



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta stavební
Katedra mechaniky

Metody pro tvorbu rovnoměrně rozprostřených návrhů

Methods for space-filling designs of experiments

Bakalářská práce

Studijní program: Stavební inženýrství

Studijní obor: Materiálové inženýrství

Vedoucí práce: Ing. Matěj Lepš, Ph.D.

Eva Myšáková

Praha 2012

Zde je prostor pro zadání.

Čestné prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pouze za odborného vedení vedoucího bakalářské práce 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 Ing. Matěji Lepšovi, Ph.D. za jeho cenné připomínky, nekonečnou trpělivost a ochotu při vedení mé bakalářské práce.

Abstrakt

Tato práce se zabývá metodami pro tvorbu návrhů experimentů - *designs of experiments* (DoEs). Ty jsou základní součástí každého experimentování, ale jsou nepostradatelné rovněž pro spolehlivostní výpočty nebo citlivostní analýzy. V této práci se věnujeme tvorbě návrhů experimentů v regulárních návrhových prostorech tvaru hyperkrychlí. Základním požadavkem na návrhy experimentů je jejich schopnost rovnoměrně pokrýt návrhový prostor (*space-filling*). Pro hodnocení kvality návrhů, ale též jako cílová funkce v procesu optimalizace během jejich tvorby je použita *Euklidovská maximální vzdálenost* (*EMM*). Její použití je vhodné u návrhů jak pro klasické reálné (fyzikální, chemické, biologické, ...) experimenty, tak pro experimenty virtuální - tzv. simulace (např. i armádní bojové scénáře).

V práci jsou představeny metody pro tvorbu návrhů s fixním, předem stanoveným počtem návrhových bodů, ale i metody schopné doplňovat již vytvořené návrhy dalšími návrhovými body. Dále můžeme metody rozdělit na ty, které tvoří návrhy splňující podmínky *Latin Hypercube Sampling* (LHS), a na ty, jež tyto podmínky nespĺňují. LHS návrhy jsou vhodné především pro stochastické výpočty, neboť zaručují určitou pravděpodobnost výskytu bodů. Návrhy nespĺňující LHS podmínky jsou pak vhodné pro aproximace nebo optimalizace reálných konstrukcí (např. v automobilovém průmyslu).

Práce představuje řadu metod pro tvorbu rovnoměrně rozprostřených návrhů experimentů spolu s porovnáním jejich časové náročnosti (implementovány byly v programu MATLAB) a kvality jimi vytvořených návrhů.

Abstract

This thesis is focused on methods for *designs of experiments* (DoEs). DoEs constitute an essential part of any experimentation. They are also indispensable for reliability or sensitivity analyses. We are interested in designs of experiments in regular design spaces within hypercubes. The main objective placed on the designs is their *space-filling* property. The quality of resulting designs is measured by *Euclidean Maximin distance* (*EMM*). This metric is used as an objective function in optimization process during the design creation in several presented methods. The use of this metric is suitable for classical real (physical, chemical, biological, ...) experiments as well as for virtual experiments called simulations (e.g. army's combat scenarios).

Both methods that create designs with a fix, predetermined number of design points and methods with ability of adding points into already created designs are presented in this thesis. We can also divide the methods into two categories. Ones with designs complying the *Latin Hypercube Sampling* (LHS) conditions and ones with designs uncomplying these conditions. LHS designs are suitable mainly for stochastic computations because they guarantee certain probability of points distribution. Non-LHS designs are suitable for approximations or structure optimization (e.g. in car industry).

The thesis presents several methods for creation of space-filling designs of experiments. It also provides a comparison of their computational demands (algorithms were implemented in MATLAB) and quality of resulting designs.

Klíčová slova

návrh experimentu, rovnoměrnost pokrytí, Latin Hypercube Sampling, Delaunayova triangulace, Distmesh, maximin, hyperkrychle, regulární návrhové prostory, implementace

Keywords

design of experiment, space-filling, Latin Hypercube Sampling, Delaunay triangulation, Distmesh, maximin, hypercube, regular design spaces, implementation

Obsah

1	Úvod	13
1.1	Návrhy experimentů	13
1.2	Cílové funkce	16
1.3	Rozdělení metod	17
2	Metody s fixním počtem bodů	18
2.1	Metody splňující podmínky LHS	18
2.1.1	Metoda <code>standard</code> LHS	18
2.1.2	Výměny	19
2.1.2.1	Metoda výměna <code>náhodný_náhodný</code>	20
2.1.2.2	Metoda výměna <code>Cmin_náhodný</code>	20
2.1.3	Výsledky	21
2.2	Metody nesplňující podmínky LHS	23
2.2.1	Odebírání z přehuštěného LHS návrhu	23
2.2.1.1	Metoda ubírání	23
2.2.1.2	Metoda ubírání <code>NEW</code>	24
2.2.2	Nástroj <code>Distmesh</code>	24
2.2.3	Výsledky	26
3	Sekvenční metody	28
3.1	Metody splňující podmínky LHS	28
3.1.1	Metoda <code>LHS_sequential</code>	28
3.1.1.1	<code>smooth off</code> - trojnásobek počtu bodů	28
3.1.1.2	<code>smooth on</code> - dvojnásobek počtu bodů	30
3.1.2	Výsledky	31
3.2	Metody nesplňující podmínky LHS	32
3.2.1	Metody založené na Delaunayově triangulaci	32
3.2.1.1	Stálá triangulace	33
3.2.1.2	Rozdělování	36
3.2.1.3	Heuristická procedura pro rozdělování	36

3.2.1.4	Periodické okrajové podmínky	37
3.2.2	Generátory	39
3.2.2.1	Generátory pseudonáhodných čísel	39
3.2.2.2	Generátory kvazináhodných sekvencí	40
3.2.3	Výsledky	41
4	Implementace	44
5	Výsledky	48
6	Závěr	54
A	Výpočet objemu simplexu	59
B	Výpočet souřadnic těžiště simplexu	60
C	Přehled prvků (0-5)-dimenzionální hyperkrychle	61
D	Parametry použitého počítače a softwaru	62
E	Obsah příloženého CD	63

Seznam tabulek

1.1	Rozdělení metod.	17
C.1	Přehled prvků hyperkrychle pro 0-5 dimenzí.	61
D.1	Použitá počítačová sestava.	62

Seznam obrázků

1.1	Diskrétní návrhový prostor.	14
1.2	Nerovnoměrně a rovnoměrně rozprostřený DoE.	14
1.3	Spojitý návrhový prostor.	15
1.4	Tvary návrhových prostorů ve 3D.	15
2.1	Příklad LHS návrhu.	18
2.2	Nevhodný LHS návrh.	19
2.3	Výměna dvou náhodných bodů návrhu.	20
2.4	Výměna bodu z dvojice s <i>EMM</i> a náhodně vybraného bodu.	21
2.5	Porovnání rychlosti a kvality - fixní, LHS - 5D.	22
2.6	Odebrání náhodně zvoleného bodu z dvojice s <i>EMM</i>	23
2.7	Odebrání „horšího“ bodu z dvojice s <i>EMM</i>	24
2.8	Průběh metody využívající nástroj <i>Distmesh</i>	25
2.9	Směr posunu bodů zpět do přípustné domény.	26
2.10	Porovnání rychlosti a kvality - fixní, neLHS - 2D, 5D.	27
3.1	Metoda <code>LHS_sequential</code> verze <code>smooth off</code>	29
3.2	Průběh metody <code>LHS_sequential</code> verze <code>smooth off</code>	29
3.3	Tvorba výchozího návrhu pro metodu <code>LHS_sequential</code> verze <code>smooth off</code>	30
3.4	Metoda <code>LHS_sequential</code> verze <code>smooth on</code>	30
3.5	Průběh metody <code>LHS_sequential</code> verze <code>smooth on</code>	31
3.6	Porovnání rychlosti a kvality - sekvenční, LHS - 2D.	32
3.7	Delaunayova triangulace.	32
3.8	Dělení stěny 3D domény.	33
3.9	Rozdíl v hustotě dělení hran.	34
3.10	První varianta metody založené na Delaunayově triangulaci.	35
3.11	Druhá varianta metody založené na Delaunayově triangulaci.	35
3.12	Heuristická procedura pro DT metodu s rozdělováním.	36
3.13	Použití periodických okrajových podmínek.	37
3.14	Výstup typický pro rozdělování bez retriangulace při PBC.	38

3.15	Varianta metody s PBC.	38
3.16	Sobolova a Haltonova sekvence ve 2D.	40
3.17	Sobolova a Haltonova sekvence ve 3D.	40
3.18	Vývoj <i>EMM</i> při přidávání bodů.	42
3.19	Porovnání rychlosti a kvality - sekvenční, neLHS - 2D.	43
4.1	Příklad výpočtu vzájemných vzdáleností.	45
4.2	Porovnání rychlosti a kvality - 2D - před a po optimalizaci.	47
5.1	Porovnání rychlosti a kvality - 2D.	50
5.2	Porovnání rychlosti a kvality - 3D.	51
5.3	Porovnání rychlosti a kvality - 4D.	52
5.4	Porovnání rychlosti a kvality - 5D.	53
6.1	Metoda tvorby DoE v neregulárních prostorech.	55
B.1	Plošné (trojúhelníkové) souřadnice.	60
B.2	Těžiště trojúhelníku.	60

Kapitola 1

Úvod¹

„Pokud zavoláte statistika až po provedení experimentu, je to jako byste ho požádali o provedení pitvy. Možná vám řekne, na co experiment zemřel.“² Tak zní citát R. A. Fishera (1890 - 1962), jenž je považován za zakladatele teorie plánování experimentů [Fisher, 1935]. Každý experiment je prováděn za účelem získání informací o nějakém systému nebo procesu. Přesněji řečeno, chceme znát odezvu na konkrétní kombinaci vstupních parametrů. K tomu, abychom získali o chování systému či procesu co nejlepší a nejvíce vypovídající informace, je vhodné experiment předem naplánovat - použít *návrh experimentu - design of experiment* (DoE).

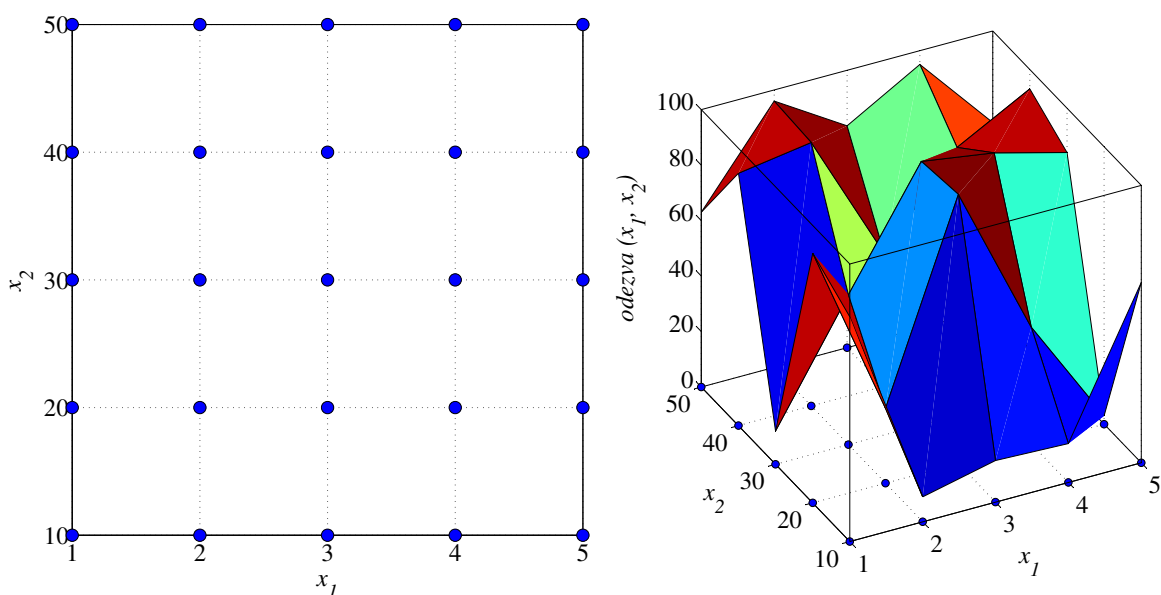
1.1 Návrhy experimentů

Všechny přípustné kombinace vstupních parametrů vytváří n -dimenzionální návrhový prostor. Souřadnice bodů návrhového prostoru (návrhových bodů) odpovídají konkrétním hodnotám jednotlivých vstupních parametrů. Na Obrázcích 1.1 a 1.3 vidíme diskrétní a spojitý 2D návrhový prostor. Z důvodů časových, finančních, ale též faktických nelze otestovat všechny kombinace vstupních parametrů, tj. provést experiment odpovídající každému bodu návrhového prostoru. Přestože máme možnost vybrat pouze omezené množství návrhových bodů, potřebujeme získat co nejlepší představu o chování testovaného systému nebo procesu; výběru daného množství návrhových bodů z návrhového prostoru (tedy tvorbě návrhu experimentu) je proto zapotřebí věnovat patřičnou pozornost. Je zřejmé, že získáme odlišnou představu o chování systému použitím návrhů experimentů zobrazených na Obrázku 1.2 vlevo a vpravo.

Oblasti lidské činnosti, ve kterých se s touto problematikou setkáváme, jsou velice

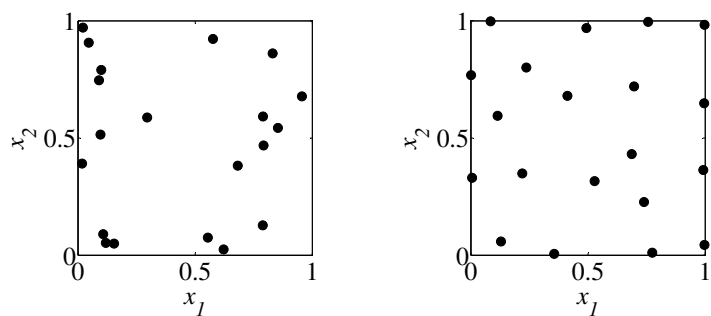
¹Část této kapitoly byla publikována v [Myšáková and Lepš, 2012a].

²Původně „*To call in the statistician after the experiment is done may be no more than asking him to perform a postmortem examination: he may be able to say what the experiment died of.*“ Sir Ronald Aylmer Fisher, Indian Statistical congress, Sankhya, c.1938.

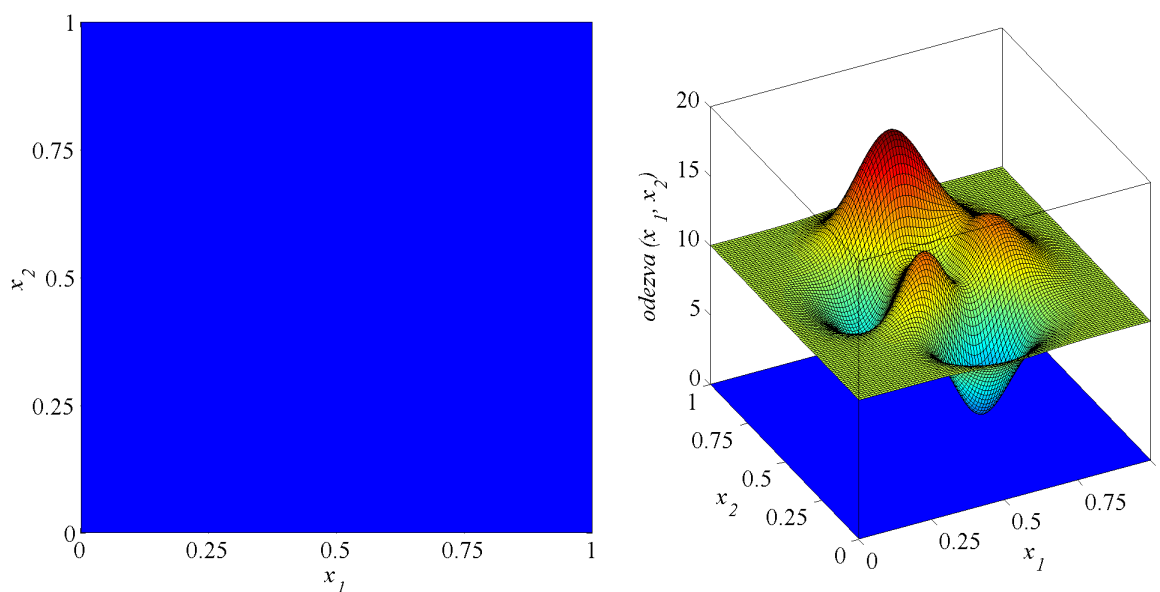


Obrázek 1.1: 2-dimenzionální diskretní návrhový prostor. Obrázek vlevo ukazuje návrhový prostor. Ten je tvořen konečným počtem návrhových bodů (modré body) - všemi kombinacemi přípustných hodnot jednotlivých proměnných (x_1 a x_2). Obrázek vpravo pak ukazuje odezvu systému nebo procesu pro jednotlivé kombinace vstupních parametrů.

rozdílné. Původně se jednalo o navrhování chemických experimentů, dnes se s návrhy experimentů setkáváme v průmyslu farmaceutickém, zpracovatelském, ve strojírenství i elektronice, ale i v oblasti služeb nebo marketingu. Optimalizují se výrobní procesy i vlastnosti výrobků samotných. Kromě reálných experimentů je však návrh experimentu nedílnou součástí experimentů počítačových, analýz spolehlivosti či analýz citlivosti. Návrhy experimentů se pro různé účely mohou výrazně lišit. Na rozdíl od počítačových experimentů, ve kterých při stejné kombinaci vstupních parametrů získáme stejný výstup, v případě reálných experimentů toto platit nemusí. Proto každá z těchto skupin nahlíží odlišně například na vhodnost výskytu duplikovaných nebo vzájemně si blízkých návrhových



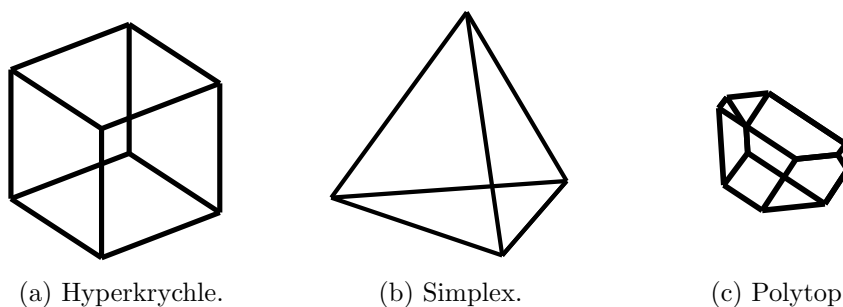
Obrázek 1.2: Nerovnoměrně (vlevo) a rovnoměrně (vpravo) rozprostřený návrh experimentu s 20 návrhovými body ve 2D.



Obrázek 1.3: 2-dimenzionální spojitý návrhový prostor. Návrhovým prostorem (obrázek vlevo) je část roviny (modrý čtverec). Je tvořen nekonečným počtem návrhových bodů odpovídajících kombinacím vstupních proměnných (x_1 a x_2). Obrázek vpravo (pro ilustraci byla použita funkce `peaks` dostupná v programu MATLAB) pak ukazuje odezvu systému nebo procesu pro jednotlivé kombinace vstupních parametrů.

bodů. O návrhu reálných experimentů a analýze jejich výsledků pojednává základní dílo [Montgomery, 2000], o počítačových experimentech se může čtenář více dozvědět v [Sacks et al., 1989] nebo ve [Fang et al., 2006].

