



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební
Katedra mechaniky

**Optimalizace uniformity počítačových návrhů
pro omezené návrhové prostory**

**Optimization of uniformity of computer experiments
for constrained design spaces**

Diplomová práce

Studijní program: Stavební inženýrství

Studijní obor: Materiálové inženýrství

Vedoucí práce: doc. Ing. Matěj Lepš, Ph.D.

Bc. Eva Myšáková

Praha 2013



ZADÁNÍ DIPLOMOVÉ PRÁCE

studijní program: Stavební inženýrství
studijní obor: Materiálové inženýrství
akademický rok: 2013/2014

Jméno a příjmení diplomanta: Eva Myšáková
Zadávací katedra: Katedra mechaniky
Vedoucí diplomové práce: doc. Ing. Matěj Lepš, Ph.D.
Název diplomové práce: Optimalizace uniformity počítačových návrhů pro omezené návrhové prostory
Název diplomové práce v anglickém jazyce: Optimization of uniformity of computer experiments for constrained design spaces


Rámcový obsah diplomové práce: Rešerše měřítek uniformity počítačových návrhů, vývoj několika algoritmů pro optimalizaci uniformity, implementace jednotlivých metod v MATLABu, porovnání výpočetní náročnosti a kvality výsledků jednotlivých algoritmů.

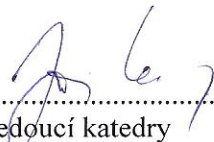
Datum zadání diplomové práce: 23. 9. 2013 Termín odevzdání: 20. 12. 2013
(vyplňte poslední den výuky přísl. semestru)

Diplomovou práci lze zapsat, kromě oboru A, v letním i zimním semestru.

Pokud student neodevzdal diplomovou práci v určeném termínu, tuto skutečnost předem písemně zdůvodnil a omluva byla děkanem uznána, stanoví děkan studentovi náhradní termín odevzdání diplomové práce. Pokud se však student řádně neomluvil nebo omluva nebyla děkanem uznána, může si student zapsat diplomovou práci podruhé. Studentovi, který při opakovaném zápisu diplomovou práci neodevzdal v určeném termínu a tuto skutečnost řádně neomluvil nebo omluva nebyla děkanem uznána, se ukončuje studium podle § 56 zákona o VŠ č.111/1998 (SZŘ ČVUT čl 21, odst. 4).

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.


.....
vedoucí diplomové práce


.....
vedoucí katedry

Zadání diplomové práce převzal dne: 23. 9. 2013


.....
diplomant

Formulář nutno vyhotovit ve 3 výtiscích – 1x katedra, 1x diplomant, 1x studijní odd. (zašle katedra)

Nejpozději do konce 2. týdne výuky v semestru odešle katedra 1 kopii zadání DP na studijní oddělení a provede zápis údajů týkajících se DP do databáze KOS.
DP zadává katedra nejpozději 1. týden semestru, v němž má student DP zapsanou.

Čestné prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně pouze za odborného vedení vedoucího diplomové práce doc. Ing. Matěje Lepše, Ph.D.

Dále prohlašuji, že veškeré podklady, ze kterých jsem čerpala, jsou uvedeny v seznamu použité literatury.

Datum:

Podpis:

Na tomto místě bych ráda srdečně poděkovala doc. Ing. Matěji Lepšovi, Ph.D. za jeho cenné připomínky, nekonečnou trpělivost a ochotu při vedení mé diplomové práce.

Abstrakt

Návrh experimentů - *design of experiments (DoE)* představuje základní a nedílnou část experimentování v mnoha oblastech vědy a výzkumu stejně jako v procesu optimalizace výrobních procesů i výrobků samotných. Má zásadní význam při tvorbě a testování metamodelů, v citlivostní analýze či ve spolehlivostních výpočtech. V této práci se věnujeme případům, v nichž jsou na vstupní parametry experimentu kladeny omezující podmínky, jež činí tyto parametry vzájemně závislými. Takové omezující podmínky mění tvar návrhového prostoru z regulárního prostoru hyperkrychle na prostor obecně nepravidelný. Speciálním případem experimentu s omezením je *návrh směsi*, ve kterém tvoří jednotlivé vstupní parametry (zastoupení složek směsi) jednotkový objem nebo hmotnost.

V práci se věnujeme metodám tvorby návrhů experimentů v omezených návrhových prostorech a jejich hodnocení. Pro to jsou zvolena dvě kritéria zaměřená na rovnoměrnost pokrytí návrhového prostoru - kritérium *Euklidovské Maximin vzdálenosti (EMM)* a kritérium *miniMax (mM)*. Každé z těchto kritérií je schopné jasně poukázat na závažné nedostatky návrhu. Vyhodnocení druhého ze jmenovaných kritérií navíc samo o sobě představuje problém, jemuž se v práci věnujeme.

Jednotlivé metody tvorby návrhů aplikované na několik konkrétních ilustračních příkladů jsou v práci porovnány z pohledu kvality výsledných návrhů (pomocí uvedených kritérií) a časové náročnosti.

Abstract

Design of experiments (DoE) creates an essential and integral part of any experimentation in many fields of science research and optimization of production processes and products themselves. It is also essentially important for meta-models development, sensitivity analysis or probability calculations. This thesis is focused on cases with limiting conditions put on input parameters like dependency of input parameters. Presence of limiting conditions changes shape of design space from regular hypercube to a generally irregular constrained design space. A special case of this type of experiment is a mixture experiment, where individual parameters (ingredients) form a unity volume or weight.

In this thesis we are focused on methods for creation of experimental designs in constrained design spaces as well as on their ranking. For this purpose two space-filling criteria are chosen - *Euclidean Maximin distance* (*EMM*) and criterion *miniMax* (*mM*). Both these criteria are able to point out serious shortcomings of the design. Evaluation of the second mentioned criterion is a problem itself which is also discussed in this thesis.

Individual methods for creation of experimental designs applied on several illustrative examples are compared in terms of quality of resulting designs (evaluated by mentioned criteria) and computational demands.

Klíčová slova

návrh experimentů, omezené návrhové prostory, omezující podmínky, návrh směsi, simplex, konvexní polytop, rovnoměrnost pokrytí, Maximin, miniMax, evoluční strategie, Delaunayova triangulace, plošné souřadnice, implementace

Keywords

design of experiments, constrained design spaces, limiting conditions, mixture experiment, simplex, convex polytope, space-filling, Maximin, miniMax, evolution strategy, Delaunay triangulation, barycentric coordinates, implementation

Obsah

Obsah	8
Seznam obrázků	10
1 Úvod	12
1.1 Návrhové prostory	14
1.2 Návrh směsi	16
1.3 Návrhy v obecných omezených spojitých prostorech	17
2 Kritéria uniformity	19
2.1 Euklidovská Maximin vzdálenost (EMM)	20
2.2 Kritérium miniMax (mM)	20
2.2.1 Přesná hodnota kritéria miniMax	21
2.2.2 Evoluční strategie pro odhad hodnoty kritéria miniMax	23
3 Metody pro tvorbu návrhů v omezených návrhových prostorech	25
3.1 Generátor náhodných bodů	27
3.2 Metoda LHS na opsaném hyperkvádru	28
3.3 Metoda odebrání bodů z přehuštěného návrhu	29
3.3.1 Metoda ubírání	29
3.3.2 Metoda ubírání _{NEW}	29
3.4 Nástroj Distmesh	30
3.5 Simplexové mřížkové (<i>lattice</i>) návrhy	31
4 Způsoby vracení bodů opustivších doménu	33
4.1 Kolmý průmět na facetu	34
4.2 Nulování záporných plošných souřadnic	35
4.3 Pomocná sada bodů uvnitř domény	36
4.4 Porovnání	37

5	Příklady aplikace metod	39
5.1	Příklady návrhů s malým počtem bodů ve 2D a ve 3D	39
5.2	Návrh směsi	41
5.2.1	Trojsložková směs	41
5.2.2	Šestisložková směs	43
6	Implementace	44
7	Výsledky	45
7.1	Porovnání metod vyhodnocení kritéria miniMax	45
7.2	Porovnání metod tvorby návrhů experimentů	47
8	Závěr	51
	Literatura	52
A	Přehled prvků (0-5)-dimenzionálního simplexu a hyperkrychle	57
B	Algoritmus pro výpočet <i>EMM</i>	58
C	Delaunayova triangulace	59
D	Plošné souřadnice	61
E	Výpočet hodnoty kritéria miniMax v regulárním prostoru	63
E.1	Přesné řešení	63
E.2	Sériová evoluční strategie	64
E.3	Paralelní evoluční strategie	66
E.4	Porovnání a shrnutí	67
F	Ruční výpočet hledání vrcholů polytopu pomocí LP	70

Seznam obrázků

1.1	Návrh experimentů.	13
1.2	Porovnání návrhů experimentů.	13
1.3	Tvary návrhových prostorů ve 3D.	14
1.4	Transformace návrhu z regulárního prostoru.	15
1.5	Spojité a diskrétní omezený návrhový prostor ve 2D.	15
1.6	Návrhový prostor pro návrh směsi.	16
1.7	Návrhový prostor pro návrh směsi s dalším omezením.	17
1.8	Doporučený postup četby.	18
2.1	Kritéria Maximin a miniMax.	19
2.2	Výpočet přesné hodnoty kritéria miniMax.	22
2.3	Sériová evoluční strategie.	23
2.4	Paralelní evoluční strategie.	24
3.1	1. skupina metod tvorby	25
3.2	2. skupina metod tvorby	26
3.3	Vrcholový simplex.	26
3.4	Varianty rozdělení bodů do simplexů.	27
3.5	Příklady LHS návrhu.	28
3.6	Použití LHS v opsaném hyperkvádru.	29
3.7	Odebírání z přehuštěného návrhu.	30
3.8	Tvorba rovnoměrné sítě z náhodně vytvořených bodů nástrojem <i>Distmesh</i>	31
3.9	Příklady simplexových <i>lattice</i> návrhů.	31
3.10	Aplikace simplexových <i>lattice</i> návrhů ve ztriangulované doméně.	32
4.1	Posun bodů - kolmý průmět.	33
4.2	Posun bodů - nejbližší bod ve vrcholu.	34
4.3	Posun bodů - kolmý průmět - řešení na celé doméně.	34
4.4	Posun bodů - nulování záporné pl. s.	35
4.5	Posun bodů - pomocná sada.	36

4.6	Půlení intervalu.	37
4.7	Porovnání způsobů vracení bodů.	38
5.1	Příklady návrhů s malým počtem bodů ve 2D a 3D.	40
5.2	Množina přípustných řešení v kartézské soustavě.	42
5.3	Množina přípustných řešení v trojosé soustavě.	42
7.1	Výsledky evoluční strategie pro vyčíslení mM	46
7.2	Výsledky metod tvorby na 2D a 3D příkladech.	48
7.3	Výsledky metod tvorby na příkladech návrhů směsí.	49
C.1	Triangulace.	59
C.2	Dualita DT a VD.	60
D.1	Plošné souřadnice.	61
D.2	Trojosý souřadnicový systém a plošné souřadnice.	62
D.3	Určení polohy bodu pomocí plošných souřadnic.	62
E.1	miniMax I.	63
E.2	miniMax II.	64
E.3	Sériová evoluční strategie.	65
E.4	Paralelní evoluční strategie - způsoby dělení na subdomény.	66
E.5	Paralelní evoluční strategie ve 2D.	66
E.6	Paralelní evoluční strategie.	67
E.7	Speed-up paralelní ES.	68
E.8	Boxplot hodnot mM	68
E.9	Vývoj hodnoty mM v průběhu výpočtu.	69
F.1	Simplexová tabulka obecně.	70
F.2	Simplexové tabulky č. 1-3.	71
F.3	Simplexové tabulky č. 4-6.	72
F.4	Simplexové tabulky č. 7-9.	73
F.5	Simplexové tabulky č. 10-12.	74

Kapitola 1

Úvod

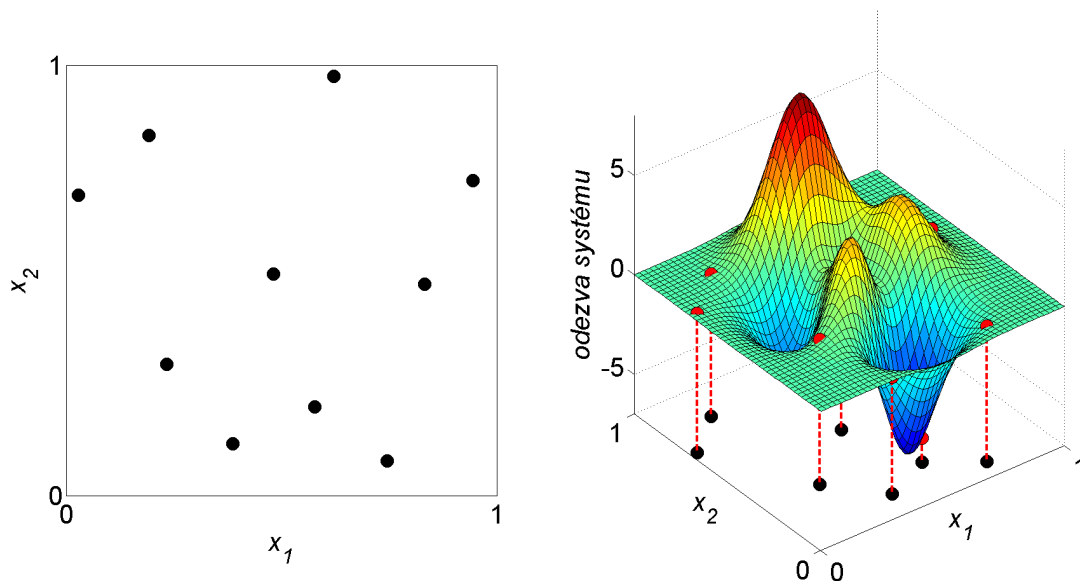
„Žádné množství experimentů nemůže prokázat, že mám pravdu; jediný stačí k prokázání opaku.“¹

Albert Einstein.

Provádění experimentů je nepostradatelnou a nesčetněkrát opakovanou činností nejen každého vědce, programátora, návrháře, výrobce, ale i člověka v každodenním životě. Experimentem chápeme provedení testu za určitých podmínek s cílem zjištění odezvy. Příkladem nám může být chemický experiment, ve kterém volíme zastoupení jednotlivých sloučenin a zjišťujeme například množství uvolněné energie při reakci. Nebo návrh výrobku, kdy volíme rozměry jednotlivých prvků a zajímá nás vliv takové volby například na pevnost nebo životnost. Za experiment ovšem můžeme považovat i pečení nového moučníku - volíme množství (poměry) jednotlivých surovin a zjišťovanou odezvou může být zpracovatelnost těsta nebo chutnost výsledného produktu. Experiment tedy slouží ke zjištění odezvy (reakce) nějakého systému na určité hodnoty vstupních parametrů, respektive k získání představy o chování takového systému (a následně k vyšší pravděpodobnosti, že budeme schopni chování systému v různých situacích správně předvídat). V ideálním případě bychom tedy provedli co největší množství experimentů (s různými vstupními parametry) a získali tak informaci, jak systém zareaguje v různých situacích. Ve skutečnosti bývá ovšem počet experimentů, jež je možné provést, značně limitován (z důvodů finančních, časových, či faktických). Je proto zapotřebí věnovat značnou pozornost vhodnému naplánování experimentů, konkrétně volbě kombinací hodnot vstupních parametrů pro jednotlivá provedení. K tomu slouží *návrh experimentů - design of experiments* (DoE).

Návrh experimentů je soubor návrhových bodů (jejich počet odpovídá počtu možných uskutečnění experimentu), jejichž souřadnice odpovídají konkrétním hodnotám vstupních

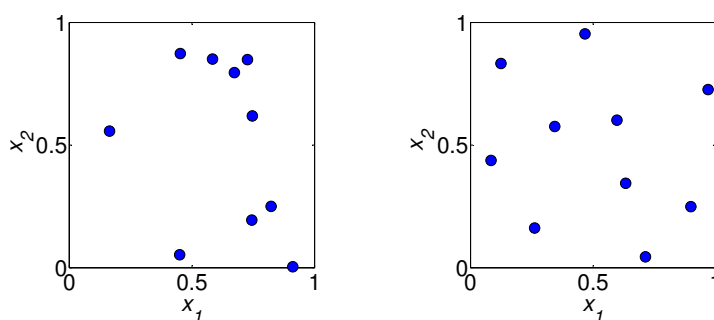
¹Původně „No amount of experimentation can ever prove me right; a single experiment can prove me wrong.“ Albert Einstein [Wynn and Wiggins, 1997].



Obrázek 1.1: Návrh experimentů. Obrázek vlevo ukazuje návrh experimentů ve 2D. Obrázek vpravo pak zachycuje aplikaci návrhu experimentů - určení odezvy systému (červené body) v návrhových bodech (černé body). Pro ilustraci byla použita funkce `peaks` dostupná v programu MATLAB.

parametrů (proměnných). V těchto bodech je následně provedeno vyčíslení (proveden experiment s odpovídajícími hodnotami parametrů). Obrázek 1.1 znázorňuje návrh experimentů se dvěma vstupními parametry.

Tvorbě návrhu experimentů je třeba věnovat značnou pozornost; je zřejmé, že v závislosti na použitém návrhu můžeme získat značně odlišnou představu o chování systému. Obrázek 1.2 ukazuje rozdílné návrhy z pohledu rovnoměrnosti pokrytí návrhového prostoru. Použitím návrhu vpravo bychom jistě získali o chování systému lepší představu.²



Obrázek 1.2: Porovnání návrhů experimentů. Vlevo nekvalitní návrh, vpravo návrh rovnoměrně pokrývající návrhový prostor (návrhovou doménu).

²Existují samozřejmě případy, kdy není jediným kritériem pro hodnocení návrhu rovnoměrnost pokrytí návrhového prostoru. V mnoha aplikacích je záměrně „zahušťován“ návrhovými body určitý konkrétní prostor v návrhové doméně.

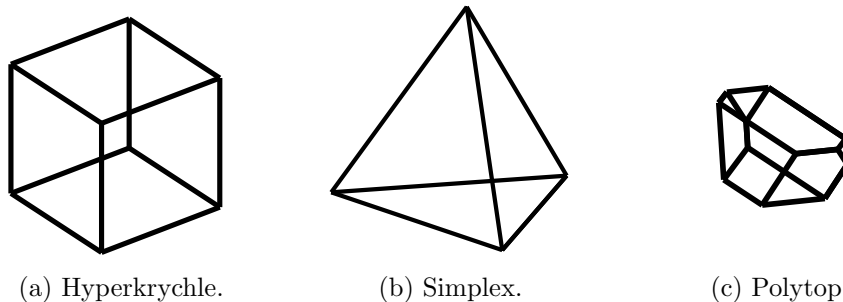
Obecně je u návrhů experimentů důraz kladen především na zmíněnou rovnoměrnost pokrytí (*space-filling*) a ortogonalitu.

Je třeba zmínit, že návrhy experimentů se liší rovněž podle toho, pro jakou aplikaci jsou určeny. Jak již bylo zmíněno, setkáváme se s touto problematikou v mnoha oblastech lidské činnosti. Původně byly návrhy experimentů použity v chemii, dnes se s nimi setkáváme kromě vědeckých disciplín v mnoha odvětvích průmyslu i ve službách či v marketingu. Nezastupitelnou roli hrají v experimentech počítačových (simulacích), ve spolehlivostních výpočtech, či citlivostních analýzách. O reálných experimentech (fyzikálních, chemických [Liang et al., 2001]...), jejich navrhování i analýze pojednává kniha [Montgomery, 2000], počítačovým experimentům se pak věnují například autoři v [Sacks et al., 1989] nebo [Fang et al., 2006]. Rozdíl mezi experimenty reálnými a počítačovými je možné demonstrovat například na tom, zda pro stejné kombinace vstupních parametrů obdržíme stejnou odezvu systému. U počítačových experimentů tomu tak zpravidla je (pokud není do modelu záměrně vložen nějaký prvek náhodnosti), zatímco u reálných experimentů toto platit nemusí. Proto je například odlišně nahlíženo na přítomnost duplikovaných návrhových bodů v návrzích experimentů pro jednotlivá odvětví.

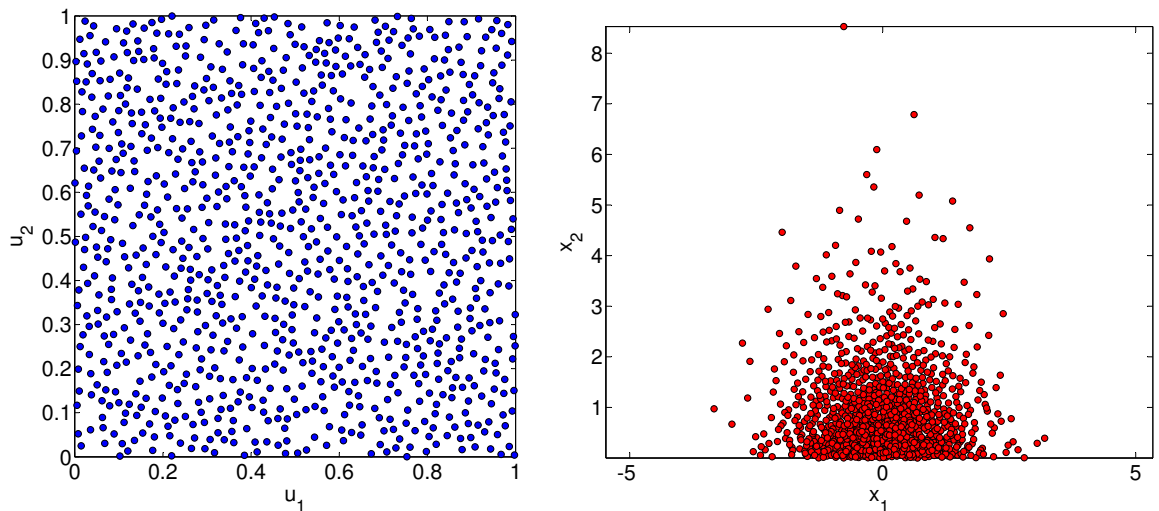
1.1 Návrhové prostory

Tvorba návrhů experimentů se liší především v závislosti na tvaru návrhového prostoru. Ten je dán všemi přípustnými kombinacemi hodnot vstupních parametrů. V případě, že jsou jednotlivé návrhové parametry (proměnné) nezávislé a každý z nich má zadané určité rozdělení, je návrhovým prostorem hyperkrychle (Obrázek 1.3a). I v případě, že parametry nemají stejné rozdělení, můžeme návrh experimentů vytvořit rovnoměrný v jednoduché hyperkrychli $[0, 1]^n$ (kde n je dimenze/počet parametrů) a jednotlivé parametry pak pomocí inverzní kumulativní distribuční funkce převést do zadaného rozdělení [Myšáková and Lepš, 2013c], jak je znázorněno na Obrázku 1.4.

Velmi častým případem experimentu s omezením je *návrh směsi* (*mixture experiment*), ve kterém tvoří jednotlivé vstupní parametry jednotkovou váhu či objem. Tomuto typu



Obrázek 1.3: Tvary návrhových prostorů ve 3D.



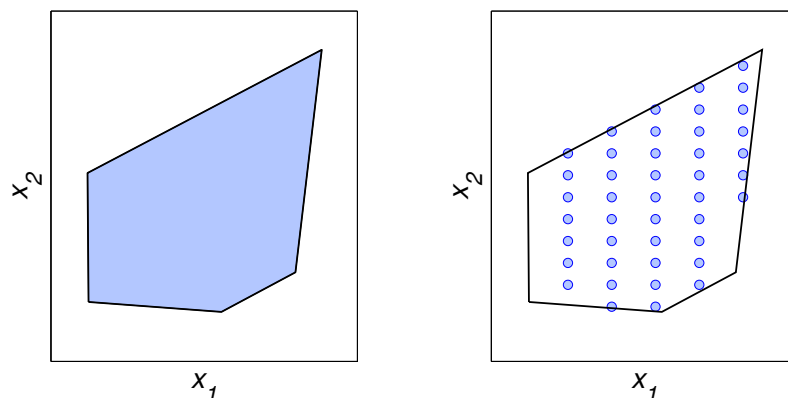
(a) Rovnoměrný návrh v regulárním prostoru hyperkrychle (ve 2D: čtverec). (b) Převedení do zadaných rozdělení: 1. parametr (osa x_1) - normální rozdělení ($\mu = 0, \sigma = 1$), 2. parametr (osa x_2) - exponenciální rozdělení ($\mu = 1$).

Obrázek 1.4: Transformace návrhu z regulárního prostoru hyperkrychle $[0, 1]^n$ (kde n je dimenze/počet parametrů) do prostoru daného zadanými rozděleními jednotlivých parametrů.

experimentu odpovídá návrhový prostor ve tvaru simplexu (Obrázek 1.3b). Více se tomuto druhu experimentu věnuje následující Sekce.

Jsou-li zadány jiné nebo další omezující podmínky, je návrhovým prostorem *polytop*³ (Obrázek 1.3c).

Zásadní rozdíl v tvorbě návrhů experimentů (a v podobě návrhového prostoru) je



(a) Spojitý návrhový prostor je dán nekonečným počtem přípustných kombinací vstupních parametrů (modrá plocha).

(b) Diskrétní návrhový prostor je dán konečným počtem přípustných kombinací vstupních parametrů (modré body).

Obrázek 1.5: Příklady spojitého a diskrétního omezeného návrhového prostoru ve 2D.

³V této práci uvažujeme pouze konvexní polytoxy.

rovněž dán tím, zda mohou jednotlivé parametry nabývat libovolných hodnot v rámci svých mezí - jsou spojité, či zda je počet přípustných hodnot konečný - jsou diskrétní. Rozdíl ukazuje Obrázek 1.5.

