použít popsanou kontrakci.

5.0.5 Aplikace na náš problém

Máme kvadratické podmínky pouze s jedním kvadratickým členem a máme bilineární podmínky. Protože dosud navržené kontrakce bilineárních podmínek fungovaly hůře než jednoduchá kontrakce podle kvadratických podmínek, mohla by kombinace obou podmínek pomocí součtu fungovat lépe.

Vezmeme kvadratickou podmínku pro i-tý řádek

$$\sum_{j=1}^{n} x_{ij}^2 = 1$$

a bilineární podmínku pro skalární součin i-tého a k-tého řádku

$$\sum_{j=1}^{n} x_{ij} x_{kj} = 0$$

a sečteme je. Tím dostaneme kvadratickou podmínku s nenulovým lineárním členem.

$$\sum_{j=1}^{n} (x_{ij}^2 + x_{ij} x_{kj}) = 1 \tag{5.8}$$

Proměnné x_{kj} budeme považovat při propagaci podmínky (5.8) za konstanty. Díky tomu získáme separovanou kvadratickou podmínku.

Budeme se zabývat pouze zpětnou propagací. Dopředná propagace nemá smysl, protože na pravé straně podmínky je číslo 1.

Z podmínky (5.8) vyjádříme

$$x_{ij}^2 + x_{ij} x_{kj} = 1 - \sum_{l=1, l \neq j}^{n} (x_{il}^2 + x_{il} x_{kl})$$

Zavedeme substituce, abychom zjednodušili zápis

$$x^2 + \boldsymbol{b} x \subseteq \boldsymbol{c} \qquad \text{kde} \quad \boldsymbol{b} = \boldsymbol{x}_{kj} \quad \text{a} \quad \boldsymbol{c} = 1 - \sum_{l=1, l \neq j}^{n} (\boldsymbol{x}_{il}^2 + \boldsymbol{x}_{il} \boldsymbol{x}_{kl}) \tag{5.9}$$

Nyní chceme najít množinu

$$X = \{x \in \boldsymbol{x} \mid x^2 + \boldsymbol{b} x \subseteq \boldsymbol{c}\}$$

Mohou nastat tyto tři situace:

- Pokud je $\underline{x} \geq 0$ hledáme $Y = \{x \geq 0 \mid x^2 + \boldsymbol{b} x \subseteq \boldsymbol{c}\}$
- Pokud je interval $\boldsymbol{x}$ celý v záporné části reálné osy, dosadíme $-x$ do předpisu množiny X, čímž dostaneme předpis nové množiny

  $$Z = \{x \geq 0 \mid x^2 - \boldsymbol{b} x \subseteq \boldsymbol{c}\}$$

  Když čísla z množiny Z vynásobíme -1, dostaneme hledanou množinu X. Tím jsme problém převedli na předchozí případ.

- Jestliže $0 \in \boldsymbol{x}$, rozdělíme interval $\boldsymbol{x}$ na dva podintervaly $[\underline{x}, 0]$ a $[0, \overline{x}]$.

Interval $\boldsymbol{x}$ rozdělíme na kladný a záporný interval a hledáme množiny

$$Y = \{x \geq 0 \mid x^2 + \boldsymbol{b}x \subseteq \boldsymbol{c}\}$$

$$Z = \{x \geq 0 \mid x^2 - \boldsymbol{b}x \subseteq \boldsymbol{c}\}$$

Množina Y je řešení pro kladnou část intervalu $\boldsymbol{x}$ a množina Z je řešení pro zápornou část intervalu $\boldsymbol{x}$ a odpovídá množině (5.6).

Nyní ještě převedeme množiny Y a Z na průnik dvou množin určených nerovnicí s reálnými koeficienty, viz vztah (5.7). Označíme

$$Y_1 = \{x \geq 0 \mid x^2 + \overline{b}x \geq \underline{c}\} \qquad Y_2 = \{x \geq 0 \mid -x^2 - \underline{b}x \geq -\overline{c}\}$$

$$Z_1 = \{x \geq 0 \mid x^2 - \underline{b}x \geq \underline{c}\} \qquad Z_2 = \{x \geq 0 \mid -x^2 + \overline{b}x \geq -\overline{c}\}$$

pak

$$Y = Y_1 \cap Y_2 \qquad Z = Z_1 \cap Z_2$$

a výsledná množina X je

$$X = Y \cup Z$$

Tvar množin X_1, X_2, Y_1, Y_2 je trochu specifický. Množiny dosadíme do obecného řešení problému.

1. $Y_1 = \{x \geq 0 \mid x^2 + \overline{b}x \geq \underline{c}\}$, pak je $a = 1$, $b = \frac{\overline{b}}{2}$, $c = \underline{c}$ a řešení je

$$Y_1 = [0, \infty) - [t_1, t_2]$$

kde

$$\Delta = b^2 + c$$

$$t_1 = -b - \sqrt{\Delta}$$

$$t_2 = -b + \sqrt{\Delta}$$

Dosadíme do univerzálního řešení.

$$Y_1 = \begin{cases} \left[-b + \sqrt{\Delta}, \infty\right) & c > 0 \\ \left[0, -b - \sqrt{\Delta}\right) \cup \left(-b + \sqrt{\Delta}, \infty\right) & c \leq 0, b \leq 0 \\ [0, \infty) & c \leq 0, b \geq 0 \end{cases}$$