Samotné experimenty můžeme dělit dle přítomnosti a podoby podmínek omezujících hodnoty vstupních parametrů (proměnných). V případě, že takové podmínky nejsou, mluvíme o regulárním návrhovém prostoru tvaru hyperkrychle (Obrázek 1.4a). Jednotlivé parametry (odpovídající dimenzím hyperkrychle) mohou nabývat celé škály hodnot bez ohledu na hodnoty ostatních parametrů. Nejčastějším typem experimentu s omezením je *návrh směsi*, ve kterém součet hodnot vstupních parametrů odpovídá jednot-



Obrázek 1.4: Tvary návrhových prostorů ve 3D.

kové hmotnosti nebo objemu. V takovém případě má návrhový prostor tvar simplexu³ (Obrázek 1.4b). Jsou-li zadány ještě další podmínky omezující hodnoty jednotlivých vstupních parametrů, získáváme návrhový prostor ve tvaru *polytopu* (Obrázek 1.4c).

1.2 Cílové funkce

Kvalitu návrhů můžeme hodnotit celou řadou cílových funkcí/kritérií hodnotících ortogonalitu nebo rovnoměrnost pokrytí. Do první skupiny patří cílové funkce:

- *číslo podmíněnosti (CN)*,
- *Pearsonův korelační koeficient (PMCC)*,
- *Spearmanův korelační koeficient (SRCC)* nebo
- *Kendallův korelační koeficient (KRCC)*.

Odkazy na kritéria zaměřená na ortogonalitu je možné nalézt např. v publikacích [Cioppa and Lucas, 2007, Hofwing and Strömberg, 2010]. Rovnoměrnost pokrytí pak hodnotí následující funkce:

- *Audze-Eglais (AE)*,
- *Euklidovská maximin vzdálenost (EMM)*,
- *modifikovaná L_2 diskrepance (ML_2)* nebo
- *D-optimalita (Dopt)*.

O těch se lze více dozvědět v [Toropov et al., 2007] nebo [Crombecq et al., 2009]. Přehled a porovnání jmenovaných cílových funkcí uvádí též [Janouchová and Kučerová, 2011].

V této práci je pro porovnávání návrhů vytvořených jednotlivými metodami, ale i jako cílová funkce v optimalizačním cyklu některých metod použita *Euklidovská maximin vzdálenost (EMM)*. Je zvolena pro svou jednoduchost a snadnost zobrazení. Zároveň se jedná o metriku, která nejlépe poukazuje na vzájemnou blízkost experimentů (návrhových bodů). *EMM* je nejkratší ze všech vzájemných vzdáleností mezi body návrhu:

$$E^{EMM} = \min\{\dots, L_{ij}, \dots\}, \quad i = 1, \dots, np; \quad j = (i + 1), \dots, np, \quad (1.1)$$

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

³Hodnoty jednotlivých proměnných pak odpovídají plošným (trojúhelníkovým) souřadnicím návrhových bodů (viz Příloha B). Dimenze návrhového prostoru tu tudíž není shodná s počtem proměnných, nýbrž o jednu nižší. Jak vidíme i v Příloze B, trojúhelník je 2-dimenzionální objekt, ale popíšeme-li jednotlivé body plošnými souřadnicemi, stanovíme tak hodnoty tří proměnných.

	splňující LHS	nesplňující LHS
Fixní počet bodů	standard LHS výměna_náhodný_náhodný výměna_Cmin_náhodný	Distmesh ubírání ubírání_NEW
Sekvenční	LHS_sequential	Delaunayova triangulace generátory

Tabulka 1.1: Rozdělení prezentovaných metod.

1.3 Rozdělení metod

Metod tvorby návrhů experimentů existuje velké množství. Nejjednodušší je metoda *Monte Carlo*, ve které jsou body voleny zcela náhodně. My se však snažíme o tvorbu návrhů rovnoměrně pokrývajících návrhový prostor. To nelze u metody *Monte Carlo* zaručit. Lepších výsledků z tohoto pohledu dosahuje populární metoda *Latin Hypercube Sampling* (LHS), o níž se více dozvíme záhy v Sekci 2.1.1 a z které řada metod popsaných v této práci vychází. Použijeme Delaunayovu triangulaci [Crombecq et al., 2009, Del, 2001] k vyhledávání nedostatečně pokrytých částí návrhového prostoru a prozkoumáme možnosti nástroje *Distmesh* [Persson and Strang, 2004] pro tvorbu rovnoměrně rozprostřených návrhů. Pozornost bude věnována též generátorům pseudonáhodných čísel a kvazináhodných sekvencí.

V této práci se budeme zabývat metodami pro tvorbu návrhů experimentů bez omezujících podmínek; návrhovým prostorem nám tedy budou hyperkrychle. Uvedené metody jsou schopné vytvářet návrhy v n -dimenzionálním prostoru. Z důvodů výpočetní náročnosti jsou v této práci uvedeny výsledky metod na hyperkrychlích ve 2D - 5D, a to vždy pro 100 návrhových bodů. Návrhy experimentů jsou zde tvořeny v bezrozměrných doménách $\langle 0, 1 \rangle^n$, kde n je počet dimenzí. Skutečné návrhy se vytvoří lineární transformací do uživatelem daných mezí. Je nutné poznamenat, že se zabýváme experimenty se spojitými proměnnými v oboru reálných čísel⁴.

Jako výchozí je v práci použito dělení na metody s fixním, předem daným počtem bodů (Kapitola 2) a metody s doplňováním bodů do již vytvořených návrhů (Kapitola 3). V obou kapitolách jsou dále metody rozděleny podle toho, zda návrhy jimi vytvořené splňují (Sekce 2.1 a 3.1) či nesplňují (Sekce 2.2 a 3.2) podmínky LHS. Tabulka 1.1 přináší přehled a základní rozdělení prezentovaných metod. V závěru každé sekce jsou představeny dílčí výsledky jednotlivých metod. Kompletní výsledky jsou pak uvedeny v Kapitole 5.

⁴Jednotlivé proměnné mohou nabývat libovolné hodnoty v rámci svých mezí - zde interval $\langle 0, 1 \rangle$. Naproti tomu v případě návrhů v diskrétních prostorech jsou hodnoty jednotlivých proměnných vybírány z konečného počtu možností.

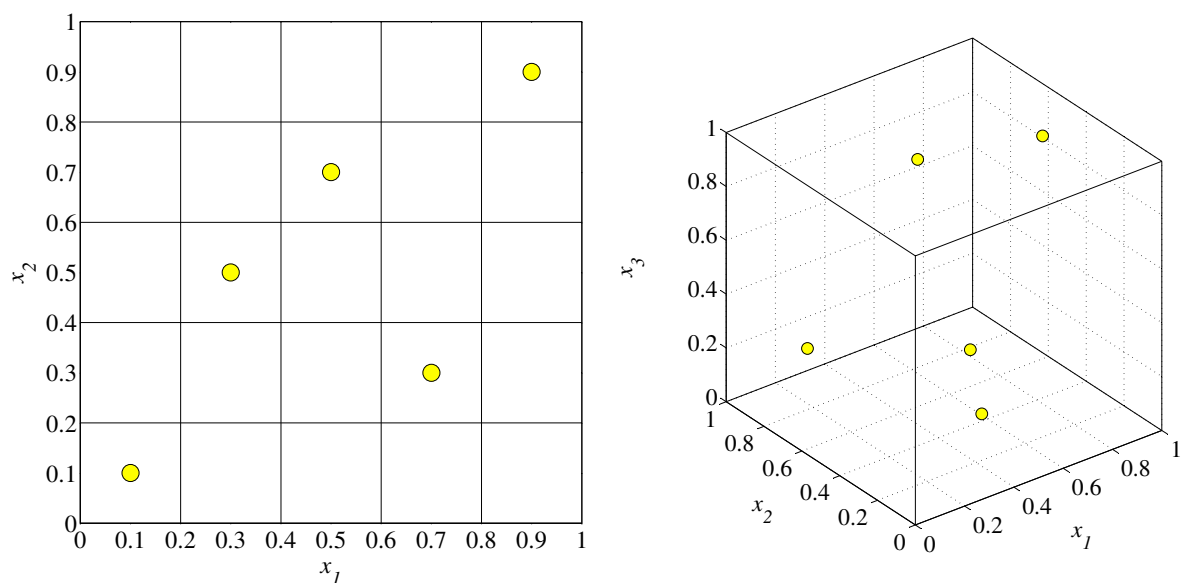
Kapitola 2

Metody s fixním počtem bodů⁵

V této kapitole se budeme věnovat metodám, pomocí kterých tvoříme návrhy s fixním, předem stanoveným počtem bodů.

2.1 Metody splňující podmínky LHS

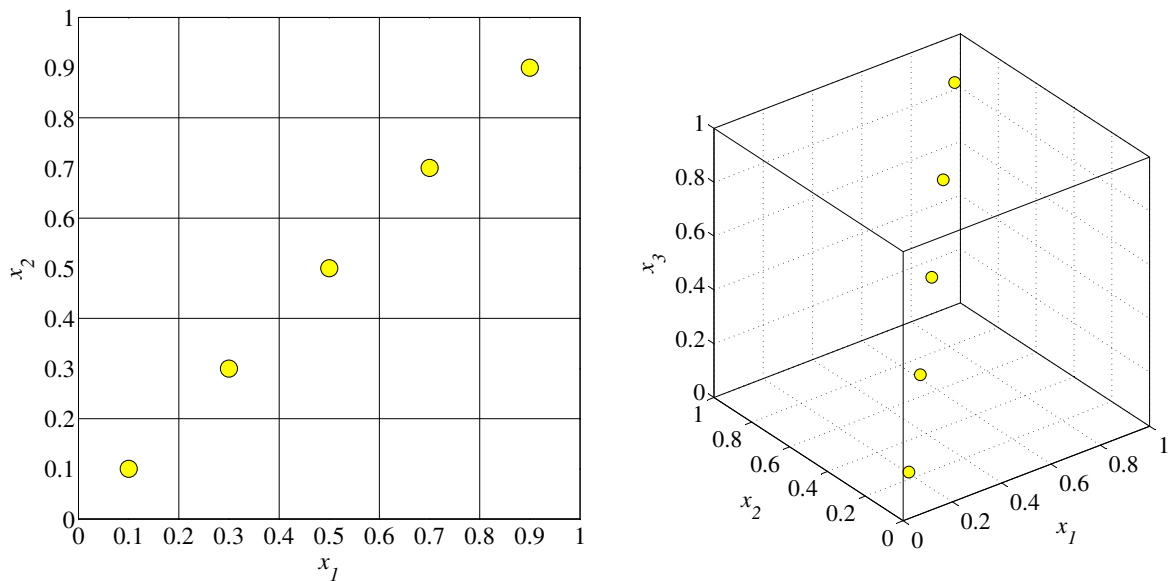
2.1.1 Metoda standard LHS



Obrázek 2.1: Příklad LHS návrhu pro 2 (vlevo) a 3 (vpravo) proměnné s 5 body umístěnými do středů intervalů (`smooth off`).

Latin Hypercube Sampling (LHS) je jedním z nejpopulárnějších algoritmů pro tvorbu návrhů, zejména z důvodu jednoduchosti použití a rychlosti výpočtu. Tímto algoritmem

⁵Část této kapitoly byla publikována v [Myšáková and Lepš, 2011].



Obrázek 2.2: Nevhodný LHS návrh.

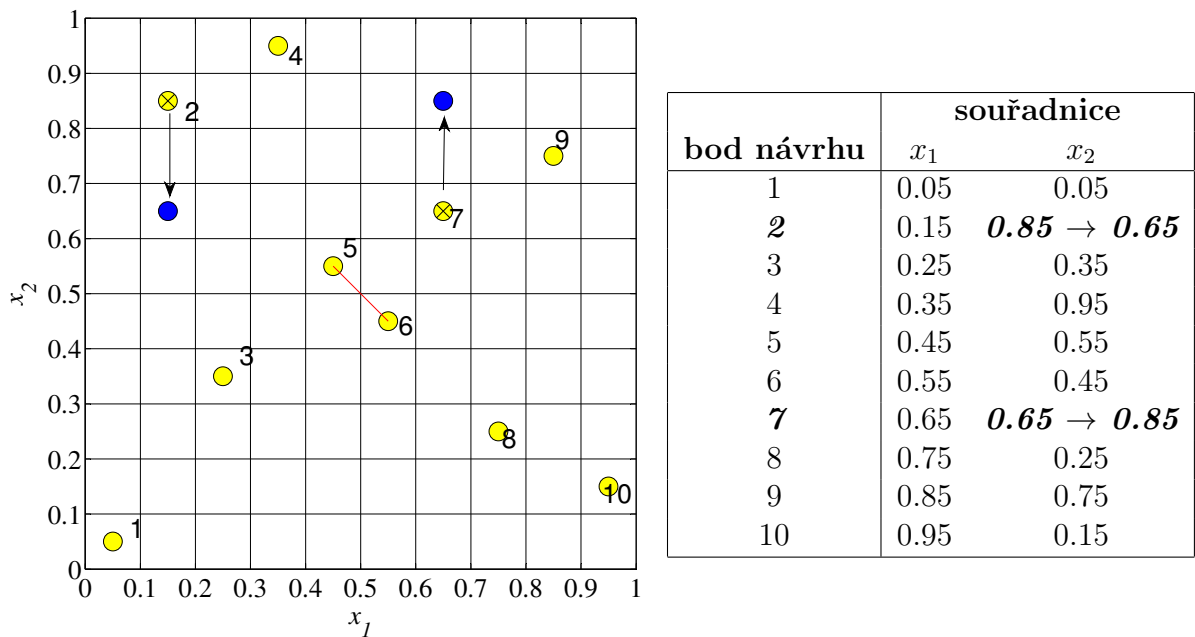
vytvořené návrhy však z pohledu rovnoměrnosti pokrytí návrhového prostoru nedosahují vždy přesvědčivých výsledků. V LHS je každá proměnná (dimenze) rozdělena na np intervalů, kde np odpovídá požadovanému počtu bodů návrhu. V rámci každého intervalu je bod umístěn náhodně⁶ nezávisle pro každou proměnnou. V každém intervalu každé proměnné (dimenze) je umístěn pouze jeden bod (ve 2D „pravidlo sudoku“). To vede k regulárnímu DoE, jak je patrné z Obrázku 2.1. Nejhorší případ LHS návrhu je ten, kdy body vytvoří diagonálu (Obrázek 2.2). Souřadnice jednotlivých návrhových bodů jsou pak lineárně závislé, rovněž rovnoměrnost pokrytí návrhového prostoru je velmi špatná. Nejjednodušším řešením takového nedostatku je vytvoření zcela nového LHS návrhu, to jest užití „metody hrubé síly“. Takový postup je užit například v prostředí MATLAB ve funkci `lhsdesign`⁷. Metoda `standard LHS` tedy v každé iteraci (počet iterací volí uživatel) vytváří nový návrh a jako výsledný představí ten s nejlepší (nejvyšší) hodnotou *EMM*.

2.1.2 Výměny

Špatný LHS návrh (z pohledu *EMM*) může být vylepšen výměnou pozic jednotlivých bodů (ovšem při zachování vlastností LHS), viz například [Novák and Lehký, 2006, Kučerová, 2007]. Jsou vybrány 2 body, jedna z dimenzí a prohozeny odpovídající souřadnice.

⁶Nebo může být umístěn do středu intervalu - v prostředí programu MATLAB parametr `smooth off`.

⁷Podle přednastavení je vytvořeno 5 návrhů a nejlepší (z pohledu zvolené cílové funkce - v našem případě *EMM*) je představen uživateli.



Obrázek 2.3: Výměna dvou náhodných bodů návrhu (zde na souřadnici x_2). Červená úsečka vyznačuje EMM . Nedochozí ke zvýšení hodnoty EMM , tato výměna tedy nebude přijata.

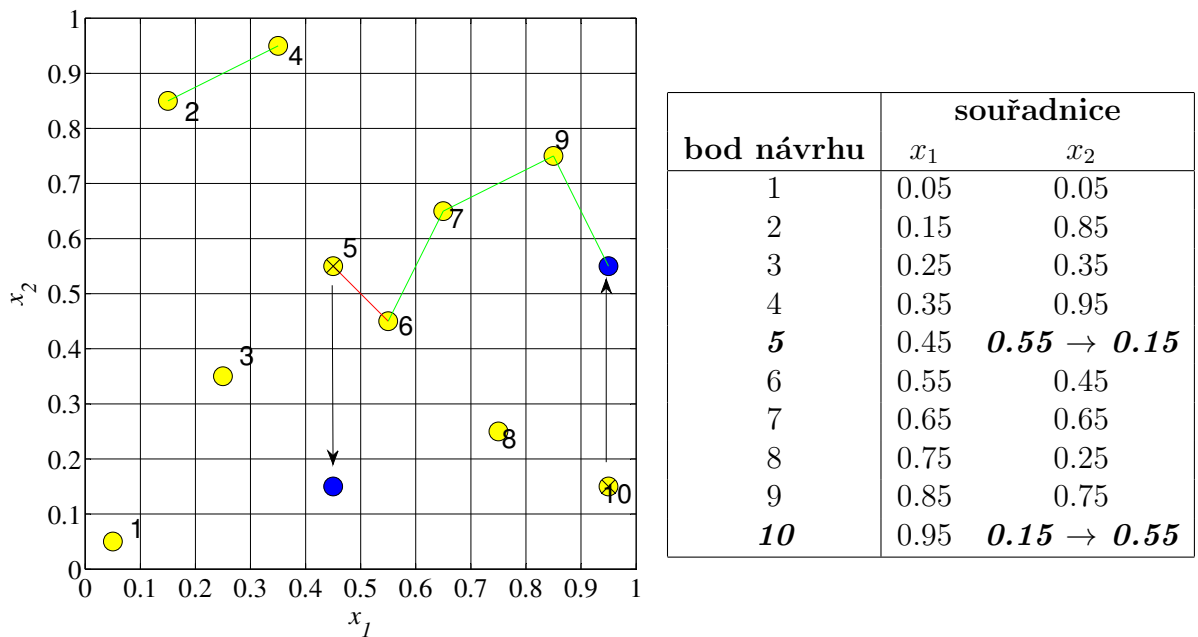
2.1.2.1 Metoda výměna náhodný náhodný

V první variantě metody jsou všechny tři volby (2 body a konkrétní souřadnice) zcela náhodné. Po každé iteraci, oproti výše zmíněným odkazům, kde je použit algoritmus simulovaného žíhání (*Simulated Annealing*), přijímáme pouze úspěšný krok (kdy došlo ke zvýšení hodnoty EMM). Proto tuto metodu můžeme označit jako náhodný horolezecký (*Hill Climbing*) algoritmus. Jak je patrné i z Obrázku 2.3, úspěšná může být pouze výměna, do níž vstupuje jeden z bodů z dvojice s minimální vzájemnou vzdáleností - to je zajištěno v následující metodě.

2.1.2.2 Metoda výměna Cmin náhodný

Jelikož víme, které dvojici bodů odpovídá hodnota EMM , aplikujeme heuristickou proceduru, ve které se snažíme měnit pozice bodů z této dvojice (tj. zajistit konec existence této příliš si blízké dvojice). Algoritmus vyměňuje jednu (náhodně zvolenou) souřadnici bodu z dvojice s EMM (který z dvojice to bude, je voleno náhodně⁸) a jiného, náhodně zvoleného bodu, dokud nedojde ke zlepšení - ke zvýšení hodnoty EMM . Poté algoritmus

⁸Zde je prostor pro vylepšení metody tak, aby byl vždy do výměny poslán „horší“ z dvojice bodů s EMM , tj. bod, jehož druhá nejkratší vzdálenost k jinému bodu je menší než u druhého z dvojice s EMM ; popsaná metoda by byla časově náročnější z důvodu nutnosti obsáhlejší analýzy vektoru vzájemných vzdáleností; u použité verze stačí nalézt minimální prvek tohoto vektoru.