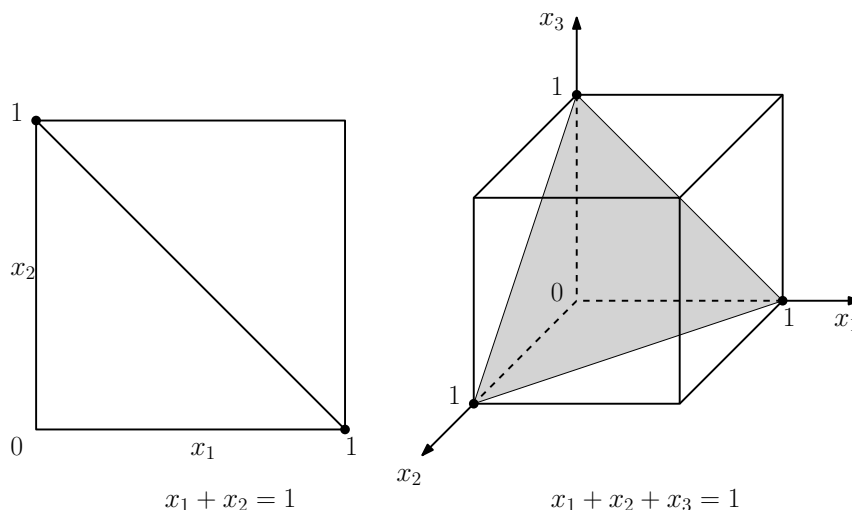
1.2 Návrh směsi

Speciální, ale velmi častý případ experimentu s omezeným návrhovým prostorem je *návrh směsi*. Je dán omezující podmínkou

$$x_1 + x_2 + \dots + x_n = 1, \quad 0 \leq x_i \leq 1, \quad i = 1, \dots, n, \quad (1.1)$$

kde n odpovídá počtu složek dané směsi. Tato samotná podmínka určuje tvar návrhového prostoru jako $(n - 1)$ -dimenzionální simplex (Obrázek 1.3b a 1.6). Hodnoty jednotlivých parametrů (zastoupení složek směsi) je dáno plošnými (trojúhelníkovými, barycentrickými) souřadnicemi bodů v simplexu [Cruise, 1966] (více v Příloze D). Jejich počet je vždy o jednu vyšší než dimenze odpovídajícího simplexu. Například návrhu směsi s pěti složkami tedy odpovídá návrhový prostor ve tvaru 4-dimenzionálního simplexu; každý bod v tomto simplexu je možné popsat pěti barycentrickými souřadnicemi, jejichž součet je vždy roven jedné.

Kromě základní omezující podmínky návrhu směsi (Rovnice 1.1) bývají zpravidla přítomny ještě další omezující podmínky: ty nejčastěji stanovují limity zastoupení jednotlivých složek směsi. Každá taková podmínka mění tvar návrhového prostoru - „ořezává“ původní simplex, jak je vidět například na Obrázku 1.7. Návrhovým prostorem se pak



Obrázek 1.6: Návrhový prostor pro návrh směsi se dvěma (vlevo) a třemi (vpravo) složkami.

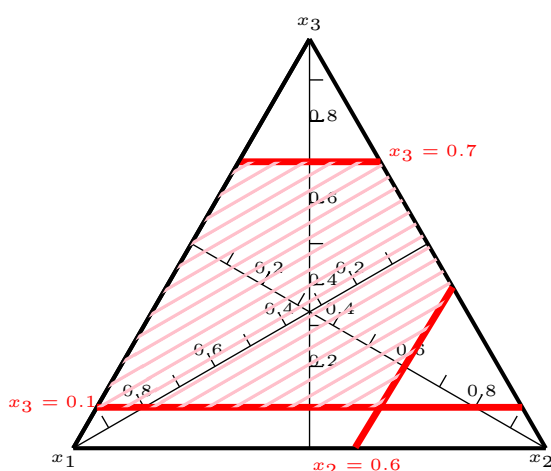
stává obecně nepravidelný, ale vždy konvexní polytop.

Návrhu experimentů pro návrh směsi je věnována pozornost již několik desetiletí, jak je popsáno v [Khuri, 2006, Kapitola 12] nebo v [Chan, 2000]. Před nástupem počítačových experimentů byly používány klasické šablony o několika návrhových bodech [Cornell, 1973, Cornell, 1979]. Například mřížkové návrhy (více v Kapitole 3) se ve vhodných případech [JMP, 2005] stále často využívají. V dnešní době se však pozornost zaměřuje spíše na počítačově optimalizované návrhy experimentů [Lepš, 2009, Kapitola 2].

1.3 Návrhy v obecných omezených spojitých prostorech

Tato práce se zabývá návrhy experimentů v omezených spojitých prostorech. Návrhovými doménami tedy jsou simplexu nebo obecné konvexní polytopy. Jelikož uvažujeme tvorbu návrhů pro experimenty počítačové, je pro nás základním požadavkem rovnoměrnost pokrytí návrhového prostoru.

Metodami pro tvorbu takových návrhů se zabývá řada autorů. Například autoři v článku [Hofwing and Strömberg, 2010] optimalizují *D-optimality* návrhů v omezených prostorech pomocí sekvenčního lineárního programování v kombinaci s genetickým algoritmem. [Draguljic et al., 2012] se zaměřuje na projekční vlastnosti návrhů experimentů v omezených prostorech. Podmíněné rozdělení pravděpodobnosti v metodách Monte Carlo je použito pro tvorbu rovnoměrných návrhů v [Fang and Yang, 2000]. [Petelet et al., 2010] při tvorbě vychází z LHS návrhu (viz Sekci 3.2), na kterém provádí permutace, dokud



Podmínka návrhu směsi:

$$x_1 + x_2 + x_3 = 1$$

Limity pro zastoupení složek:

$$x_2 \leq 0,6$$

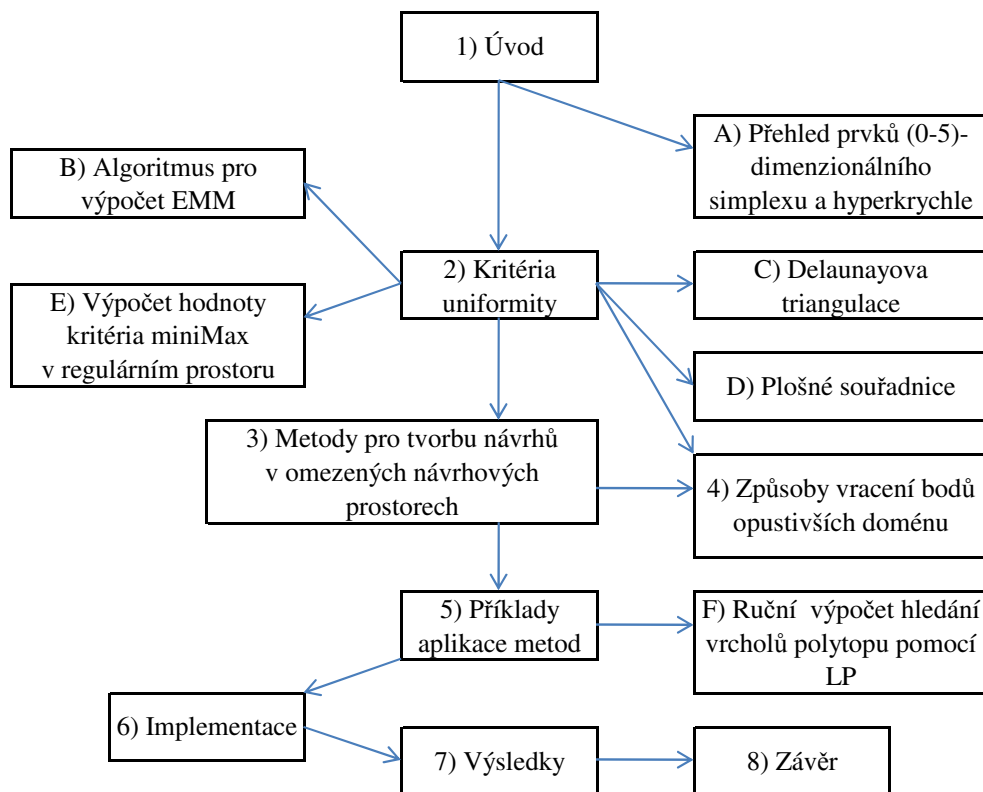
$$x_3 \geq 0,1$$

$$x_3 \leq 0,7$$

Obrázek 1.7: Změna podoby návrhového prostoru pro návrh směsi při zavedení limitů pro zastoupení jednotlivých složek. Červené úsečky značí hraniční přímky podmínek, růžově šrafovaná oblast pak výsledný návrhový prostor.

nejdou splněny omezující podmínky. Dva přístupy jsou popsány autory v [Prescott, 2008]: pro domény ve tvaru simplexu nebo simplexu blízkém je navržen postup založený na tzv. *sphere packing* - tedy plnění domény daným počtem hyperkoucí. Návrhovými body jsou pak středy těchto hyperkoucí. Pro dlouhé a úzké domény podobné obdélníkům potom autoři navrhují přizpůsobení pravidelné síte.

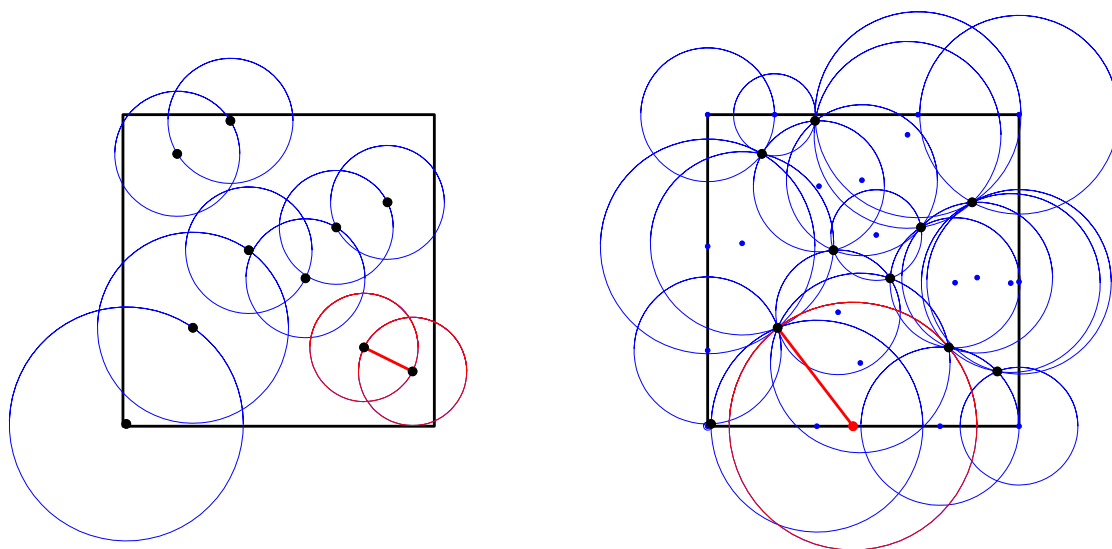
V této práci se věnujeme tvorbě optimalizovaných (z pohledu rovnoměrnosti pokrytí) návrhů experimentů ve spojitém omezeném návrhovém prostoru. Členění práce je následující. V Kapitole 2 představíme dvě kritéria zvolená pro hodnocení i optimalizaci jednotlivých návrhů. Vyhodnocení jednoho z nich je samo o sobě náročné, je mu proto věnována značná pozornost. Kapitola 3 se věnuje vlastním metodám tvorby návrhů. Dále je v Kapitole 4 řešen problém vracení bodů, které opustí návrhovou doménu, zpět na její povrch. To může být zapotřebí jak při tvorbě návrhu, tak při jeho hodnocení. Příklady, na něž jsou představené metody aplikovány, jsou uvedeny v Kapitole 5. Kapitola 6 se věnuje popisu implementace uvedených algoritmů. V Kapitole 7 nalezneme výsledky získané použitím metod na uvedených příkladech následované shrnutím v Kapitole 8. Práce obsahuje rovněž řadu příloh, Obrázek 1.8 proto ukazuje doporučený postup její četby. Text je rovněž doprovázen řadou obrázků, které slouží k ilustraci, v mnoha případech jsou proto vynechány popisy os.



Obrázek 1.8: Doporučený postup četby, respektive návaznost jednotlivých kapitol a sekcí.

Kapitola 2

Kritéria uniformity



(a) Euklidovská Maximin vzdálenost (EMM).
Legenda: modré kružnice - kruhy se středy v návrhových bodech s poloměry rovnými vzdálenostem k nejbližšímu z ostatních návrhových bodů; červené kružnice - nejmenší z těchto kruhů; červená úsečka - spojnice vzájemně si nejbližších návrhových bodů; délka červené úsečky - EMM .

(b) Kritérium miniMax (mM).
Legenda: modré body - vrcholy Voronoiova diagramu (kandidátní body); modré kružnice - největší prázdné kruhy se středy v modrých bodech; červená kružnice - největší prázdný kruh; červený bod - střed největšího prázdného kruhu; délka červené úsečky - mM .

Obrázek 2.1: Kritéria Maximin a miniMax. Obrázky slouží pro ilustraci, popisky os jsou proto vynechány. Legenda: černé úsečky - hranice návrhové domény; černé body - návrhové body.

Kvalitu návrhu můžeme hodnotit řadou kritérií zaměřených především na ortogonalitu [Cioppa and Lucas, 2007, Hofwing and Strömberg, 2010] nebo rovnoměrnost pokrytí [Crombecq et al., 2009, Myšáková and Lepš, 2011, Janouchová and Kučerová, 2013]. V té

to práci jsme pro hodnocení jednotlivých návrhů zvolili kritérium *Euklidovská Maximin vzdálenost* (EMM) a kritérium *miniMax* (mM) [Johnson et al., 1990, Auffray et al., 2012]. Tato kritéria byla vybrána pro jejich snadnou zobrazitelnost a jednoduchou interpretaci. Zároveň se jedná o kritéria, která jasně upozorňují na závažné nedostatky návrhu. V případě Euklidovské Maximin vzdálenosti je to nežádoucí blízkost experimentů. V případě kritéria miniMax pak nedostatečné pokrytí některých oblastí návrhového prostoru.

2.1 Euklidovská Maximin vzdálenost (EMM)

Prvním kritériem použitým v této práci pro hodnocení návrhů experimentů je *Euklidovská Maximin vzdálenost* (EMM). Jedná se o jednoduchou a snadno zobrazitelnou metriku. Toto kritérium nejlépe poukazuje na vzájemnou blízkost experimentů (návrhových bodů). Euklidovská Maximin vzdálenost je vhodná pro hodnocení návrhů experimentů klasických reálných (fyzikálních, chemických, biologických, ...) stejně jako virtuálních - tzv. simulací (například armádních bojových scénářů). Jedná se o kritérium často používané k optimalizaci návrhů experimentů, jak můžeme vidět například v [Grosso et al., 2009], [Morris, 2014] nebo v [Pronzato and Walter, 1988]. EMM je nejkratší ze všech vzájemných vzdáleností mezi body návrhu:

$$EMM = \min\{\dots, L_{ij}, \dots\}, \quad i = 1, \dots, np; \quad j = (i + 1), \dots, np \quad , \quad (2.1)$$

kde np je počet návrhových bodů a L_{ij} je euklidovská vzdálenost mezi body i a j . Snažíme se tedy o maximalizaci hodnoty EMM .

K určení hodnoty EMM je nutné spočítat vzájemné vzdálenosti mezi všemi body návrhu. Efektivní algoritmus je použit ve funkci `lhsdesign` ze statistického toolboxu programu MATLAB. Díky vhodné indexaci jsou všechny vzdálenosti spočteny během jediného *for*-cyklu. Algoritmus je uveden v Příloze B. Význam kritéria je ilustrován na Obrázku 2.1a.

2.2 Kritérium miniMax (mM)

Problematika kritéria *miniMax* je v literatuře též označována jako *the largest empty sphere problem* (LES) [Dickerson and Eppstein, 1995] - „problém největší prázdné koule“ (Obrázek 2.1b). Je-li dána sada (návrhových) bodů v omezeném prostoru, hodnota miniMax je poloměr největší hyperkoule, která neobsahuje žádný z bodů sady (maximálně se dotýká) a jejíž střed leží uvnitř daného prostoru. Kritérium miniMax tedy slouží k hodnocení kvality pokrytí návrhového prostoru tím, že ukazuje největší nepokrytou oblast.

Máme-li zadánu sadu np návrhových bodů $D = d_i, i = 1, \dots, np$ v omezeném prostoru

S , hledáme bod $p \in S$ takový, že jeho vzdálenost k nejbližšímu návrhovému bodu je nejdelší:

$$mM = \max_{p \in S} \min_i d(p, d_i), \quad i = 1, \dots, np \quad , \quad (2.2)$$

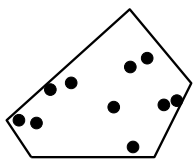
kde $d(x, y)$ je euklidovská vzdálenost mezi body x a y . Pro návrh rovnoměrně pokrývající návrhový prostor je tedy žádoucí co nejnižší hodnota mM .

Vyčíslení hodnoty tohoto kritéria je výrazně náročnější než v případě Euklidovské Maximin vzdálenosti. Jednou z možností pro nalezení středu největší prázdné koule (a poté hodnoty mM) je prověření všech vrcholů Voronoiova diagramu [Okabe et al., 2000] (více v Příloze C) sestaveného pro danou sadu návrhových bodů. Počet těchto vrcholů však roste $O(np^{\lceil dim/2 \rceil})$ v případě, kdy prostor není ohraničen, a v ohraničeném prostoru je počet bodů nutných k prověření ještě vyšší. Ačkoli můžeme problém hraničních oblastí domény efektivně řešit pomocí zrcadlení [Pronzato and Müller, 2012], ve vyšších dimenzích jsou paměťové a časové nároky algoritmů pro sestavení Voronoiova diagramu velmi vysoké. Ve vyšších dimenzích je tedy nutné spokojit se s odhadem hodnoty kritéria miniMax. Tento odhad nám může poskytnout například metoda založená na evoluční strategii.

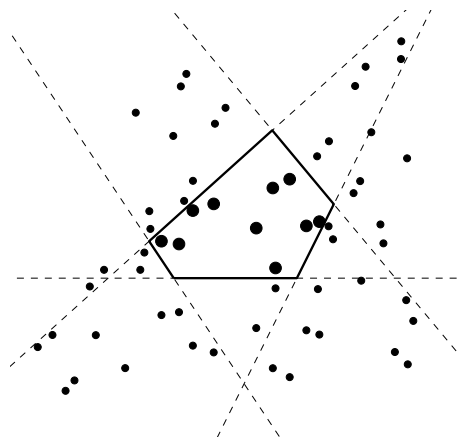
Způsoby vyčíslení kritéria miniMax v regulárním návrhovém prostoru byly předmětem předchozího studia a jsou uvedeny v Příloze E. Přístupy aplikované zde - v omezeném návrhovém prostoru - jsou jejich rozšířením.

2.2.1 Přesná hodnota kritéria miniMax

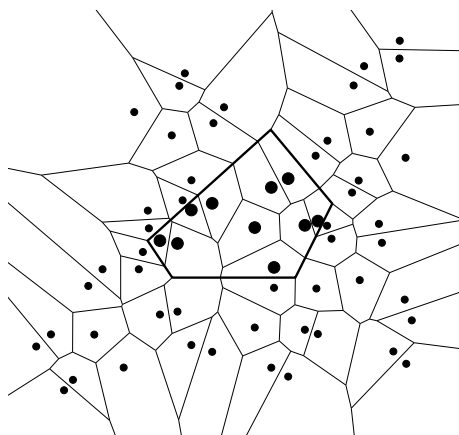
Přesnou hodnotu kritéria miniMax, resp. polohu bodu v návrhovém prostoru, který je nejvíce vzdálen od návrhových bodů, je možné nalézt pomocí Voronoiova diagramu. Kandidátními body jsou vrcholy Voronoiova diagramu, samotné vrcholy řešené domény a průsečíky hran diagramu s hraničními objekty řešené domény. Efektivnější, než hledat zmíněné průsečíky, je použití zrcadlení návrhových bodů skrze všechny $(n - 1)$ -dimenzionální facety (kde n označuje dimenzi) návrhového prostoru a až následné sestavení Voronoiova diagramu [Pronzato and Müller, 2012]. Tento přístup je funkční v regulárním návrhovém prostoru (viz opět Přílohu E). Zdá se ovšem použitelný (zde bez důkazu; správnost testována porovnáním výsledků získaných touto metodou s výsledky získanými pomocí dále popsanych metod) i ve zde řešených konvexních omezených návrhových prostorech. Postup ilustruje Obrázek 2.2.



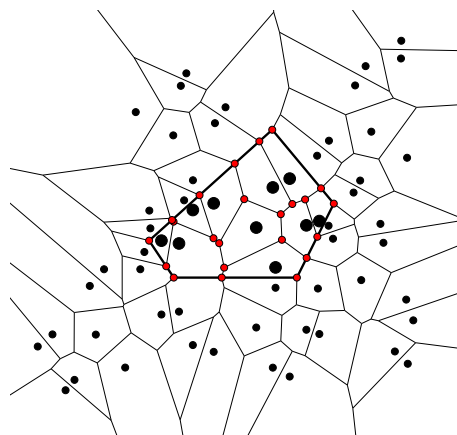
(a) Návrhové body v omezeném návrhovém prostoru.



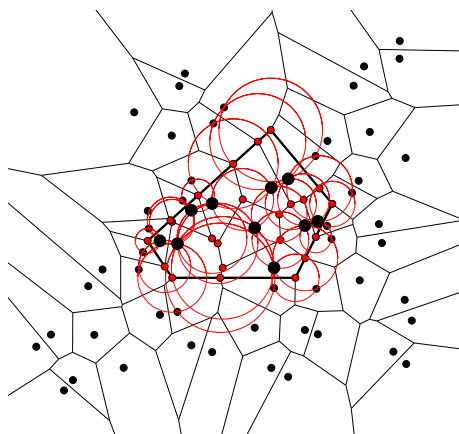
(b) Zrcadlení návrhových bodů skrze všechny $(n-1)$ -dimenzionální facety.



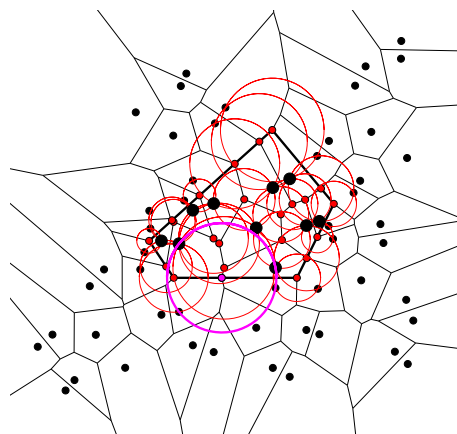
(c) Sestrojení Voronoiova diagramu.



(d) Označení vrcholů Voronoiova diagramu spadajících dovnitř nebo na hranici návrhového prostoru - kandidátní body.



(e) Sestrojení největších prázdných hyperkoulí (ve 2D: kruhů) se středy v kandidátních bodech.

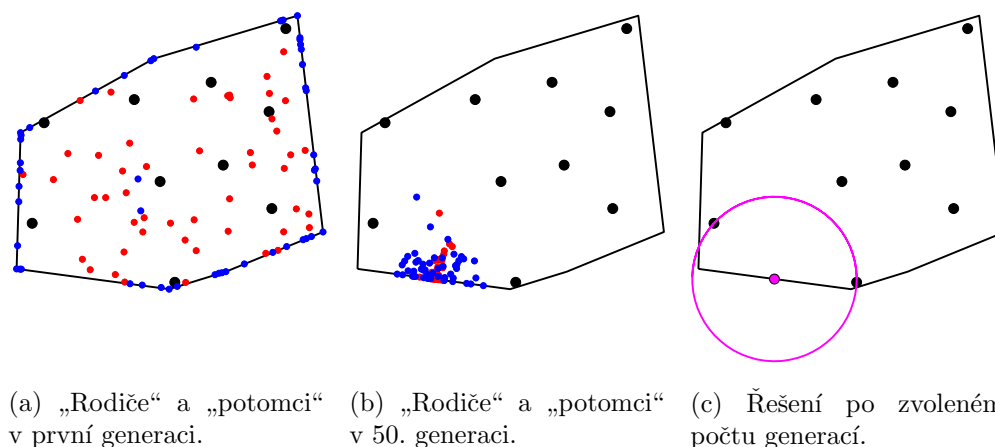


(f) Nalezení největší z těchto koulí. Její střed je bod nejvzdálenější od návrhových bodů. Poloměr této koule odpovídá hodnotě mM .

Obrázek 2.2: Výpočet přesné hodnoty kritéria miniMax. Ilustrace ve 2D.

2.2.2 Evoluční strategie pro odhad hodnoty kritéria miniMax

Ve vyšších dimenzích (problémy nastávají zpravidla již v šestiřozměrném návrhovém prostoru) je získání přesné hodnoty kritéria miniMax výše uvedeným způsobem a za použití běžných výpočetních prostředků nemožné. Odhad hodnoty mM a pozice „nejosamělejšího“ bodu v doméně však můžeme získat pomocí stochastické metody evoluční strategie (ES). Tento způsob byl úspěšně použit v regulárních návrhových prostorech (Příloha E), pokusíme se jej tedy aplikovat i zde.