2. $Y_2 = \{x \geq 0 \mid -x^2 - \underline{b}x \geq -\overline{c}\}$, pak je $a = -1$, $b = \frac{-\underline{b}}{2}$, $c = -\overline{c}$ a řešení je množina $Y_2 = [t_2, t_1] \cap [0, \infty)$.

$$t_1 = b + \sqrt{\Delta}$$

$$t_2 = b - \sqrt{\Delta}$$

Opět dosadíme do univerzálního řešení

$$Y_2 = \begin{cases} \left[0, b + \sqrt{\Delta}\right] & c \leq 0 \\ \emptyset & c > 0, b \leq 0 \\ \left[b - \sqrt{\Delta}, b + \sqrt{\Delta}\right] & c > 0, b \geq 0 \end{cases}$$

3. $Z_1 = \{x \geq 0 \mid x^2 - \underline{b}x \geq \underline{c}\}$, pak je $a = 1$, $b = \frac{-\underline{b}}{2}$, $c = \underline{c}$ a řešení je $Z_1 = [0, \infty) - [t_1, t_2]$. Univerzální řešení pro Z_1 vypadá stejně jako pro Y_1.

4. $Z_2 = \{x \geq 0 \mid -x^2 + \overline{b}x \geq -\overline{c}\}$, pak je $a = -1$, $b = \frac{\overline{b}}{2}$, $c = -\overline{c}$ a řešení je $Z_2 = [t_2, t_1] \cap [0, \infty)$. Dosazení do univerzálního řešení je stejné jako u množiny Y_2.

Zaokrouhlovat budeme t_1 nahoru a t_2 dolů, abychom neztratili žádné řešení. Z toho plyne, že Δ a $\sqrt{\Delta}$ zaokrouhlíme vždy dolů, protože v t_1 je $\sqrt{\Delta}$ s mínusem a chceme, aby t_1 bylo zaokrouhleno nahoru. Podobně t_2 chceme zaokrouhlit dolů a $\sqrt{\Delta}$ přičítáme.

Preudokód

SpojeníPodmínek()
  for $i = 1$ **to** n **do**
    for $k = i + 1$ **to** n **do**
      for $j = 1$ **to** n **do**
        $\boldsymbol{b} = \boldsymbol{x}_{kj}$
        $\boldsymbol{c} = 1 - \sum_{l=1, l \neq j}^{n} (\boldsymbol{x}_{il}^2 + \boldsymbol{x}_{il}\boldsymbol{x}_{kl})$
        SpočtiY$_1$($\boldsymbol{b}$,$\boldsymbol{c}$)
        SpočtiY$_2$($\boldsymbol{b}$,$\boldsymbol{c}$)
        SpočtiZ$_1$($\boldsymbol{b}$,$\boldsymbol{c}$)
        SpočtiZ$_2$($\boldsymbol{b}$,$\boldsymbol{c}$)
        $X = (Y_1 \cap Y_2) \cup (Z_1 \cap Z_2)$
        $\boldsymbol{x}_{ij} = \boldsymbol{x}_{ij} \cap X$
  end

SpočtiY$_1$($\boldsymbol{b}$,$\boldsymbol{c}$)
  $b = \frac{\overline{b}}{2}$
  $c = \underline{c}$
  if $c > 0$
    $Y_1 = \left[-b + \sqrt{\Delta}, \infty\right)$
  else
    if $b \leq 0$
      $Y_1 = \left[0, -b - \sqrt{\Delta}\right] \cup \left[-b + \sqrt{\Delta}, \infty\right)$
    else
      $Y_1 = [0, \infty)$

Metody pro Y_2, Z_1 a Z_2 budou vypadat velice podobně.

6. Kontrakce podle determinantu

6.0.6 Teoretická příprava

Definice 12. *Nechť $A \in \mathbb{R}^{n\times n}$ je čtvercová matice nad tělesem $\mathbb{R}$, pak charakteristickou maticí nazveme matici*

$$\lambda I - A = \begin{pmatrix} \lambda - a_{11} & -a_{12} & \dots & -a_{1n} \\ -a_{21} & \lambda - a_{22} & \dots & -a_{2n} \\ \dots & \dots & \dots & \dots \\ -a_{n1} & -a_{n2} & \dots & \lambda - a_{nn} \end{pmatrix}$$

Definice 13. *Charakteristický polynom A nazveme determinant charakteristické matice, tj.:* $\det(\lambda I - A)$.

Definice 14. *Kořeny charakteristického polynomu matice A nazveme vlastní čísla. Násobností vlastního čísla budeme chápat násobnost kořene v charakteristickém polynomu.*

Spektrum matice A nazveme soubor všech vlastních čísel. Každé vlastní číslo se v souboru vyskytuje tolikrát kolik je jeho násobnost.

Spektrální radius $\rho(A)$ je absolutní hodnota největšího vlastního čísla.

Tvrzení 1. *Ortogonální matice má vlastní čísla na jednotkové kružnici v komplexní rovině.*

Proof. Důkaz v [12]. □

Důsledek 1. *Ortogonální matice může mít pouze dvě reálná vlastní čísla a to jsou 1 a -1.*

Tvrzení 2. *Q je ortogonální matice. Je-li* $\det(Q) = -1$*, tak -1 je vlastním číslem liché násobnosti.*

Proof. Protože matice Q je reálná, má její charakteristický polynom pouze reálné koeficienty. Vlastní čísla jsou kořeny charakteristického polynomu. Pokud jsou komplexní, musí být komplexně sdružené.