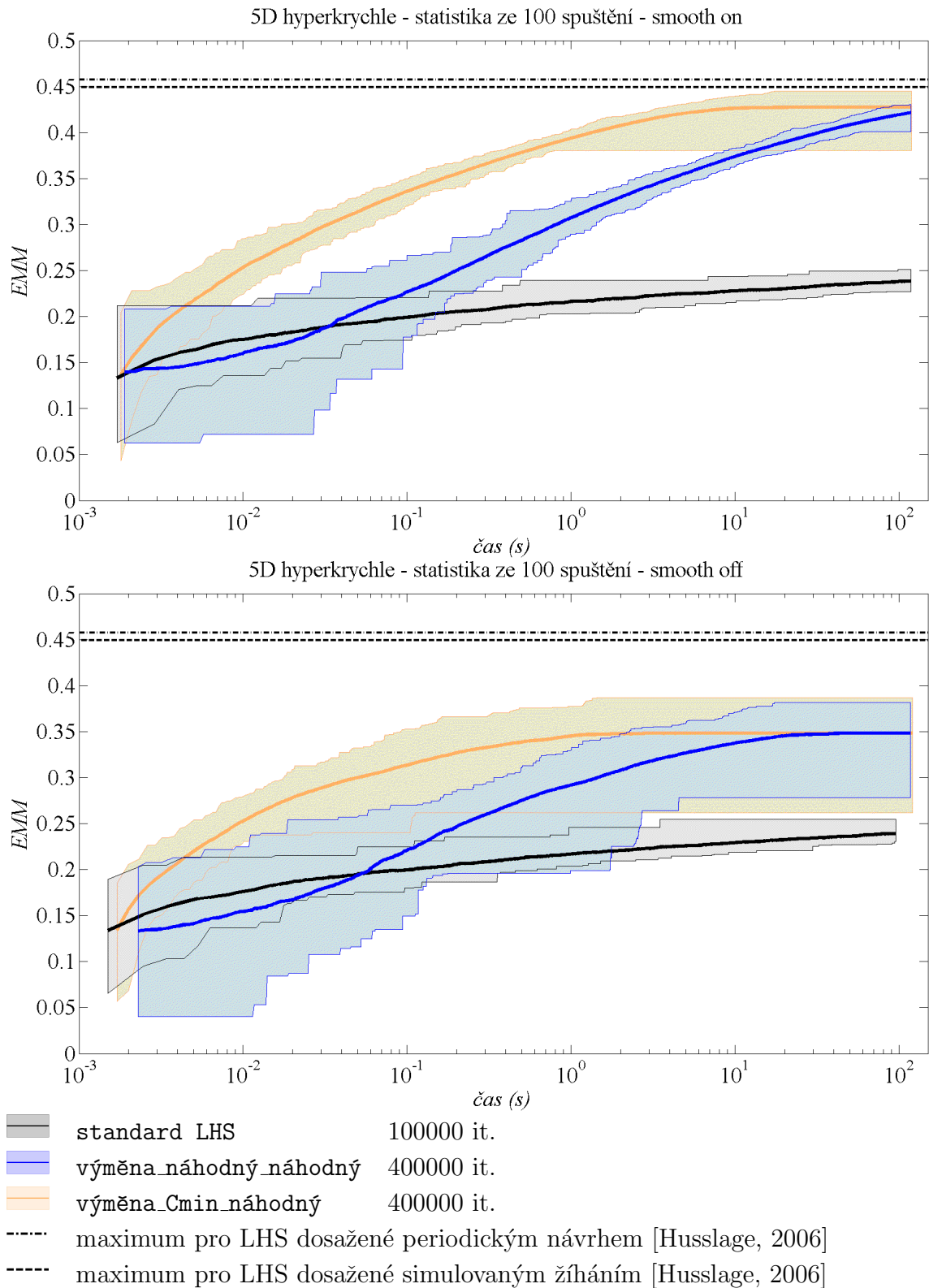
Obrázek 2.4: Výměna bodu z dvojice s *EMM* a náhodně vybraného bodu (zde na souřadnici x_2). Červená úsečka vyznačuje *EMM* před výměnou, zelená po výměně. Dochází ke zvýšení hodnoty *EMM*, tato výměna tedy bude přijata.

najde novou dvojici s minimální vzdáleností a pokračuje stejným způsobem. Metodu přibližuje Obrázek 2.4.

2.1.3 Výsledky

Obrázek 2.5 ukazuje výsledky uvedených metod v 5D, a to ve verzi `smooth on` a `smooth off`. Vidíme, že metoda `standard LHS` (černá křivka) je výrazně pomalejší (za stejný časový interval provedla čtvrtinový počet iterací), neboť v každé iteraci vytváří zcela nový návrh, zatímco metody s výměnami bodů pracují s původním jednou vytvořeným návrhem a v něm pouze mění určité souřadnice. Dále je znatelný rozdíl mezi variantami LHS návrhů s body umístěnými v rámci jednotlivých intervalů náhodně (`smooth on`) a s body ve středech intervalů (`smooth off`). Vidíme celkově nižší dosaženou hodnotu *EMM* a větší rozptyl hodnot u druhé jmenované varianty. Metoda `standard LHS` ztrácí v porovnání s metodami s výměnami bodů. Z těch je lepší metoda `výměna.Cmin.náhodný` (okrová křivka), ve které je v každé iteraci výrazně vyšší pravděpodobnost na zlepšení. Ovšem jak vidíme, při dostatečném počtu iterací se modrá křivka (`výměna.náhodná.náhodný`) té okrové přibližuje, neboť nakonec zřejmě provede ty samé výměny.

Klíč k Obrázku 2.5: tenké barevné křivky znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty ze 100 spuštění.



Obrázek 2.5: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritmu s fixním počtem bodů splňujících podmínky LHS na 5D doméně. Obrázek nahoře - **smooth on** (náhodně v rámci intervalu), obrázek dole - **smooth off** (uprostřed intervalu).

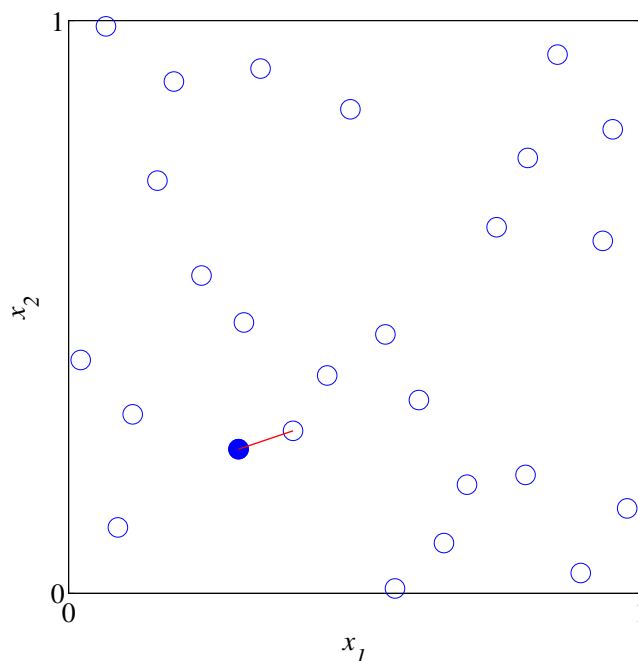
2.2 Metody nesplňující podmínky LHS

2.2.1 Odebírání z přehuštěného LHS návrhu

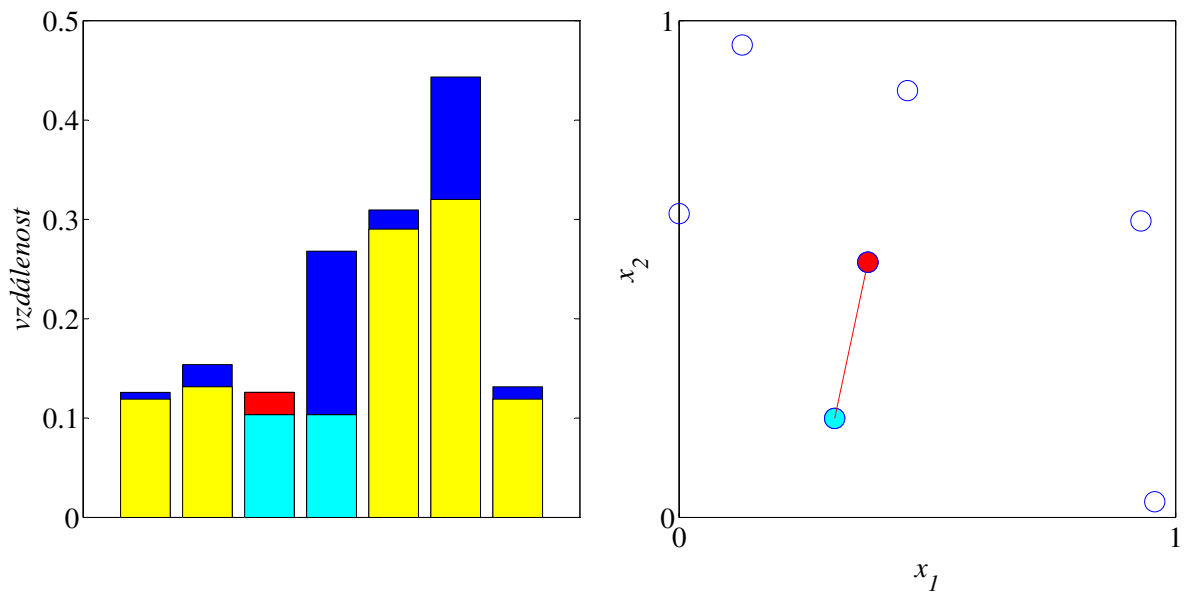
Výsledné návrhy vytvářené následujícími dvěma metodami podmínky LHS nesplňují. Metody však využívají rychlosti tvorby LHS návrhů a ty jsou tak použity jako výchozí pro tyto metody. Je vytvořen LHS návrh s vyšším počtem bodů, než je požadováno pro výsledný návrh, a z takto přehuštěného návrhu opakovaně odebírány body až do dosažení požadovaného počtu bodů.

2.2.1.1 Metoda ubírání

V přehuštěném LHS návrhu (počet výchozích bodů, stejně jako požadovaný počet bodů ve výsledném návrhu je volen uživatelem) je nalezena dvojice bodů, které odpovídá hodnota EMM , tedy dvojice bodů s nejkratší vzájemnou vzdáleností. Právě jeden z těchto bodů je z návrhu odstraněn. Je tak zajištěno zlepšení hodnoty EMM v každé iteraci, tj. vždy po odstranění bodu. Který z dvojice bodů bude odebrán, je voleno náhodně. Metodu přibližuje Obrázek 2.6. Jak je patrné i na tomto obrázku, může nastat situace, při níž je zřejmé, že by bylo vhodnější odebrat druhý z dvojice bodů s minimální vzdáleností. V následující metodě je proto volbě mezi těmito dvěma body věnována větší pozornost.



Obrázek 2.6: Odebrání náhodně zvoleného bodu z dvojice s EMM . Modrý bod = náhodně zvolený z dvojice s EMM - bude odebrán, Červená úsečka = EMM .



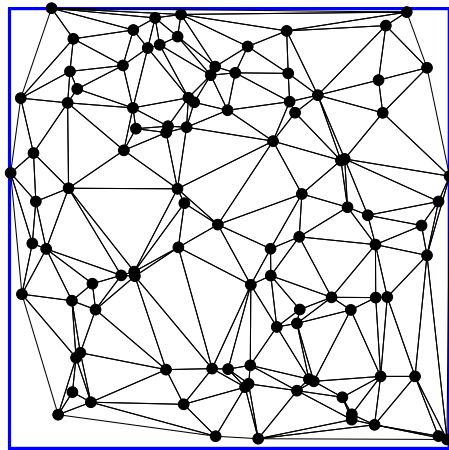
Obrázek 2.7: Odebrání „horšího“ bodu z dvojice s *EMM*. Graf vlevo znázorňuje pro každý bod jeho první a druhou nejkratší vzdálenost k jinému bodu. Azurové sloupce značí minimální vzájemnou vzdálenost (*EMM*), červený sloupec označuje kratší z druhých nejkratších vzdáleností dvojice bodů s *EMM* - odpovídající bod bude odebrán. Klíč: Žluté sloupce = první nejkratší vzdálenosti k jinému bodu, Modré sloupce = druhé nejkratší vzdálenosti k jinému bodu, Azurové sloupce = *EMM*, Červený sloupec = kratší z druhých nejkratších vzdáleností dvojice bodů s *EMM*, Azurový bod = „lepší“ z dvojice s *EMM*, Červený bod = „horší“ z dvojice s *EMM* - bude odebrán, Červená úsečka = *EMM*.

2.2.1.2 Metoda ubírání_{NEW}

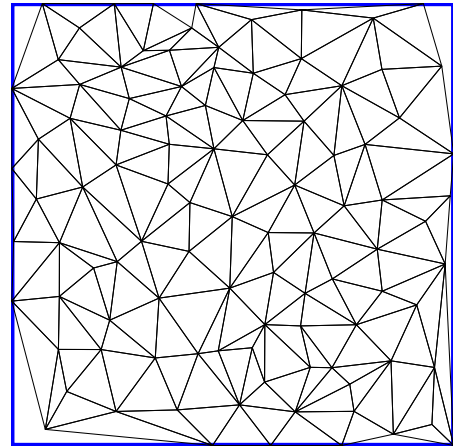
V principu se tato metoda shoduje s metodou předchozí. Liší se však tím, jak je vybírán bod z dvojice s minimální vzájemnou vzdáleností (*EMM*), který bude z návrhu odebrán. V předchozí metodě je bod vybrán náhodně. Je však patrné, že výsledný návrh může dosáhnout vyšší kvality, pokud budeme z dvojice odebírat bod „horší“. Za takový je v této metodě považován ten, jehož druhá nejkratší vzdálenost k ostatním bodům návrhu je menší. Je tedy nalezena dvojice bodů s nejkratší vzájemnou vzdáleností a pro každý z těchto bodů zjištěna ještě druhá nejkratší vzdálenost. Princip metody ilustruje Obrázek 2.7.

2.2.2 Nástroj Distmesh

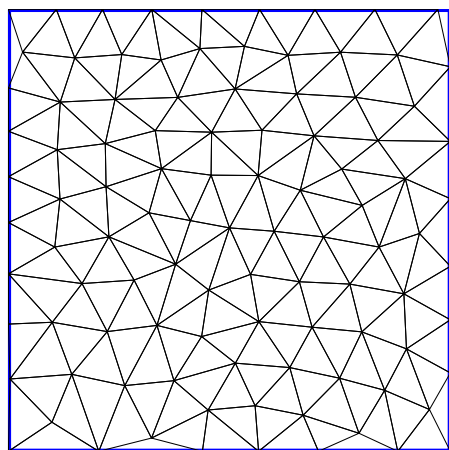
Další metoda pro tvorbu návrhů experimentů je založena na aplikaci nástroje *Distmesh* (DM) detailně popsáném v [Persson and Strang, 2004]. Jedná se o heuristický algoritmus používaný k tvorbě sítí pro metodu konečných prvků (MKP) - *Finite Element Method* (FEM). Je známé, že tyto sítě dosahují vysoké kvality, jsou-li jejich uzly rovnoměrně



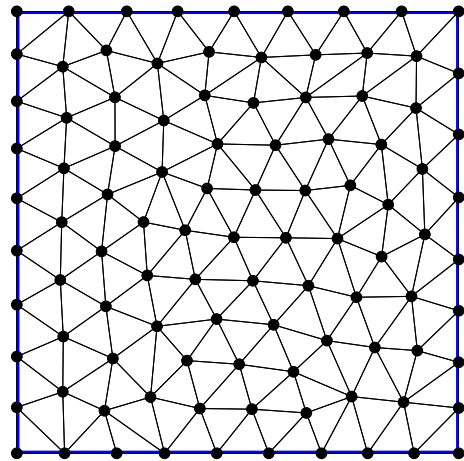
(a) Triangulace náhodných bodů vytvářející systém prutů.



(b) Stav po 5ti iteracích.



(c) Stav po 50ti iteracích.



(d) Výsledný návrh.

Obrázek 2.8: Průběh metody využívající nástroj *Distmesh*.

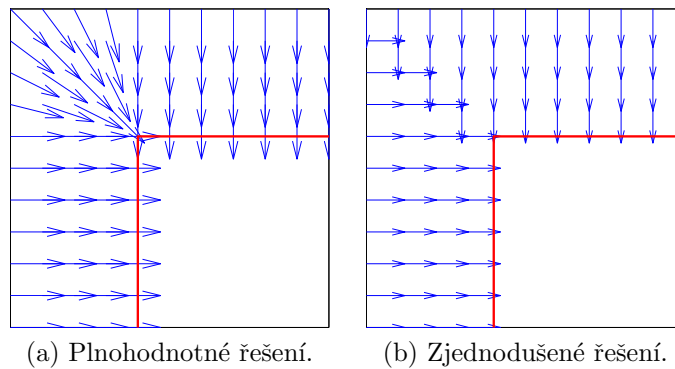
rozprostřené. Proto nástroj *Distmesh* používáme i k tvorbě návrhů experimentů.

Výchozí body jsou ztriangulovány pomocí Delaunayovy triangulace⁹ [Del, 2001], čímž se vytvoří prutová konstrukce. Algoritmus následně posunuje příliš si blízké uzly (spojené příliš krátkými pruty) směrem od sebe tak, aby pruty výsledné konstrukce byly stejně dlouhé¹⁰. Obrázek 2.8 ilustračně znázorňuje průběh metody. V průběhu dochází k opakovaní triangulace pro zajištění fyzikální konzistence vyvíjející se příhradové konstrukce.

Body, které během rozpínání opustí hranice přípustné domény, je nutné vrátit zpět. DM obsahuje návratové procedury pro jednoduché objekty, v případě hyperkrychle použí-

⁹Delaunayova triangulace je často používaným typem triangulace. Doména tvořená návrhovými body je v ní rozdělena na simplexu tak, že hyperkoule opsaná jednotlivým simplexem tvořeným body návrhu neobsahuje žádný další bod návrhu (viz Obrázek 3.7).

¹⁰V popisované metodě je použita varianta s parametrem *uniform*, ve které je požadována stejná délka všech prutů. Nástroj však nabízí i možnost zadání proměnné délky prutů v různých oblastech řešené domény, např. zahuštění u okrajů, otvorů apod.



Obrázek 2.9: Směr posunu bodů zpět do přípustné domény. Zjednodušená varianta je použita z důvodu úspory; pro plnohodnotné řešení je nutné zjistit, v jaké oblasti se bod nachází. Ve 2D doméně (čtverci) jsou možnosti posunu dvě: kolmo ke straně nebo směrem k vrcholu; ve 3D doméně (krychli) už jsou možnosti tři: kolmo ke stěně, kolmo k hraně nebo směrem k vrcholu; s vyšší dimenzí tak přibývá i těchto možností.