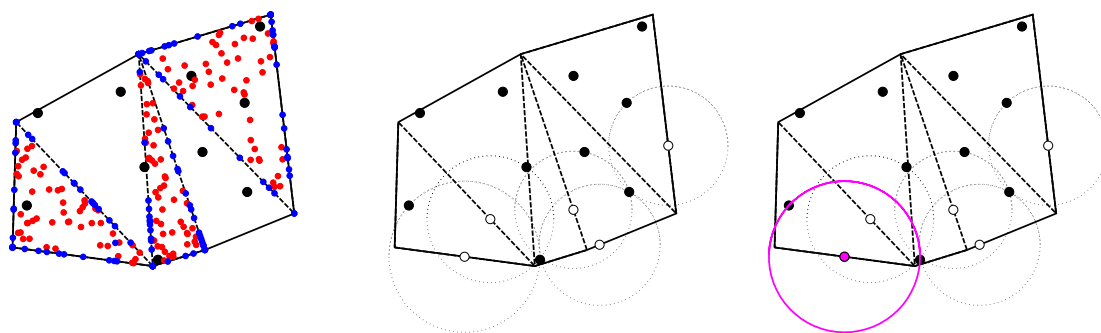


Obrázek 2.3: Sériová evoluční strategie pro vyhodnocení miniMax kritéria v omezeném návrhovém prostoru. Legenda: černé body - návrhové body; červené body - „rodiče“; modré body - „potomci“; purpurový bod - střed největšího prázdného kruhu; purpurová kružnice - největší prázdný kruh.

Sériová verze evoluční strategie je znázorněna na Obrázku 2.3. Střed největší prázdné hyperkoule (bod v doméně nejvíce vzdálený od návrhových bodů) je hledán v rámci celé této domény. Populace „potomků“ je z „rodičů“ odvozována mutací, konkrétně přičtením normálně rozdělených náhodných čísel⁴, což vede k tomu, že jsou vytvořeny body mimo návrhový prostor. Ty je třeba posunout zpět dovnitř, aby byly platné při hledání středu prázdné koule. Způsoby, jak toho dosáhnout, jsou popsány v Kapitole 4. Právě toto je však největším úskalím sériové verze evoluční strategie v omezeném návrhovém prostoru v porovnání s jejím použitím v prostoru regulárním (tam stačí souřadnici nabývající hodnoty mimo dané meze položit rovné právě hodnotě limitu).

Praktičtější je zde použití paralelní verze evoluční strategie pro vyhodnocení miniMax kritéria. A to nejen z důvodu možnosti výpočtu na větším počtu výpočetních jednotek zároveň. Zparalelizování provedeme rozdělením řešené návrhové domény na menší celky - subdomény - a to pomocí Delaunayovy triangulace (viz Přílohu C). Střed největší prázdné hyperkoule poté hledáme paralelně nezávisle v každé této subdoméně. Získáme

⁴Střední hodnota $\mu = 0$; směrodatná odchylka σ a) klesající s pořadím generace, b) řízena tzv. „pětinovým pravidlem“.



(a) „Rodiče“ a „potomci“ v první generaci v souběžně řešených subdoménách. (b) Kandidátní řešení z jednotlivých subdomén. (c) Výsledné řešení vybrané z kandidátních řešení.

Obrázek 2.4: Paralelní evoluční strategie pro vyhodnocení miniMax kritéria v omezeném návrhovém prostoru. Legenda: černé body - návrhové body; červené body - „rodiče“; modré body - „potomci“; bílé body - v každé subdoméně bod nejvíce vzdálený od návrhových bodů; tečkované kružnice - největší prázdné kruhy se středy v bílých bodech; purpurový bod - střed největšího prázdného kruhu; purpurová kružnice - největší prázdný kruh.

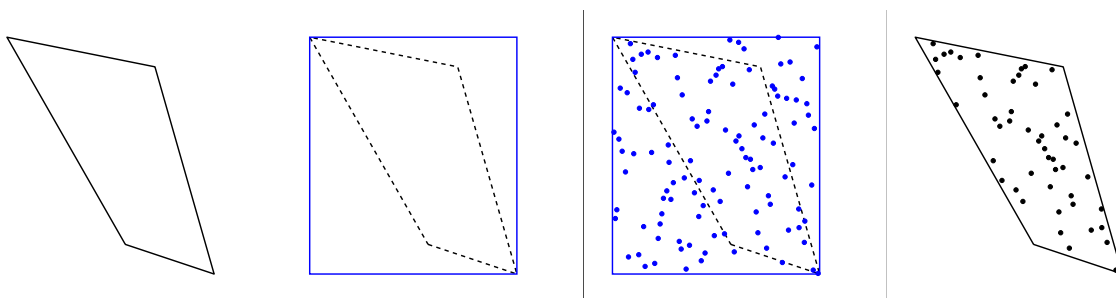
tak množinu kandidátních řešení z každé z nich a jako konečné řešení vybereme z těchto bodů ten, jehož vzdálenost k nejbližšímu návrhovému bodu je nejdelší. Metodu ilustruje Obrázek 2.3. V této verzi ES je snazší vracení bodů, které opustí řešenou subdoménu. To právě proto, že „držíme“ body v jednom konkrétním simplexu a do něho je lze jednoduše vracet pomocí nulování záporných plošných souřadnic, jak je popsáno v Sekci 4.2.

Kapitola 3

Metody pro tvorbu návrhů v omezených návrhových prostorech

Vytvořit návrh experimentů v omezeném návrhovém prostoru je možné v zásadě dvěma způsoby:

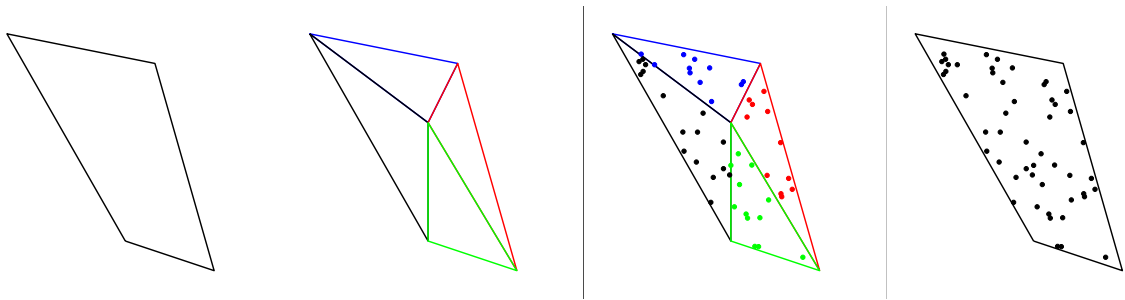
1. „obalit“ nepravidelnou doménu regulární doménou (hyperkrychlí/hyperkvádrem) - *bounding boxem*, v tomto prostoru vytvořit návrhové body a z těchto bodů následně vybrat ty, které leží v původní řešené doméně (viz Obrázek 3.1). V tomto případě lze použít jakékoli metody tvorby návrhu experimentů v regulárním návrhovém prostoru. Těm se věnuje práce [Mysáková, 2012] a některé z metod uvedených v této práci budou aplikovány i zde.



Obrázek 3.1: 1. skupina metod tvorby - *bounding box*.

2. rozdělit doménu na jednodušší objekty (simplexy), v těchto objektech vytvořit návrhové body a tyto body sjednotit (viz Obrázek 3.2)

Oba přístupy mají svá úskalí. U prvního je to nemožnost zaručení, že do omezené domény „padne“ právě požadované množství bodů. Rovněž není zaručeno, že kvalitní návrh v opsaném regulárním prostoru přinese i kvalitní návrh v omezené doméně. Největším úskalím se však může stát poměr objemu omezené domény a opsaného hyperkvádrů.



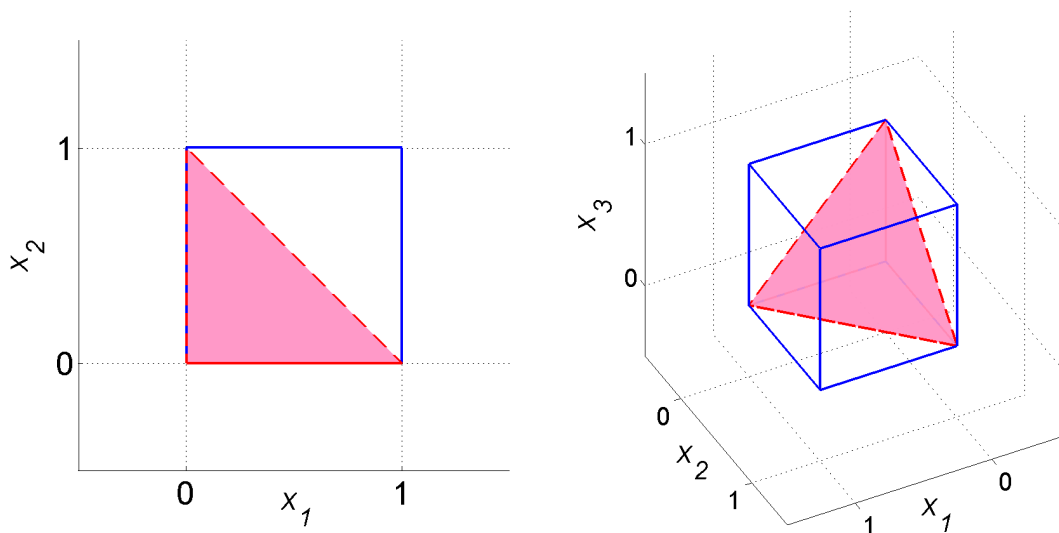
Obrázek 3.2: 2. skupina metod tvorby - rozdělení na simplexu.

Obzvláště pak se zvyšující se dimenzionalitou návrhového prostoru. Pro vrcholový simplex a jemu odpovídající opsanou hyperkrychli (viz Obrázek 3.3) je to:

$$V_{simplex}/V_{hyperkrychle} = \frac{1}{dim!} \quad , \quad (3.1)$$

kde dim označuje dimenzi prostoru. Například v 10dimenzionálním prostoru by tedy bylo nutno vytvořit v opsané hyperkrychli téměř 4 miliony rovnoměrně rozprostřených bodů, aby bylo pravděpodobné, že alespoň jeden z nich bude ležet v řešeném simplexu.

Problémy druhého přístupu spočívají především v rozdělení domény. Námi používaná Delaunayova triangulace (více informací v Příloze C) je se zvyšující se dimenzionalitou časově i paměťově náročná. Druhý přístup může rovněž vést k nekvalitním návrhům především v místech spojení jednotlivých simplexů.



Obrázek 3.3: Vrcholový simplex ve 2D a ve 3D.

3.1 Generátor náhodných bodů

První metoda tvorby využívá generátor náhodných bodů v simplexu. Metoda spadá do druhé skupiny (dle rozdělení výše), neboť jsou body vytvářeny v jednotlivých simplexech poté, co byla zadaná návrhová doména ztriangulována pomocí Delaunayovy triangulace. Námi používaný generátor [Devroye, 1986] je uveden níže. Souřadnice bodu jsou vytvořeny s exponenciálním rozdělením a následně normalizovány.

```

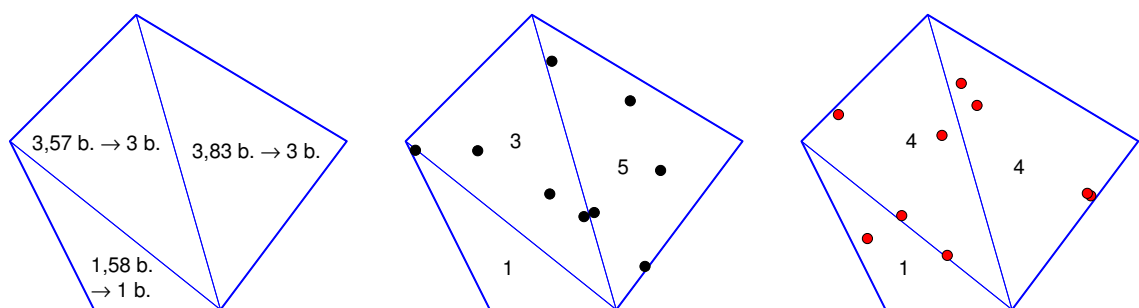
1 y=rand(1,n); % funkce rand - tvoří bod z rovnoměrného rozdělení [0-1]^(dim+1)
2 x=-log(y);
3 S=sum(x')';
4 X=x./repmat(S,1,n);
5 L=X*T; % v T jsou uloženy souřadnice vrcholů simplexu, v němž tvoříme náhodný bod

```

Počet návrhových bodů vytvořených v jednotlivých simplexech je závislý na poměru objemů těchto simplexů. Teoretický počet bodů $np_{simplex_i}$ v simplexu i je:

$$np_{simplex_i} = np \cdot \frac{V_{simplex_i}}{V_{domena}} = np \cdot \frac{V_{simplex_i}}{\sum_i V_{simplex_i}}, \quad (3.2)$$

kde np značí požadovaný počet bodů v celé doméně, $V_{simplex_i}$ objem simplexu i a V_{domena} objem celé domény. Při určování počtu bodů skutečně umístěných do jednotlivých simplexů je použita spodní celá část z teoretické hodnoty a zbylé body jsou a) vytvořeny v největším ze simplexů triangulace, b) rozmístěny po jednom do simplexů seřazených sestupně dle objemu (Obrázek 3.4). Takovýchto variant by bylo možno vyzkoušet celou řadu, na výsledek mají vliv pouze v případě malého počtu požadovaných bodů nebo podobného počtu bodů a počtu simplexů v triangulaci, případně pokud by triangulace obsahovala simplexu s výrazně odlišnými objemy.



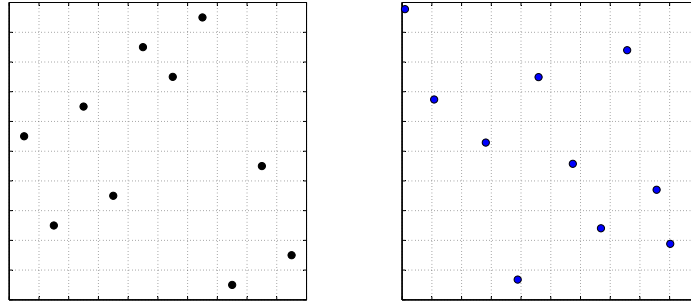
(a) Triangulace návrhové domény s označením počtu bodů teoreticky náležejících jednotlivým simplexům a počtu bodů skutečně umístěných do jednotlivých simplexů v prvním kroku.

(b) Zbytek bodů umístěn do největšího simplexu.

(c) Zbylé body umístěny po jednom postupně do simplexů seřazených sestupně dle objemu.

Obrázek 3.4: Varianty rozdělení bodů do simplexů. Počet bodů: 9.

3.2 Metoda LHS na opsaném hyperkvádru



Obrázek 3.5: Příklady LHS návrhu ve 2D s 5 body umístěnými do středů intervalů (vlevo) a náhodně v rámci intervalu (vpravo).

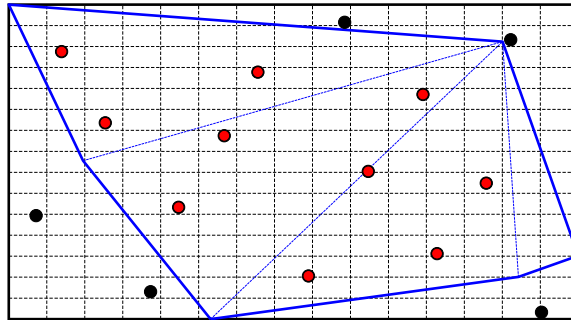
Další metoda spadá do 1. kategorie dle rozdělení v úvodu této kapitoly. Nepravidelné návrhové doméně opíšeme pravoúhlý hyperkvádr a v něm vytvoříme návrh experimentů.

Tento návrh by měl být co nejkvalitnější a co nejrovnoměrněji pokrývat zvolený prostor. Proto je vhodné zvolit návrh splňující podmínky LHS - *latin hypercube sampling* [Cioppa and Lucas, 2007, Toropov et al., 2007], neboť ten již ze své podstaty zajišťuje určitou úroveň rovnoměrnosti pokrytí. LHS je často používaný typ návrhu, v němž je každá z dimenzí (proměnných) rozdělena na np intervalů (kde np odpovídá požadovanému počtu bodů) a do každého intervalu každé proměnné je umístěn právě jeden bod. Běžně se používají dvě varianty, v první je bod umístěn do středu intervalu (Obrázek 3.5 vlevo), ve druhé pak náhodně v rámci celého intervalu (Obrázek 3.5 vpravo).

Použití LHS návrhu zde tedy probíhá tak, že jej vytvoříme v opsaném hyperkvádru a následně vybereme body, které leží uvnitř řešené omezené domény. Nelze tedy předem zaručit, kolik bodů „padne“ do cílového prostoru, vodítkem pro volbu počtu bodů v LHS návrhu nám může být poměr mezi objemem opsaného hyperkvádru a samotné omezené domény, jak ilustruje Obrázek 3.6. Je tedy zřejmé, že použití takovéto metody je vhodné pouze v případech, kdy řešená doména vyplňuje značnou část opsaného hyperkvádru.

Pro srovnání byly vyzkoušeny dvě varianty této metody - prostý neoptimalizovaný LHS návrh a LHS návrh optimalizovaný prohazováním pozic jednotlivých návrhových bodů vzhledem ke kritériu Euklidovské Maximin vzdálenosti⁵.

⁵V každé iteraci je nalezena dvojice vzájemně si nejbližších bodů (těm odpovídá aktuální hodnota *EMM*). Z této dvojice je náhodně vybrán jeden bod a k němu opět náhodně jeden bod ze všech ostatních bodů návrhu. Poté je zvolena jedna ze souřadnic (opět náhodně) a u zvolených bodů je hodnota této souřadnice prohozena. Pokud dojde ke zlepšení (zvýšení hodnoty *EMM*), je taková výměna přijata. Tato metoda je podrobně popsána v [Myšáková, 2012, Kapitola 2.1.2.2]



Obrázek 3.6: Použití LHS v opsaném hyperkvádru. Omezené návrhové doméně je opsán hyperkvádr, v němž je vytvořen LHS návrh (černé body). Následně jsou vybrány pouze body spadající do omezeného prostoru (červené body). Požadovaný počet bodů v omezeném prostoru je 10 a poměr objemů omezené domény a opsaného hyperkvádru je $0,6796 = 1/1,4714$; LHS návrh byl tedy vytvořen s 15 body a vidíme, že právě požadovaných 10 bodů leží uvnitř omezené domény.

3.3 Metoda odebírání bodů z přehuštěného návrhu

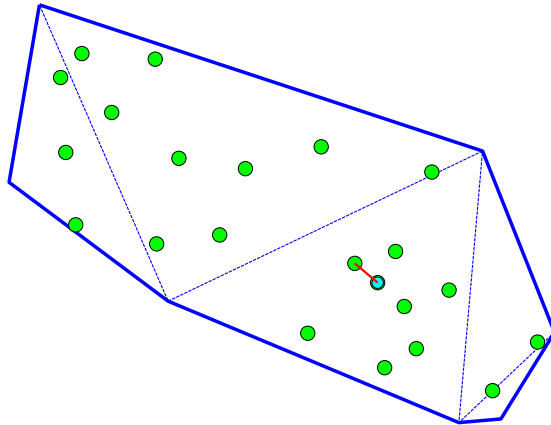
Další metoda pracuje na principu odebírání bodů ze záměrně přehuštěného návrhu. Tento přístup byl rovněž úspěšně použit v regulárním prostoru ([Myšáková and Lepš, 2011, Myšáková, 2012]). Pomocí generátoru náhodných bodů (Sekce 3.1) je v návrhovém prostoru vytvořen návrh s vyšším než požadovaným počtem bodů. Nadbytečné body jsou následně z návrhu odebírány a to nikoli náhodně, nýbrž heuristicky na základě kritéria Maximin. Metodu představujeme ve dvou dále popsanych verzích.

3.3.1 Metoda ubírání

V první variantě metody je v každé iteraci (počet iterací odpovídá počtu nadbytečných bodů) nalezena dvojice vzájemně si nejbližších návrhových bodů. Této dvojici odpovídá aktuální hodnota *EMM*. Náhodně zvolený bod z této dvojice je z návrhu odebrán. Je tedy zřejmé, že s každým dalším odebraným bodem se zlepšuje (zvyšuje) hodnota kritéria Maximin. Idea této metody je navíc taková, že po odebrání všech nadbytečných bodů získáme návrh, jehož kvalita bude vyšší než u návrhu, který je vytvořen najednou přímo s požadovaným počtem bodů.

3.3.2 Metoda ubírání NEW

Druhá varianty metody se liší způsobem volby odebraného bodu z dvojice s aktuální nejmenší vzájemnou vzdáleností. Ta zde není náhodná jako v první variantě, nýbrž deterministická. Odebrán je ten z bodů z dvojice, jehož druhá nejkratší vzdálenost k bodům



Obrázek 3.7: Odebírání z přehuštěného návrhu. Zelené body značí přehuštěný návrh. V každém kroku je označena dvojice vzájemně si nejbližších bodů (červená úsečka odpovídá hodnotě EMM). Z této dvojice je jeden bod odebrán (zvýrazněn azurově).

návrhu (tedy vzdálenost k nejbližšímu bodu návrhu jinému, než je druhý z dvojice) je kratší. Rozdíl je patrný i z Obrázku 3.7, kdy odebrání azurově zvýrazněného bodu je jistě výhodnější než odebrání druhého z bodu dvojice.

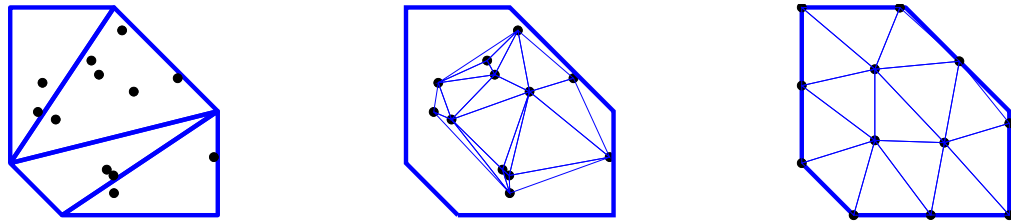
3.4 Nástroj Distmesh

Nástroj `Distmesh` detailně popsáný v [Persson and Strang, 2004] je heuristický algoritmus pro tvorbu rovnoměrných sítí [Chen and Holst, 2011] primárně určených pro metodu konečných prvků (MKP) - *Finite Element Method (FEM)*. Je známé, že v rovnoměrných sítích jsou jednotlivé uzly vzájemně srovnatelně vzdáleny. Proto tohoto nástroje využíváme rovněž pro tvorbu návrhů experimentů jak v regulárních [Myšáková, 2012], tak v omezených návrhových prostorech [Myšáková and Lepš, 2012, Myšáková et al., 2012].

Algoritmus je založen na jednoduchém dynamickém systému rozpínající se příhradové konstrukce. Příliš krátké pruty vyvolávají síly, jež tlačí navzájem blízké uzly směrem od sebe. Často tak ovšem nastává situace, že je uzel (návrhový bod) vysunut z návrhové domény. V takovém případě je třeba jej vrátit zpět. Nástroj `Distmesh` obsahuje tyto procedury pro jednoduché objekty, například 2D polygon, pro složitější útvary jako je simplex nebo polytop v ND je třeba vložit vlastní řešení, například některé z uvedených v Kapitole 4.

Postup metody využívající nástroje `Distmesh` je znázorněn na Obrázku 3.8. Do návrhové domény je třeba nejprve vložit prvotní sadu bodů⁶ a na tu aplikovat popsáný nástroj.

⁶Kvalita této výchozí sady nemá velký vliv na výsledek, zpravidla tedy postačí použití jednoduchého generátoru náhodných bodů v simplexech ztriangulované omezené návrhové domény.



(a) Triangulace návrhové domény a umístění výchozích bodů do jednotlivých simplexů.

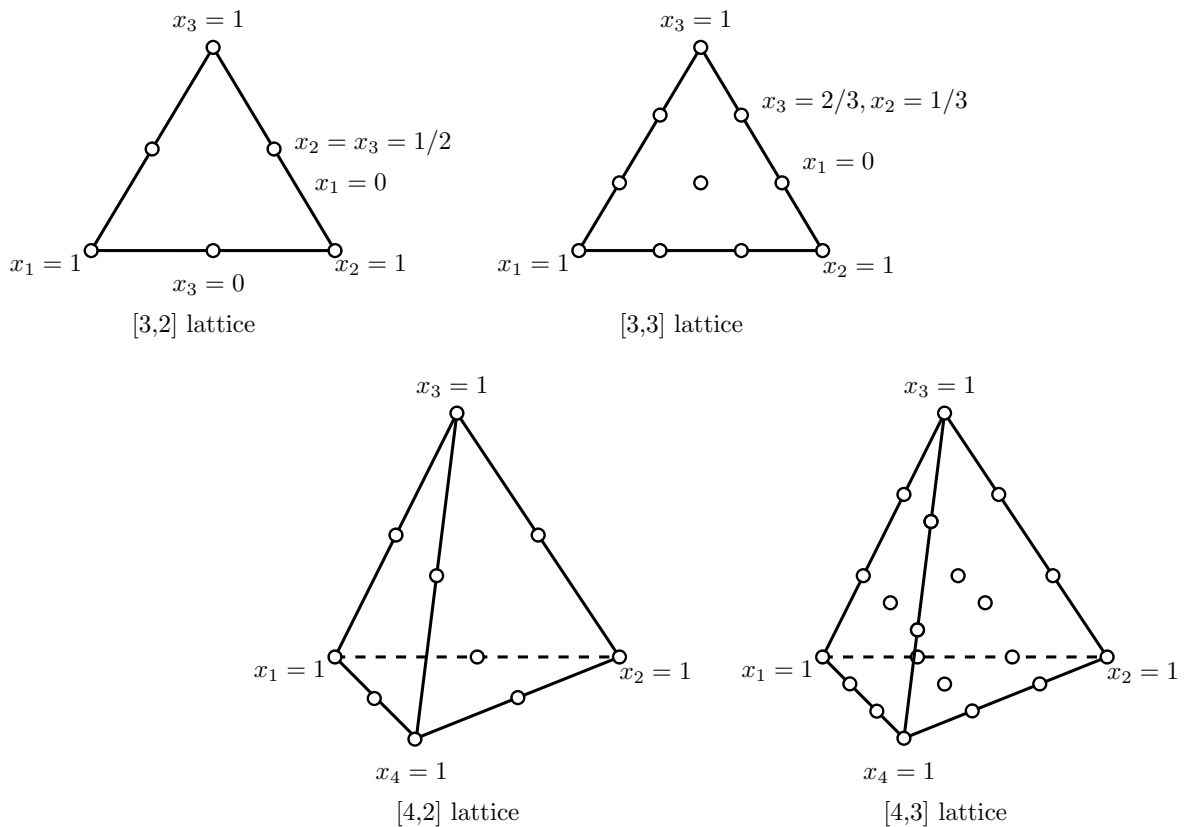
(b) Tvorba prutové konstrukce triangulací výchozích bodů.