Nechť $x = a + bi$ je vlastní číslo matice Q a $x' = a - bi$ je vlastní číslo k němu komplexně sdružené. Pak platí $x \cdot x' = (a + bi)(a - bi) = a^2 + b^2 \geq 0$. Součin dvou komplexně sdružených čísel je vždy nezáporný.

Determinant matice je součin jejích vlastních čísel. Protože je determinant záporný musí existovat alespoň jedno záporné vlastní číslo s lichou násobností. Jediné možné záporné vlastní číslo je -1. □

Tvrzení 3. *λ je vlastní číslo matice A právě tehdy když je* $\det(A - \lambda I) = 0$

Proof. Důkaz najdeme v [6], str. 86, věta 10.3. □

Tvrzení 4. *Nechť Q je ortogonální matice,* $\det(Q) = -1$ *a I je jednotková matice. Pak $Q + I$ je singulární.*

Proof. Z tvrzení 3 plyne, že $\det(Q - (-1)I) = \det(Q + I) = 0$.
Matice s nulovým determinantem je singulární (Důkaz najdeme v [5], str.: 168, 14.7.(xi).) Pak je matice $Q + I$ je singulární. □

6.0.7 Postup kontrakce

Pokud je matice Q regulární, pak je matice $Q + I$ singulární. Díky tomu můžeme kontrakci intervalové matice $\boldsymbol{Q}$ převézt na kontrakci singulární matice $\boldsymbol{Q} + I$. Z matice $\boldsymbol{Q} + I$ budeme odstraňovat regulární matice. Přitom nesmíme odstranit žádnou singulární, protože tato matice by odečtením I mohla dát ortogonální matici.

Po kontrakci převedeme matici $\boldsymbol{Q} + I$ zpět na $\boldsymbol{Q}$ a dostaneme kontrahovanou ortogonální matici $\boldsymbol{Q}$.

Matici $\boldsymbol{Q} + I$ označím jako $\boldsymbol{R}$, tj.: $\boldsymbol{R} := \boldsymbol{Q} + I$ a

$$R_c = Q_c + I$$

$$R_\Delta = Q_\Delta$$

Matici $\boldsymbol{R}$ budeme procházet po jednotlivých členech a kontrahovat intervaly.

Pro další výklad budeme pracovat s pevnými indexy i, j, které fixují člen v matici.

Definice 15. *Pro reálnou matici $M \in \mathbb{R}^{n \times n}$ a reálné číslo r označíme $M(r)$ matici s prvky*

$$\begin{array}{ll} M(r)_{kl} = r & \textit{jestliže } i = k \textit{ a } j = l \\ M(r)_{kl} = m_{kl} & \textit{jinak} \end{array}$$

Analogické značení použijeme pro intervalové matice.

Interval $[\underline{r}_{ij}, \overline{r}_{ij}]$ na pozici i, j v matici $\boldsymbol{R}$ nahradíme zkráceným intervalem $[\underline{r}_{ij}, \overline{r}_{ij} - \delta]$, tak aby všechny odřezané matice $\boldsymbol{R}([\overline{r}_{ij} - \delta, \overline{r}_{ij}])$ byly regulární. Naším úkolem je nalézt co největší δ, tak aby matice $\boldsymbol{R}([\overline{r}_{ij} - \delta, \overline{r}_{ij}])$ byla regulární.

6.0.8 Podmínka pro kontrakci

Definice 16. *Intervalová matice $\boldsymbol{A}$ je regulární pokud $\forall A \in \boldsymbol{A}$ platí, že A je regulární nad $\mathbb{R}$.*

Tvrzení 5. *Intervalová matice $\boldsymbol{B}$ je regulární, jestliže pro nějakou matici $M \in \mathbb{R}^{n \times n}$ platí:*

$$\rho(|I - MB_c| + |M|B_\Delta) < 1 \tag{6.1}$$

Proof. Důkaz v [7]. □

Následující věta dává podmínku pro δ.

Věta 1. *Pro libovolnou nenulovou matici M a matici K, která má na fixní pozici i, j jedničku a jinde nuly, platí:*

$$\delta < \frac{2 - \rho(|I - MR_c(\overline{r}_{ij})| + |I - MR_c(\overline{r}_{ij})|^T + |M|R_\Delta(0) + (|M|R_\Delta(0))^T)}{\rho(|M|K + K^T|M|^T)} \tag{6.2}$$

Proof. Do vzorce (6.1) dosadíme za matici $\boldsymbol{B}$ intervalovou matici $\boldsymbol{R}'$, která představuje všechny odřezávané matice z intervalové matice $\boldsymbol{R}$.

$$\boldsymbol{R}' = \boldsymbol{R}([\overline{r}_{ij} - \delta, \overline{r}_{ij}])$$

$$R_c' = R_c(\overline{r}_{ij} - \frac{\delta}{2})$$

$$R_\Delta' = R_\Delta(\frac{\delta}{2})$$

Po dosazení dostaneme:

$$\rho(|I - MR_c(\overline{r}_{ij} - \frac{\delta}{2})| + |M|R_\Delta(\frac{\delta}{2})) < 1 \tag{6.3}$$