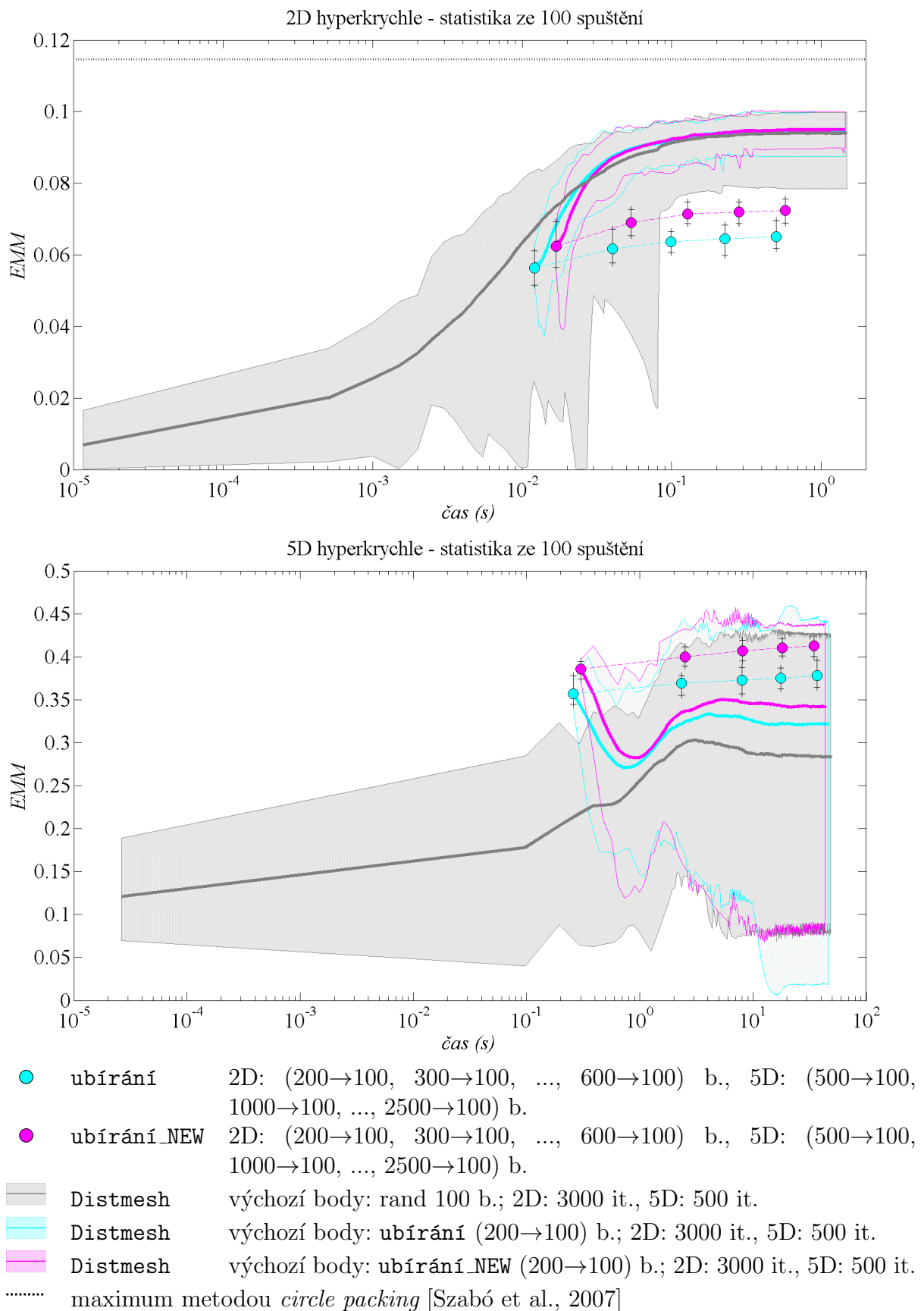
váme zjednodušené řešení znázorněné na Obrázku 2.9b.

V původní verzi nástroje `Distmesh` uživatel zadává požadovanou délku prutů a nástroj si sám vytvoří odpovídající počet výchozích bodů. V naší metodě algoritmu předkládáme již vytvořenou sadu výchozích bodů, na kterou je poté algoritmus aplikován. Vyzkoušeny byly tři varianty lišící se původem výchozí sady bodů. V prvním případě jsou body vytvořeny zcela náhodně, ve druhém případě byly vytvořeny metodou `ubírání`, ve třetím metodou `ubírání_NEW`.

2.2.3 Výsledky

Výsledky na 2D a 5D doménách ukazuje Obrázek 2.10. Je patrné, že pozornost věnovaná volbě odebraného bodu v metodě `ubírání_NEW` přináší zlepšení oproti volbě náhodně v metodě `ubírání`. Zároveň u těchto dvou metod vidíme, že ačkoli má výsledný návrh vždy 100 bodů, je patrná závislost jeho kvality (dosažené hodnoty *EMM*) na počtu bodů ve výchozím prehuštěném návrhu. Podle výsledků nástroje `Distmesh` lze konstatovat, že způsob tvorby výchozích bodů pro jeho aplikaci nemá výrazný vliv na výsledek. Nástroj dosahuje prakticky stejných výsledků bez ohledu na způsob tvorby výchozí sady bodů (náhodně vytvořené, vytvořené metodou `ubírání` nebo metodou `ubírání_NEW`). Dále vidíme velký rozdíl v použitelnosti nástroje `Distmesh` v závislosti na počtu dimenzí. V 5D je znatelné kromě poklesu kvality na začátku i určité zacyklení nástroje.

Klíč k Obrázku 2.10: ($np \rightarrow 100$) označuje 100 bodů zbylých po odebrání z np původně vytvořených; tenké barevné křivky a znaménka $+$ znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty ze 100 spuštění.



Obrázek 2.10: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritmů s fixním počtem bodů nesplňujících podmínky LHS na 2D a 5D doménách.

Kapitola 3

Sekvenční metody

V této kapitole se budeme zabývat metodami potřebnými v situacích, při nichž nezískáme dostatečné informace o chování testovaného systému ani po provedení předem stanoveného počtu experimentů. Je tedy nutné provést experimenty další. Jednou možností je vytvořit nový návrh experimentu s vyšším počtem bodů. To je ovšem nevhodné v případě, že samotné provedení experimentu je nákladné (ať už finančně nebo časově). V takovém případě chceme již získané informace využít a původní návrh pouze doplnit dalšími body (kombinacemi vstupních parametrů pro provedení experimentu). Budeme se tedy věnovat metodám, které umožňují přidání bodů do již vytvořených návrhů.

3.1 Metody splňující podmínky LHS¹¹

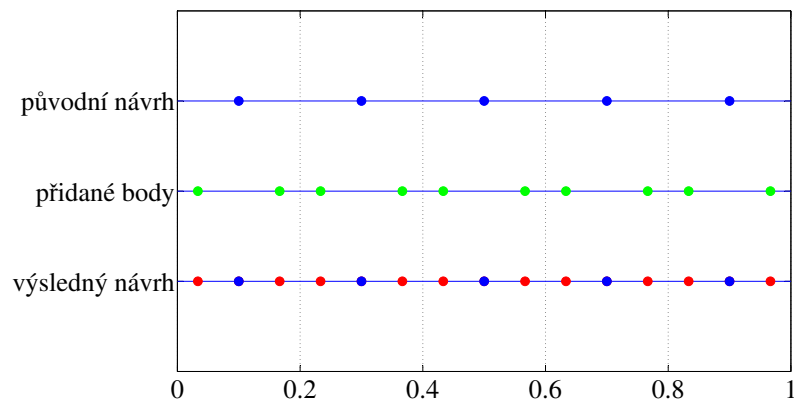
3.1.1 Metoda `LHS_sequential`

Tato metoda je schopná přidávat k již vytvořenému LHS návrhu další body tak, že i výsledný návrh splňuje podmínky LHS [Vořechovský, 2009]. Rozlišujeme 2 varianty podle toho, zda byl výchozí návrh vytvořen metodou LHS s body umístěnými ve středu jednotlivých intervalů (v MATLABu parametr `smooth off`), nebo náhodně v rámci jednotlivých intervalů (v MATLABu parametr `smooth on`). Metoda je kombinována s přístupem `výměna_Cmin_náhodný`, jak bude popsáno níže.

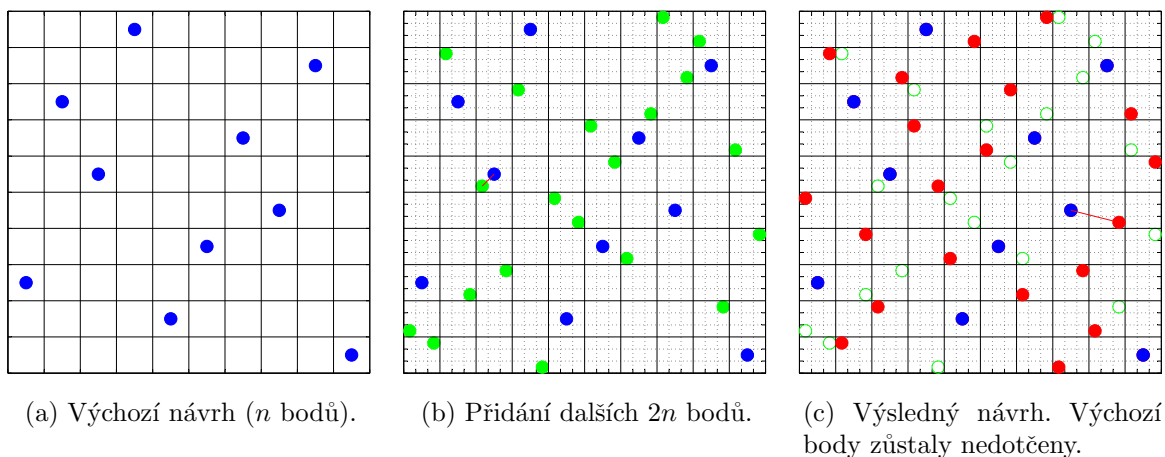
3.1.1.1 `smooth off` - trojnásobek počtu bodů

V první variantě metody je původní LHS návrh vytvořen s parametrem `smooth off`, body jsou tedy umístěny do středů jednotlivých intervalů. Aby byly i po přidání bodů zachovány podmínky LHS, je nutné přidat dvojnásobek počtu bodů tvořících původní návrh. Výsledný návrh bude mít tedy trojnásobný počet bodů než návrh původní. Každý inter-

¹¹Část této sekce byla prezentována příspěvkem v [Myšáková and Lepš, 2012b].

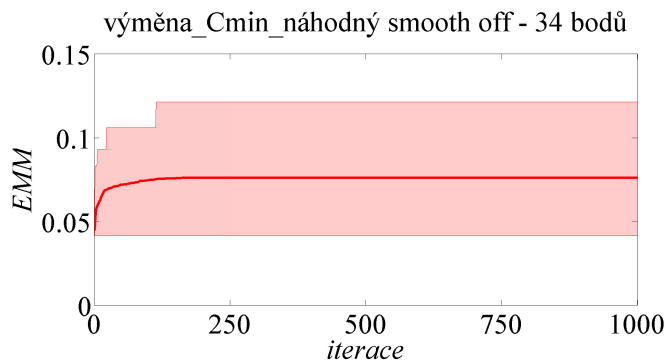


Obrázek 3.1: Metoda `LHS_sequential` verze `smooth off`. Obrázek znázorňuje princip metody na jedné proměnné (dimenzi).



Obrázek 3.2: Průběh metody `LHS_sequential` verze `smooth off`. Červená úsečka značí minimální vzdálenost mezi body návrhu (*EMM*).

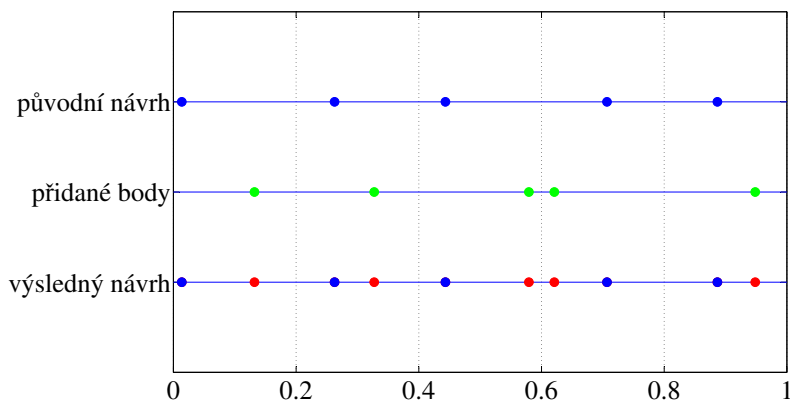
val každé proměnné původního LHS návrhu (na Obrázcích 3.2a, 3.1 znázorněn modrými body) je rozdělen na třetiny. Jelikož jsou původní body umístěny do středů intervalů, je vždy druhý z vzniklých tří podintervalů obsazen původním bodem, zatímco dva krajní podintervaly jsou volné a mohou do nich být přidány body nové. Následně je vytvořen nový LHS návrh s dvojnásobkem počtu původních bodů a ten distribuován do připravených volných intervalů (na Obrázcích 3.2b, 3.1 znázorněn zelenými body). K zaručení určité kvality takto vzniklého doplněného návrhu je následně použita metoda `výměna_C-min_náhodný` popsaná v 2.1.2.2, ovšem s tím omezením, že do výměn mohou vstupovat pouze body přidané, zatímco body původního návrhu zůstávají na svých pozicích. Výsledné návrhy jsou znázorněny na Obrázcích 3.2c, 3.1. Metoda byla testována tak, že byl nejprve vytvořen návrh s 34 body pomocí metody `výměna_Cmin_náhodný` popsané



Obrázek 3.3: Tvorba výchozího návrhu pro metodu `LHS_sequential` verze `smooth off`. Graf závislosti metriky EMM na počtu iterací použitých v metodě `výměna_Cmin_náhodný` pro tvorbu 34 bodů ve 2D. Graf obsahuje výsledky ze 100 spuštění, tenká horní, resp. dolní linie odpovídá maximální, resp. minimální hodnotě metriky EMM v dané iteraci, silná linie odpovídá hodnotě průměrné.

v 2.1.2.2¹² a následně přidány body pomocí zde popsané metody. U této metody mají tedy výsledné návrhy 102 bodů.

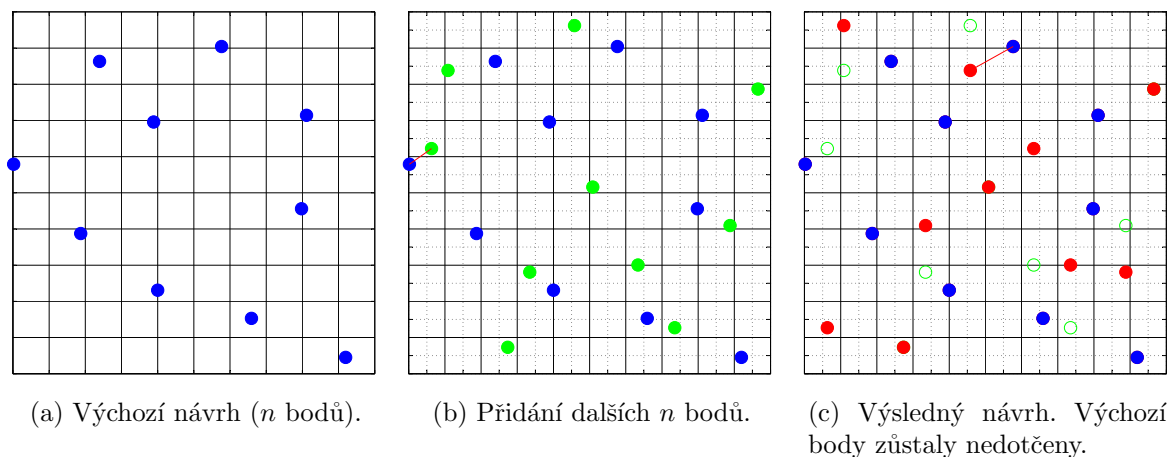
3.1.1.2 smooth on - dvojnásobek počtu bodů



Obrázek 3.4: Metoda `LHS_sequential` verze `smooth on`. Obrázek znázorňuje princip metody na jedné proměnné (dimenzi).

Druhá varianta metody se liší tím, že původní LHS návrh je vytvořen s parametrem `smooth on`, body jsou tedy umístěny v rámci jednotlivých intervalů náhodně. Aby byly i po přidání bodů zachovány podmínky LHS, je nutné přidat počet bodů stejný jako

¹²Aby byl výchozí návrh sám o sobě dostatečně kvalitní (neboť ve skutečnosti by byl původní „menší“ návrh rovněž optimalizován), volba počtu iterací pro jeho tvorbu vychází z Obrázku 3.3. Je patrné, že vhodná volba počtu iterací je cca 250. Vyšší počet již nepřináší zlepšení. Takováto analýza je provedena pro volbu počtu iterací pro všechny příklady (2D - 5D) variant `smooth off` i `smooth on`.



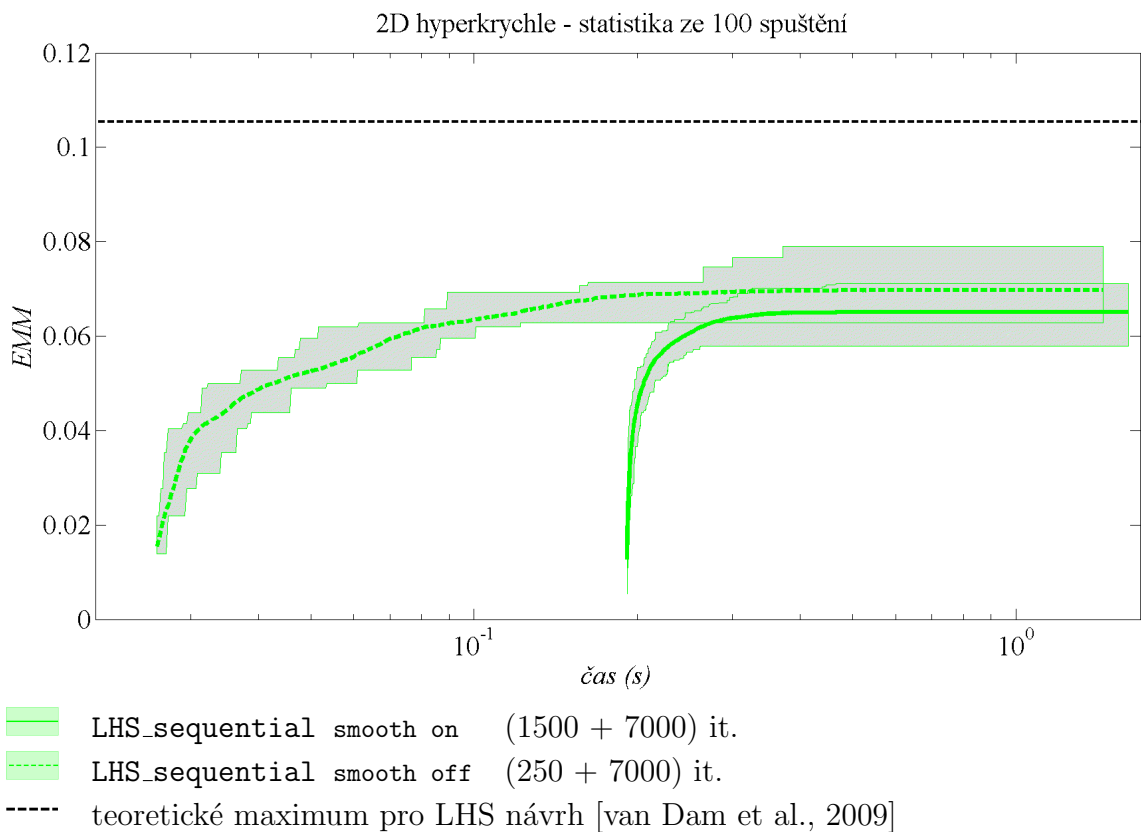
Obrázek 3.5: Průběh metody `LHS_sequential` verze `smooth on`. Červená úsečka značí minimální vzdálenost mezi body návrhu (EMM).

měl původní návrh. Výsledný návrh bude mít tedy dvojnásobný počet bodů než návrh původní. Každý interval každé proměnné původního LHS návrhu (na Obrázcích 3.5a, 3.4 znázorněn modrými body) je rozdělen na poloviny. Jelikož jsou původní body umístěny v intervalech náhodně, je nutné určit, které z vzniklých podintervalů jsou již obsazené původními body a které jsou volné pro body nové. Následně je vytvořen nový LHS návrh, který má stejný počet bodů jako má původní návrh, a ten je distribuován do připravených volných intervalů (na Obrázcích 3.5b, 3.4 znázorněn zelenými body). Poté je použita metoda `výměna_Cmin_náhodný` popsaná v 2.1.2.2, ovšem s tím omezením, že do výměn mohou vstupovat pouze body přidané, zatímco body původního návrhu zůstávají na svých pozicích. Výsledné návrhy jsou znázorněny na Obrázcích 3.5c, 3.4. Metoda byla testována tak, že byl nejprve vytvořen návrh s 50 body pomocí metody `výměna_Cmin_náhodný` popsané v 2.1.2.2 a následně přidány body pomocí zde popsané metody.