(c) Finální návrh po aplikaci nástroje Distmesh.

Obrázek 3.8: Tvorba rovnoměrné sítě z náhodně vytvořených bodů pomocí nástroje Distmesh.

3.5 Simplexové mřížkové (*lattice*) návrhy

Klasickým způsobem tvorby návrhů v simplexovém prostoru jsou simplexové mřížkové (*lattice*) návrhy [Montgomery, 2000, Kapitola 11-5]. Vznikají pravidelným rozdělením jednotlivých barycentrických souřadnic a jsou tak obdobou plně faktoriálních návrhů v regulárních doménách. Při rozdělení každé z n barycentrických souřadnic (kde n odpovídá počtu složek) na m stejně dlouhých intervalů vznikne simplexový *lattice* návrh obsahující

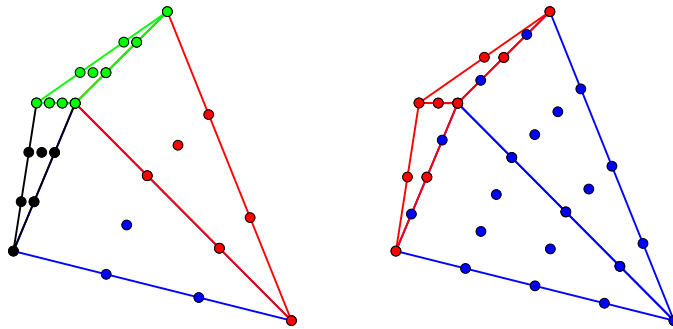


Obrázek 3.9: Příklady simplexových *lattice* návrhů se třemi a čtyřmi složkami.

$$M = \frac{(n + m - 1)!}{m!(n - 1)!} \quad (3.3)$$

návrhových bodů. Příklady návrhů jsou uvedeny na Obrázku 3.9 (označeny $[n, m]$).

Tyto návrhy lze tedy chápat jako šablony, které můžeme použít jak v jakémkoli simplexu, tak i v jednotlivých simplexech vzniklých triangulací libovolné domény, jak je znázorněno například na Obrázku 3.10 vlevo. Uvedený příklad má ovšem špatné vlastnosti z hlediska rovnoměrnosti pokrytí. Bylo by však možné použít různé *lattice* návrhy (s různými počty intervalů, na něž jsou rozděleny jednotlivé souřadnice) v závislosti na objemu jednotlivých simplexů tvořících triangulaci domény. To je znázorněno na Obrázku 3.10 vpravo, kde jsou ve „větších“ simplexech použity *lattice* návrhy $[3, 4]$ (modré body), zatímco v těch „menších“ návrhy $[3, 2]$ (červené body).

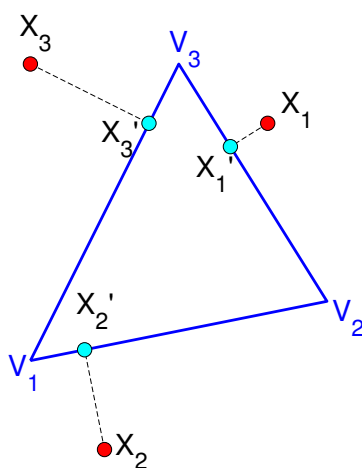


Obrázek 3.10: Aplikace simplexových *lattice* návrhů ve ztriangulované doméně. Vlevo použit stejný *lattice* návrh ve všech simplexech triangulace, vpravo použity různé návrhy v závislosti na objemech simplexů (resp. obsazích trojúhelníků).

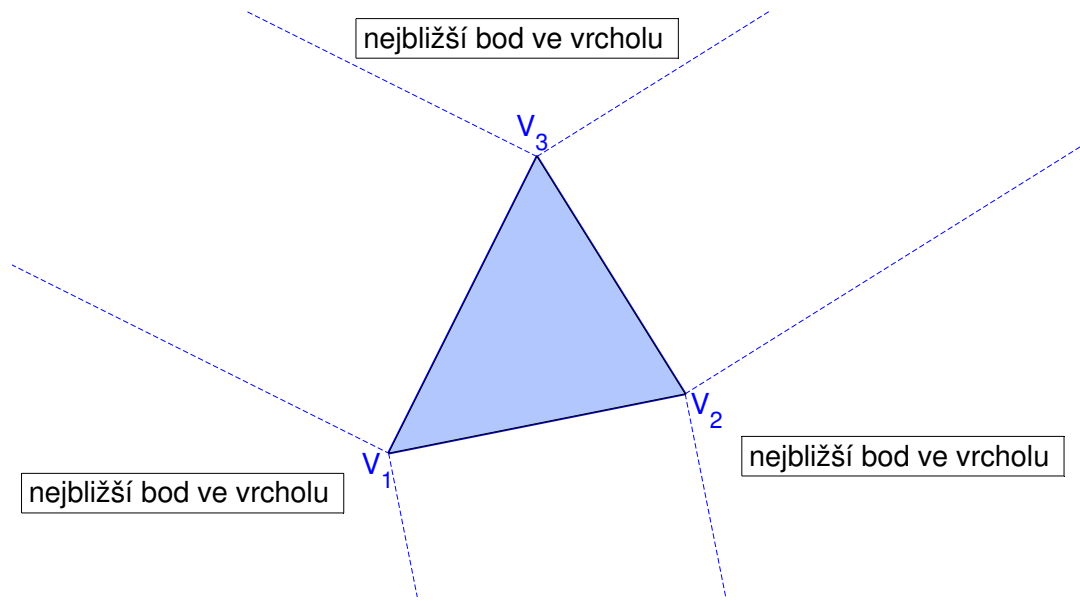
Kapitola 4

Způsoby vracení bodů opustivších doménu

Při tvorbě návrhů (při použití nástroje `Distmesh`) stejně jako při hodnocení kvality návrhů (například evoluční strategií pro vyhodnocení kritéria `miniMax`) dochází k situacím, kdy je potřeba posunout body, které z nějakého důvodu opustily návrhovou doménu, zpět do ní, respektive na její povrch. V regulárním návrhovém prostoru se jedná o triviální úkon. V omezeném návrhovém prostoru to představuje problém, k jehož řešení lze přistupovat mnoha způsoby. Tři z nich jsou uvedeny v této kapitole.



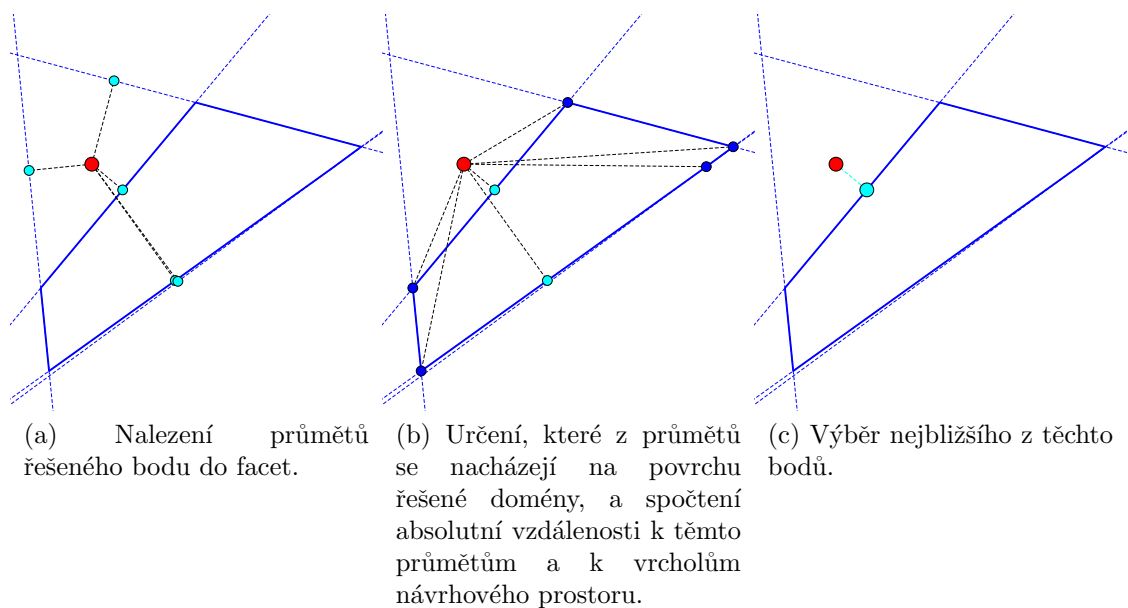
Obrázek 4.1: Posun bodů - kolmý průmět.



Obrázek 4.2: Posun bodů - nejbližší bod ve vrcholu.

4.1 Kolmý průmět na facetu

Nejpřesněji lze problém řešit analyticky - najít nejbližší bod návrhového prostoru. Tedy nejbližší z kolmých průmětů řešeného bodu (bodu ležícího mimo návrhový prostor) do všech facet návrhové domény; n označuje dimenzi návrhového prostoru. Kolmý průmět hledáme stejně jako ve 2D: pomocí vektorového součinu nalezneme vektor kolmý na facetu, dále přímku procházející řešeným bodem s nalezeným vektorem coby svým normálovým



(a) Nalezení průmětů řešeného bodu do facet.

(b) Určení, které z průmětů se nacházejí na povrchu řešené domény, a spočtení absolutní vzdálenosti k těmto průmětům a k vrcholům návrhového prostoru.

(c) Výběr nejbližšího z těchto bodů.

Obrázek 4.3: Posun bodů - kolmý průmět - řešení na celé doméně.

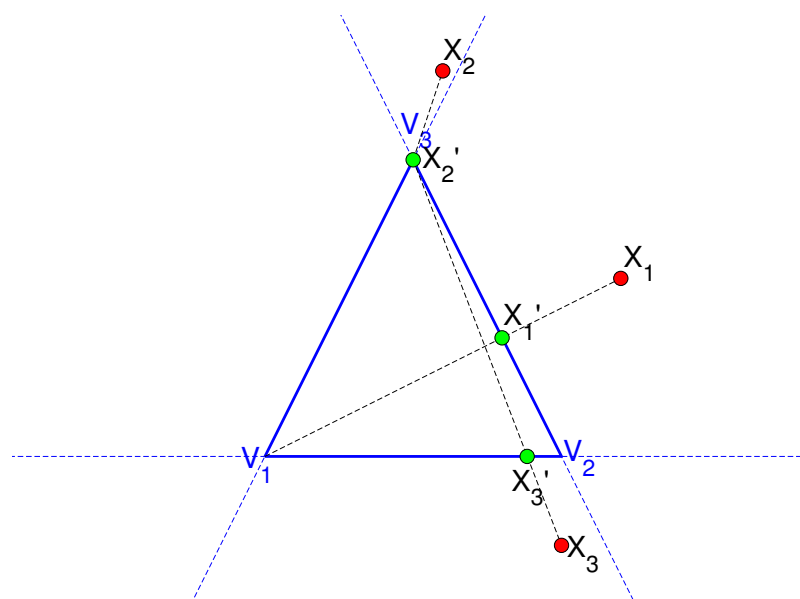
a kolmý průmět řešeného bodu je poté průsečíkem této přímky s facetou. Obrázek 4.1 ukazuje několik příkladů ve 2D.

Je tedy nutné nalézt průmět řešeného bodu do všech facet návrhové domény a vybrat ten nejbližší. Výše zmíněný postup hledání kolmého průmětu ovšem nerozliší, zda nalezený průmět leží na facetě návrhové domény nebo „až na jejím prodloužení“. Je tedy nutné nejprve množinu nalezených kolmých průmětů zredukovat na ty, které leží skutečně na povrchu návrhového prostoru (více o tomto rozlišení v Příloze D).

Nyní je ještě třeba ošetřit případ, kdy bod leží v některé z oblastí znázorněných na Obrázku 4.2. Pro body z těchto oblastí je nejbližším bodem domény její vrchol. Všechny vrcholy návrhového prostoru tedy přidáme k množině kolmých průmětů ležících na jejím povrchu, pro tyto body vypočteme absolutní vzdálenost k řešenému bodu a do nejbližšího z nich posuneme řešený bod. Postup ilustruje Obrázek 4.3.

4.2 Nulování záporných plošných souřadnic

Jiný způsob vrácení bodu zpět na povrch domény využívá plošných souřadnic (viz Přílohu D). Máme-li daný simplex a bod, můžou nastat v zásadě dvě situace: a) bod leží uvnitř daného simplexu - v tom případě jsou všechny plošné souřadnice tohoto bodu vůči tomuto simplexu kladné; b) bod leží mimo daný simplex - minimálně jedna (maximálně všechny kromě jedné) plošná souřadnice je záporná. Posun bodu na povrch simplexu je pak následující: záporné plošné souřadnice položíme rovny 0 a zbylé (kladné) plošné souřadnice změníme v jejich původním vzájemném poměru tak, aby celkový součet



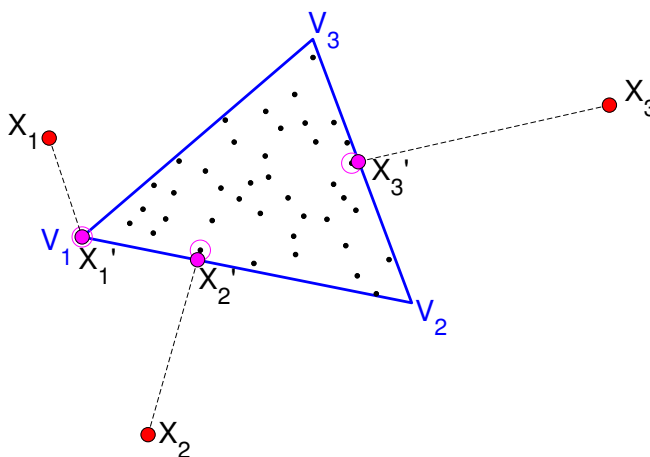
Obrázek 4.4: Posun bodů - nulování záporné plošné souřadnice.

plošných souřadnic zůstal roven 1. Posun bodů na povrch simplexu touto metodou ukazuje Obrázek 4.4.

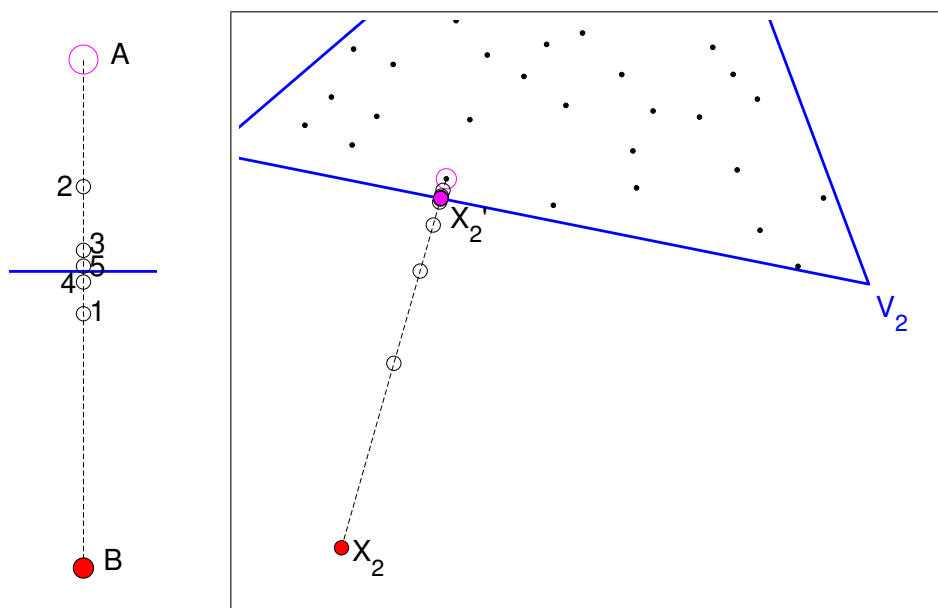
Problém s tímto přístupem nastává v případě, kdy neřešíme jen posun na povrch simplexu, nýbrž na povrch nepravidelné domény. Doménu v takovém případě můžeme rozdělit na jednotlivé simplexu pomocí Delaunayovy triangulace (viz Přílohu C), poté ovšem nejsme schopni určit, vzhledem ke kterému ze simplexů této triangulace provést onen posun pomocí úpravy plošných souřadnic. Z číselných hodnot plošných souřadnic vztahených k jednotlivým simplexům to zjistit nelze, neboť ty jsou v jistém smyslu relativní. Je tedy nutné provést posun řešeného bodu na povrch všech simplexů triangulace výše popsáním způsobem a následně spočítat absolutní vzdálenosti k těmto bodům a vybrat ten nejbližší.

4.3 Pomocná sada bodů uvnitř domény

Další možností, jak posunout bod zpět do řešené domény, je využití pomocné sady bodů [Lepš, 2009]. Tu vytvoříme uvnitř návrhového prostoru některým z rychlých algoritmů. Následně je pro body, které chceme posunovat, nalezen nejbližší z pomocné sady bodů (do této sady jsou přidány rovněž všechny vrcholy návrhového prostoru). Pomocí metody „půlení intervalu“ (viz Obrázek 4.6) nalezneme bod na hranici řešené domény a do tohoto bodu přesuneme bod z vnějšku domény. Je třeba zmínit, že nalezený bod neleží přesně na hranici domény, nýbrž uvnitř řešené domény libovolně (ovlivněno počtem iterací při aplikaci metody půlení intervalu) blízko jejímu povrchu. Příklad je znázorněn



Obrázek 4.5: Posun bodů - pomocná sada.



Obrázek 4.6: Použití metody půlení intervalu. Hledáme bod mezi dvěma zadanými body (které leží každý na opačné straně zadané hranice) takový, aby ležel co nejbližně zvolené hranici a navíc na dané straně této hranice. V prvním kroku nalezneme bod uprostřed mezi původními body; zjistíme, na které straně hranice nový bod leží; použijeme jej v dalším kroku místo předchozího bodu na této straně hranice.

na Obrázku 4.5.

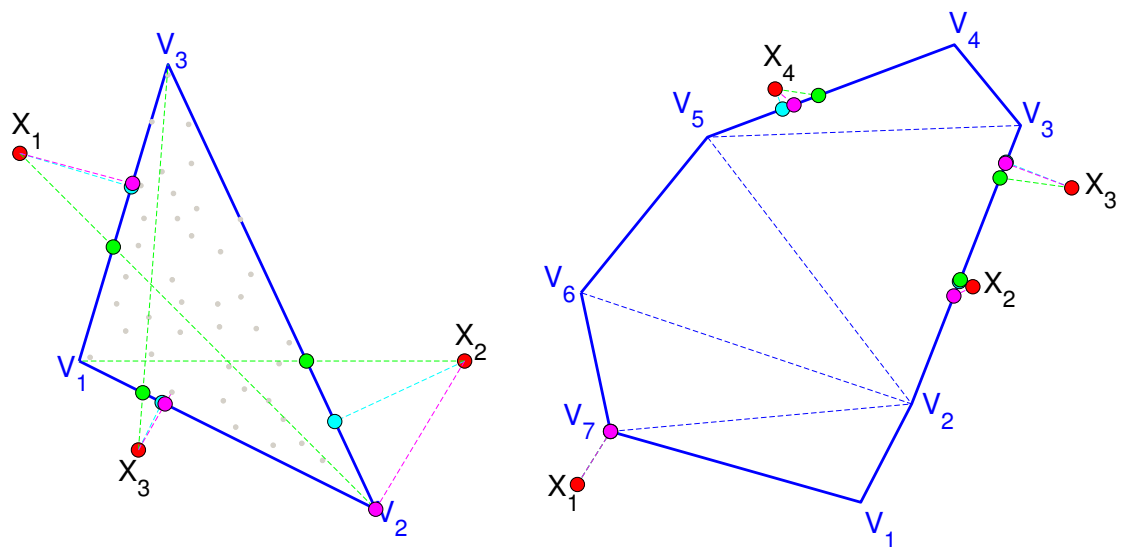
Pokud je nejbližším bodem z pomocné sady vrchol návrhového prostoru, neprovádíme již půlení intervalu, nýbrž řešený bod posuneme rovnou do vrcholu domény. Pro zrychlení je rovněž nejprve kontrolováno, jak daleko je nejbližší pomocný bod od hranice domény (jak hluboko je pod jejím povrchem): pokud by se bod z pomocné sady při posunu o tisícinu (možno zvolit libovolně) vzdálenosti k řešenému bodu dostal již mimo návrhový prostor, je řešený bod posunut přímo do nejbližšího pomocného bodu.

K omezení vzniku duplikovaných bodů (obzvláště ve vrcholech) bylo rovněž zavedeno opatření, které vyškrtává již použitý pomocný bod z výběru.

Tato metoda vracení bodů je jednoduchá a přináší další prvek náhodnosti do algoritmů, v nichž je aplikována. Je však časově náročná z důvodu nutnosti výpočtu velkého množství vzájemných vzdáleností. Může též docházet k posunu „nepřirozeným“ směrem (viz bod $X_1 \rightarrow X'_1$ na Obrázku 4.5).

4.4 Porovnání

Porovnání uvedených řešení posunu bodů na povrch řešené domény ve 2D ukazuje Obrázek 4.7. Vidíme, že výsledek může být značně rozdílný. To však nemusí být na škodu v případě, kdy si určitou náhodnost a nepravidelnost v algoritmu přejeme. Kritériem



Obrázek 4.7: Porovnání způsobů vracení bodů do simplexu (vlevo) a do nepravidelné domény (vpravo). Legenda: azurové body - metoda kolmého průmětu; zelené body - metoda nulování záporných plošných souřadnic; purpurové body - metoda pomocné sady bodů.

výběru metody je navíc (a především) její časová náročnost. Srovnání časové náročnosti posunu bodu zpět do nepravidelné domény ve 2D (příklad odpovídá Obrázku 4.7 vpravo) a 5D (70 vrcholů, 970 4-dimenzionálních facet, 2909 simplexů triangulace) ukazuje Tabulka 4.1. Je však nutno připomenout, že časová náročnost metody s pomocnou sadou bodů je určena právě počtem bodů v této sadě a rovněž zvoleným počtem iterací v algoritmu půlení intervalu. Výsledek naznačuje dominanci metody nulování záporných plošných souřadnic, zatímco metoda využívající kolmého průmětu bodu do facet domény ztlačně ztrácí.

	„kolmý průmět“	„nulování“	„pomocná sada“
2D příklad	0,0032 s	0,0007 s	0,0087 s
5D příklad	66,3531 s	0,3350 s	5,2703 s

Tabulka 4.1: Časová náročnost způsobů posunu bodů.

Kapitola 5

Příklady aplikace metod

Aplikací, na něž je možné uведенé přístupy aplikovat, je celá škála a prostupují nejrůznějšími odvětvími. Typickým a častým příkladem experimentu, v němž jsou jednotlivé vstupní parametry limitovány omezujícími podmínkami, je navrhování směsi. S tím se setkávají chemici ve svých laboratořích, stejně tak tvůrci receptur v jakémkoli podniku vyrábějícím materiály. Jiným příkladem experimentu s omezením může být navrhování nosníku, kde je zadána podmínka vzájemného poměru výšky a šířky apod.

Ve zde uváděných metodách jsme vždy uvažovali úlohu zadanou pomocí vrcholů omezené domény. Tak to ale nemusí být vždy, úloha může být zadána jen omezujícími podmínkami a vrcholy návrhové domény je pak třeba najít.