Přepíšeme vzorec (6.3) s využitím matice K, která je definovaná tak že:

$$\begin{array}{ll} K_{kl} = 1 & \text{jestliže } i = k \text{ a } j = l \\ K_{kl} = 0 & \text{jinak} \end{array}$$

$$\rho(|I - M(R_c(\overline{r}_{ij}) - \frac{\delta}{2}K)| + |M|(R_\Delta(0) + \frac{\delta}{2}K)) < 1$$

Roznásobíme.

$$\rho(|I - MR_c(\overline{r}_{ij}) + \frac{\delta}{2}MK| + |M|R_\Delta(0) + \frac{\delta}{2}|M|K) < 1$$

Použijeme tvrzení o nezáporných maticích a jejich spektru vlastních číslech.

Tvrzení 6. *Pro nezáporné reálné matice A,B platí*

$$0 \leq A \leq B \Rightarrow \rho(A) \leq \rho(B)$$

.

Proof. Důkaz v [8]. □

Díky tomuto tvrzení můžeme omezit výraz shora.

$$\begin{aligned} \rho(|I - MR_c(\overline{r}_{ij}) + \frac{\delta}{2}MK| + |M|R_\Delta(0) + \frac{\delta}{2}|M|K) \leq \\ \leq \rho(|I - MR_c(\overline{r}_{ij})| + \delta|M|K + |M|R_\Delta(0)) \end{aligned} \tag{6.4}$$

Tvrzení 7. *Pro reálnou matici $B \geq 0$ platí*

$$\rho(B) = \frac{1}{2}\rho(B + B^T)$$

Proof. Důkaz najdeme v [9]. □

Tvrzení 7 aplikujeme na výraz (6.4).

$$\begin{aligned} \frac{1}{2}\rho(|I - MR_c(\overline{r}_{ij})| + |I - MR_c(\overline{r}_{ij})|^T + |M|R_\Delta(0) + (|M|R_\Delta(0))^T + \\ + \delta(|M|K + K^T|M|^T)) \end{aligned} \tag{6.5}$$

Ve výrazu (6.5) máme součet šesti matic. Pokud sečteme matici s její transpozicí dostaneme symetrickou matici. Výraz si můžeme představit jako součet tří symetrických matic.

Pro spektrum symetrických matic platí Weylova věta, která říká:

Věta 2 (Weylova). *Pro A,B symetrické matice platí*

$$\rho(A+B) \leq \rho(A) + \rho(B)$$

Proof. Důkaz je v [10] nebo v [9]. □

Využitím této věty dostaneme následující.

$$\frac{1}{2}\rho(|I - MR_c(\overline{r}_{ij})| + |I - MR_c(\overline{r}_{ij})|^T + |M|R_\Delta(0) + (|M|R_\Delta(0))^T) +$$

$$+\frac{\delta}{2}\rho(|M|K + K^T|M|^T) \tag{6.6}$$

Pokud bude tento výraz menší než jedna, musí být matice $\boldsymbol{R}'$ nutně regulární, protože pro ni bude splněna nutná podmínka regularity (tvrzení 5). Výraz (6.6) položíme menší než jedna, vyjádříme δ a dostaneme podmínku (6.2).

$$\delta < \frac{2 - \rho(|I - MR_c(\overline{r}_{ij})| + |I - MR_c(\overline{r}_{ij})|^T + |M|R_\Delta(0) + (|M|R_\Delta(0))^T)}{\rho(|M|K + K^T|M|^T)}$$

Pokud bude tato podmínka splněna, tak $\boldsymbol{R}'$ bude regulární a můžeme ji odříznout.

Aby byl zlomek v podmínce (6.2) definovaný, musí být jeho jmenovatel nenulový.

Pokud vynásobím libovolnou matici M maticí K, bude j-tý sloupec výsledné matice tvořen i-tým sloupcem matice M. Ostatní členy budou nulové. Matice $K^T|M|^T = (|M|K)^T$ bude mít v j-tém řádku i-tý sloupec matice $|M|$.

Charakteristický polynom matice

$$|M|K + K^T|M|^T \tag{6.7}$$

bude vypadat takto:

$$(2|m_{ji}| - \lambda)\lambda^{(n-1)} + \sum_{k=1,k\neq j}^{n} \lambda^{n-2}|m_{ik}|^2$$

Charakteristický polynom upravíme

$$\lambda^{n-2}(-\lambda^2 + 2|m_{ij}|\lambda + \sum_{k=1,k\neq j}^{n} |m_{ik}|^2)$$

Matice (6.7) má vlastní číslo 0 s násobností $n-2$. Nyní rozložíme zbývající kvadratický polynom

$$(\lambda^2 - 2|m_{ij}|\lambda - \sum_{k=1,k\neq j}^{n} |m_{ik}|^2)$$

podle vzorce pro diskriminant.

$$D = 4|m_{ij}|^2 + 4\sum_{k=1,k\neq j}^{n} |m_{ik}|^2 = 4\sum_{k=1}^{n} |m_{ik}|^2$$

$$\lambda_{1,2} = \frac{2|m_{ij}| \pm \sqrt{D}}{2} = |m_{ij}| \pm \sqrt{\sum_{k=1}^{n} |m_{ik}|^2}$$

Pokud je ve sloupci i v matici M alespoň jedno nenulové číslo, má matice (6.7) alespoň jedno vlastní číslo větší než 0, takže ρ je taky větší než nula a jmenovatel zlomku (6.2) je nenulový.