3.1.2 Výsledky

K ilustraci výsledků metod slouží Obrázek 3.6. Ukazuje výsledky na 2D doméně. Ve variantě `smooth on` bylo nejprve vytvořeno 50 bodů a následně přidáno dalších 50, ve variantě `smooth off` měl výchozí návrh 34 bodů a přidáno bylo bodů 68. Jak naznačují výsledky v Sekci 2.1.3, `smooth off` varianty LHS návrhů dosahují nižších výsledných hodnot. To zde neplatí, pravděpodobně proto, že `smooth on` varianta má vyšší počet výchozích (následnými výměnami nedotknutelných) bodů a nemá tak dostatečný prostor k dosažení vyšší hodnoty pomocí metody `výměna_Cmin_náhodný`.

Klíč k Obrázku 3.6: tenké barevné křivky znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty ze 100 spuštění.



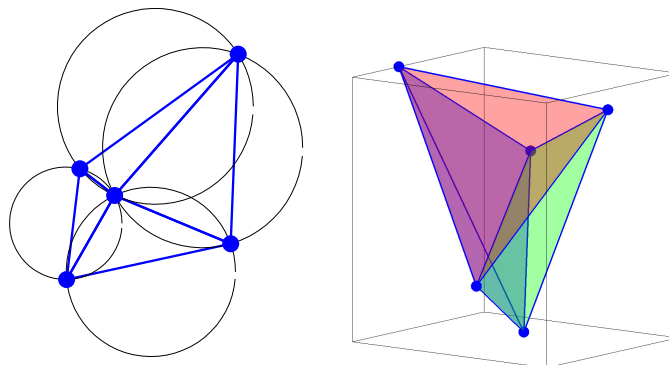
Obrázek 3.6: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) sekvenčních algoritmů splňujících podmínky LHS na 2D doméně.

3.2 Metody nesplňující podmínky LHS¹³

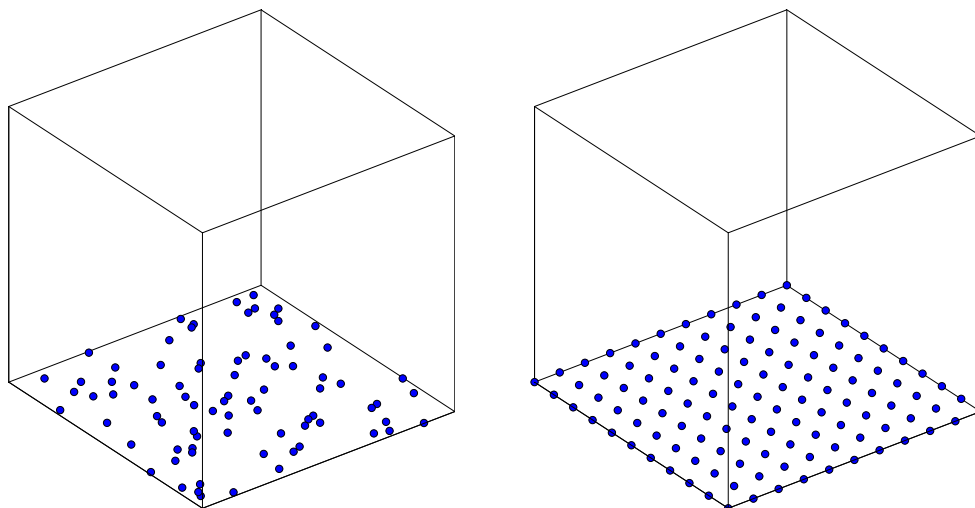
3.2.1 Metody založené na Delaunayově triangulaci

Další skupina metod je založena na Delaunayově triangulaci (Obrázek 3.7) přípustné domény (*triangulace*, o níž jsme se již krátce zmínili v sekci o nástroji *Distmesh*, je termín

¹³Část této sekce byla publikována v [Myšáková and Lepš, 2011].



Obrázek 3.7: Delaunayova triangulace ve 2D (vlevo) a ve 3D (vpravo).



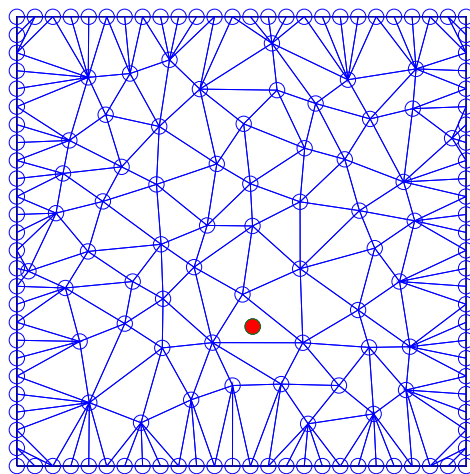
Obrázek 3.8: Dělení stěny 3D domény. Na prvním obrázku je LHS návrh, na druhém plně faktoriální návrh.

vhodný pro 2D, obecně jde o rozdělení domény na simplexu) [Crombecq et al., 2009]. Díky relativně jednoduchému výpočtu objemu simplexu můžeme získat hrubý odhad toho, kde je největší nevyplněný prostor v řešené doméně. Postup výpočtu objemu simplexu je uveden v Příloze A. Ve všech variantách metody začínáme výchozím LHS návrhem, následně je provedena Delaunayova triangulace [Del, 2001], v každém kroku nalezen simplex s největším objemem a do jeho těžiště umístěn nový bod návrhu (viz Příloha B o výpočtu souřadnic těžiště simplexu). Tento postup je opakován, dokud není docíleno požadovaného počtu návrhových bodů.

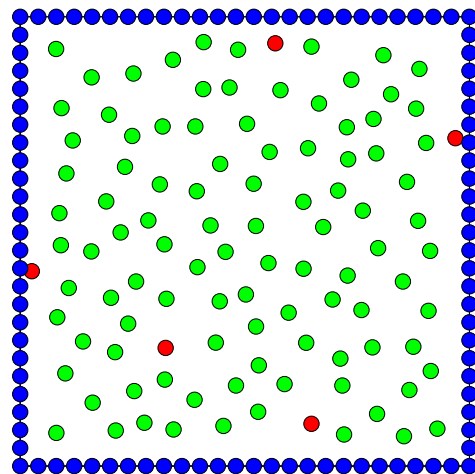
Zároveň je řešena otázka způsobu a hustoty dělení hran (ve 2D), stěn (ve 3D), respektive $(n - 1)$ -facet (viz Příloha C) v řešené doméně; n značí počet dimenzí. Důležitost správné hustoty dělení hran dokumentuje Obrázek 3.9. Co se týče způsobu dělení, porovnali jsme dvě varianty - v jedné byly hraniční objekty pokryty LHS návrhy, v druhé byly použity tzv. plně faktoriální (*fullfact*) návrhy (viz Obrázek 3.8). Tento způsob se ukázal jako výhodnější. Rozdělení hraničních objektů je prováděno dynamicky během výpočtu v závislosti na aktuálním počtu bodů ve vytvářeném návrhu.

3.2.1.1 Stálá triangulace

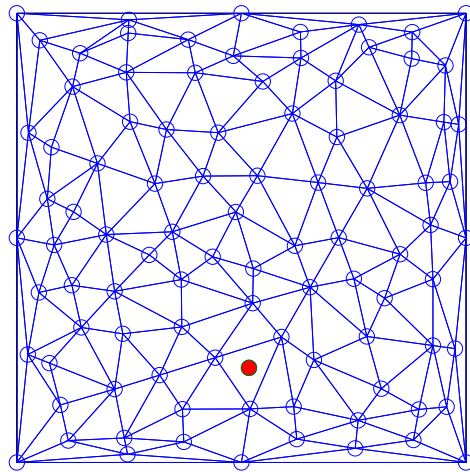
První varianta metody poskytuje nejlepší představu o tom, které oblasti domény jsou dosud málo pokryté a kam je tedy nejvhodnější přidat nový bod návrhu. V každé iteraci je provedena Delaunayova triangulace, nalezen simplex s největším objemem a do jeho těžiště přidán nový návrhový bod. Postup dokumentuje Obrázek 3.10. Hraniční objekty jsou pokryty pomocnými plně faktoriálními návrhy s každou osou rozdělenou na intervaly, jejichž počet odpovídá spodní celé části n -té odmocniny z aktuálního počtu bodů v návrhu,



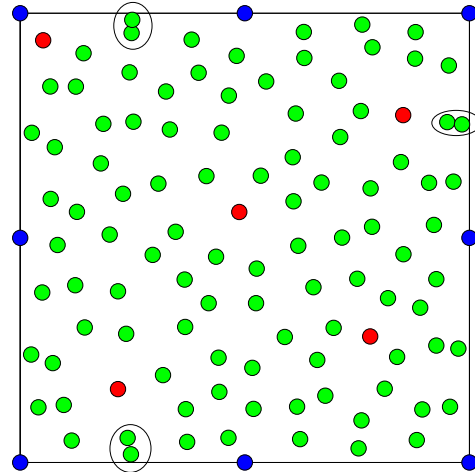
(a) Použití hustého dělení hrany. Vznikají typické úzké, do středu domény orientované simplexxy.



(b) Výsledný obrázek po vytvoření celého návrhu. Vlivem příliš hustého dělení hrany nedojde k dostatečnému pokrytí okrajových částí domény.



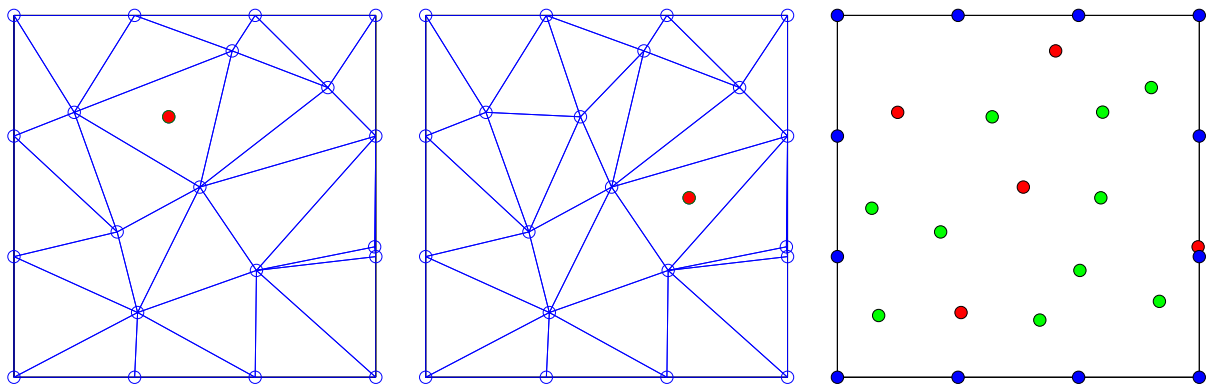
(c) Použití řídkého dělení hrany. Vznikají typické široké simplexxy podél hran domény.



(d) Výsledný obrázek po vytvoření celého návrhu. Vlivem příliš řídkého dělení hrany mohou vznikat shluky v okrajových částech domény.

Obrázek 3.9: Rozdíl v hustotě dělení hran. Použita varianta plně faktoriálního návrhu - hrana je pravidelně rozdělena. Pro Obrázky 3.9b, 3.9d: modré body označují hranice domény (jsou jen pomocné během výpočtu, nejsou součástí výsledného návrhu), červené body byly vytvořeny LHS metodou, zelené přidány v každé iteraci do těžiště simplexu s největším objemem.

kde n je dimenze návrhového prostoru. Provádění Delaunayovy triangulace v každém kroku je ovšem, zvláště ve vyšších dimenzích, výpočetně náročné. V následující variantě je proto retriangulováno jen po určitém počtu iterací.

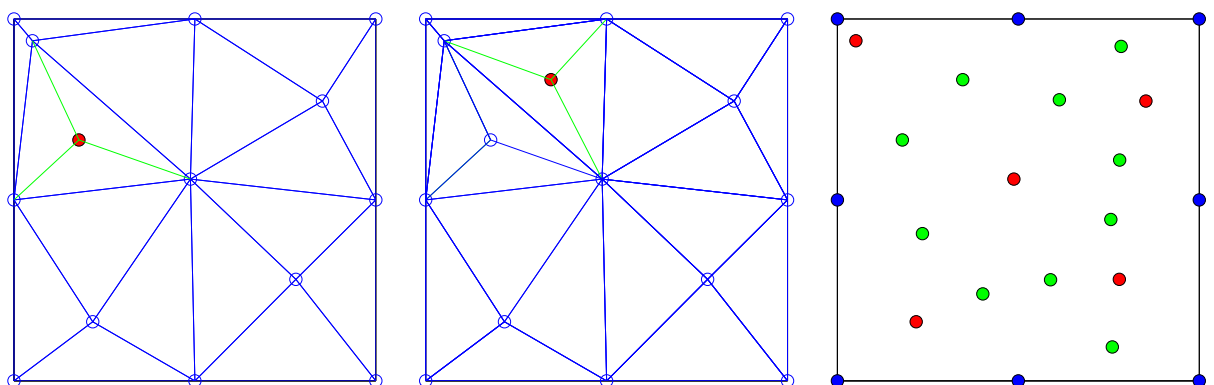


(a) Červený bod znázorňuje těžiště největšího simplexu - bude přidán jako nový bod návrhu.

(b) Následující krok. Došlo k retriangulaci, opět vyznačen bod v těžišti největšího simplexu.

(c) Výsledný obrázek po vytvoření celého návrhu.

Obrázek 3.10: První varianta metody založené na Delaunayově triangulaci (ve 2D) - po přidání bodu je provedena nová triangulace celé přípustné domény. Význam barev v Obrázku 3.10c: modré body označují hranice domény (jsou jen pomocné během výpočtu, nejsou součástí výsledného návrhu), červené body byly vytvořeny LHS metodou, zelené přidány v každé iteraci do těžiště simplexu s největším objemem.



(a) Červený bod znázorňuje těžiště největšího simplexu - bude přidán jako nový bod návrhu. Zelené úsečky vyznačují rozdělení simplexu na 3 nové.

(b) Následující krok. Nedošlo k retriangulaci. Opět vyznačen bod v těžišti největšího simplexu a rozdělení tohoto simplexu.

(c) Výsledný obrázek po vytvoření celého návrhu.

Obrázek 3.11: Druhá varianta metody založené na Delaunayově triangulaci (ve 2D) - po přidání bodu není provedena nová triangulace celé přípustné domény, největší simplex je pouze rozdělen. Význam barev v Obrázku 3.11c: modré body označují hranice domény (jsou jen pomocné během výpočtu, nejsou součástí výsledného návrhu), červené body byly vytvořeny LHS metodou, zelené přidány v každé iteraci do těžiště simplexu s největším objemem.

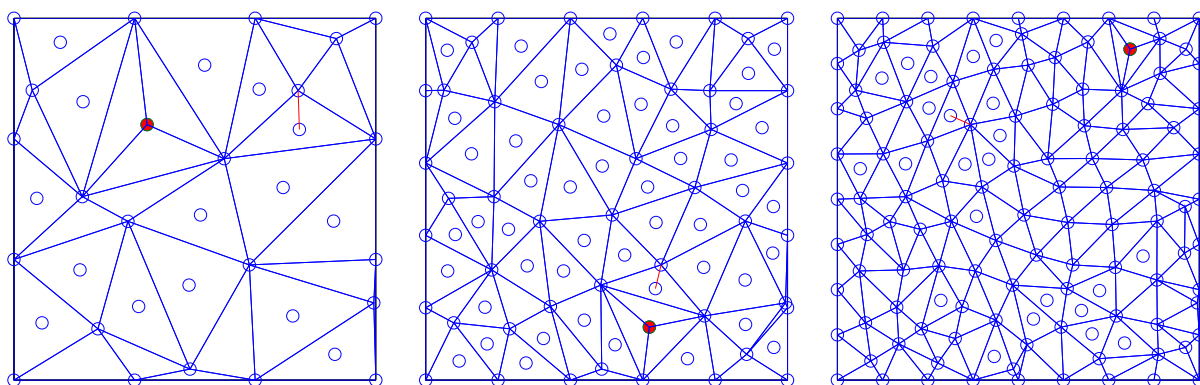
3.2.1.2 Rozdělování

Ve druhé variantě metody je retriangulace prováděna jen po určitém počtu iterací. V ostatních krocích je simplex, do jehož těžiště byl přidán nový bod, pouze rozdělen na $(n + 1)$ simplexů, které jsou považovány za plnohodnotné nové simplexu pro následný výpočet objemu¹⁴; n značí počet proměnných, tj. dimenzí. Tato varianta metody je znázorněna na Obrázku 3.11. Způsob a hustota dělení hraničních objektů jsou stejné jako u předchozí varianty metody. K retriangulaci dochází zároveň se změnou v hustotě dělení hraničních objektů (tedy v závislosti na aktuálním počtu bodů ve vytvářeném návrhu). Zda neexistuje vhodnější volba okamžiku pro retriangulaci se dozvíme vzápětí.

3.2.1.3 Heuristická procedura pro rozdělování

U varianty algoritmu, kde dochází k retriangulaci až po určitém počtu kroků, je stěžejní právě toto určení - kdy retriangulovat. V základní verzi je retriangulace provedena zároveň se změnou hustoty dělení hraničních objektů. Je však patrné, že kvalita návrhu klesá v okamžiku, kdy je rozdělen simplex, který již sám vznikl rozdělením. Proto je zavedena heuristická procedura zaručující, že takový simplex již rozdělen nebude. Jsou tedy rozděleny (přidáním nového bodu do těžiště) ty simplexu, jejichž objem je větší než $1/(n + 1)$ objemu největšího (prvního rozděleného po dosud poslední provedené triangulaci) simplexu, kde n značí počet dimenzí (Obrázek 3.12). Teprve po rozdělení všech

¹⁴Může dojít k situaci, kdy je simplex s největším objemem tak velký, že po jeho rozdělení jsou i jeho části při hledání dalšího simplexu pro přidání bodu vyhodnoceny jako největší v doméně.

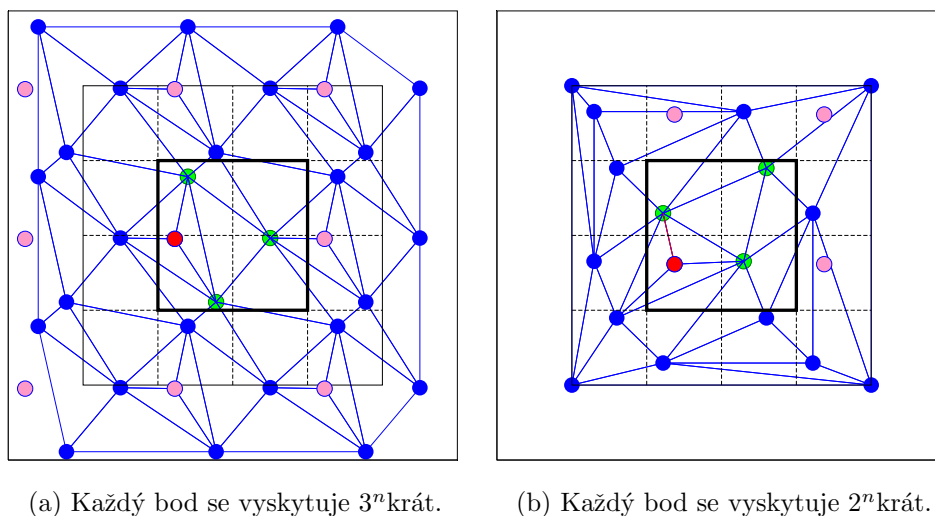


(a) Červený bod značí těžiště největšího simplexu - prvního rozděleného. Nové body jsou přidány do těžišť všech simplexů, jejichž objem je větší než $1/3$ (ve 2D) největšího ze simplexů (prvního rozděleného).

(b) Následně provedena retriangulace a opakován postup.

(c) Opět provedena retriangulace a opakován postup - zde již je přidáno jen takové množství bodů, aby jejich celkový počet odpovídal požadavku uživatele.

Obrázek 3.12: Použití heuristické metody k určení iterace, ve které má být provedena retriangulace. Červená úsečka značí minimální vzájemnou vzdálenost bodů návrhu.