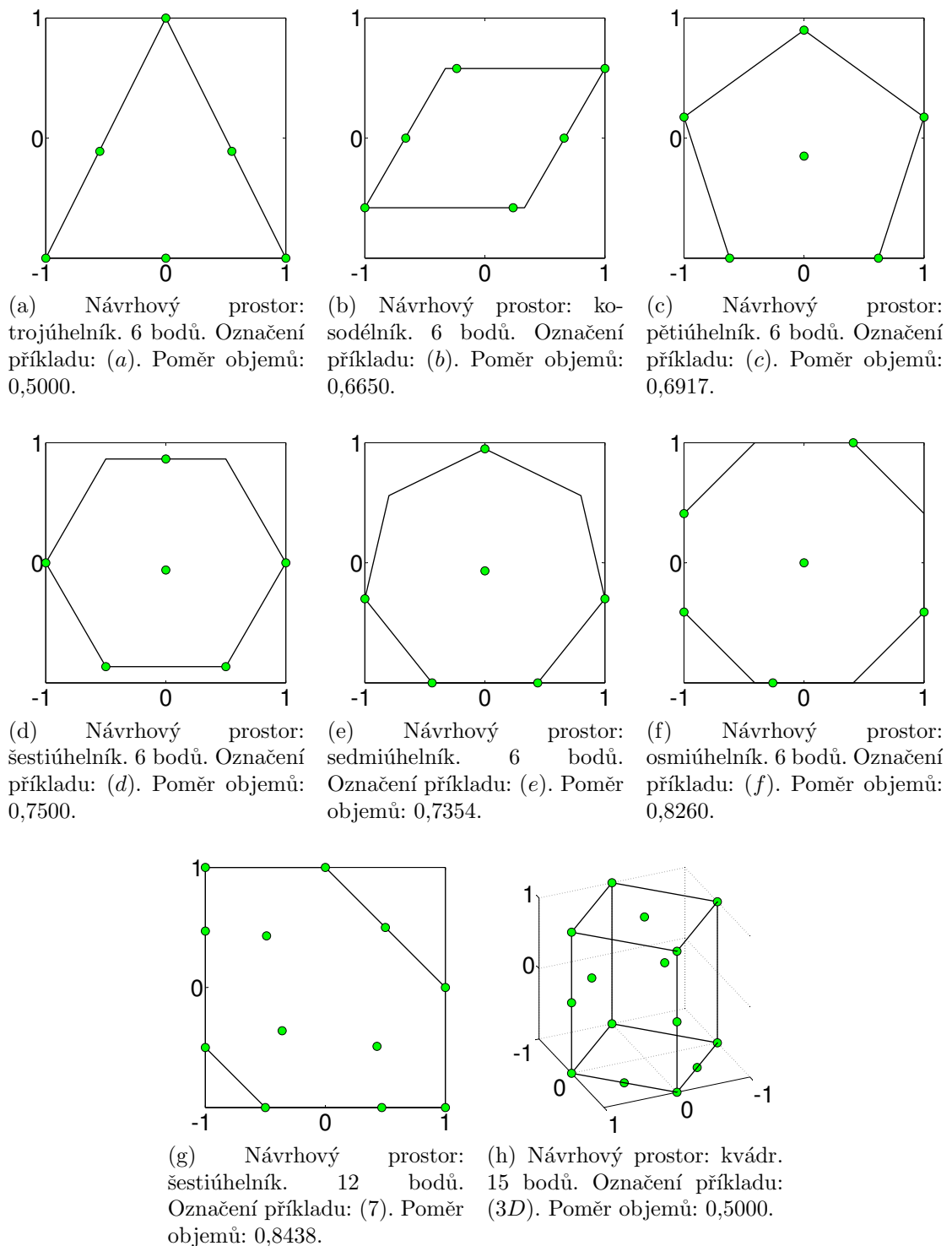
Je rovněž třeba rozdílný přístup k problému, pokud konkrétní omezující podmínky známe, nebo neznáme. Například určení, zda bod leží uvnitř nebo vně návrhového prostoru je snadné v prvním případě, kdy jeho souřadnice stačí dosadit do jednotlivých omezujících podmínek a zjistit, zda jsou splněny.

Následuje několik konkrétních příkladů, na nichž otestujeme uvedené metody pro tvorbu návrhů experimentů.

5.1 Příklady návrhů s malým počtem bodů ve 2D a ve 3D

Nejprve budeme metody aplikovat na sadě příkladů zobrazených na Obrázku 5.1. Ty jsou převzaty z článku [Hofwing and Strömberg, 2010] odečtením souřadnic vrcholů. Neuvažujeme zde znalost konkrétních omezujících podmínek.

Poměry objemů daného návrhového prostoru a opsaného hyperkvádrů se pohybují v rozmezí 0,5 ((a) a $(3D)$) a 0,8438 ((7)). Návrhový prostor tedy vyplňuje značnou část opsaného hyperkvádrů; při použití metod tvořících body v opsaném hyperkvádrů tedy není nutné tvořit výrazně více bodů, než je požadováno v zadané omezené doméně.



Obrázek 5.1: Příklady návrhů s malým počtem bodů ve 2D a 3D. Na obrázcích jsou vyznačeny návrhové prostory a zelenými body referenční návrhy (získané metodou popsanou v [Hofwing and Strömberg, 2010]). Dále pak poměr objemu návrhové domény a opsaného obdélníku (ve 3D kvádru). Jednotky na osách jsou bezrozměrné, popisky os jsou vynechány.

5.2 Návrh směsi

Následující příklady spadají do skupiny experimentů nazývaných návrh směsi. Jak již bylo řečeno dříve, tento typ experimentu je dán omezující podmínkou danou Rovnicí 1.1. Suma zastoupení jednotlivých složek směsi je tedy pevně dána. Návrhovým prostorem je simplex nebo polytop vzniklý ze simplexu díky přítomnosti dalších omezujících podmínek. Zastoupení jednotlivých složek takového experimentu je pak dáno plošnými souřadnicemi návrhového bodu. Jelikož se však (na rozdíl od filmového průmyslu, který vykazuje tendenci přidávat jedno „D“ za druhým) snažíme snížit dimenzionalitu řešeného problému, tvoříme návrh experimentů nikoli v plošných, ale v kartézských souřadnicích. Těch je vždy o jednu méně než plošných, tedy než složek dané směsi. Transformace mezi souřadnými soustavami je uvedena v Příloze D.

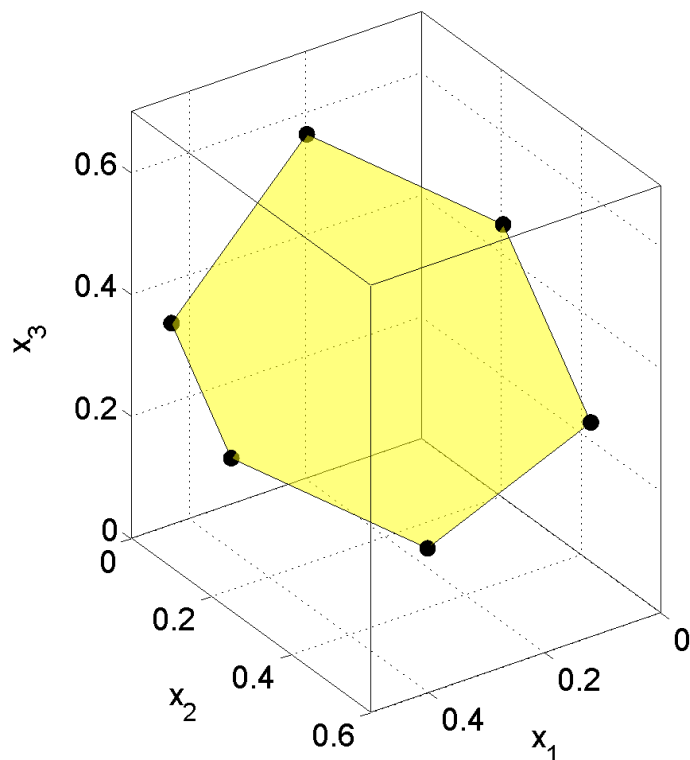
5.2.1 Trojsložková směs

První příklad návrhu směsi je převzat z referencí [Snee, 1979, Piepel, 1988]. Jedná se o trojsložkovou směs zadanou omezujícími podmínkami:

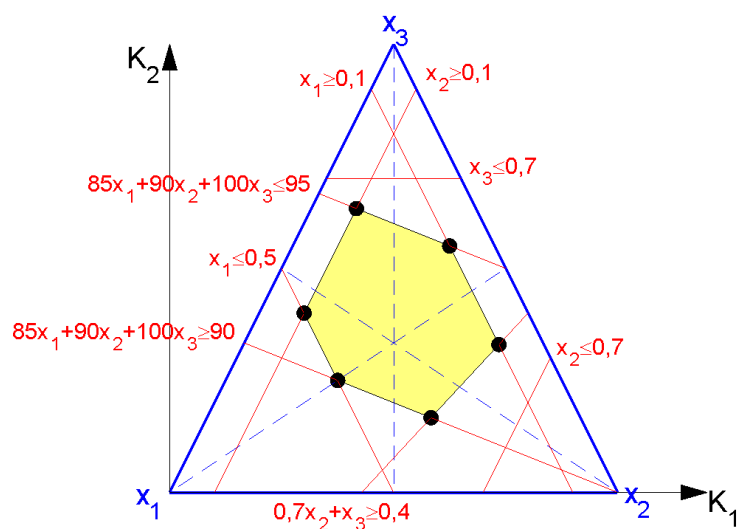
$$\begin{array}{rclclcl}
 x_1 & + & x_2 & + & x_3 & = & 1 \\
 x_1 & & & & & \geq & 0,1 \\
 x_1 & & & & & \leq & 0,5 \\
 & & x_2 & & & \geq & 0,1 \\
 & & x_2 & & & \leq & 0,7 \\
 & & & & x_3 & \leq & 0,7 \\
 85 \cdot x_1 & + & 90 \cdot x_2 & + & 100 \cdot x_3 & \geq & 90 \\
 85 \cdot x_1 & + & 90 \cdot x_2 & + & 100 \cdot x_3 & \leq & 95 \\
 0,7 \cdot x_1 & & & + & x_3 & \geq & 0,4
 \end{array}$$

Dříve, než můžeme začít tvořit návrh experimentů, je třeba najít vrcholy návrhové domény. To je možné provést pomocí simplexové metody, neboť se jedná o úlohu lineárního programování [Demel, 2011]. Hledáme krajní body množiny přípustných řešení. Celý postup „ručního“ řešení je uveden v Příloze F. Výsledkem je 6 vrcholů o třech souřadnicích. Ty po vynesení do 3D prostoru (Obrázek 5.2) tvoří ovšem pouze 2-dimenzionální objekt - část roviny. To je dáno právě onou součtovou podmínkou. Návrhový prostor jde převést do 2D a body popisovat stejnou trojicí souřadnic jako v prostoru. Ve 2D se však bude jednat o trojici souřadnic plošných. Pomocí transformace mezi plošnými a kartézskými souřadnicemi však můžeme body popisovat pouze dvěma kartézskými souřadnicemi (označeny K_1 a K_2), jak je znázorněno na Obrázku 5.3. Takto s body zacházíme v této práci.

Poměr objemů návrhové domény a jednoduchého opsaného obdélníku je v tomto případě roven 0,6236; i zde je tedy možné použít metodu *bounding boxu*.



Obrázek 5.2: Krajní body množiny přípustných řešení získané pomocí simplexové metody (Příloha F).



Obrázek 5.3: Krajní body množiny přípustných řešení v trojosé soustavě se znázorněním jednotlivých omezujících podmínek.

5.2.2 Šestisložková směs

Poslední příklad popisuje návrh šestisložkové směsi, konkrétně vysokopevnostního betonu. Je převzatý z referencí [Simon et al., 1997, Simon, 2003]. Je rovněž zadán pouze pomocí následujících omezujících podmínek:

Podmínka návrhu směsi	$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 1$
Voda (Water)	$0,16 \leq x_1 \leq 0,185$
Cement	$0,13 \leq x_2 \leq 0,15$
Křemičitý úlet (Silica fume)	$0,013 \leq x_3 \leq 0,027$
Plastifikátor HRWRA	$0,0046 \leq x_4 \leq 0,0074$
Hrubé kamenivo (Coarse aggregate)	$0,40 \leq x_5 \leq 0,4424$
Jemné kamenivo (Fine aggregate)	$0,25 \leq x_6 \leq 0,2924$

Hledat vrcholy návrhového prostoru příslušného těmto omezujícím podmínkám „ručně“ (jako u předchozího příkladu) by bylo značně zdlouhavé. Proto byl zvolen způsob založený na stejné metodě, ovšem pomocí funkce `linprog` z programového prostředí MATLAB. Ta ovšem není určena pro hledání krajních bodů množiny přípustných řešení, nýbrž pro hledání optimálního řešení při zadané cílové funkci (směru hledání). Hledání krajních bodů bylo tedy provedeno tak, že byl řešič spuštěn mnohokrát s různou cílovou funkcí (aby „nahlédl do mnoha různých směrů“). Tak bylo získáno velké množství krajních bodů. Po odstranění duplikátů je za množinu vrcholů návrhové domény považováno 51 bodů.

I zde uvádíme poměr objemů návrhové domény a opsaného hyperkvádru, ten je roven 0,1630. Návrh v opsaném hyperkvádru tedy vytváříme se 7násobkem bodů, než je cílem v omezeném prostoru.

V obou příkladech návrhu směsi bylo pro porovnání metod tvorby experimentů tvořeno 10 a 100 návrhových bodů.

Kapitola 6

Implementace

Veškeré metody popsané v této práci byly implementovány v programovém prostředí MATLAB R2010a. Kromě běžných funkcí byly používány funkce `lhsdesign` pro tvorbu neoptimalizovaného LHS návrhu v regulárním prostoru, funkce `delaunayn` pro sestavení Delaunayovu triangulace, funkce `voronoin` pro konstrukci Voronoiova diagramu nebo funkce `linprog` pro hledání krajních bodů množiny přípustných řešení. Nástroj `Distmesh` byl stažen z webové stránky <http://persson.berkeley.edu/distmesh/distmesh.zip>.

Kapitola 7

Výsledky

Následující sekce přináší výsledky a srovnání metod představených v této práci.

7.1 Porovnání metod vyhodnocení kritéria miniMax

Na všech příkladech, v nichž bylo možno použít exaktní metodu pro vyhodnocení miniMax kritéria (všechny 2D a 3D příklady, na příklad se šestisložkovou směsí již nepostačovaly výpočetní prostředky použitého počítače) tato metoda poskytla výsledek následně potvrzený evoluční strategií. Byla rovněž testována na řadě dalších dvou- až čtyřdimenzionálních příkladů s pozitivním výsledkem. Lze tedy konstatovat, že se exaktní metoda pro vyhodnocení miniMax kritéria využívající zrcadlení návrhových bodů a Voronoiova diagramu zdá použitelná jak v regulárním, tak v konvexním omezeném návrhovém prostoru.

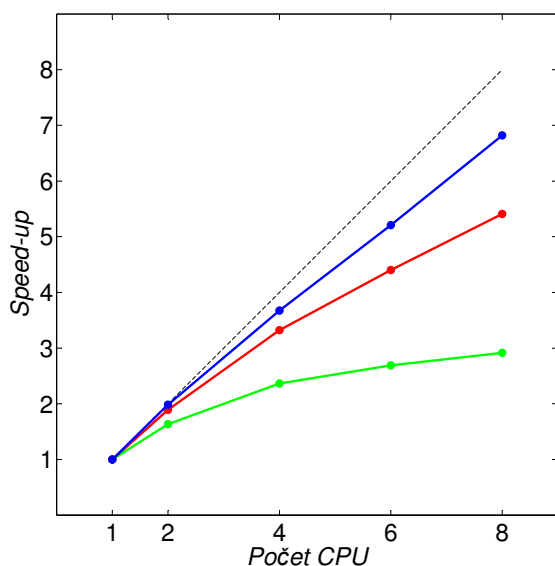
Ve vyšších dimenzích je třeba spokojit se s odhadem hodnoty kritéria miniMax získaným pomocí evoluční strategie. V omezeném návrhovém prostoru je její sériová verze v současné úrovni implementace v praxi nepoužitelná. V rámci testování byla použita na jeden z návrhů experimentů vytvořených pro příklad šestisložkové směsi z Kapitoly 5. Bylo tedy možné využít znalosti konkrétních omezujících podmínek a zjištění, zda body

metoda	vracení bodů	čas [s]
sériová verze	pomocná sada bodů	1308 s
sériová verze	nulování záporných pl. s.	7225 s
sériová verze	kolmý průmět	-
paralelní verze (1 CPU)	nulování záporných pl. s.	35 s

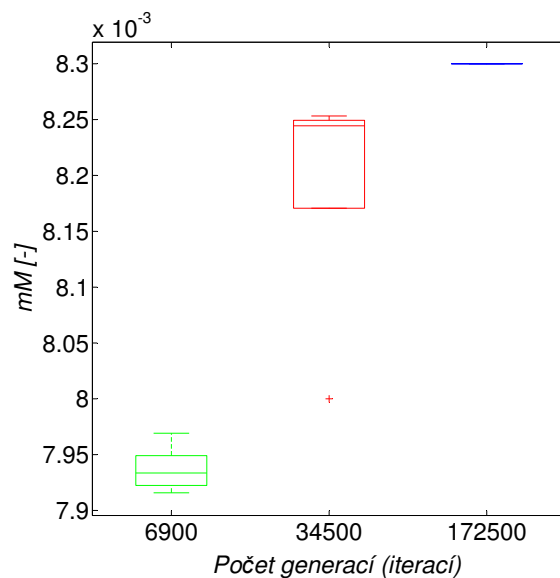
Tabulka 7.1: Časová náročnost verzí evoluční strategie pro vyčíslení mM . Testováno na návrhu se 100 body v návrhovém prostoru šestisložkové směsi popsané v Kapitole 5. Použito bylo 50 chromozomů a celkový počet 6900 generací.

leží vně či uvnitř návrhové domény, neprovádět „výpočetně náročnými způsoby“, nýbrž pouze ověřením (ne)splnění těchto podmínek. Přesto byl čas potřebný k vyhodnocení nesrovnatelně vyšší než u verze paralelní, byť s použitím jediné výpočetní jednotky (CPU), jak uvádí Tabulka 7.1.

Výrazně lepších výsledků dosahuje paralelní verze evoluční strategie pro vyhodnocení miniMax kritéria. Pro vracení bodů, jež mutací opustily jednotlivé subdomény (simplexy ztriangulované domény), byla použita metoda nulování záporných plošných souřadnic popsaná v Sekci 4.2. Obrázek 7.1a zobrazuje speed-up při použití 2, 4, 6 a 8 výpočetních jednotek. Za výchozí vyhodnocení (bod [1,1]) není z výše popsaných důvodů považována sériová verze výpočtu, nýbrž verze výpočtu v subdoménách s použitím jedné výpočetní jednotky. Jednotlivé křivky odpovídají různým volbám počtu generací ve výpočtu. Vidíme, že čím „větší“ je úloha, tím výraznějšího zrychlení dosahujeme pomocí použití více výpočetních jednotek. To je dáno především tím, že čas nutný ke zparalelizování úlohy hraje u úlohy s nízkým počtem generací značnou roli, zatímco u úlohy celkově déle trvající se ztrácí. Obrázek 7.1b zobrazuje boxplot odhadů hodnoty miniMax získaných pomocí paralelní verze s různým počtem generací. Vidíme, že je splněn předpoklad, že čím více generací je použito, tím lepšího (vyššího, tedy skutečnosti bližšího) výsledku dosáhneme;



(a) Speed-up paralelní verze evoluční strategie.



(b) Boxplot odhadů hodnot kritéria miniMax pomocí paralelní verze evoluční strategie.

Obrázek 7.1: Výsledky paralelní evoluční strategie pro vyčíslení mM . Při testování bylo použito 50 chromozomů. Zelené objekty odpovídají celkovému počtu 6900 generací ve výpočtu (20 generací v každé subdoméně), červené objekty 34500 generací ve výpočtu (100 generací v každé subdoméně), modré objekty 172500 generací ve výpočtu (500 generací v každé subdoméně).

samozřejmě za cenu delšího času výpočtu.

Lze tedy konstatovat, že k vyhodnocení kritéria miniMax v omezených prostorech je vhodné použití exaktní metody v nízkých dimenzích a paralelní verze metody založené na algoritmu evoluční strategie (s použitím co nejvyššího počtu výpočetních jednotek) v dimenzích vyšších. Je třeba pamatovat na to, že pomocí evoluční strategie získáme pouze odhad hodnoty kritéria miniMax, skutečná hodnota může být vždy vyšší.

7.2 Porovnání metod tvorby návrhů experimentů

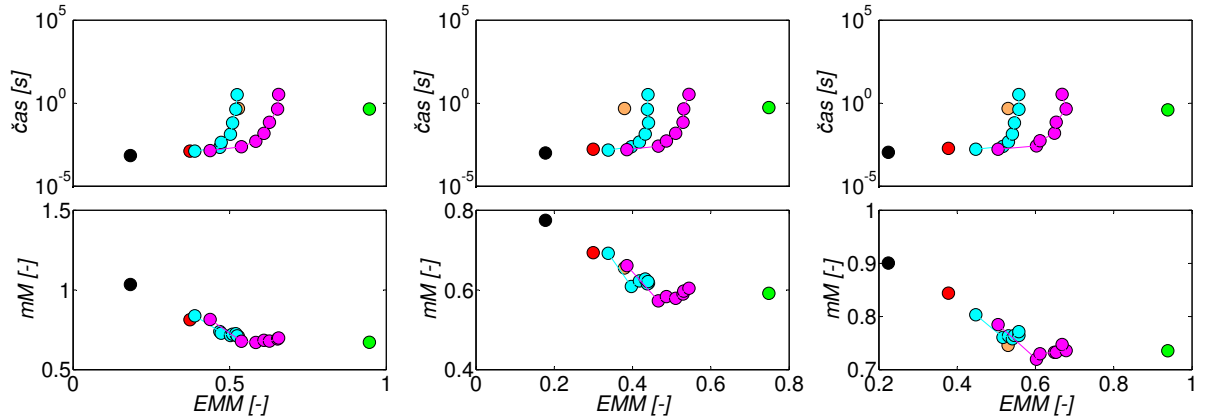
Výsledky tvorby návrhů experimentů v návrhových prostorech odpovídajících příkladům uvedeným v Kapitole 5 jsou uvedeny na Obrázcích 7.2 a 7.3. Jedná se o statistiku ze 100, respektive 10 (u časově náročnějších metod) spuštění. Metody porovnáváme z hlediska tří kritérií - metrik *EMM* a *mM* a časové náročnosti. Kompletní informaci o výsledcích by tedy přinášel trojrozměrný graf. Pro lepší orientaci však byly pro vizualizaci výsledků zvoleny dva grafy dvojrozměrné. Spodní část jednotlivých obrázků ukazuje kvalitu vzniklých návrhů - tedy porovnání kritérií *EMM* a *mM*. Kritérium Maximin se snažíme maximalizovat, zatímco u kritéria miniMax je žádoucí co nejnižší hodnota - nejkvalitnější návrhy z pohledu rovnoměrnosti pokrytí se tedy nalézají v pravém dolním rohu tohoto grafu. Horní část obrázků potom přináší informace o časové náročnosti jednotlivých metod tvorby. Na vodorovné ose je, stejně jako na spodním grafu, vynesena hodnota *EMM*, i u tohoto grafu tedy leží nejvýhodnější návrhy v pravém dolním rohu. Legenda k obrázkům je uvedena níže:

- generátor náhodných bodů (Sekce 3.1)
- neoptimalizovaný LHS návrh v opsaném hyperkvádru (Sekce 3.2)
- optimalizovaný LHS návrh v opsaném hyperkvádru (Sekce 3.2)
- ubírání (Sekce 3.3.1) ($2^2 \cdot np \rightarrow np$, $2^3 \cdot np \rightarrow np$, $2^4 \cdot np \rightarrow np$, ...) b.
- ubírání_{NEW} (Sekce 3.3.2) ($2^2 \cdot np \rightarrow np$, $2^3 \cdot np \rightarrow np$, $2^4 \cdot np \rightarrow np$, ...) b.
- nástroj Distmesh I (Sekce 3.4)
- nástroj Distmesh II (Sekce 3.4)

pozn.: u metod s odebráním bodů z přehuštěného návrhu bylo vždy vyzkoušeno několik úrovní přehuštění výchozího návrhu, označení $(x \rightarrow y)$ znamená y bodů v návrhu zbylých po odebrání z x původně vytvořených

Výsledky jasně naznačují korelaci mezi kritérii Maximin a miniMax. Dá se tedy říci, že optimalizací jednoho z nich se zlepšuje kvalita návrhu i z pohledu druhého kritéria.

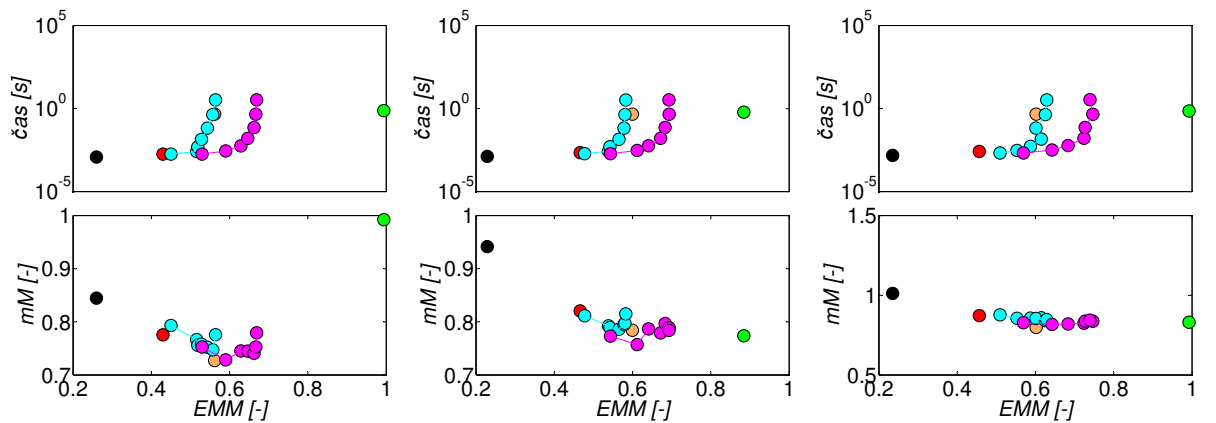
Ve 2D jednoznačně dominuje tvorba návrhu experimentu pomocí nástroje Distmesh, ačkoli je jeho časová náročnost vyšší než u ostatních metod, což je dáno opakovanou triangulací během běhu algoritmu. Již ve 3D se však jeho použití nevyplatí, neboť jeho



(a) Návrhový prostor: trojúhelník. 6 bodů. Označení příkladu: (a).