Tím je důkaz věty dokončen.

□

6.0.9 Kontrakce intervalu zespodu

Všechny členy matice $\boldsymbol{R}$ vynásobíme -1. Interval $[\underline{r}_{ij}, \overline{r}_{ij}]$ se mění na interval $[-\overline{r}_{ij}, -\underline{r}_{ij}]$. Tento interval budeme kontrahovat stejným způsobem jako jsme kontrahovali původní interval shora. Získáme hledané δ, takové že $[-\overline{r}_{ij}, -\underline{r}_{ij} - \delta]$. Zpětným vynásobením všech členů matice -1 dostaneme kontrakci intervalu zespoda $[\underline{r}_{ij} - \delta, \overline{r}_{ij}]$.

6.0.10 Algoritmus kontrakce

Nyní vysvětlíme, jak bude algoritmus kontraktoru podle determinantu pracovat.

Vypočteme matici intervalovou $\boldsymbol{R}$ a budeme postupně procházet všechny její členy $\boldsymbol{R}_{ij}$. Pro každý člen spočteme velikost δ a zkrátíme interval $\boldsymbol{R}_{ij}$.

K výpočtu δ podle vzorce (6.2) potřebujeme vhodně zvolit matici M. Pokud vybereme M vhodně, vzorec (6.2) pro výpočet δ se může zjednodušit. Díky tomu potřebujeme k jeho výpočtu méně matematických operaci a dosáhneme větší přesnosti a rychlosti. Je pravděpodobné, že s volbou speciální matice M bude kontraktor fungovat lépe než s náhodně generovanou maticí.

Pseudokód kontraktoru

Pseudokód popisuje jednu smyčku kontratoru, tj.: kontrakce pro všechny členy matice $\boldsymbol{Q}$ proběhne jednou. Tento algoritmus budeme pouštět střídavě s propagací.

KontraktorPodleDeterminantu($\boldsymbol{Q}$)
  $\boldsymbol{R} \leftarrow \boldsymbol{Q} + I$
  for $i = 1$ **to** n **do begin**
    for $j = 1$ **to** n **do begin**
      Spočítej M
      Spočítej δ z $\boldsymbol{R}$ dle vzorce (6.2)
      Aktualizuj $\boldsymbol{R}_{ij} \leftarrow [\underline{r}_{ij}, \overline{r}_{ij} - \delta]$
    end
  end
  $\boldsymbol{Q} \leftarrow \boldsymbol{R} - I$
  return $\boldsymbol{Q}$
end

Kontrakci zespoda dostaneme když budeme pracovat s maticí $-\boldsymbol{R}$ místo matice $\boldsymbol{R}$. První krok algoritmu musíme upravit na $\boldsymbol{R} = -(\boldsymbol{Q} + I)$ a předposlední krok na $\boldsymbol{Q} = -(\boldsymbol{R} - I)$. Vlastní kontrakce zůstane stejná.

6.0.11 Volba matice M

Inverzní matice

Pokud položíme $M := R_c(\overline{r}_{ij})_{ij}^{-1}$, můžeme vzorec (6.2) upravit do jednoduššího tvaru.

$$\delta < \frac{2 - \rho(|M|R_\Delta(0) + (|M|R_\Delta(0))^T)}{\rho(|M|K + K^T|M|^T)} \tag{6.8}$$

Tato volba má z hlediska výpočetního času nevýhodu, že musíme při každé změně $\boldsymbol{R}$ znova počítat inverzní matici $R_c(\overline{r}_{ij})_{ij}^{-1}$.

Jednotková matice

Další možnost volby M je jednotková matice. Pak $M = I$ a $|M| = I$ a dosadíme do vzorce (6.2).

$$\delta < \frac{2 - \rho(|I - R_c(\overline{r}_{ij})| + |I - R_c(\overline{r}_{ij})|^T + R_\Delta(0) + R_\Delta(0)^T)}{\rho(K + K^T)} \tag{6.9}$$

Ve jmenovateli zůstane matice

$$K + K^T \tag{6.10}$$

která má na pozici ij a ji jedničku a jinde nuly.

Určíme vlastní čísla této matice, abychom mohli určit hodnotu jmenovatele. Charakteristická matice k matici (6.10) je tvaru

$$\begin{pmatrix} -\lambda & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & & & & \vdots \\ \vdots & & -\lambda & \dots & 1 & & \vdots \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ \vdots & & 1 & \dots & -\lambda & & \vdots \\ \vdots & & & & & \ddots & \vdots \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & -\lambda \end{pmatrix} \tag{6.11}$$

Vlastní čísla určíme jako kořeny charakteristického polynomu, který je roven determinantu matice (6.11). Determinant vypočítáme rozvojem podle i-tého řádku.