Obrázek 3.13: Použití periodických okrajových podmínek.

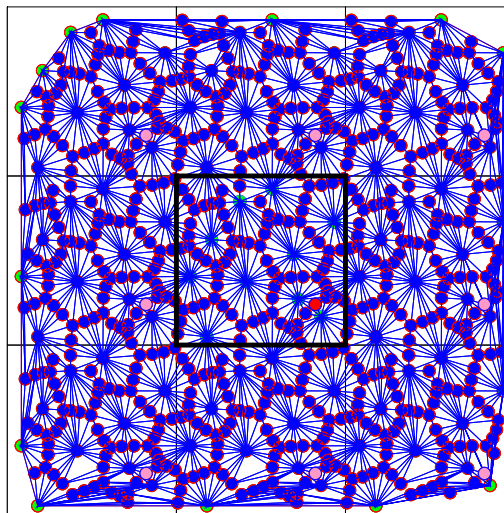
těchto přípustných simplexů je provedena retriangulace. A právě až při retriangulaci je aktualizováno i dělení hraničních objektů.

3.2.1.4 Periodické okrajové podmínky

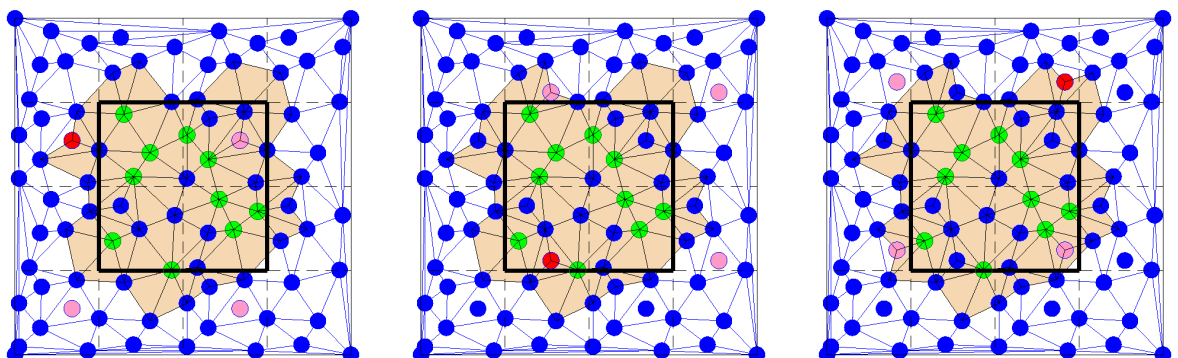
U skupiny algoritmů založené na Delaunayově triangulaci je důležitá otázka způsobu a hustoty dělení hraničních objektů řešené domény. Vyzkoušeli jsme dva způsoby - LHS návrh na hraničních objektech nebo plně faktoriální návrh (Obrázek 3.8). Tento problém lze obejít použitím periodických okrajových podmínek - *periodic boundary conditions* (PBC). Konkrétně se jedná o periodicitu bodů, nikoli sítě. Výchozí LHS návrh vytvořený v základní (cílové) buňce (tou je řešená doména - v této práci hyperkrychle $\langle 0, 1 \rangle^n$) je zkopírován do sousedních buněk, následně je provedena Delaunayova triangulace všech těchto bodů a metoda pokračuje hledáním simplexu, do jehož těžiště má být přidán nový bod. Ten je také zkopírován do okolních buněk. Nejprve byla zkoušena varianta, kdy se každý bod vyskytuje 3^n krát (Obrázek 3.13a), tato varianta je však časově náročnější a její výsledky nejsou lepší než u varianty, ve které se každý bod vyskytuje jen 2^n krát (Obrázek 3.13b), kde n značí počet dimenzí. Byla vyzkoušena celá řada verzí lišících se v tom, jak často je prováděna retriangulace, zda je největší simplex (do jehož těžiště má být přidán nový bod) vyhledáván v celé doméně, nebo jen v základní buňce, zda jsou rozdělovány všechny simplexu, do nichž je vložen nový bod, nebo jen ten ležící v základní buňce atd. Pro porovnání bylo vybráno 6 verzí, tři (označeny 1, 2, 5) jsou založeny na principu stálé triangulace, tři (označeny 12, 15, 18) pak vycházejí z metody s rozdělováním simplexů. Ilustrační Obrázek 3.14 znázorňuje variantu, v níž je provedena pouze první triangulace a nadále jsou simplexu pouze rozdělovány. Na obrázku jsou jasně patrné původní, LHS metodou vytvořené body, kolem nichž vznikají nové body návrhu. Z hlediska met-

riky *EMM* je však takový návrh zcela nevhodný. Obrázek tak naznačuje i motivaci pro vytvoření heuristické procedury popsané v předchozí sekci. Na Obrázku 3.15 vidíme verzi metody, ve které je největší simplex hledán pouze mezi simplexey, které mají alespoň jeden vrchol uvnitř výchozí buňky. Rovněž rozdělovány jsou pouze tyto simplexey (označeny okrovou barvou). Můžeme si všimnout postupného zužování okrové oblasti kolem výchozí buňky.

Klíč k Obrázkům 3.13 - 3.15: Zelené body = body původně vytvořené pomocí LHS, Červený bod = těžiště největšího simplexu v dané iteraci - přidáno jako nový bod návrhu, Růžové body = „kopie“ těžiště v okolních buňkách - přidány jako nové body, Modré body = body návrhu přidané v předchozích iteracích.



Obrázek 3.14: Výstup typický pro rozdělování bez retriangulace při použití periodických okrajových podmínek ve verzi, v níž se každý bod vyskytuje 3^n krát.



Obrázek 3.15: Varianta metody s periodickými okrajovými podmínkami ve verzi s každým bodem vyskytující se 2^n krát. Simplex s největším objemem je hledán pouze mezi těmi, které mají alespoň jeden vrchol uvnitř cílové buňky.

3.2.2 Generátory

Jako metodu pro tvorbu návrhů experimentů nelze opomenout v oboru optimalizace často používané generátory pseudonáhodných čísel a kvazináhodných sekvencí. Následuje přehled obou skupin generátorů, více informací lze nalézt v [Maaranen et al., 2007] nebo v [Press et al., 2007].

3.2.2.1 Generátory pseudonáhodných čísel

Generátory pseudonáhodných čísel jsou typické používáním matematické operace *modulo* - zbytek po celočíselném dělení. Lze je dělit do dvou skupin podle toho, zda během cyklu používají hodnotu pouze z poslední, nebo z více předchozích iterací.

První skupinou jsou **kongruenční generátory**. Jejich název je odvozen z výrazu *kongruence*, jejíž nejznámější případ je spojen právě s operací *modulo*, např. číslo 3 je kongruentní s číslem 7 modulo 4, protože $(3 \bmod 4 = 3)$ a rovněž $(7 \bmod 4 = 3)$. Tyto generátory využívají hodnotu pouze z předchozí iterace. Patří sem lineární, kvadratický, inverzní, aditivní nebo paralelní lineární generátory. Jednoduchý příklad kongruenčního generátoru je dán vztahem:

$$m_i = m_{i-1} \cdot K \bmod M, \quad i = 0, 1, 2, \dots, \quad (3.1)$$

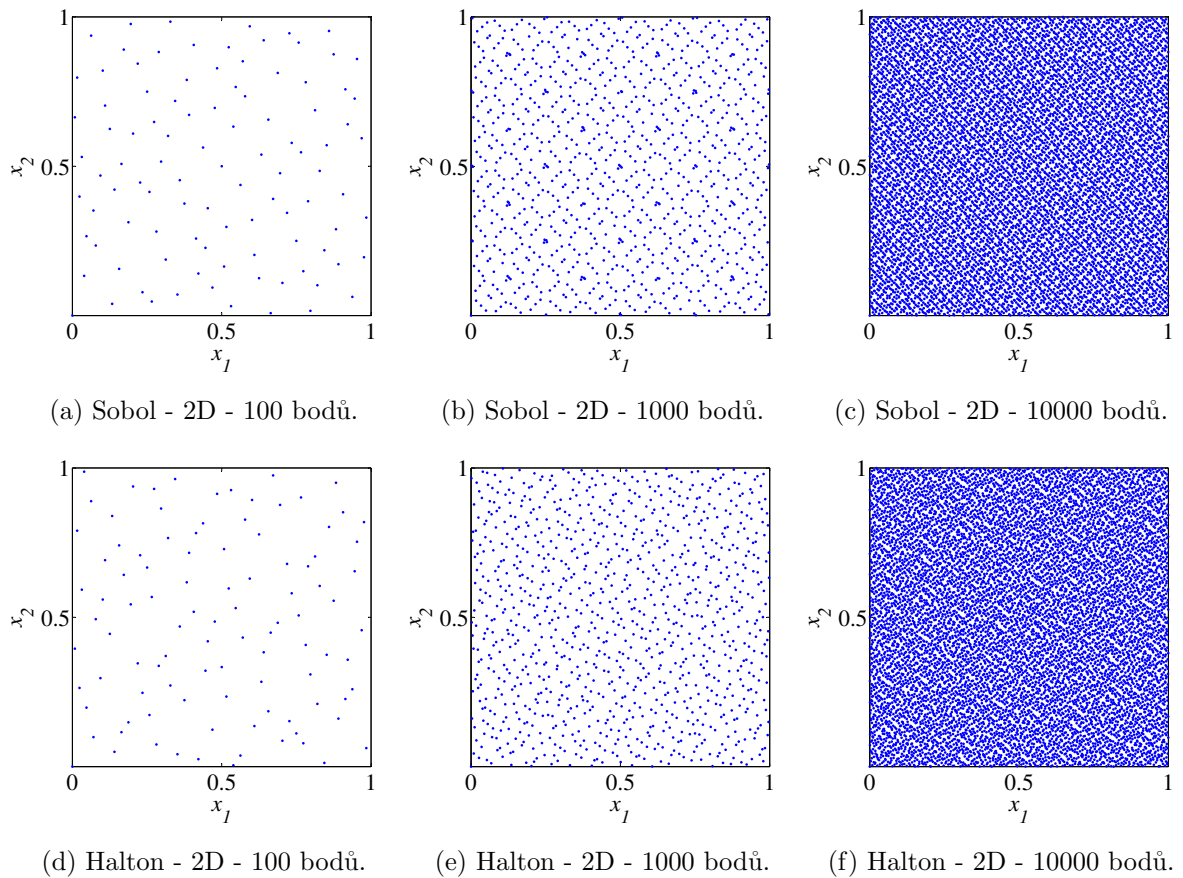
ve kterém je číslo m_i z intervalu $(0, M-1)$ vytvořeno pomocí čísla m_{i-1} z předchozí iterace. M je celé číslo, K konstanta generátoru [Klvaňa, 2005]. Náhodné číslo z intervalu $(0, 1)$ poté získáme vydělením čísla m celým číslem M .

Druhou skupinu generátorů pseudonáhodných čísel tvoří **rekurzivní generátory**. Ty počítají s hodnotami z několika předchozích iterací. Zařazujeme sem multiplikační rekurzivní, zpožděný Fibonacciho, AWC (*add-with-carry*), SWB (*subtract-with-borrow*) nebo MWC (*multiply-with-carry*) generátory. Příklad rekurzivního generátoru pak vypadá následovně:

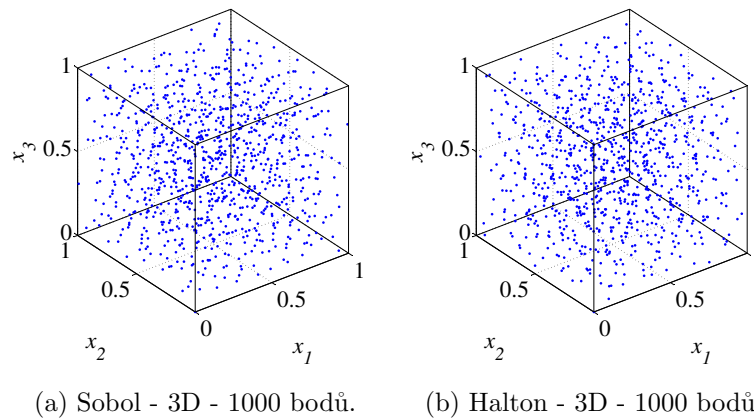
$$m_i = (m_{i-1} \cdot K_1 + \dots + m_{i-k} \cdot K_k) \bmod M, \quad i = 0, 1, 2, \dots, \quad (3.2)$$

ve kterém je číslo m_i z intervalu $(0, M-1)$ vytvořeno pomocí čísel z k předchozích iterací. M je celé číslo. Náhodné číslo z intervalu $(0, 1)$ poté získáme vydělením čísla m celým číslem M .

Ačkoli se od předchozích dvou skupin liší, zařazujeme do skupiny generátorů náhodných čísel i generátory s využitím posuvných registrů se zpětnou vazbou **feedback shift register generators**. Ty také používají operaci *modulo*, ale narozdíl od výše uvedených generátorů, ve kterých se dělí velkým přirozeným číslem, zde je M zpravidla rovno dvěma, čímž vznikají sekvence jedniček a nul.



Obrázek 3.16: Sobolova a Haltonova sekvence ve 2D.



Obrázek 3.17: Sobolova a Haltonova sekvence ve 3D.

3.2.2.2 Generátory kvazináhodných sekvencí

Druhou velkou skupinou jsou generátory kvazináhodných sekvencí. Z nich můžeme jmenovat van der Corputovu, Hammersleyovu, Haltonovu, Sobolovu, Faureho nebo Niederreiterovu sekvenci. Jejich významnou vlastností je, že pokud je prvních 100 bodů sekvence rovnoměrně rozprostřeno, pak i dalších 100 bodů sekvence je rovněž rovnoměrně

rozprostřeno. Navíc i návrh vzniklý překrytím těchto dvou dílčích návrhů je rovnoměrně rozprostřen.

Kvazináhodné sekvence se vyznačují nízkou diskrepancí a jsou navrženy tak, aby se jejich body co nejvíce „navzájem vyhýbaly“. Jsou založeny na elementárním intervalu s bází b :

$$E = \prod_{i=1}^n \left[\frac{a_i}{b^{d_i}}, \frac{a_i + 1}{b^{d_i}} \right), \quad (3.3)$$

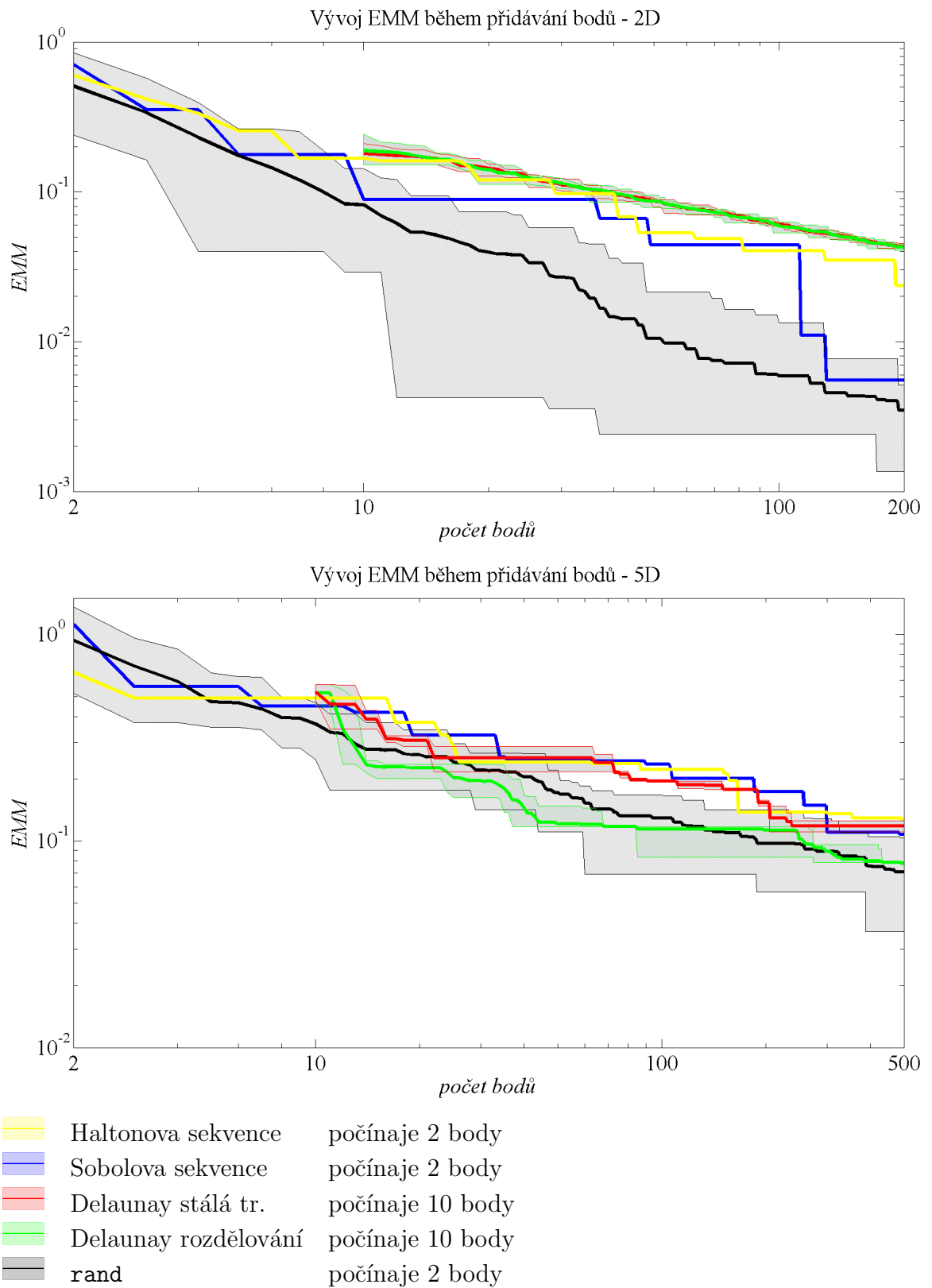
kde a_i , d_i a b jsou přirozená čísla. E je subinterval n -dimenzionální jednotkové hyperkrychle. Sobolova sekvence má bází rovnou dvěma, báze Haltonovy sekvence je prvočíslo.

Program MATLAB nabízí funkce k tvorbě těchto dvou sekvencí. Prvních 100, 1000 a 10000 bodů sekvencí ve 2D ukazuje Obrázek 3.16, Obrázek 3.17 potom prvních 1000 bodů sekvencí ve 3D.