(b) Návrhový prostor: kosodélník. 6 bodů. Označení příkladu: (b).

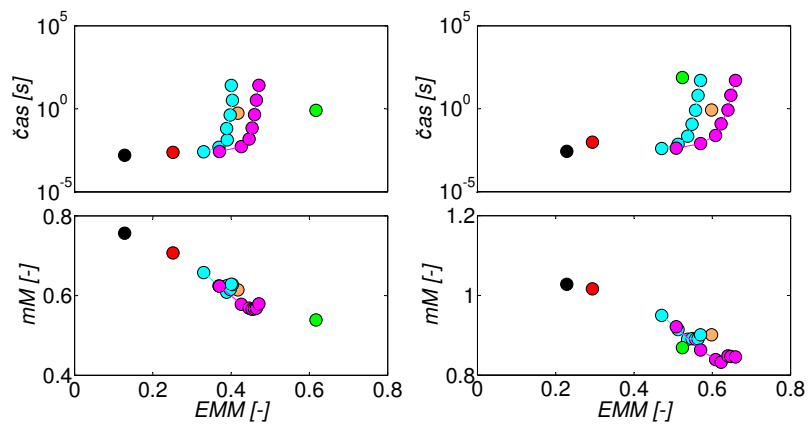
(c) Návrhový prostor: pětúhelník. 6 bodů. Označení příkladu: (c).



(d) Návrhový prostor: šestiúhelník. 6 bodů. Označení příkladu: (d).

(e) Návrhový prostor: sedmiúhelník. 6 bodů. Označení příkladu: (e).

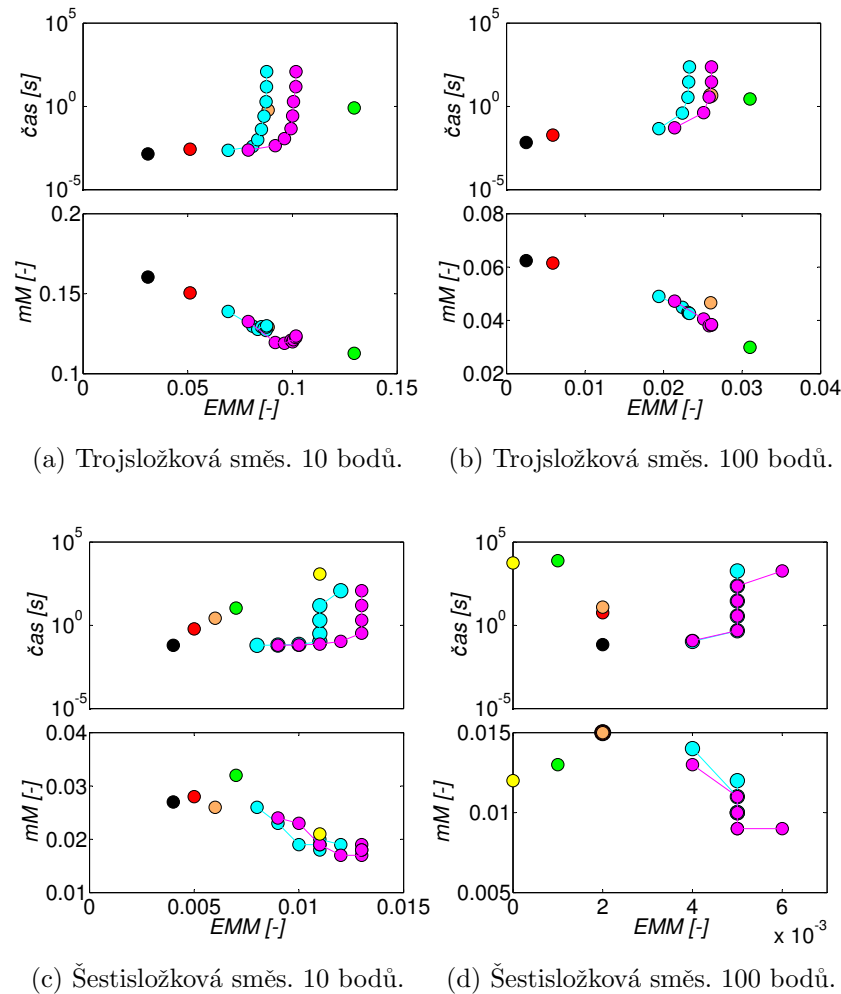
(f) Návrhový prostor: osmiúhelník. 6 bodů. Označení příkladu: (f).



(g) Návrhový prostor: šestiúhelník. 12 bodů. Označení příkladu: (7).

(h) Návrhový prostor: kvádr. 15 bodů. Označení příkladu: (3D).

Obrázek 7.2: Výsledky metod tvorby na 2D a 3D příkladech.



Obrázek 7.3: Výsledky metod tvorby na příkladech návrhů směsí.

výpočetní nároky rostou, zatímco kvalita jím tvořených návrhů v porovnání s jinými metodami klesá. V případě šestisložkové směsi byl dokonce metodou vytvořen návrh s duplikovanými body.

Generátor náhodných bodů v jednotlivých simplexech ztriangulované omezené domény byl ve všech příkladech nejrychlejší metodou, kvalita jím vytvořených návrhů však byla až na výjimky nejnižší.

Lepších výsledků dosahuje metoda vybírání bodů z LHS návrhu vytvořeném v hyperkvádru omezené doméně. Výsledky potvrzují, že použití optimalizovaného LHS návrhu místo neoptimalizovaného zajišťuje lepší výsledek i v omezené doméně. Zůstává však otázka počtu bodů, jež tvořit v omezeném regulárním prostoru. Zde jsme vycházeli z pomocného výpočtu poměru objemů. Takto jsme se však přesně do požadovaného počtu bodů uvnitř omezené domény „trefili“ průměrně jednou z 29 pokusů (v jednom z příkladů dokonce jednou z 263 pokusů). Jelikož stejně není možné o návrhu takto vytvořeném v omezené doméně hovořit jako o návrhu LHS, nabízí se možnost zkombinování

této metody s metodou odebrání bodů z přehuštěného návrhu. Přebytečné body v omezené doméně se jednoduše odeberou pomocí této metody.

Samotná metoda odebrání bodů ze záměrně přehuštěného návrhu dosahuje velmi dobrých výsledků. V posledním příkladu je Pareto-front dokonce tvořen právě jen touto metodou. Druhá varianta, kde je pozornost věnována konkrétní volbě odebraného bodu z aktuálně vzájemně si nejbližší dvojice bodů, dosahuje lepších výsledků při naprosto srovnatelném čase. Uživatel si může snadno zvolit míru přehuštění výchozího návrhu a tak ovlivnit jak čas, tak kvalitu výsledného návrhu. Je patrná pozitivní korelace mezi mírou přehuštění výchozího návrhu a kvalitou výsledného návrhu, ovšem jen do jisté míry. Ve zde uváděných příkladech nemá význam tvořit ve výchozím návrhu více než 30násobek požadovaných bodů.

Na základě předložených výsledků můžeme jako nejvhodnější metody pro tvorbu návrhů experimentů v omezených návrhových prostorech v nízké dimenzi označit použití nástroje `Distmesh`. Ve vyšších dimenzích se potom nejvíce osvědčilo vytvoření přehuštěného návrhu a následné heuristické odebrání bodů až k dosažení jejich požadovaného počtu.

Kapitola 8

Závěr

Návrh experimentů, jímž se zabývala tato diplomová práce, slouží k naplánování konkrétních provedení experimentů v případech, kdy je možno volit hodnoty jednotlivých vstupních parametrů. Cílem je pokrýt zvoleným počtem návrhových bodů (jejichž souřadnice odpovídají kombinacím konkrétních hodnot vstupních parametrů) zadaný prostor, jak nejrovnoměrněji je to možné, abychom získali co nejlepší představu o chování systému, který je na těchto vstupních parametrech závislý. Návrh experimentů je tak nepostradatelným nástrojem v mnoha oblastech vědy, výzkumu i průmyslu. Aplikací kvalitního návrhu je snad možno zmírnit obavu mezi řádky vyplývající z citátu, jímž byla tato práce uvedena.

Práce se věnovala návrhům experimentů v omezených návrhových prostorech, tedy v situacích, kdy jsou zadány omezující podmínky a jednotlivé vstupní parametry jsou na sobě závislé. Bylo představeno několik metod pro tvorbu návrhů v odpovídajících návrhových prostorech a návrhy těmito metodami vytvořené byly hodnoceny dvěma kritérii. Rovněž byla hodnocena časová náročnost těchto metod.

V další práci se hodláme zaměřit na zefektivnění některých z metod tvorby návrhů i jejich hodnocení. Například zrychlení vyčíslení odhadu kritéria miniMax pomocí evoluční strategie s jinou formou mutace chromozomů.

Samo kritérium miniMax je námětem pro další práci. Střed největší prázdné hyperkoule se jeví jako ideální kandidát při adaptivním samplování, tedy v situacích, kdy je třeba přidávat další body do návrhu experimentů. Samotná vzdálenost od již umístěných návrhových bodů navíc nemusí být jediným kritériem pro umístění dalšího bodu. Například v oboru stochastické spolehlivosti konstrukcí může být dalším kritériem vzdálenost k hranici poruchy (*limit state function*), tedy k rozmezí mezi oblastí bezpečnou a oblastí poruchy. V tomto prostoru je zpravidla žádoucí vyšší zahuštění návrhu, jak je patrné i na předběžných výsledcích v [Myšáková and Lepš, 2013b, Pospíšilová and Lepš, 2013] nebo v [Myšáková et al., 2013, Pospíšilová et al., 2013].

Literatura

- [Sim, 2002] (2002). Hypercube. www stránky: <http://en.wikipedia.org/wiki/Simplex/>. Poslední změna 3. listopadu 2013.
- [Hyp, 2002] (2002). Hypercube. www stránky: <http://en.wikipedia.org/wiki/Hypercube/>. Poslední změna 31. října 2013.
- [JMP, 2005] (2005). *JMP design of experiments, release 6*. SAS Institute Inc., Cary, NC, USA.
- [Auffray et al., 2012] Auffray, Y., Barbillon, P., and Marin, J.-M. (2012). Maximin design on non hypercube domains and kernel interpolation. *Statistics and Computing*, 22(3):703–712.
- [Bäck and Schwefel, 1995] Bäck, T. and Schwefel, H.-P. (1995). Evolution Strategies I: Variants and their computational implementation. In Periaux and Winter, editors, *Genetic Algorithms in Engineering and Computer Science*, chapter 6, pages 111–126. John Wiley & Sons Ltd.
- [Chan, 2000] Chan, L.-Y. (2000). Optimal designs for experiments with mixtures: A survey. *Communications in Statistics. Theory and Methods*, 29(9-10):2281–2312.
- [Chen and Holst, 2011] Chen, L. and Holst, M. (2011). Efficient mesh optimization schemes based on Optimal Delaunay Triangulations. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):967–984.
- [Cioppa and Lucas, 2007] Cioppa, T. M. and Lucas, T. (2007). Efficient nearly orthogonal and space-filling latin hypercubes. *Technometrics*, 49(1):45–55.
- [Cornell, 1973] Cornell, J. A. (1973). Experiments with Mixtures: A Review. *Technometrics*, 15(3):437–455.
- [Cornell, 1979] Cornell, J. A. (1979). Experiments with Mixtures: An Update and Bibliography. *Technometrics*, 21(1):95–106.
- [Crombecq et al., 2009] Crombecq, K., Couckuyt, I., Gorissen, D., and Dhaene, T. (2009). Space-filling sequential design strategies for adaptive surrogate modelling. In Topping, B. H. V. and Tsompanakis, Y., editors, *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK.
- [Cruise, 1966] Cruise, D. R. (1966). Plotting the composition of mixtures on simplex coordinates. *Journal of Chemical Education*, 43(1):30.

- [Demel, 2011] Demel, J. (2011). *Operační výzkum*. Skriptum, Katedra aplikované informatiky, Fakulta stavební, České vysoké učení technické v Praze.
- [Devadoss and O'Rourke, 2011] Devadoss, S. and O'Rourke, J. (2011). *Discrete and Computational Geometry*. Princeton University Press.
- [Devroye, 1986] Devroye, L. (1986). *Non-uniform random variate generation*. Springer-Verlag.
- [Dickerson and Eppstein, 1995] Dickerson, M. and Eppstein, D. (1995). Algorithms for proximity problems in higher dimensions. *Comput. Geom.*, 5:277–291.
- [Draguljic et al., 2012] Draguljic, D., Dean, A. M., and Santner, T. J. (2012). Non-collapsing space-filling designs for bounded nonrectangular regions. *Technometrics*, 54(2):169–178.
- [Fang et al., 2006] Fang, K., Li, R.-Z., and Sudjianto, A. (2006). *Design And Modeling for Computer Experiments*. Computer Science & Data Analysis. CRC PressINC.
- [Fang and Yang, 2000] Fang, K.-T. and Yang, Z.-H. (2000). On uniform design of experiments with restricted mixtures and generation of uniform distribution on some domains. *Statistics & Probability Letters*, 46(2):113–120.
- [Fauvel et al., 1993] Fauvel, J., Flood, R., and Wilson, R. (1993). *Möbius and his band: mathematics and astronomy in nineteenth-century Germany*. Oxford University Press.
- [Grosso et al., 2009] Grosso, A., Jamali, A. R. M. J. U., and Locatelli, M. (2009). Finding maximin latin hypercube designs by iterated local search heuristics. *European Journal of Operational Research*, 197(2):541–547.
- [Hofwing and Strömberg, 2010] Hofwing, M. and Strömberg, N. (2010). D-optimality of non-regular design spaces by using a bayesian modification and a hybrid method. *Structural and multidisciplinary optimization (Print)*, 42(1):73–88.
- [Janouchová and Kučerová, 2013] Janouchová, E. and Kučerová, A. (2013). Competitive comparison of optimal designs of experiments for sampling-based sensitivity analysis. *Computers & Structures*, 124(0):47 – 60.
- [Joe and Wang, 1993] Joe, B. and Wang, C. (1993). Duality of constrained Voronoi diagrams and Delaunay triangulations. *Algorithmica*, 9(2):142–155.
- [Johnson et al., 1990] Johnson, M. E., Moore, L. M., and Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148.
- [Khuri, 2006] Khuri, A. (2006). *Response Surface Methodology And Related Topics*. World Scientific.
- [Lee et al., 2004] Lee, J.-S., Cho, T.-S., Lee, J., Jang, M.-K., Jang, T.-K., Nam, D., and Park, C. H. (2004). A stochastic search approach for the multidimensional largest empty sphere problem.

- [Lepš, 2009] Lepš, M. (2009). Soft computing methods in concrete mix performance approximation and optimization. Associate Professor Thesis, Faculty of civil engineering, Czech technical university in Prague.
- [Liang et al., 2001] Liang, Y., Fang, K., and Xu, Q. (2001). Uniform design and its applications in chemistry and chemical engineering. *Chemometrics and Intelligent Laboratory Systems*, 58(1):43 – 57.
- [Montgomery, 2000] Montgomery, D. C. (2000). *Design and Analysis of Experiments, 5th Edition*. Wiley.
- [Morris, 2014] Morris, M. D. (2014). Maximin distance optimal designs for computer experiments with time-varying inputs and outputs. *Journal of Statistical Planning and Inference*, 144(0):63 – 68.
- [Myšáková, 2012] Myšáková, E. (2012). Metody pro tvorbu rovnoměrně rozprostřených návrhů. Bakalářská práce, Katedra mechaniky, Fakulta stavební, České vysoké učení technické v Praze.
- [Myšáková, 2013] Myšáková, E. (2013). Paralelní evoluční strategie pro vyhodnocení miniMax kritéria. Soutěžní práce, Katedra mechaniky, Fakulta stavební, České vysoké učení technické v Praze.
- [Myšáková and Lepš, 2011] Myšáková, E. and Lepš, M. (2011). Comparison of space-filling design strategies. In *Proceedings of the 17th International Conference Engineering Mechanics*, pages 399–402. Ústav termomechaniky AV ČR.
- [Myšáková and Lepš, 2012] Myšáková, E. and Lepš, M. (2012). Method for constrained designs of experiments in two dimensions. In *Proceedings of the 18th International Conference Engineering Mechanics*, pages 222–224. Ústav termomechaniky AV ČR.
- [Myšáková and Lepš, 2013b] Myšáková, E. and Lepš, M. (2013b). Multiobjective minimax designs of experiments. In *Proceedings of the 19th International Conference Engineering Mechanics*, pages 371–379. Ústav termomechaniky AV ČR.
- [Myšáková and Lepš, 2013c] Myšáková, E. and Lepš, M. (2013c). Uniform Designs of Experiments for Asymptotic Sampling. In *Proceedings of the 4th Conference Nano & Macro Mechanics*, pages 143–150. Praha: Czech Technical University in Prague.
- [Myšáková and Lepš, 2013a] Myšáková, E. and Lepš, M. (Sent for publication, 2013a). Comparison of parallel evolutionary approaches for minimax metric of designs of experiments. *Advances in Engineering Software*.
- [Myšáková et al., 2012] Myšáková, E., Lepš, M., and Kučerová, A. (2012). A Method for Maximin Constrained Design of Experiments. In Topping, B. H. V., editor, *Proceedings of the Eighth International Conference on Engineering Computational Technology*. Civil-Comp Press, Stirlingshire, UK.
- [Myšáková et al., 2013] Myšáková, E., Pospíšilová, A., and Lepš, M. (2013). Multiobjective Adaptive Updating of Surrogate Models. In *Proceedings of the 19th International Conference Engineering Mechanics*, pages 87–88. Ústav termomechaniky AV ČR.

- [Okabe et al., 2000] Okabe, A., Boots, B., Sugihara, K., and Chiu, S. N. (2000). *Spatial tessellations: concepts and applications of Voronoi diagrams*. Wiley series in probability and statistics: Applied probability and statistics. Wiley.
- [Persson and Strang, 2004] Persson, P.-O. and Strang, G. (2004). A simple mesh generator in MATLAB. *SIAM Review*, 46(2):329–345.
- [Petelet et al., 2010] Petelet, M., Iooss, B., Asserin, O., and Loredo, A. (2010). Latin hypercube sampling with inequality constraints. *ASTA Advances in Statistical Analysis*, 94:325–339. 10.1007/s10182-010-0144-z.
- [Piepel, 1988] Piepel, G. (1988). Programs for generating extreme vertices and centroids of linearly constrained experimental regions. *Journal of Quality Technology*, 20(2):125–139.
- [Pospíšilová and Lepš, 2013] Pospíšilová, A. and Lepš, M. (2013). Multi-Objective Optimization with Asymptotic Sampling for RBDO. In *Proceedings of The Third International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK.
- [Pospíšilová et al., 2013] Pospíšilová, A., Myšáková, E., and Lepš, M. (2013). Multi-objective Adaptive DoE for RBDO. In *Proceedings of the 11th International Probabilistic Workshop*, pages 325–336. Ing. Vladislav Pokorný - LITERA, Brno.
- [Prescott, 2008] Prescott, P. (2008). Nearly uniform designs for mixture experiments. *Communications in Statistics - Theory and Methods*, 37(13):2095–2115.
- [Pronzato and Müller, 2012] Pronzato, L. and Müller, W. G. (2012). Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701.
- [Pronzato and Walter, 1988] Pronzato, L. and Walter, E. (1988). Robust experiment design via maximin optimization. *Mathematical Biosciences*, 89(2):161 – 176.
- [Rechenberg, 1973] Rechenberg, I. (1973). *Evolution strategy: Optimization of technical systems by means of biological evolution*. Fromman-Holzboog, Stuttgart.
- [Sacks et al., 1989] Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435.
- [Sanches, 2005] Sanches, S. M. (2005). Nohdesigns spreadsheet. Available online via http://diana.cs.nps.navy.mil/SeedLab/LinkedFiles/NOLHdesigns_v4.xls.
- [Schuster, 2008] Schuster, M. (2008). The largest empty circle problem. In *Proceedings of the Class of 2008 Senior Conference, Computer Science Department, Swarthmore College*, pages 28–37.
- [Simon, 2003] Simon, M. J. (2003). *Concrete mixture optimization using statistical methods: final report*. U.S. Dept. of Transportation, Federal Highway Administration, Research, Development and Technology, Turner-Fairbank Highway Research Center.

- [Simon et al., 1997] Simon, M. J., Lagergreen, E. S., and Snyder, K. A. (1997). Concrete Mixture Optimization Using Statistical Mixture Design Methods. In *Proc. of the PCI/FHWA, New Orleans, Louisiana*, pages 230 – 244.
- [Snee, 1979] Snee, R. D. (1979). Experimental designs for mixture systems with multicomponent constraints. *Communications in Statistics - Theory and Methods*, 8(4):303–326.
- [Toropov et al., 2007] Toropov, V. V., Bates, S. J., and Querin, O. M. (2007). Generation of extended uniform latin hypercube designs of experiments. In Topping, B. H. V., editor, *Proceedings of the Ninth International Conference on the Application of Artificial Intelligence to Civil, Structural and Environmental Engineering*. Civil-Comp Press, Stirlingshire, UK.
- [Toussaint, 1983] Toussaint, G. (1983). Computing largest empty circles with location constraints. *International Journal of Computer & Information Sciences*, 12:347–358.
- [van Dam, 2005] van Dam, E. R. (2005). Two-dimensional minimax latin hypercube designs. Discussion Paper 2005-105, Tilburg University, Center for Economic Research.
- [Wynn and Wiggins, 1997] Wynn, C. and Wiggins, A. (1997). *The five biggest ideas in science*. Barnes & Noble Books.

Příloha A

Přehled prvků (0-5)-dimenzionálního simplexu a hyperkrychle

dimenze	název	vrcholy	hrany	stěny	buňky	4-facety	5-facety
0	Bod	1					
1	Úsečka	2	1				
2	Trojúhelník	3	3	1			
3	Čtyřstěn Tetrahedron	4	6	4	1		
4	4-simplex	5	10	10	5	1	
5	5-simplex	6	15	20	15	6	1

Tabulka A.1: Přehled prvků simplexu pro 0-5 dimenzí.

dimenze	název	vrcholy	hrany	stěny	buňky	4-facety	5-facety
0	Bod	1					
1	Úsečka	2	1				
2	Čtverec Tetragon	4	4	1			
3	Krychle Hexahedron	8	12	6	1		
4	Teserakt Octachoron	16	32	24	8	1	
5	Penterakt Decateron	32	80	80	40	10	1

Tabulka A.2: Přehled prvků hyperkrychle pro 0-5 dimenzí.

V Tabulkách A.1 a A.2 uvádíme přehled prvků simplexů a hyperkrychlí do pěti dimenzí. Více informací na [Sim, 2002, Hyp, 2002].

Příloha B

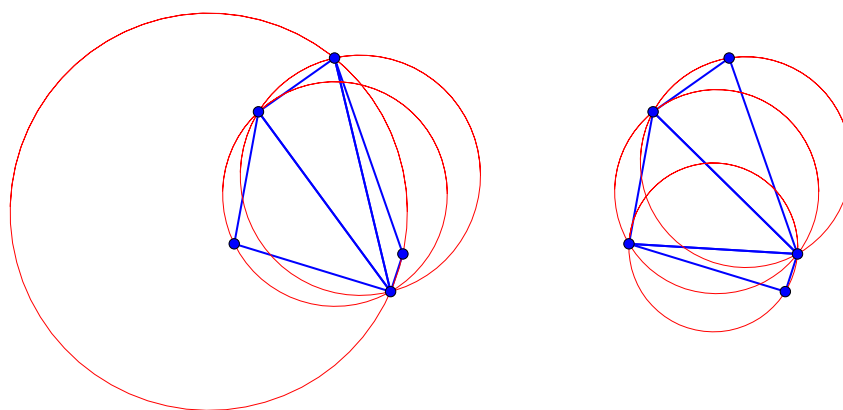
Algoritmus pro výpočet hodnoty *EMM*

V této příloze uvádíme algoritmus pro výpočet nejkratší vzájemné vzdálenosti mezi body návrhu. Díky vhodné indexaci (proměnné I a J) je vyčíslení možné za pomoci jediného *for*-cyklu. Algoritmus je součástí funkce `lhsdesign` statistického toolboxu programu MATLAB.

```
1 function s=metrika_EMM_N(x)
2
3     [m,p] = size(x);
4     pp = (m-1):-1:2;
5     I = zeros(m*(m-1)/2,1);
6     I(cumsum([1 pp])) = 1;
7     I = cumsum(I);
8     J = ones(m*(m-1)/2,1);
9     J(cumsum(pp)+1) = 2-pp;
10    J(1)=2;
11    J = cumsum(J);
12
13    d = zeros(size(I));
14
15    for j=1:p
16        d = d + (x(I,j)-x(J,j)).^2;
17    end
18
19    s = sqrt(min(d));
```

Příloha C

Delaunayova triangulace



(a) Náhodná triangulace.