$$\det(K + K^T) = (-1)^{2i} \cdot (-\lambda) \cdot \det \begin{pmatrix} -\lambda & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -\lambda \end{pmatrix}^{n-1} + (-1)^{i+j} \cdot \det(B)$$

kde matice B vypadá následovně

$$B = \begin{pmatrix} -\lambda & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots & & & & & & \vdots \\ \vdots & & -\lambda & 0 & & & & & & \vdots \\ \vdots & & 0 & 0 & -\lambda & \cdots & 0 & & & \vdots \\ \vdots & & & \vdots & \ddots & \ddots & \vdots & & & \vdots \\ \vdots & & & 0 & & \ddots & -\lambda & & & \vdots \\ 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & & 0 & & & 0 & -\lambda & & \vdots \\ \vdots & & & \vdots & & & & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & -\lambda \end{pmatrix}^{n-1}$$

Pomocí rozvoje dle $(j-1)$-tého řádku určíme determinant matice B.

$$\det(B) = (-1)^{i+j-1} \cdot \det \begin{pmatrix} -\lambda & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -\lambda \end{pmatrix}^{n-2} = (-1)^{i+j-1} \cdot (-\lambda)^{n-2}$$

Determinant matice (6.10) je roven

$$\begin{aligned} \det(K + K^T) &= (-\lambda)^n + (-1)^{i+j} \cdot \det(B) = (-\lambda)^n + (-1)^{2i+2j-1} \cdot (-\lambda)^{n-2} \\ &= (-\lambda)^n - (-\lambda)^{n-2} = (-\lambda)^{n-2} \cdot (\lambda^2 - 1) \end{aligned}$$

Vlastní čísla matice (6.10) jsou 1 a -1, takže $\rho(K + K^T) = 1$, což dosadíme do vzorce (6.9).

$$\delta < 2 - \rho(|I - R_c(\overline{r}_{ij})| + |I - R_c(\overline{r}_{ij})|^T + R_\Delta(0) + R_\Delta(0)^T) \tag{6.12}$$

7. Implementace

7.0.12 Prostředí

Implementaci provedeme v Matlabu. (Konkrétně byl použit Matlab R2012a.) Pro intervalové počítání využijeme knihovnu Intlab [18]. V Intlabu jsou funkce pro základní práci s intervaly a intervalovou aritmetikou. Umožňuje některé pokročilejší výpočty a práci s nastavitelnou přesností a další. Základní úvod do Intlabu v češtině najdeme zde [11].

Dále využijeme knihovnu Versoft [19], která umí řešit některé problémy numerické lineární algebry, obsahuje různé funkce pro matice, lineární systémy a optimalizační problémy.

Všechny výpočty v obou knihovnách jsou verifikované. To znamená, že hledané řešení opravdu patří do našeho výsledku.

Obě knihovny jsou napsané v Matlabu a volně dostupné na internetu.

7.0.13 Uživatelská dokumentace

Všechny algoritmy jsme uvedli v předešlém textu v pseudokódu. Popsali jsme vždy jednu smyčku algoritmu, ve které proběhne kontrakce každé proměnné v matici $\boldsymbol{Q}$ právě jednou. Všechny metody dostanou na vstupu intervalovou matici $\boldsymbol{Q}$ a vracejí (kontrahovanou) matici $\boldsymbol{Q}$ a proměnou r, která je rovna jedné v případě, že jsme zjistili, že zadaná matice $\boldsymbol{Q}$ neobsahuje žádnou ortogonální matici. V opačném případě je r rovna nule. V matici, kterou algoritmy vrací, když je $r = 1$, se vyskytuje nedefinovaná hodnota (NaN), která vznikla většinou z prázdného průniku. Tato konvence odpovídá metodě `emptyintersect()`, kterou používáme při výpočtu průniků.

Dále máme naprogramovány kontraktory, které využívají kombinace více dílčích kontrakcí k dosažní lepšího konečného výsledku. Jejich základní kostra odpovídá pseudokódu v sekci 3.2. Na vstupu dostanou, podobně jako dílčí smyčky, intervalovou matici $\boldsymbol{Q}$ a vrací opět kontrahovanou matici $\boldsymbol{Q}$ a proměnnou r, která má stejný význam jako výše.

Z důvodu testování a porovnání jednotlivých metod jsme vytvořili další funkce, které generují testovací data nebo porovnávají výsledné kontrahované matice a provádí testy.

Metody generující testovací data

Tyto metody generují matice, na kterých testujeme jednotlivé metody.

1. `crmatrand()` vytváří náhodné matici. Vygeneruje náhodně středy intervalu mezi -1 a 1 a poloměry mezi 0 a 1.

2. `crmatortjedn()` jako středovou matici vezme jednotkovou matici, která je ortogonální, a náhodně k ní přidáme poloměry.

3. $crmatortgen()$ vyjde z matice

$$\begin{pmatrix} \frac{-7}{9} & \frac{4}{9} & \frac{4}{9} \\ \frac{4}{9} & \frac{-1}{9} & \frac{8}{9} \\ \frac{4}{9} & \frac{8}{9} & \frac{-1}{9} \end{pmatrix}$$

   která je ortogonální, a náhodně vygeneruje poloměry intervalů, čímž z matice dělá intervalovou matici, která nutně obsahuje ortogonální matici.

Všechny intervaly matic generujeme, tak aby byly podintervaly $[-1, 1]$.

Metody na porovnání

Pro porovnání jsme si vytvořili pomocné metody comp(A,B) a compp(A,B). První metoda spočítá rozdíl kontrahované a původní matice. Metoda compp() přepočítá výši kontrakce na procenta vzhledem k délkám původních intervalu v matici. Tuto metodu používáme pro další porovnání.