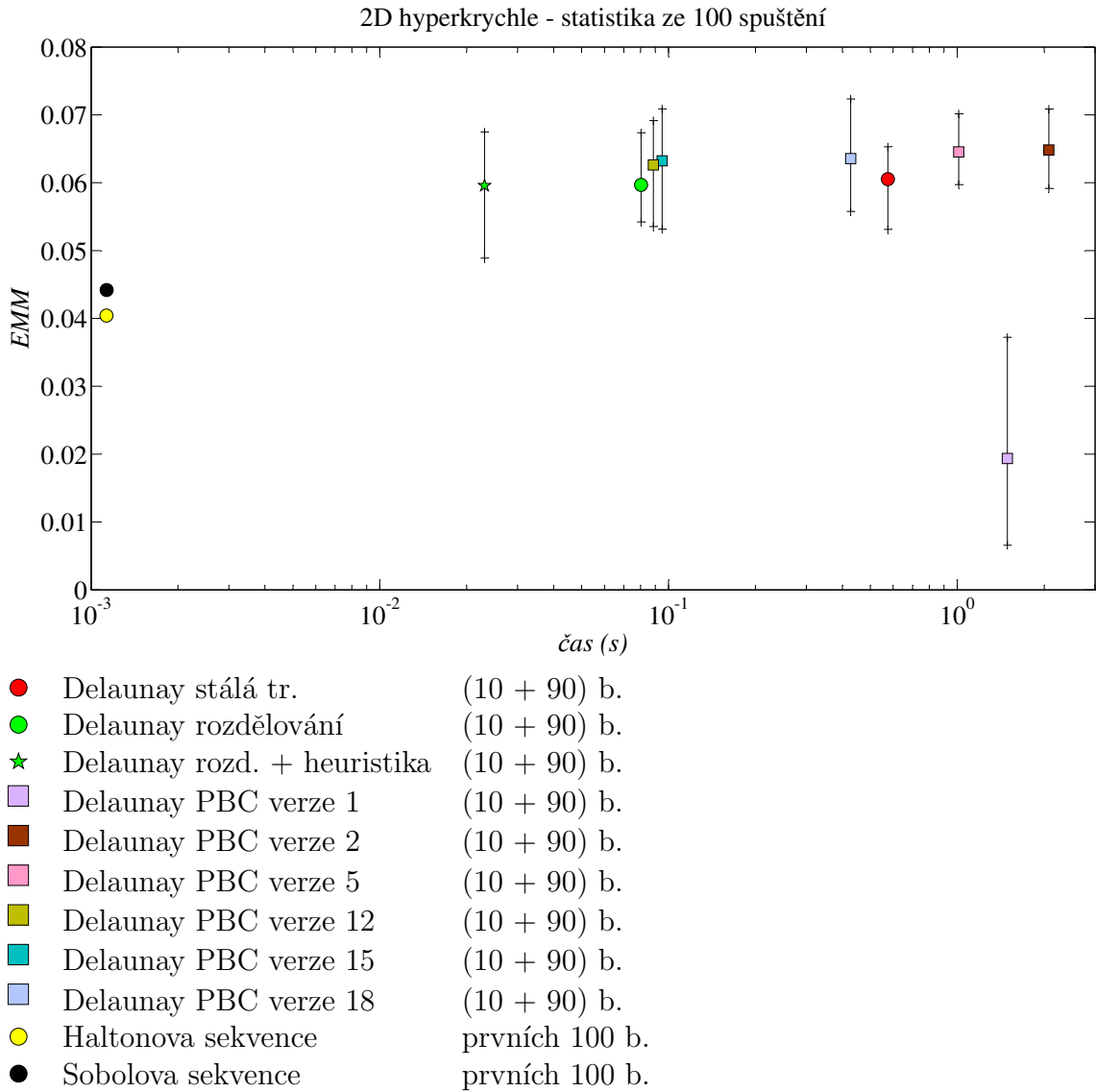
3.2.3 Výsledky

Obrázek 3.18 ukazuje vývoj hodnoty EMM v průběhu přidávání návrhových bodů na 2D a 5D doméně. Je zřejmé, že hodnota stále klesá, můžeme si však všimnout skoků v případě Haltonovy a Sobolovy sekvence. Generátory přidávají body v určitých sekvencích tak, že návrh s np body je rovnoměrně rozprostřený a návrh s $(np + 1)$ body rovněž. Můžeme si též všimnout vyrovnané kvality metod založených na Delaunayově triangulaci ve 2D, ovšem výrazného poklesu kvality varianty s rozdělováním simplexů v dimenzi vyšší. Pro srovnání je uvedena metoda přidávání bodů pomocí funkce `rand` dostupné v programu MATLAB. Tato funkce je založena na generátoru pseudonáhodných čísel Mersenne twister. Na Obrázku 3.19 uvádějícím výsledky na 2D doméně ovšem vidíme značný rozdíl v časové náročnosti metod založených na Delaunayově triangulaci. Z těch pak nejlépe dopadá varianta s rozdělováním s aplikací heuristické procedury k určení iterace, ve které provádíme retriangulaci. Tedy varianta, v níž nelze rozdělit simplex, který sám vznikl rozdělením. Použití periodických okrajových podmínek se ve 2D jeví jako úspěšné (kromě verze 1, ve které je největší simplex hledán ve všech buňkách, což vede díky periodicitě bodů, nikoli sítě k nebezpečí vzniku nekvalitního návrhu ve výchozí buňce), v kompletních výsledcích v Kapitole 4 však uvidíme, že ve vyšších dimenzích, zvláště kvůli výpočetní náročnosti, tato metoda spíše ztrácí.

Klíč k Obrázkům 3.18 a 3.19: tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty ze 100 spuštění.



Obrázek 3.18: Vývoj EMM při přidávání bodů.



Obrázek 3.19: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) sekvenčních algoritmů nespĺňujících podmínky LHS na 2D doméně.

Kapitola 4

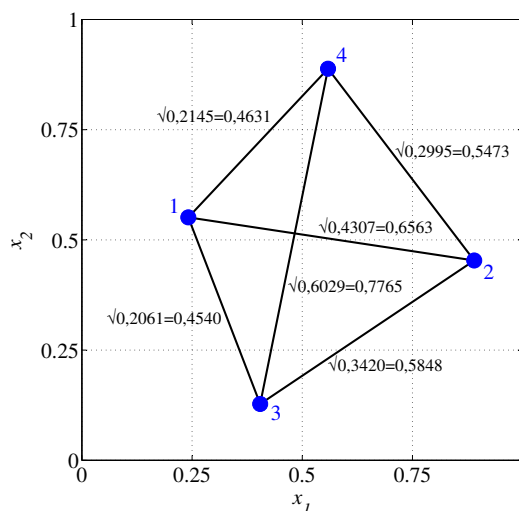
Implementace

Algoritmy byly implementovány v programu MATLAB R2010a. Používáme funkce `lhsdesign`, `delaunayn`, `haltonset` a `sobolset`, které program MATLAB, případně jeho statistický toolbox obsahuje. K výpočtu vzájemných vzdáleností bodů a tedy i hodnoty *EMM* byl použit algoritmus, který je součástí funkce `lhsdesign` statistického toolboxu programu MATLAB verze R2007b. Jelikož se nejedná o běžně používaný algoritmus, spolu s příkladem ho zde uvádíme pro seznámení:

```
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));
20
```

Do funkce vstupuje matice \mathbf{x} , která obsahuje souřadnice bodů (počet řádků odpovídá počtu bodů, počet sloupců odpovídá počtu dimenzí):

```
x =
    0.2414    0.5512
    0.8903    0.4533
    0.4046    0.1276
    0.5586    0.8886
```



Obrázek 4.1: Příklad výpočtu vzájemných vzdáleností.

Body z matice \mathbf{x} spolu s jejich vzájemnými vzdálenostmi, jejichž výpočet tu rozebíráme, zobrazuje Obrázek 4.1.

Algoritmus vytvoří vektory indexů I a J , které následně použije v jediném *for*-cyklu (probíhající přes dimenze) k určení pořadových čísel bodů, jejichž kvadráty odpovídajících souřadnic se mají odečíst. Samotné souřadnice bodů jsou tedy použity až v samém závěru v tomto *for*-cyklu.

I =	J =
1	2
1	3
1	4
2	3
2	4
3	4

Je vytvořen vektor d kvadrátů vzájemných vzdáleností mezi body. Obsahuje kvadráty prvků horní trojúhelníkové matice \mathbf{D} vzájemných vzdáleností mezi body řazené po řádcích této trojúhelníkové matice (zobrazená matice \mathbf{D} je pro orientaci doplněna o pořadová čísla bodů):

d =	sqrt_d =	D =																									
0.4307	0.6563	<table style="border: none; text-align: center;"> <tr> <td></td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td>1</td> <td>0</td> <td>0.6563</td> <td>0.4540</td> <td>0.4631</td> </tr> <tr> <td>2</td> <td>0.6563</td> <td>0</td> <td>0.5848</td> <td>0.5473</td> </tr> <tr> <td>3</td> <td>0.4540</td> <td>0.5848</td> <td>0</td> <td>0.7765</td> </tr> <tr> <td>4</td> <td>0.4631</td> <td>0.5473</td> <td>0.7765</td> <td>0</td> </tr> </table>		1	2	3	4	1	0	0.6563	0.4540	0.4631	2	0.6563	0	0.5848	0.5473	3	0.4540	0.5848	0	0.7765	4	0.4631	0.5473	0.7765	0
	1	2	3	4																							
1	0	0.6563	0.4540	0.4631																							
2	0.6563	0	0.5848	0.5473																							
3	0.4540	0.5848	0	0.7765																							
4	0.4631	0.5473	0.7765	0																							
0.2061	0.4540																										
0.2145	0.4631																										
0.3420	0.5848																										
0.2995	0.5473																										
0.6029	0.7765																										

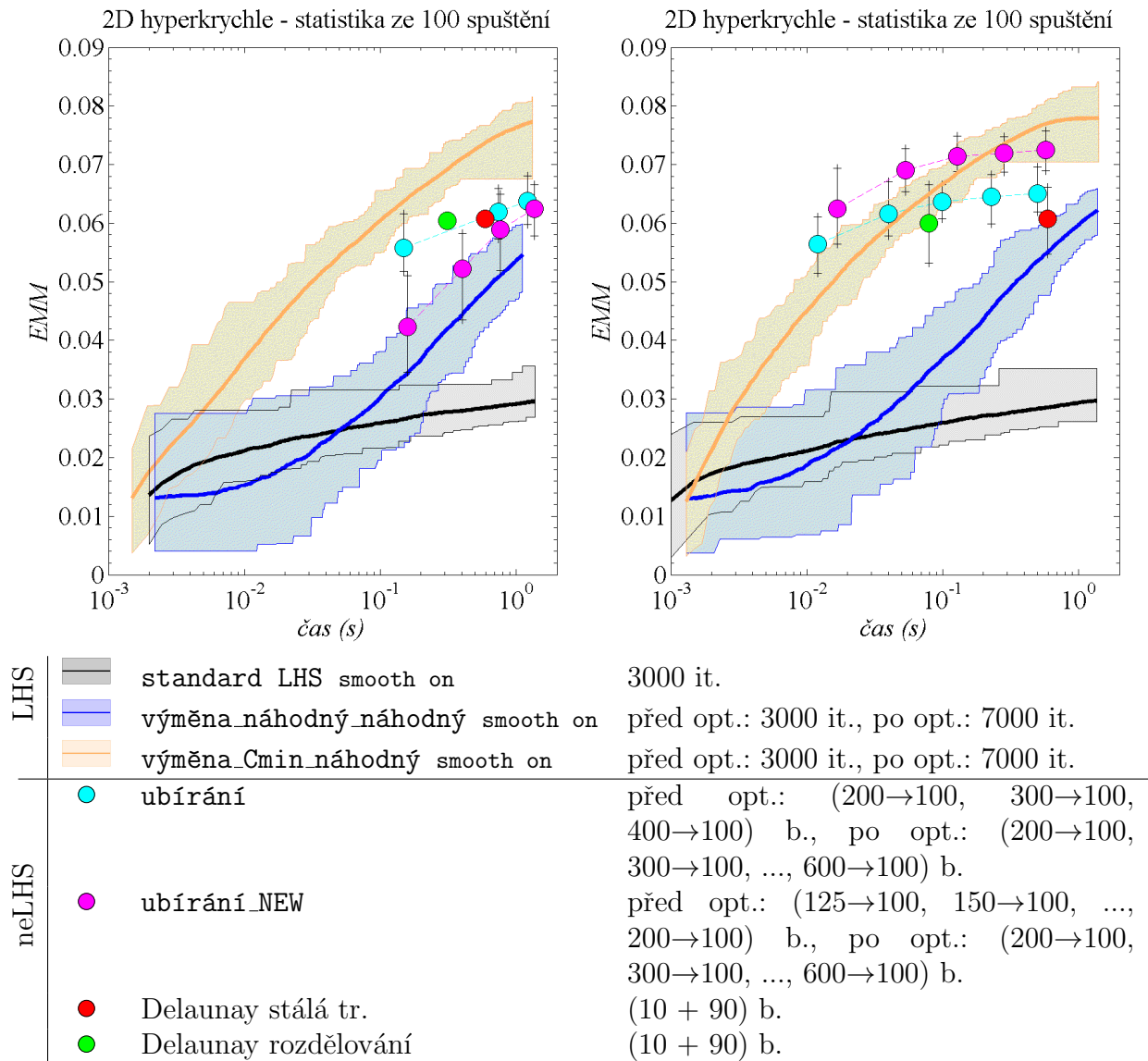
Nyní je již zřejmý i obsah vektorů I a J , vektor I obsahuje indexy bodů odpovídající číslům řádků prvků horní trojúhelníkové matice v matici \mathbf{D} , vektor J potom indexy od-

povídající číslům sloupců.

Hodnota *EMM* se vypočte jako odmocnina z minimálního prvku vektoru *d*.

Některé algoritmy byly podrobeny optimalizaci z pohledu implementace, zlepšení jejich výsledků ukazuje Obrázek 4.2. Výsledky uvedené na Obrázku 4.2 vlevo byly publikovány v [Myšáková and Lepš, 2011]. Můžeme si všimnout výrazného zrychlení algoritmů, stojí za upozornění navýšení počtu iterací provedených v jednotlivých metodách za obdobný časový interval.

V případě metod `výměna_náhodný_náhodný`, `výměna_Cmin_náhodný`, `ubírání` a `ubírání_NEW` bylo dosaženo zrychlení především zamezením přepočítávání vzájemných vzdáleností mezi všemi body návrhu v každé iteraci. V algoritmech s výměnami bodů nejsou po prohození přepočítávány vzájemné vzdálenosti mezi všemi body, nýbrž pouze ty, které jsou změnou dotčeny. Zde byly vyzkoušeny dvě varianty - výpočet v rámci vektoru vzájemných vzdáleností *d* a v rámci plné matice vzájemných vzdáleností *D*. Varianta s plnou maticí je rychlejší, jelikož ve vektoru *d* je časově náročné vyhledávání prvků dotčených výměnou bodů - v plné matici jsou to zřejmě prvky ve sloupcích a řádcích odpovídajících prohozeným bodům. V metodách založených na odebrání bodů z přehuštěného návrhu je pak kompletní matice vzájemných vzdáleností vytvořena pouze jednou a následně vždy jen odebrán řádek a sloupec odpovídající odebranému návrhovému bodu. V případě metody založené na Delaunayově triangulaci s rozdělováním simplexu (s retriangulací až po určitém počtu kroků) bylo zrychlení dosaženo zamezením přepočítávání objemů všech simplexů v každé iteraci. Nyní je po triangulaci vytvořen vektor objemů jednotlivých simplexů a až do příští triangulace je pracováno pouze s tímto vektorem. Objem rozděleného simplexu je vynulován a do vektoru je přidán odpovídající počet $(n + 1)$ objemů nových (rozdělením vzniklých) simplexů s objemem rovným $1/(n + 1)$ objemu rozděleného simplexu; *n* značí počet dimenzí.



Obrázek 4.2: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritmů na 2D doměně před a po provedení optimalizace z pohledu implementace algoritmů. Klíč: *smooth on* umísťuje body náhodně v rámci intervalu; (10 + 90) znamená 10 náhodných počátečních bodů a 90 přidaných bodů, (*np* → 100) označuje 100 bodů zbylých po odebrání z *np* původně vytvořených; tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty ze 100 spuštění.

Kapitola 5

Výsledky

Obrázky 5.1 - 5.4 ukazují kompletní výsledky prezentovaných metod na 2D - 5D hyperkrychách. Vytvořeno bylo vždy 100 návrhových bodů. Pouze výsledky `smooth off` variant metod `standard LHS`, `výměna_náhodný_náhodný` a metody `výměna_Cmin_náhodný` zde kvůli zachování přehlednosti výsledných grafů nejsou uvedeny. Jejich chování a vztah k odpovídajícím `smooth on` variantám je však jasně patrný z dílčích výsledků v Sekci 2.1.3.

V případě metod tvořících návrhy s fixním počtem bodů splňující podmínky LHS vidíme jasnou ztrátu metody `standard LHS`, ve které je použita „metoda hrubé síly“, tedy opakované tvoření nových návrhů. Nejen, že je proto metoda pomalejší (za stejný časový interval stihne cca třetinový počet iterací), ale nedosahuje zdaleka výsledků metod s výměnami bodů v jednom vytvořeném návrhu. Z těch má lepší výsledky metoda, ve které v každé iteraci najdeme dvojici s minimální vzájemnou vzdáleností a její existenci se snažíme ukončit posláním jednoho z těchto bodů do výměny. Je ovšem patrné, že s dostatečným počtem iterací dosáhne obdobné hodnoty *EMM* i metoda, ve které vyměňujeme pozice dvou zcela náhodných bodů. Pravděpodobnost úspěšné výměny (zvýšení hodnoty *EMM*) je v ní však výrazně nižší (a klesá s rostoucím počtem návrhových bodů).

Metody `ubírání` a `ubírání_NEW` dosahují stabilních velmi dobrých výsledků ve všech řešených dimenzích. Kvalita výsledného návrhu sice roste s počtem bodů v návrhu výchozím (s mírou přehuštění výchozího návrhu), ovšem poměrně pomalu a zdá se tak zbytečné tvořit výchozí návrh s mnohonásobně vyšším počtem bodů, než má mít návrh výsledný.

Jasně nejlepších výsledků ve 2D a ve 3D dosahuje nástroj `Distmesh`. Je schopný velice rychle vytvořit kvalitní návrh bez ohledu na kvalitu výchozí sady bodů, na kterou byl aplikován. Ve 4D je již však jeho chování poněkud znepokojivé a výsledky z 5D naznačují, že je nutno věnovat se více jeho nastavení pro použití ve vyšších dimenzích. Nástroj má problémy především s povrchovou sítí [Chen and Holst, 2011].

Výsledky testování metod `LHS_sequential` ukazují, že jsou velice dobrým nástrojem v případě, že je požadováno zahuštění již vytvořeného LHS návrhu (s tím, že i výsledný návrh má podmínky LHS splňovat). Výsledné návrhy vytvořené verzemi `smooth on` (vý-

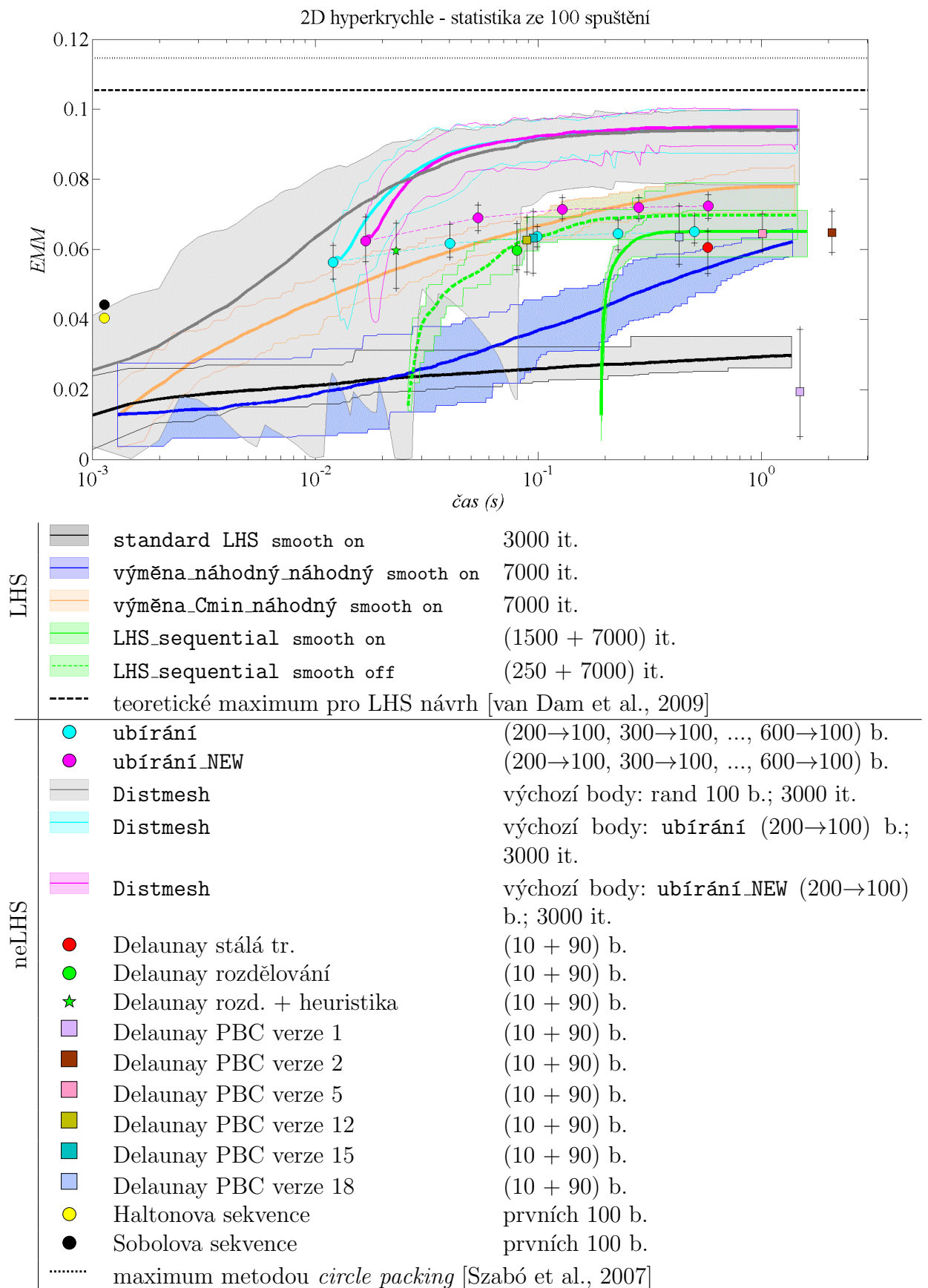
chozí návrh má poloviční počet bodů) i `smooth off` (výchozí návrh má třetinový počet bodů) dosahují výsledků srovnatelných s návrhy s předem daným počtem bodů.