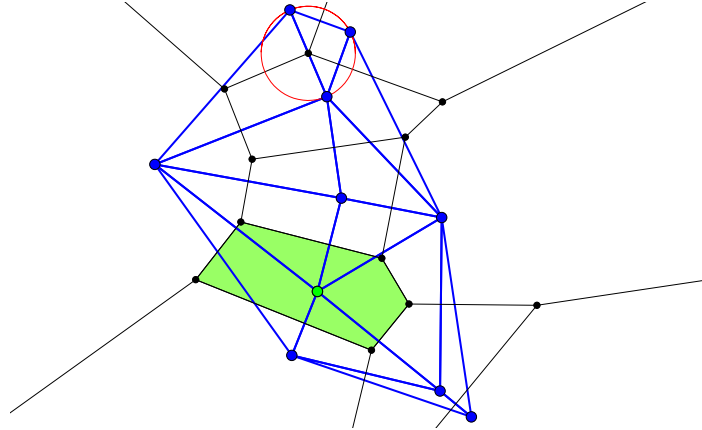
(b) Delaunayova triangulace.

Obrázek C.1: Triangulace sady bodů ve 2D. Legenda: modré body - sada návrhových bodů; modré trojúhelníky - triangulace; červené kružnice - kružnice opsané jednotlivým trojúhelníkům triangulace.

Triangulací chápeme rozdělení části roviny dané sadou bodů na trojúhelníky. Samotný pojem je vhodný pro 2D, princip je však použitelný v n -dimenzionálním prostoru: takový prostor poté rozdělujeme na n -dimenzionální simplex (trojúhelník je 2-dimenzionální simplex). Pro danou sadu bodů existuje velké množství takových řešení. Obrázek C.1 ukazuje dvě možnosti. *Delaunayova triangulace* (DT) [Devadoss and O'Rourke, 2011] je důležitým a často používaným typem triangulace. Mezi její vlastnosti patří, že kružnice opsaná každému z trojúhelníků triangulace (v n D: hyperkoule opsaná každému ze simplexů) tvořenému třemi body (v n D: $n + 1$ body) ze zadané sady neobsahuje žádný další z těchto bodů. To je patrné na srovnání Obrázků C.1a a C.1b. Delaunayova triangulace maximalizuje minimální úhel v jednotlivých trojúhelnících pro zabránění vzniku úzkých trojúhelníků.

Další ze známých vlastností Delaunayovy triangulace je její dualita s *Voronoiovým diagramem* (VD) [Joe and Wang, 1993]. Voronoiův diagram rozděluje rovinu se zadanou sadou bodů na oblasti (Voronoiovy buňky) podle toho, ke kterému ze zadaných bodů mají body roviny nejbližší. Jinými slovy, máme-li zadánou sadu bodů $P = \{p_1, \dots, p_i, \dots, p_n\}$, Voronoiova buňka VC_i je množina bodů v rovině takových, že jejich vzdálenost k za-

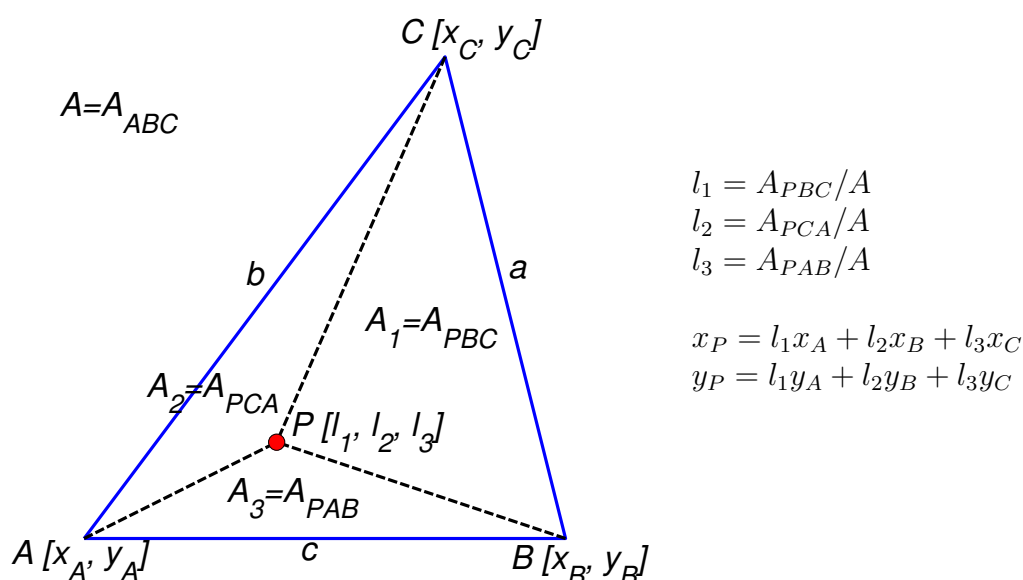
danému bodu p_i je menší než ke všem ostatním zadaným bodům p_j , $j \neq i$. To je ukázáno na Obrázku C.2. Jednotlivé Voronoiovy buňky se stýkají v hranách a vrcholech. Právě tyto Voronoiovy vrcholy odpovídají středům opsaných kružnic jednotlivých trojúhelníků Delaunayovy triangulace.



Obrázek C.2: Dualita Delaunayovy triangulace a Voronoiova diagramu. Legenda: modré body - sada návrhových bodů; modré trojúhelníky - triangulace; černé úsečky - Voronoioův diagram (Voronoiovy hrany); černé body - Voronoiovy vrcholy; zelená oblast - Voronoiova buňka odpovídající zelenému návrhovému bodu; červená kružnice - kružnice opsaná jednomu z trojúhelníků triangulace.

Příloha D

Plošné souřadnice



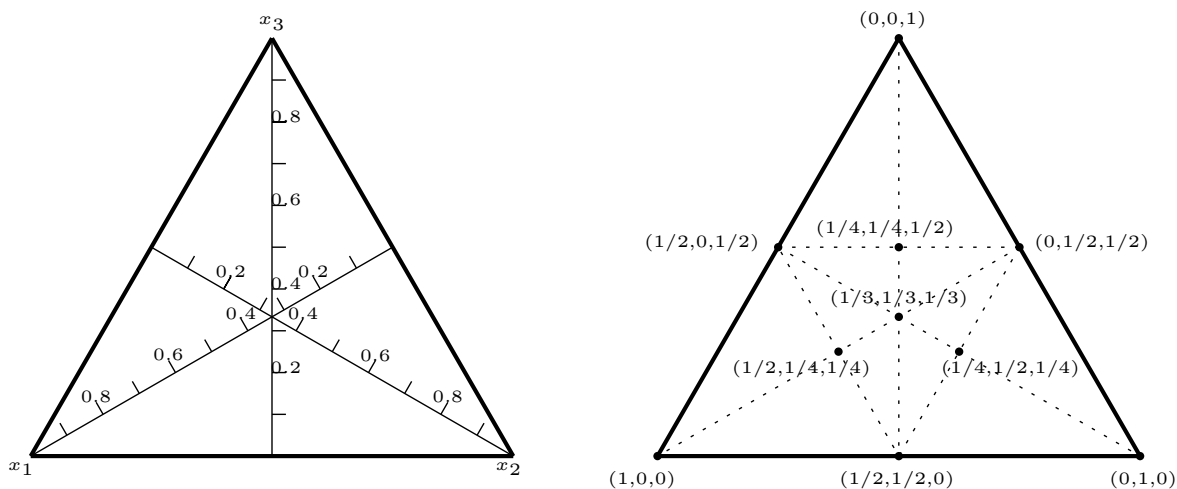
Obrázek D.1: Plošné souřadnice.

Plošné (trojúhelníkové, barycentrické) souřadnice poprvé představil August Ferdinand Möbius (1790 - 1816) ve své knize *The Barycentric Calculus* [Fauvel et al., 1993]. Máme-li daný trojúhelník ABC , plošné souřadnice libovolného bodu P jsou dány poměrem plochy trojúhelníku tvořeného bodem P a jednotlivými stranami trojúhelníku ABC a plochy samotného trojúhelníku ABC (Obrázek D.1).

Plošné souřadnice tak odpovídají souřadnicím v trojosé soustavě (Obrázek D.2 vlevo). Transformace mezi kartézskými souřadnicemi a plošnými souřadnicemi je dána rovnicí:

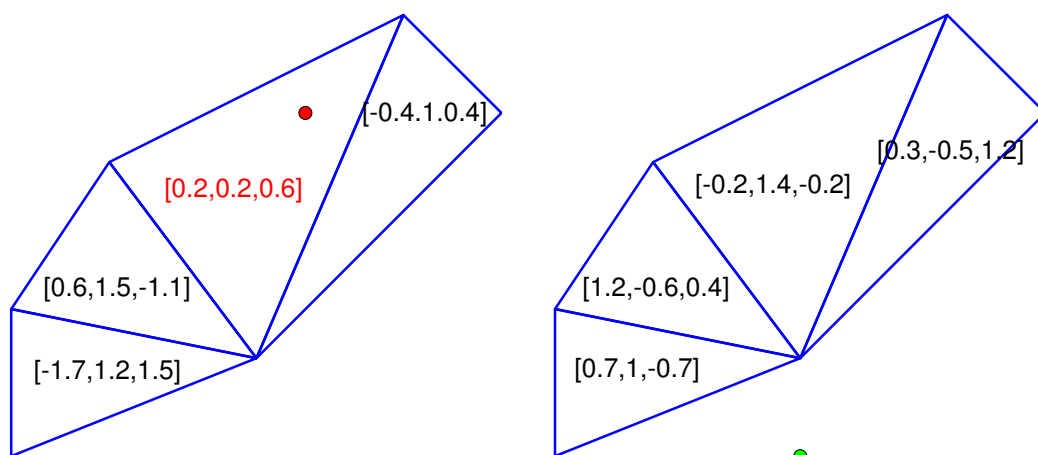
$$\begin{Bmatrix} 1 \\ x_p \\ y_p \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_A & x_B & x_C \\ y_A & y_B & y_C \end{bmatrix} \begin{Bmatrix} l_1 \\ l_2 \\ l_3 \end{Bmatrix} \quad (\text{D.1})$$

Součet plošných souřadnic bodu je vždy roven jedné (viz Obrázek D.2 vpravo s příklady bodů a jejich plošných souřadnic). Bod ležící mimo daný trojúhelník má některé (jednu



Obrázek D.2: Trojosý souřadnicový systém (vlevo) a plošné souřadnice (vpravo).

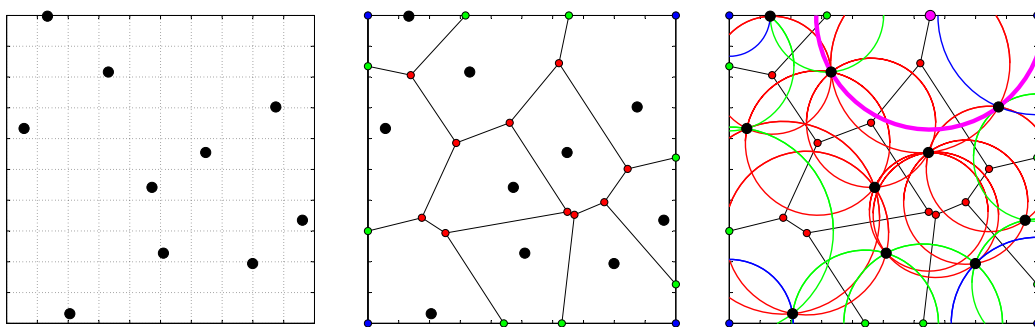
nebo dvě) plošné souřadnice záporné. Toho lze využít právě pro určení, zda bod leží uvnitř nebo vně zadaného trojúhelníku nebo obecně simplexu (při rozšíření do nD). Dále, máme-li ztriangulovaný návrhový prostor a bod, u něhož chceme zjistit, zda leží uvnitř nebo vně této domény, můžeme spočítat plošné souřadnice ke všem simplexům triangulace a rozhodnout: bod leží uvnitř návrhové domény, pokud v triangulaci existuje simplex takový, že plošné souřadnice k němu vztažené jsou všechny kladné (Obrázek D.3).



Obrázek D.3: Určení polohy bodu pomocí plošných souřadnic. Červený bod leží uvnitř návrhového prostoru, neboť je v triangulaci trojúhelník, vzhledem ke kterému má řešení bod všechny plošné souřadnice kladné (obrázek vlevo). Zelený bod leží mimo doménu, neboť jeho plošné souřadnice ke všem trojúhelníkům triangulace obsahují záporná čísla (obrázek vpravo).

Příloha E

Výpočet hodnoty kritéria miniMax v regulárním prostoru⁷



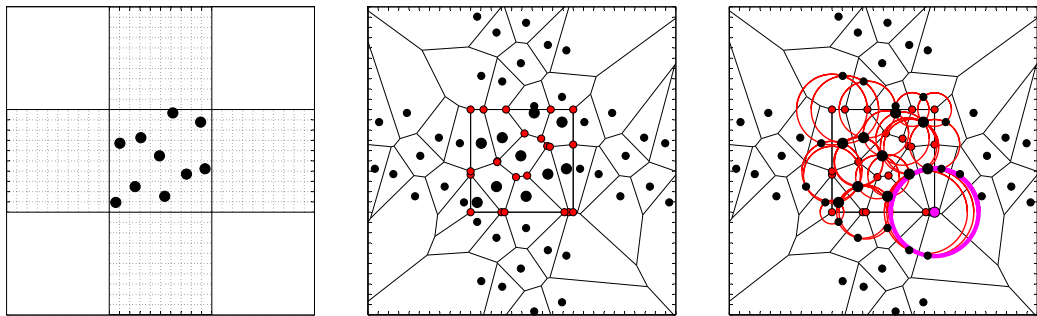
Obrázek E.1: miniMax I. Legenda: černé body - návrhové body; černé úsečky - Voronoiův diagram; červené body - vrcholy Voronoiova diagramu; zelené body - průsečíky hran Voronoiova diagramu s hranicemi řešené domény; modré body - vrcholy domény; červené, zelené a modré kružnice - největší prázdné kruhy se středy v červených, zelených a modrých bodech; purpurová kružnice - největší prázdný kruh; purpurový bod - střed největšího prázdného kruhu. Obrázky slouží pro ilustraci průběhu metody, popisky os jsou proto vynechány.

E.1 Přesné řešení

Přesnou hodnotu kritéria miniMax lze získat pomocí Voronoiova diagramu. Střed největší prázdné koule (jejíž poloměr odpovídá hodnotě mM) leží v některém z vrcholů Voronoiova diagramu, nebo průniku hrany Voronoiova diagramu s hranicí řešené domény.

Ve 2D je všechny tyto kandidátní body možné najít „ručně“. Jsou to vrcholy Voronoiova diagramu, průsečíky hran Voronoiova diagramu se čtyřmi stranami řešené domény (ve 2D čtverce) a čtyři vrcholy řešené domény [Schuster, 2008, Toussaint, 1983]. Zde tuto metodu označujeme jako miniMax I a její průběh je znázorněn na Obrázku E.1. Ve vyšších

⁷Část této kapitoly byla publikována v [Myšáková and Lepš, 2013a] a v [Myšáková, 2013].



Obrázek E.2: miniMax II. Legenda: velké černé body - návrhové body; malé černé body - ozrcadlené návrhové body; černé úsečky - Voronoioův diagram; červené body - vrcholy Voronoioiva diagramu ležící uvnitř nebo na hranici řešené domény; červené kružnice - největší prázdné kruhy se středy v červených bodech; purpurová kružnice - největší prázdný kruh; purpurový bod - střed největšího prázdného kruhu.

dimenzích je však použití této metody obtížnější. Nalezení všech průniků Voronoioiva diagramu s hraničními objekty řešené domény není triviální a je nutné vzít v úvahu i všechny vrcholy řešené hyperkrychle.

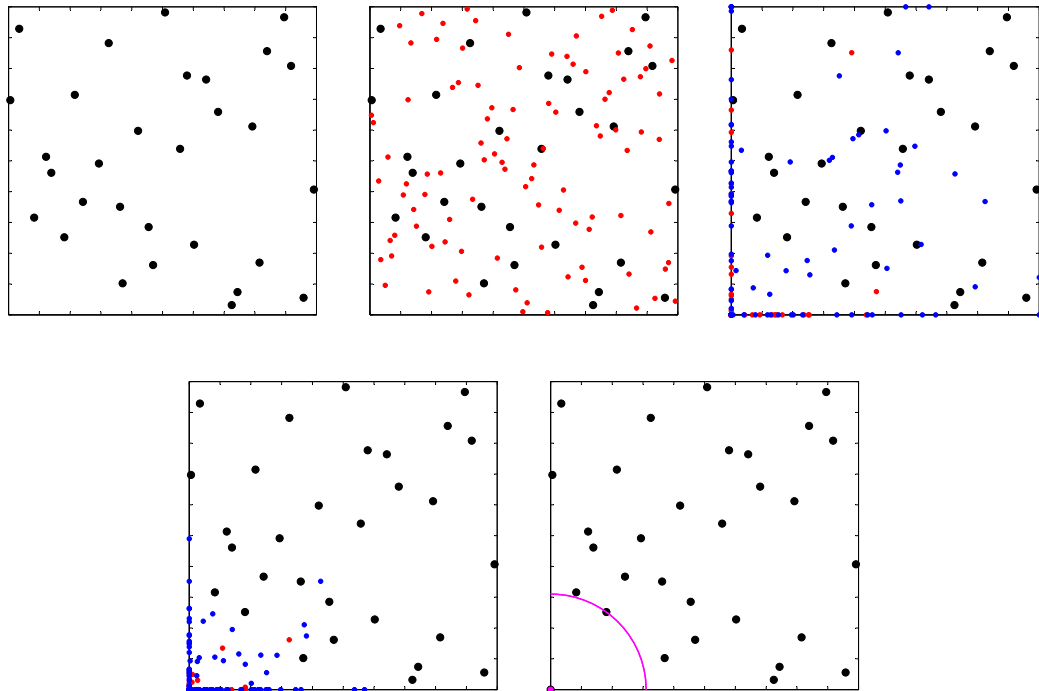
V článku [Pronzato and Müller, 2012] je popsána jiná metoda pro nalezení přesného řešení. Návrhové body jsou zrcadleny skrz všechny $(n - 1)$ -facety (kde n značí počet dimenzí prostoru) a až následně je sestrojen Voronoioův diagram. Kandidátními body jsou pak ty vrcholy Voronoioiva diagramu ležící uvnitř nebo na hranici řešené domény. Tuto metodu ilustruje Obrázek E.2 a označujeme ji jako miniMax II. Zrcadlení zaručuje, že i body na hranici a ve vrcholech domény budou uvedeny mezi kandidátními body. I tato metoda však má své limity. Sestrojení Voronoioiva diagramu je ve vyšších dimenzích velice náročné (časově i pamětově) a zrcadlení (každý bod musí být ozrcadlen $(2n)$ -krát) tyto nároky ještě zvyšuje. To dokumentuje i Tabulka E.1 - v nižších dimenzích jsou exaktní metody efektivnější, ve vyšších dimenzích ale jejich pamětové nároky výrazně rostou.

Problémy, se kterými se setkáváme v inženýrské praxi, jsou však zpravidla multidimenzionální, proto potřebujeme metody schopné pracovat i v takových návrhových prostorech.

E.2 Sériová evoluční strategie

Jednou z možností je odhadnout hodnotu miniMax pomocí heuristické nebo metaheuristické metody. My jsme zvolili evoluční strategii (ES). Jedná se o metodu založenou na principech známých z přírody - přizpůsobování (*adaptation*), mutace (*mutation*), křížení (*crossover*) a výběr (*selection*). Tato technika byla vyvinuta v 60. a 70. letech 20. století Rechenbergem, Schwefelem a jejich spolupracovníky, více informací v [Rechenberg, 1973] nebo [Bäck and Schwefel, 1995].

Základem procedury je cyklus, jehož iterace jsou nazývány generace. Každá generace obsahuje populaci složenou z jednotlivých chromosomů - bodů, které pokrývají řešenou doménu. Cílová funkce je pak vyhodnocena pro každý chromosom (s parametry odpovídajícími souřadnicím chromosomu). Nová populace („potomci“) je odvozena z předchozí populace („rodiče“) pomocí mutace a křížení. Rozlišují se dva typy: v (μ, λ) -

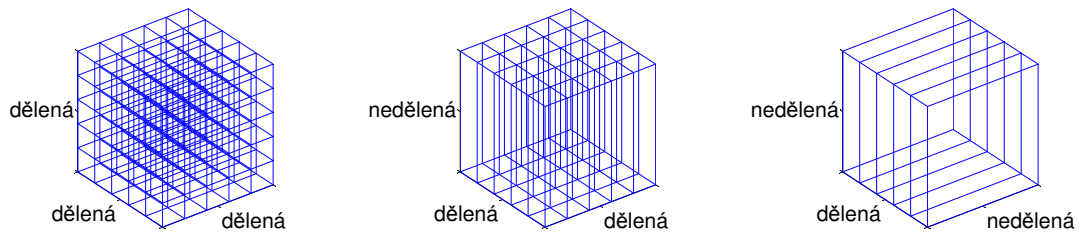


Obrázek E.3: Sériová evoluční strategie. Legenda: černé body - návrhové body; červené body - „rodiče“; modré body - „potomci“; purpurový bod - střed největšího prázdného kruhu; purpurová kružnice - největší prázdný kruh.

ES je nová generace odvozena z potomků z předchozí generace, v $(\mu + \lambda)$ -ES ze sjednocení rodičů a potomků z předchozí generace. Počet generací může být stanoven uživatelem pevně, nebo je pro algoritmus určeno zastavovací kritérium.

$(\mu + \lambda)$ -evoluční strategie použitá zde je inspirována článkem [Lee et al., 2004]. Nejprve je vytvořena výchozí populace. Ta by měla rovnoměrně pokrývat celý prostor, proto je vytvořena pomocí LHS (*latin hypercube sampling*). Jelikož se zde zaměřujeme na porovnání časové náročnosti vlastního cyklu evoluční strategie (sériové vs. paralelní), je počáteční populace vytvořena pomocí funkce `lhsdesign` ze statistického toolboxu programu MATLAB ve výchozím nastavení tak, aby nebyl celkový čas tímto procesem výrazně ovlivněn. Potomci jsou odvozováni pomocí mutace - přičtením normálně rozdělených (průměr roven nule, standardní odchylka klesající s pořadím cyklu) náhodných čísel k rodičovské populaci. Následně jsou ze sjednocení rodičovské populace a potomků vytvořeny náhodné páry a z každého tohoto páru je do nové rodičovské populace vybrán ten chromosom, který má delší vzdálenost k nejbližšímu návrhovému bodu. Toto výběrové schéma upřednostňuje lepší jednotlivce k postupu do další generace. Metoda je znázorněna na Obrázku E.3. Je na něm patrné, jak se populace přibližuje k hledanému řešení.

Metoda je robustní a uživatel může volit velikost populace i počet generací podle dimenze řešeného problému. Přesto je ve vyšších dimenzích obtížné prozkoumat celou řešenou doménu. To by měla zajistit dále popsaná paralelní metoda.



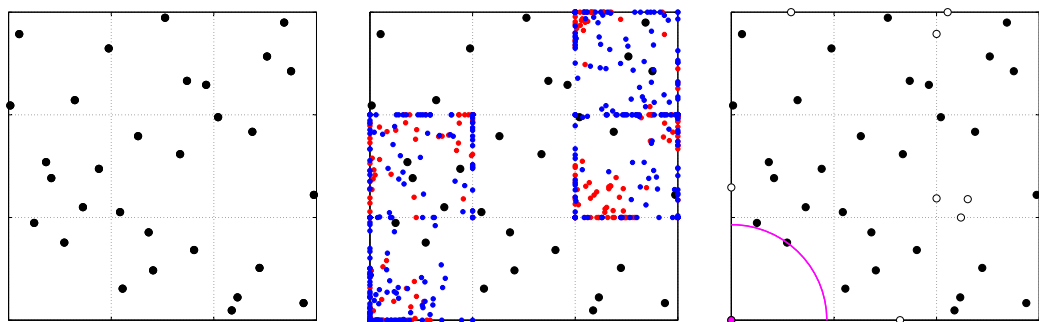
Obrázek E.4: Paralelní evoluční strategie - způsoby dělení na subdomény ve 3D s $k = 5$, kde k je počet intervalů na dělené hraně domény.

E.3 Paralelní evoluční strategie

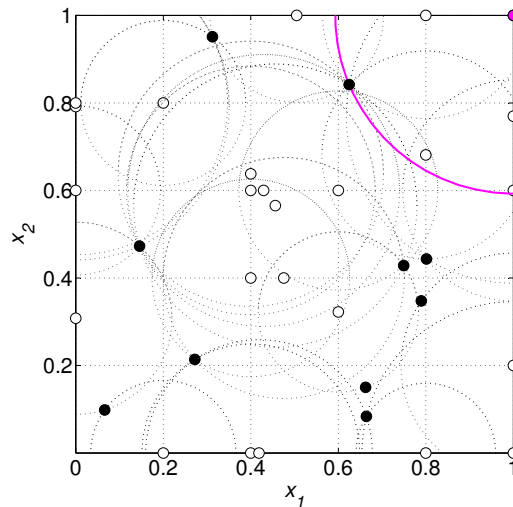
Paralelizovat evoluční strategii lze dvěma způsoby: za prvé, může probíhat několiké hledání v celé doméně paralelně nezávisle na sobě; za druhé, řešenou doménu můžeme rozdělit na menší celky - subdomény - a hledání pomocí ES pak probíhá paralelně v těchto subdoménách. Zde je zvolena druhá z možností.

Doménu lze dělit několika způsoby, jak ukazuje Obrázek E.4. První z uvedených variant je pravděpodobně nejvhodnější k zaručení prohledání celého prostoru. Počet vzniklých subdomén je potom k^n , kde k označuje počet intervalů na dělené hraně domény. Množství subdomén tedy roste velice rychle s dimenzí řešené domény a v případě vyšších domén je tedy nutné použít některou z dalších alternativ dělení. Zde je použita první varianta dělení na subdomény s $k = 2$. Počet chromosomů v populaci je zvolen $10n$.