Metody na testování

Dále máme dvě metody, které provádí testování.

1. test_oneiter() pustí jednu smyčku dané metody a pomocí metody compp() porovná, jak moc metoda matici zkontrahovala. Tato metoda je užitečná, proto abychom viděli, jestli vůbec metoda matici kontrahuje.

2. test_kontr() pustí celý kontraktor a opět metodou compp() zjišťuje, jak moc se matice zkontrahovala.

7.0.14 Výsledky testů

Implementovali jsme následující metody pro dílčí kontrakce:

1. kvadr_prop() Kontrakce podle kvadratických podmínek

2. bilin_prop_ia() Kontrakce podle bilineárních podmínek intervalovou aritmetikou

3. bilin_prop_linprog() Kontrakce pomocí lineárního programování

4. spoj_podm() Propagace podle součtu kvadratické a bilineární podmínky

5. kontr_det_inv() Kontrakce podle determinantu s inverzní maticí

6. kontr_det_jedn() Kontrakce podle determinantu s jednotkovou matici

S těmito metodami jsme provedli základní testy a porovnání, abychom zjistili, jak moc a jestli vůbec kontrahují intervalovou matici. Tyto testy nám poslouží jako nápověda, jak vhodně nakombinovat jednotlivé metody do výsledného kontraktoru, aby fungoval co nejlépe.

Vytvořili jsme soubor testovacích dat, na kterých jsme nechali proběhnout jednu smyčku kontrakce a pak jsme provedli porovnání původní a nové matice.

Metoda	Výsledek testu v [%]
kvadr_prop()	11
bilin_prop_ia()	6
bilin_prop_linprog()	15
spoj_podm()	1
kontr_det_inv()	0,1
kontr_det_jedn()	0,00

Nejlépe dopadl kontraktor založený na lineárním programování a podle kvadratických podmínek. Naproti tomu kontraktor podle determinantu s jednotkovou maticí je sice o něco málo rychlejší než kontraktor podle determinantu s inverzí, ale funguje velmi špatně. Srovnatelných výsledků dosáhly metody kontrakce podle bilineárních podmínek za využití intervalové aritmetiky a sečtení s kvadratickou podmínkou. Existuji však matice, které jedna z metod kontrahuje a druhá ne a naopak.

Matice, kterou metoda spoj_podm() zkontrahuje, ale metoda bilin_prop_ia() ne.

```
[ -0.1643,  1.0000] [ -0.2331,  0.2331] [ -0.5333,  0.5333]
[ -1.0000,  1.0000] [ -0.3091,  1.0000] [ -0.2668,  0.2668]
[ -1.0000,  1.0000] [ -1.0000,  1.0000] [ -0.4877,  1.0000]
```

Naproti tomu následující matici kontrahuje lépe bilin_prop_ia().

```
[  0.4391,  1.0000] [ -0.1268,  0.1268] [ -0.3700,  0.3700]
[ -1.0000,  1.0000] [  0.6001,  1.0000] [ -1.0000,  1.0000]
[ -1.0000,  1.0000] [ -0.3159,  0.3159] [ -0.9815,  1.0000]
```

Existují případy, kdy tato situace nastává v jedné matici s různými členy matice.

```
[  0.9311,  1.0000] [ -0.1969,  0.1969] [ -0.0741,  0.0741]
[ -0.0123,  0.0123] [ -0.6242,  1.0000] [ -1.0000,  1.0000]
[ -1.0000,  1.0000] [ -0.2439,  0.2439] [ -0.7805,  1.0000]
```

Členy, které zkrátí metoda spoj_podm(), dokáže metoda kontrakce podle kvadratických podmínek kvadr_prop() zkrátit většinou více. Když ve výsledném kontraktoru pustíme pustíme spoj_podm() po kvadr_prop() a bilin_prop_ia() pravděpodobně nedosáhneme ničeho nového.

Na různých maticích se jednotlivé metody chovají jinak. Existují matice, které některé metody nedokáží kontrahovat vůbec, ale jiné je kontrahují.

Ve výsledném kontraktoru je vhodné použít alespoň jeden kontraktor podle determinantu. Tyto kontraktory využívají ke kontrakci jiných vlastností, takže budou pravděpodobně fungovat dobře pro jiné matice než ostatní kontraktory.

Jako příklad může posloužit třeba matice

```
[  0.5528,  1.0000] [ -1.0000,  1.0000] [ -0.3873,  0.3873]
[ -0.3327,  0.3327] [  0.1464,  1.0000] [ -0.2510,  0.2510]
[ -0.2034,  0.2034] [ -1.0000,  1.0000] [ -0.0919,  1.0000]
```

Pro tuto matici fungoval velmi dobře kontraktor podle determinantu kontr_det_inv(), přestože pro většinu jiných matic tento kontraktor nedokáže vůbec matici zkrátit. Jedna smyčka kontrakce podle determinantu kontrahovala tuto matici o 42 %. Oproti tomu kontrakce podle kvadr_prop() fungovala hůře, asi 12 %, a bilin_prop_ia() nekontrahovala vůbec.

Podobně je na tom srovnání lineárního programování bilin_prop_linprog() a kontrakce podle intervalové aritmetiky bilin_prop_ia(). Lineární programování funguje v průměrném případě o trochu lépe. Na některých maticích dosahuje výrazně lepších výsledků, například matice

```
[ -0.3695,  1.0000] [ -0.4553,  0.5556] [ -0.4909, -0.2806]
[  0.6202,  1.0000] [ -0.2639,  0.0511] [ -0.9839,  0.7575]
[ -0.1328,  1.0000] [ -0.0284,  0.5819] [ -0.3485,  1.0000]
```

Výjimky jsou však poměrně časté. Například tuto matici bilin_prop_linprog() oproti od bilin_prop_ia() nekontrahuje vůbec.

```
[  0.3929,  0.4830] [  0.4052,  0.6226] [ -0.8047,  0.9071]
[  0.3644,  0.7061] [ -1.0000,  0.4992] [ -1.0000,  0.1333]
[ -0.0715,  1.0000] [  0.6093,  0.9130] [ -0.8614, -0.4657]
```

Z výsledků testů vyplývá, že každá metoda dílčí kontrakce se chová jinak, což nasvědčuje tomu, že bude výhodné použít kombinaci různých metod k dosažení nejlepšího výsledku v konečném kontraktoru.

Provedeme porovnání dvou výsledných kontraktorů, které se budou lišit kontrakcí podle bilineárních podmínek.

1. kontr_ia() Kontraktor, který bude obsahovat kvadr_prop(), bilin_prop_ia() a kontrakci podle determinantu s inverzi (kontr_det_inv()).

2. kontr_linprog() Kontraktor se bude skládat z kvadr_prop(), bilin_prop_linprog() a kontr_det_inv() stejně jako v minulém případě.

Výsledky porovnání jsou shrnuty v následující tabulce.

Metoda	Výsledek testu v [%]
kontr_ia()	22
kontr_linprog()	14

Závěr

V práci jsme navrhli několik možných metod pro ortogonální kontraktor. Tyto metody jsme naimplementovali a porovnali. Při návrhu metod jsme vyšli z vlastností ortogonálních matic. Z vlastnosti definující ortogonalitu matice jsme dostali jednoduchým použitím intervalové aritmetiky metodu kontrakce, která fungovala poměrně dobře, takže jsme ji použili jako základ navrhovaného kontraktoru. Dále jsme kontraktor doplnili o složitější metody kontrakce podle determinantu, čímž jsme dosahli v některých případech lepších výsledků. Dále jsme vyzkoušeli aplikovat lineární programování, které také dosáhlo celkem dobrých výsledků.

Jednotlivé metody je složité vzájemně porovnat, protože se na různých maticích chovají velmi odlišně. Přesto jsme nějaké porovnání provedli a předložili v poslední kapitole.

Seznam použité literatury

[1] Milan Hladík and Luc Jaulin: *An Eigenvalue Symmetric Matrix Contractor*. Reliable Computing, 16:27-37, 2011.

[2] Milan Hladík, David Daney, and Elias P. Tsigaridas: *A filtering method for the interval eigenvalue problem.* Applied Mathematics and Computation, 217(12):5236-5242, 2011.

[3] F. Domes, A. Neumaier: *Constraint propagation on quadratic constraints.* Springer Science + Business Media, LLC 2009

[4] R. E. Moore, R. B. Kearfott, M. J. Cloud : *Introdaction to Interval Analysis.* Society for Industrial and Applied Mathematics, Philadelphia 2009

[5] J. Bečvář: *Lineární algebra.* Matfyzpress, Praha 2010

[6] M. Hladík: *Lineárná algebra (nejen) pro informatiky.* Skripta MFF UK, 2012

[7] G. Rex, J. Rohn: *Sufficient conditions for regularity and singularity of interval matrices.* SIAM J. Matrix Anal. Appl. 20 (1998) 437–445.

[8] A.S. Deif: *The interval eigenvalue problem* Z. Angew. Math. Mech. 71 (1991) 61–64.

[9] R.A. Horn, C.R. Johnson: *Topics in Matrix Analysis* Cambridge Univ. Press, 1994.

[10] G.H. Golub, C.F. Van Loan: *Matrix computations* J. Hopkins Univ. Press, 1996.

[11] J. Horáček: *Přeurčené soustavy intervalových lineárních rovnic* Diplomová práce, MFF UK, 2011

[12] O. Borůvka: *Základy teorie matic* Praha Academia, 1971 dostupné na webu: www.dml.cz

[13] j. Tůma: *Skripta k předmětu Lineární algebra* MFF UK, 2004

[14] P. Kolman: *Polynomiální algoritmy pro lineární programování* Skripta MFF UK, 2011

[15] C.S.Adjiman, S. Dallwig, C.A.Floudas, A. Neumaier: *A global optimization method, αBB, for general twice-differentiable constrained NLPs - I. Theoretical advances* 1997

[16] Siegfried M. Rump: *Algebraic Computation, Numerical Computation and Verified Inclusions* In R. Janssen, editor, Trends in Computer Algebra, Lecture Notes in Computer Science 296, 1988, pp. 177–197.

[17] X. Baguenard, M. Dao, L. Jaulin et W. Khalil (2003): *Méthodes ensemblistes pour l'étalonnage géométrique*, Journal Européen des Syst?mes Automatisés, Volume 37, n°9, pages 1059-1074

[18] http://www.ti3.tu-harburg.de/rump/intlab/

[19] http://uivtx.cs.cas.cz/ rohn/matlab/