Popsanou metodu znázorňuje Obrázek E.5. Několik subdomén je tedy řešeno zároveň paralelně na jednotlivých CPU. Z každé subdomény obdržíme kandidátní řešení/bod (Obrázek E.6), jako střed největší prázdné koule (její poloměr je pak hodnota mM) je z těchto bodů označen ten, který má nejdelší vzdálenost k bodům návrhu.



Obrázek E.5: Paralelní evoluční strategie ve 2D s $k = 3$. Legenda: černé body - návrhové body; červené body - „rodiče“; modré body - „potomci“; bílé body - v každé subdoméně bod nejvíce vzdálený od návrhových bodů; purpurový bod - střed největšího prázdného kruhu; purpurová kružnice - největší prázdný kruh.



Obrázek E.6: Paralelní evoluční strategie. Legenda: černé body - návrhové body; bílé body - v každé subdoméně bod nejvíce vzdálený od návrhových bodů; tečkované kružnice - největší prázdné kruhy se středy v bílých bodech; purpurový bod - střed největšího prázdného kruhu; purpurová kružnice - největší prázdný kruh.

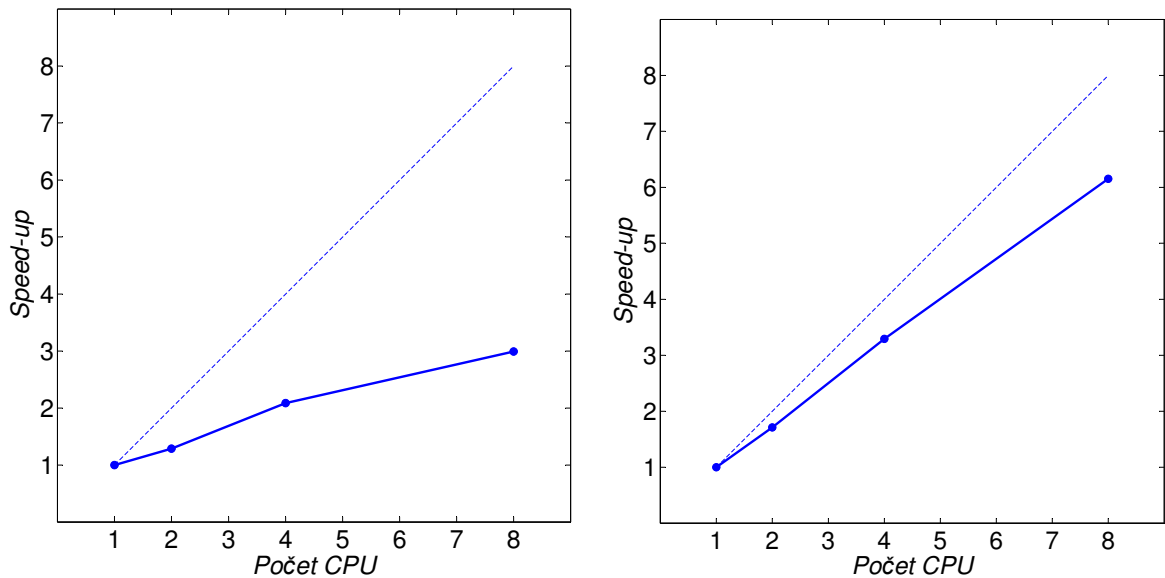
E.4 Porovnání a shrnutí

K porovnání představených metod byly použity tři typové příklady: (i) 2D návrh s 27 body převzatý z [van Dam, 2005], (ii) 6D návrh se 17 body a (iii) 12D návrh s 65 body. Tvorba uvedených 6D a 12D návrhů je popsána v [Cioppa and Lucas, 2007] a výsledné návrhy byly staženy ze [Sanches, 2005].

Nejprve byly jednotlivé metody porovnány z pohledu výpočetních nároků. Ty jsou uvedeny v Tabulce E.1. Jedná se o průměrné hodnoty z 10 nezávislých spuštění každé procedury. Je zřejmé, že v nízkých dimenzích jasně vítězí metody poskytující přesné řešení (označeny miniMax I a miniMax II). Již v 12D příkladu ovšem nestačí operační paměť kvůli extrémní náročnosti sestavení Voronoiova diagramu. Sériová verze evoluční strategie je při zvolené velikosti populace a počtu generací více než 20krát pomalejší než exaktní metody. Efektivita paralelní verze je pak ve 2D případě nulová. Komunikace v počítači nutná k rozdělení problému na jednotlivé subdomény zde časově vysoce převyšuje samotný

	2D	6D	12D
metoda	čas [s]		
miniMax I	0,007	–	–
miniMax II	0,004	2,265	- (vyčerpáno 12 GB RAM)
ES miniMax - sériová	0,183	47,193	2489,557
ES miniMax - paralelní - 2 CPU	7,338	36,634	1455,274
ES miniMax - paralelní - 4 CPU	7,489	22,617	756,122
ES miniMax - paralelní - 8 CPU	7,935	15,774	404,693

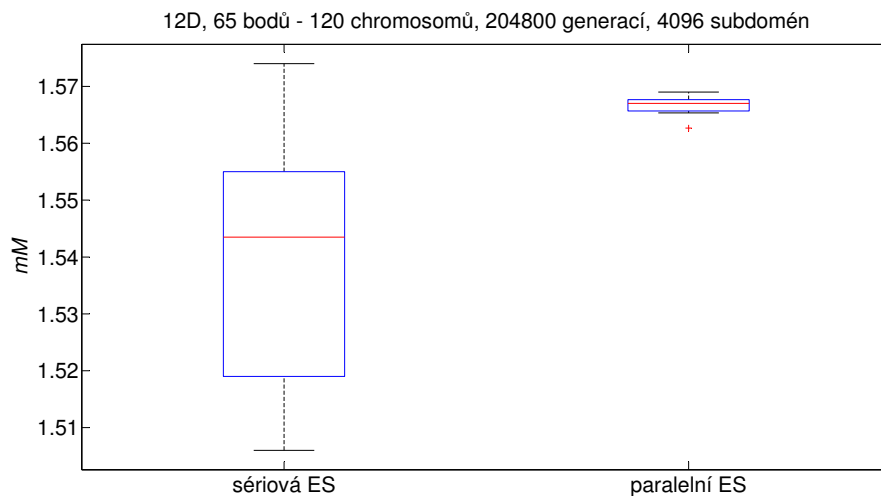
Tabulka E.1: Časová náročnost metod vyčíslení kritéria miniMax.



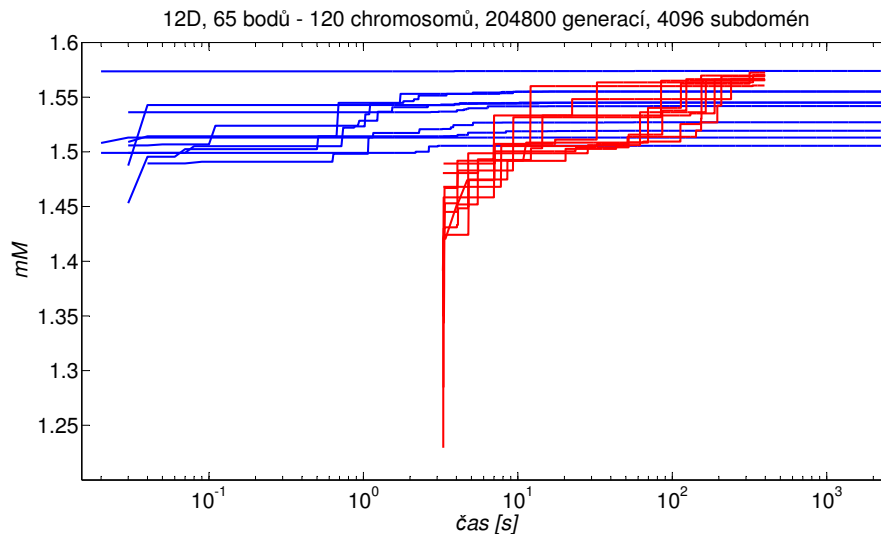
Obrázek E.7: Speed-up paralelní ES pro výpočet hodnoty miniMax: 6D, 17 bodů - 60 chromosomů, 12800 generací, 64 subdomén (vlevo) a 12D, 65 bodů - 120 chromosomů, 204800 generací, 4096 subdomén (vpravo).

výpočet a celkový potřebný čas tu dokonce roste s počtem použitých CPU. Zajímavější je však situace ve vyšších dimenzích. Obrázek E.7 ukazuje speed-up (závislost zrychlení výpočtu na počtu využitých CPU) pro 6D a 12D problémy. Se zvyšující se dimenzí se čas potřebný ke komunikaci mezi jednotlivými CPU stává zanedbatelným a paralelní metoda zde dosahuje téměř lineárního speed-upu.

Nyní metody porovnáme z pohledu kvality získaného řešení - hodnoty kritéria miniMax. Připomeňme, že evoluční strategie je stochastický algoritmus s náhodným chováním a hodnota jí získaná je tedy pouze odhadem - přesná hodnota může být vyšší. V nižších



Obrázek E.8: Boxplot hodnot mM získaných uvedenými metodami pro problém ve 12D.



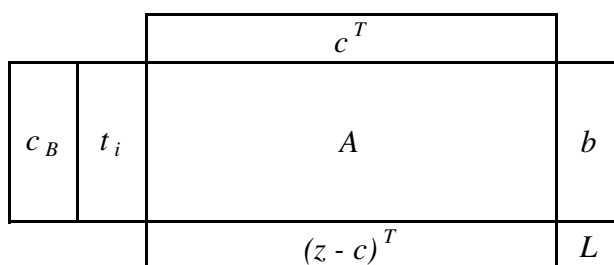
Obrázek E.9: Vývoj hodnoty mM v průběhu výpočtu pro 10 spuštění ES. Legenda: modré křivky - sériová verze; červené křivky - paralelní verze (8 CPU).

dimenzích, tedy ve 2D a 6D příkladu poskytují naše metody založené na ES (sériová a paralelní) přesné řešení při všech 10 spuštěních. Pro 12D problém přesnou hodnotu mM neznáme. Obrázek E.8 ukazuje boxplot hodnot kritéria miniMax získaných pomocí obou verzí evoluční strategie. Sériová verze je někdy schopná nalézt přesnější řešení (vyšší hodnotu odhadu mM), vykazuje však vyšší rozptyl v porovnání s verzí paralelní řešící problém v jednotlivých subdoménách. Poznamenejme, že celkový počet provedených iterací a výpočtů vzdáleností byl shodný pro sériovou verzi i pro verzi paralelní (součet počtů generací v jednotlivých subdoménách je shodný s počtem generací v sériové verzi). Srovnání metod z hlediska reálného času ukazuje Obrázek E.9. Sériová verze ES se zdá být velmi rychlá z pohledu nalezení kvalitního řešení. Výsledek je však ze značné části dán „štěstím“ v zasažení oblasti středu největší prázdné koule (ten se velmi často nachází na hranici řešené domény). Paralelní verze ztrácí určitý čas na začátku výpočtu rozdělením prostoru na subdomény a komunikací nutnou k paralelizaci. Poté však paralelní verze poskytuje robustnější řešení s menším rozptylem.

Byly představeny dvě tradiční metody poskytující přesné řešení miniMax kritéria společně s algoritmem evoluční strategie v sériové a paralelní verzi. Ten poskytuje odhad hodnoty kritéria. Naše výsledky ukazují, že v nižších dimenzích navržené metody založené na ES přesné řešení naleznou. Ve vyšších dimenzích pak bez problémů s nedostatkem operační paměti (typické pro exaktní metody využívající Voronoiův diagram) poskytují odhad hodnoty mM . Paralelní verze představuje způsob „rozparcelování“, tedy rozdělení celé řešené domény na menší subdomény. Tato verze dosahuje ve vyšších dimenzích téměř lineárního speed-upu a odhady jí získané mají menší rozptyl v porovnání se sériovou verzí (kde je řešena celá doména najednou).

Příloha F

Ruční výpočet hledání vrcholů polytopu pomocí LP



Obrázek F.1: Simplexová tabulka obecně. Základem simplexové tabulky je rozšířená matice soustavy $(A|b)$. Řádek c^T obsahuje cenové koeficienty, sloupec t_i označuje pořadí sloupců matice A , které tvoří bázi a sloupec c_B cenové koeficienty odpovídající těmto sloupcům. V řádku $(z - c)^T$ najdeme relativní ceny a v poli L aktuální hodnotu účelové funkce.

Simplexové tabulky uvedené na Obrázcích F.2, F.3, F.4 a F.5 znázorňují řešení úlohy hledání krajních bodů množiny přípustných řešení pomocí simplexové metody lineárního programování [Demel, 2011].

Uspořádání tabulek odpovídá obecnému schématu uvedenému na Obrázku F.1. Neznámé x_1, x_2, x_3 odpovídají hledaným neznámým ze zadání úlohy, neznámé $x_4 - x_{11}$ jsou tzv. doplňkové a neznámé $x_{12} - x_{16}$ nazýváme umělé.

Cenové koeficienty (odpovídající normále účelové funkce) byly zvoleny rovny 1, což onu účelovou funkci činí nepodstatnou a není tak preferován žádný směr hledání krajních bodů. Je tak zaručeno, že budou nalezeny všechny, a to možnými kombinacemi sloupců matice A v bázi (sloupce v bázi jsou zvýrazněny různým podbarvením).

Touto metodou bylo nalezeno všech šest krajních bodů množiny přípustných řešení, což odpovídá jak výsledku z referencí, tak výsledku grafické metody.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	1
-1E+06 12	1	1	1									1					0,1
-1E+06 13	1			-1									1				0,5
0 5	1				1												0,1
-1E+06 14		1				-1								1			0,7
0 7		1					1										0,7
0 8			1					1									90
-1E+06 15	85	90	100						-1							1	95
0 10	85	90	100							1							0,4
-1E+06 16	0,7		1								-1						1
	-8,8E+07	-9,2E+07	-1E+08	1000000	0	1000000	0	0	1000000	0	1000000	0	0	0	0	0	-9E+07
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	
-1E+06 12	0,3	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0,6
-1E+06 13	1			-1									1				0,1
0 5	1				1												0,5
-1E+06 14		1				-1								1			0,1
0 7		1					1										0,7
0 8	-0,7	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0,3
-1E+06 15	15	90	0	0	0	0	0	0	-1	0	100	0	0	0	1	-100	50
0 10	15	90	0	0	0	0	0	0	0	1	100	0	0	0	0	-100	55
1 3	0,7		1								-1						0,4
	-1,6E+07	-9,2E+07	0	1000000	0	1000000	0	0	1000000	0	-1E+08	0	0	0	0	0	1,02E+08
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	
-1E+06 12	1	1	0	0	0	0	0	-1	0	0	0	1	0	0	0	0	0,3
-1E+06 13	1			-1									1				0,1
0 5	1				1												0,5
-1E+06 14		1				-1								1			0,1
0 7		1					1										0,7
0 11	-0,7	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0,3
-1E+06 15	85	90	0	0	0	0	0	-100	-1	0	0	0	0	0	1	0	20
0 10	85	90	0	0	0	0	0	-100	0	1	0	0	0	0	0	0	25
1 3	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0,7
	-8,7E+07	-9,2E+07	0	1000000	0	1000000	0	1,01E+08	1000000	0	0	0	0	0	0	0	-2E+07

Obrázek F.2: Simplexové tabulky č. 1-3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000		
-1E+06 12	1	0	0	0	0	1	0	-1	0	0	0	1	0	-1	0	0	0,2	
-1E+06 13	1			-1									1				0,1	
0 5	1				1												0,5	
1 2		1				-1								1			0,1	
0 7		0	0	0	0	1	1	0	0	0	0	0	0	-1	0	0	0,6	
0 11		-0,7	0	0	0	0	0	1	0	0	1	0	0	0	0	-1	0,3	
-1E+06 15		85	0	0	0	90	0	-100	-1	0	0	0	0	-90	1	0	11	
0 10		85	0	0	0	90	0	-100	0	1	0	0	0	-90	0	0	16	
1 3		0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0,7	
		-8,7E+07	0	0	1000000	0	-9,1E+07	0	1,01E+08	1000000	0	0	0	0	92000001	0	1000000	-1E+07
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000		
-1E+06 12		0	0	0,06	0	0	0	0,11	0,01	0	0	1	-0,06	0	-0,01	0	0,0722	
1 1		1		-1									1				0,1	
0 5		0	0	0	1	1	0	0	0	0	0	0	-1	0	0	0	0,4	
1 2		0	1	0	0,94	0	0	-1,11	-0,01	0	0	0	-0,94	0	0,01	0	0,1278	
0 7		0	0	0	-0,94	0	0	1,11	0,01	0	1	0	0,94	0	-0,01	0	0,5722	
0 8		0	0	0	-0,7	0	0	1	0	0	1	0	0,7	0	0	-1	0,37	
0 6		0	0	0	0,94	0	1	-1,11	-0,01	0	0	0	-0,94	-1	0,01	0	0,0278	
0 10		0	0	0	0	0	0	0,00	1	1	0	0	0	0	-1	0	5	
1 3		0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0,7	
		0	0	0	-55555,6	0	0	-111111	-11111,1	0	0	0	1055556	1000000	1011111	1000000	-72221	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16		
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000		
-1E+06 12		0	0	0,13	0	0	0	0	0,01	0	-0,11	1	-0,13	0	-0,01	0,11	0,0311	
1 1		1		-1									1				0,1	
0 5		0	0	0	1	1	0	0	0	0	0	0	-1	0	0	0	0,4	
1 2		0	1	0	0,17	0	0	0	-0,01	0	1,11	0	-0,17	0	0,01	-1,11	0,5389	
0 7		0	0	0	-0,17	0	0	1	0	0,01	-1,11	0	0,17	0	-0,01	1,11	0,1611	
0 8		0	0	0	-0,7	0	0	0	1	0	1	0	0,7	0	0	-1	0,37	
0 6		0	0	0	0,17	0	1	0	-0,01	0	1,11	0	-0,17	-1	0,01	-1,11	0,4389	
0 10		0	0	0	0	0	0	0	1	1	0	0	0	0	-1	0	5	
1 3		0	0	1	0,7	0	0	0	0	0	-1	0	-0,7	0	0	1	0,33	
		0	0	0	-133333	0	0	0	-11111,1	0	111111,2	0	1133333	1000000	1011111	888888,8	-31110	

Obrázek F.3: Simplexové tabulky č. 4-6.

7

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	
0 4	0	0	0	1	0	0	0	0	0,08	0	-0,83	7,5	-1	0	-0,08	0,83	0,2333
1 1	1	0	0	0	0	0	0	0	0,08	0	-0,83	7,5	0	0	-0,08	0,83	0,3333
0 5	0	0	0	0	1	0	0	0	-0,08	0	0,83	-7,5	0	0	0,08	-0,83	0,1667
1 2	0	1	0	0	0	0	0	0	-0,02	0	1,25	-1,25	0	0	0,02	-1,25	0,5
0 7	0	0	0	0	0	0	1	0	0,02	0	-1,25	1,25	0	0	-0,02	1,25	0,2
0 8	0	0	0	0	0	0	0	1	0,06	0	0,42	5,25	0	0	-0,06	-0,42	0,5333
0 6	0	0	0	0	0	1	0	0	-0,02	0	1,25	-1,25	0	-1	0,02	-1,25	0,4
0 10	0	0	0	0	0	0	0	0	1	1	0	0	0	0	-1	0	5
1 3	0	0	1	0	0	0	0	0	-0,06	0	-0,42	-5,25	0	0	0,06	0,42	0,1667
	0	0	0	0	0	0	0	0	0	0	0	1000001	1000000	1000000	1000000	1000000	1

x1 0,3333
x2 0,5
x3 0,1667

hledání dalších vrcholů (změnou báze):

8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	
0 9	0	0	0	12	0	0	0	0	1	0	-10	90	-12	0	-1	10	2,8
1 1	1	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0,1
0 5	0	0	0	1	1	0	0	0	0	0	0	0	-1	0	0	0	0,4
1 2	0	1	0	0,3	0	0	0	0	0	0	1	1	-0,3	0	0	-1	0,57
0 7	0	0	0	-0,3	0	0	1	0	0	0	-1	-1	0,3	0	0	1	0,13
0 8	0	0	0	-0,7	0	0	0	1	0	0	1	0	0,7	0	0	-1	0,37
0 6	0	0	0	0,3	0	1	0	0	0	0	1	1	-0,3	-1	0	-1	0,47
0 10	0	0	0	-12	0	0	0	0	0	1	10	-90	12	0	0	-10	2,2
1 3	0	0	1	0,7	0	0	0	0	0	0	-1	0	-0,7	0	0	1	0,33
	0	0	0	0	0	0	0	0	0	0	0	1000001	1000000	1000000	1000000	1000000	1

x1 0,1
x2 0,57
x3 0,33

9

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	
0 9	0	0	0	0	0	0	0	0	1	1	0	0	0	0	-1	0	5
1 1	1	0	0	-1	0	0	0	0	0	0	0	0	1	0	0	0	0,1
0 5	0	0	0	1	1	0	0	0	0	0	0	0	-1	0	0	0	0,4
1 2	0	1	0	1,5	0	0	0	0	0	-0,1	0	10	-1,5	0	0	0	0,35
0 7	0	0	0	-1,5	0	0	1	0	0	0,1	0	-10	1,5	0	0	0	0,35
0 8	0	0	0	0,5	0	0	0	1	0	-0,1	0	9	-0,5	0	0	0	0,15
0 6	0	0	0	1,5	0	1	0	0	0	-0,1	0	10	-1,5	-1	0	0	0,25
0 11	0	0	0	-1,2	0	0	0	0	0	0,1	1	-9	1,2	0	0	-1	0,22
1 3	0	0	1	-0,5	0	0	0	0	0	0,1	0	-9	0,5	0	0	0	0,55
	0	0	0	0	0	0	0	0	0	0	0	1000001	1000000	1000000	1000000	1000000	1

x1 0,1
x2 0,35
x3 0,55

Obrázek F.4: Simplexové tabulky č. 7-9.

10

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0 9	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	5
1 1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	-1	0	0,2667
0 5	1	0	0	0	0	0,67	0	0	0	-0,07	0	6,67	0	-0,67	0	0	0,2333
1 2	0	0	0	0	1	-0,67	0	0	0	0,07	0	-6,67	0	0,67	0	0	0,1
0 7	0	1	0	0	0	-1	0	0	0	0	0	0	0	1	0	0	0,6
0 8	0	0	0	0	0	1	1	0	0	0	0	0	0	-1	0	0	0,0667
0 4	0	0	0	0	0	-0,33	0	1	0	-0,07	0	5,67	0	0,33	0	0	0,1667
0 11	0	0	0	1	0	0,67	0	0	0	-0,07	0	6,67	-1	-0,67	0	0	0,42
1 3	0	0	0	0	0	0,8	0	0	0	0,02	1	-1	0	-0,8	0	-1	0,6333
	0	0	1	0	0	0,33	0	0	0	0,07	0	-5,67	0	-0,33	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	1000001	1000000	1000000	1000000	1000000	

x1 0,2667
x2 0,1
x3 0,6333

11

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0 9	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	1,5
1 1	0	0	0	0	-15	10	0	0	1	0	0	100	0	-10	-1	0	0,5
0 10	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3,5
1 2	0	0	0	0	15	-10	0	0	0	1	0	-100	0	10	0	0	0,1
0 7	0	1	0	0	0	-1	0	0	0	0	0	0	0	1	0	0	0,6
0 8	0	0	0	0	0	1	1	0	0	0	0	0	0	-1	0	0	0,3
0 4	0	0	0	0	1	-1	0	1	0	0	0	-1	0	1	0	0	0,4
0 11	0	0	0	1	1	0	0	0	0	0	0	0	-1	0	0	0	0,35
1 3	0	0	0	0	-0,3	1	0	0	0	0	1	1	0	-1	0	-1	0,4
	0	0	1	0	-1	1	0	0	0	0	0	1	0	-1	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	1000001	1000000	1000000	1000000	1000000	

x1 0,5
x2 0,1
x3 0,4

12

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0 6	1	1	1	0	0	0	0	0	0	0	0	-1000000	-1000000	-1000000	-1000000	-1000000	0,15
1 1	0	0	0	0	-1,5	1	0	0	0,1	0	0	10	0	-1	-0,1	0	0,5
0 10	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5
1 2	0	0	0	0	0	0	0	0	1	1	0	0	0	0	-1	0	0,25
0 7	0	1	0	0	-1,5	0	0	0	0,1	0	0	10	0	0	-0,1	0	0,45
0 8	0	0	0	0	1,5	0	1	0	-0,1	0	0	-10	0	0	0,1	0	0,45
0 4	0	0	0	0	-0,5	0	0	1	0,1	0	0	9	0	0	-0,1	0	0,4
0 11	0	0	0	1	1	0	0	0	0	0	0	0	-1	0	0	0	0,2
1 3	0	0	0	0	1,2	0	0	0	-0,1	0	1	-9	0	0	0,1	-1	0,25
	0	0	1	0	0,5	0	0	0	-0,1	0	0	-9	0	0	0,1	0	1
	0	0	0	0	0	0	0	0	0	0	0	1000001	1000000	1000000	1000000	1000000	

x1 0,5
x2 0,25
x3 0,25

vrcholy přípustné domény:

$(x_1) =$	0,333	$(x_2) =$	0,100	$(x_3) =$	0,100	$(x_4) =$	0,267	$(x_5) =$	0,500	$(x_6) =$	0,500
	0,500		0,570		0,350		0,100		0,100		0,250
	0,167		0,330		0,550		0,633		0,400		0,250

Obrázek F.5: Simplexové tabulky č. 10-12.