Metody založené na Delaunayově triangulaci byly testovány s výchozím LHS návrhem o 10 bodech, ke kterému bylo přidáno dalších 90 bodů. Kvalita a použitelnost metod rychle klesá s rostoucím počtem dimenzí. Lepší hodnoty *EMM* samozřejmě dosahují varianty se stálou triangulací, to je ovšem vyváženo výpočetní náročností. Použití heuristické procedury k určení iterace, ve které provést retriangulaci u metody s rozdělováním simplexu (místo stálé triangulace) je úspěšné ve všech řešených příkladech. Jak vidíme z výsledků aplikace periodických okrajových podmínek, řešení způsobu a hustoty dělení hraničních objektů v základních verzích metody je kvalitní. Použití PBC v podstatě nepřináší žádné zlepšení, je jen logicky časově náročnější. Ve vyšších dimenzích není ve výsledných grafech uvedeno všech 6 verzí metody s PBC (verze 1, 2, 5 provádí retriangulaci v každé iteraci, verze 12, 15, 18 simplex rozděluje, retrianguluje jen v určitých krocích), neboť jsou velmi časově náročné. Obecně však v n -dimenzionálním prostoru použitelné jsou.

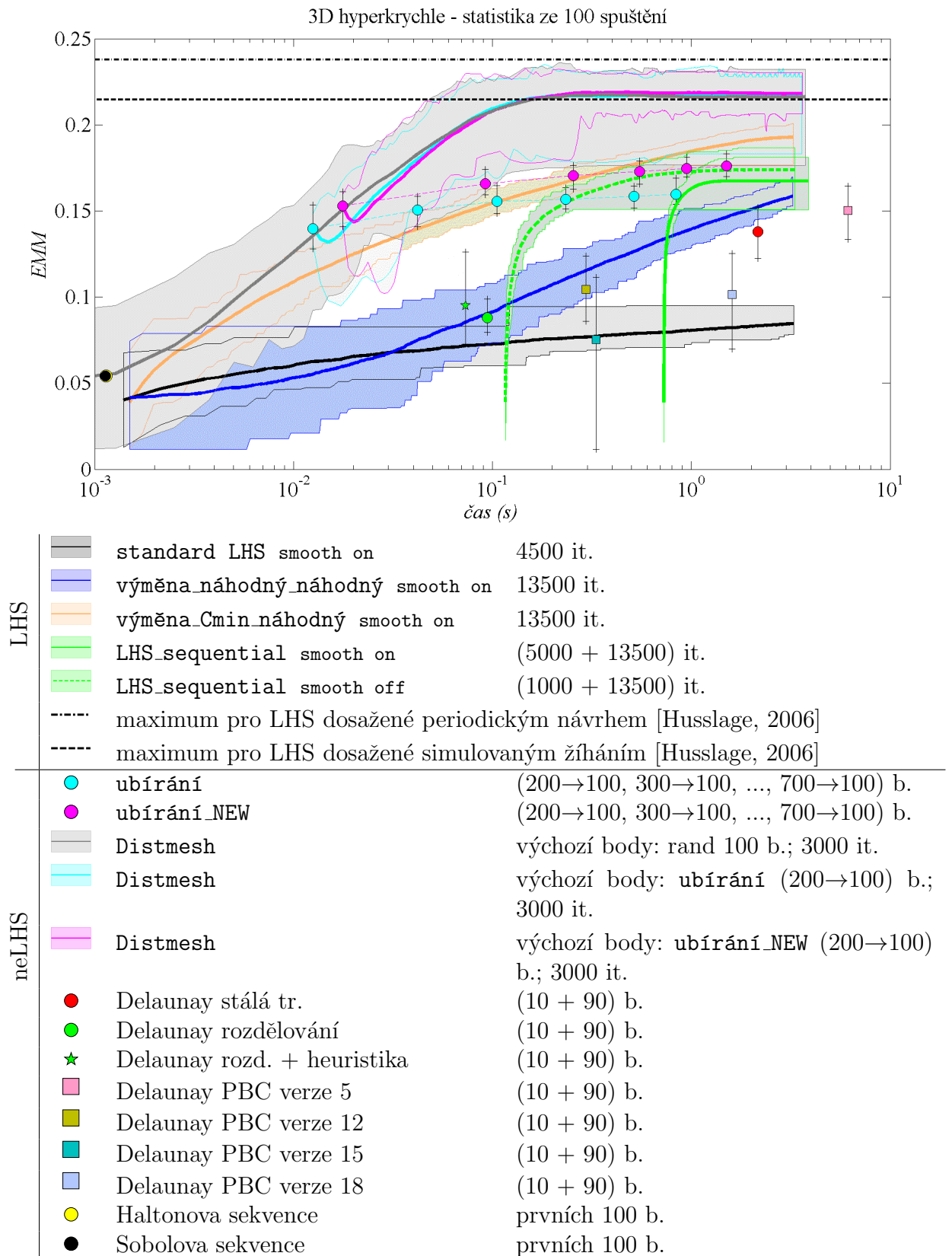
Generátory kvazináhodných sekvencí jsou rychlou metodou tvorby návrhů experimentů.

Je zřejmé, že rychlost všech metod je implementačně závislá, uvedené výsledky však jistě poskytují dobrou představu o jejich výkonnosti a přinášejí možnost jejich porovnání.

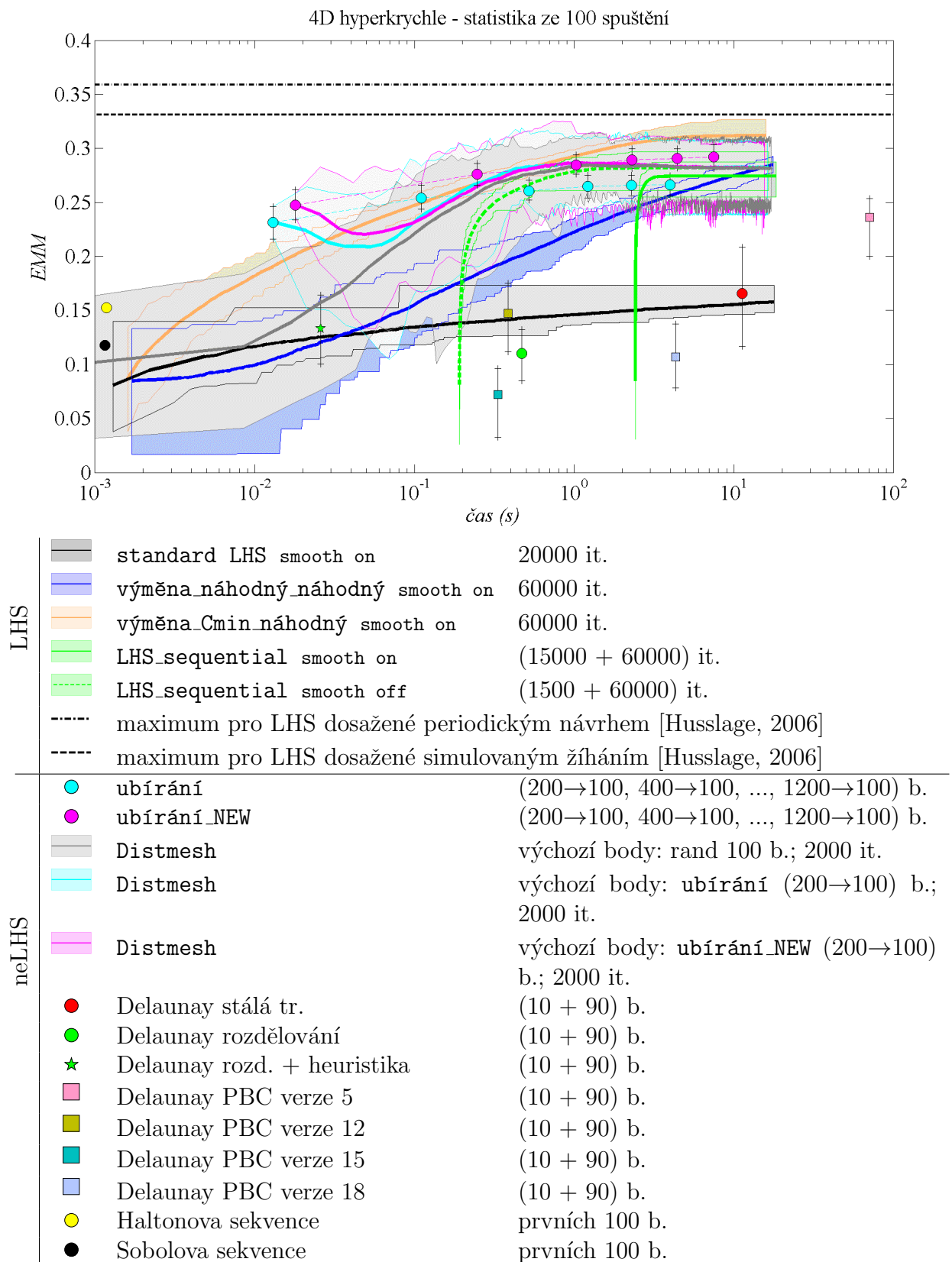
Klíč k Obrázkům 5.1 - 5.4: `smooth on` umisťuje body náhodně v rámci intervalu, `smooth off` umisťuje body doprostřed intervalu; $(10 + 90)$ znamená 10 náhodných počátečních bodů a 90 přidaných bodů, $(np \rightarrow 100)$ označuje 100 bodů zbylých po odebrání z np původně vytvořených; tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty ze 100 spuštění.



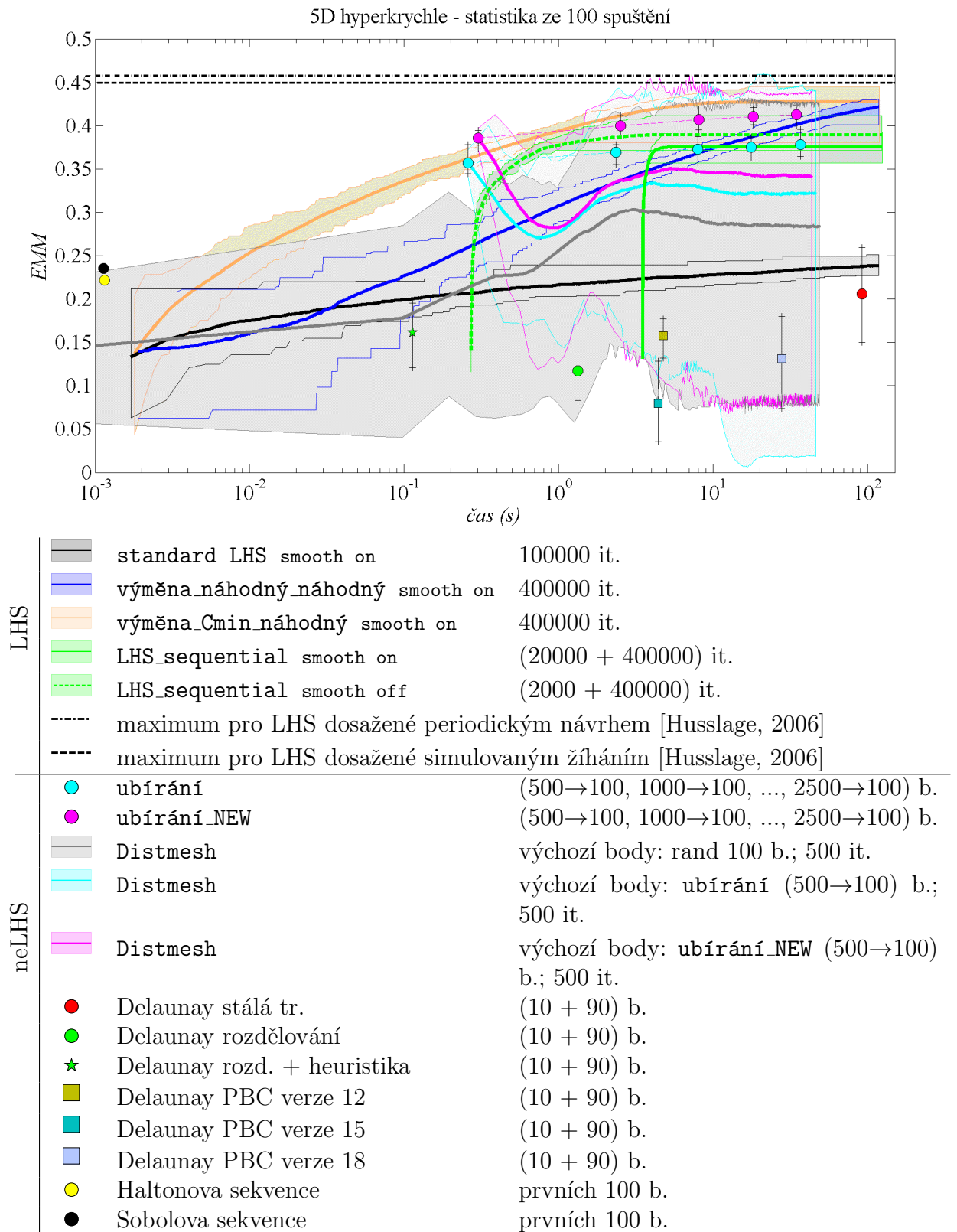
Obrázek 5.1: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritmů na 2D doměně.



Obrázek 5.2: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritů na 3D doměně.



Obrázek 5.3: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritmu na 4D doměně.



Obrázek 5.4: Porovnání rychlosti (*čas*) a kvality (*EMM* - vyšší hodnota je lepší) algoritů na 5D doměně.

Kapitola 6

Závěr

„Všechny modely jsou mylné, ale některé mohou být užitečné.“¹⁵ To jsou slova George E. P. Boxe (nar. 1919), který je shodou okolností zetěm R. A. Fishera, jehož citátem byla tato práce uvedena. K větší užitečnosti modelů slouží i návrhy experimentů - *designs of experiments* (DoEs), jež byly předmětem této práce.

Zabývali jsme se metodami pro tvorbu návrhů s fixním, předem stanoveným počtem bodů i návrhů doplňovaných. Není překvapivé, že doplňované návrhy nedosahují kvality těch optimalizovaných od začátku na daný počet návrhových bodů. Kvalita výsledného návrhu také zcela jistě závisí na kvalitě návrhu výchozího, k němuž jsou další body přidávány.

Jako kritérium hodnocení i jako cílová funkce v procesu optimalizace byla použita *Euklidovská maximin vzdálenost* (EMM). Jistě bude zajímavé porovnat představené metody i pomocí dalších kritérií a získat tak představu o jejich výkonnosti i z jiných hledisek.

Plánované je rovněž testování metod při tvorbě nižšího i vyššího počtu návrhových bodů a rozšíření počtu dimenzí návrhových prostorů. V této práci jsme se v návrzích drželi počtu 100 bodů a pohybovali jsme se ve dvou až pěti dimenzích.

Celkově lze jako nejúspěšnější hodnotit metody s výměnami bodů v LHS návrhu a metody založené na odebrání bodů ze záměrně přehustěného výchozího návrhu. Dobrých výsledků dosahuje i sekvenční metoda, ve které přidáváme body do původního LHS návrhu. Metody založené na Delaunayově triangulaci jsou ve vyšších dimenzích časově velmi náročné. To platí především pro varianty s retriangulací v každé iteraci. Na tomto místě je vhodné zmínit, že kromě použití periodických okrajových podmínek k řešení problému s dělením hraničních objektů lze využít možnosti tvorby návrhu ve „zvětšené“ hyperkrychli (tak dlouho, dokud v cílové doméně nebude požadovaný počet bodů). V takovém případě by nás kvalita dělení hraničních objektů a tedy i kvalita návrhu v okrajových oblastech této „zvětšené“ hyperkrychle nemusela znepokojoovat, neboť bychom na-

¹⁵Původně „*All models are wrong, but some are useful.*“ [Box and Draper, 1987].

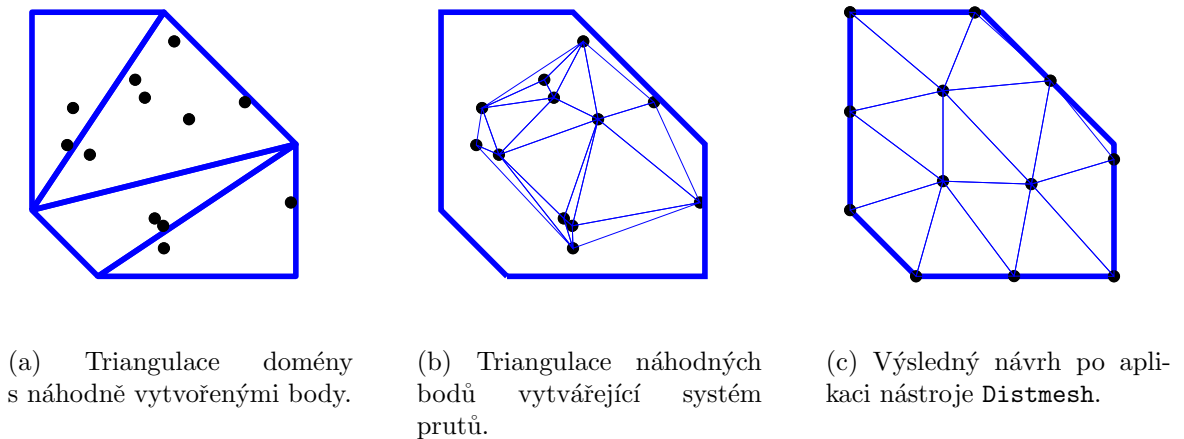
konec pouze „vyřízli“ středovou cílovou hyperkrychli. Použití nástroje `Distmesh` je velmi efektivní při nízkém počtu dimenzí.

Zajímavých výsledků bychom jistě mohli dosáhnout i zkombinováním jednotlivých metod, například použitím metody `výměna_náhodný_náhodný` nebo `výměna_Cmin_náhodný` na výchozí návrh získaný z přehuštěného návrhu metodou `ubírání` nebo `ubírání_NEW`. Výsledný návrh by nesplňoval LHS podmínky, ale rovnoměrnost pokrytí návrhového prostoru by mohla být velmi dobrá.

Kromě zde uvedených metod je další možností tvorby návrhů experimentů použití známých zoptimalizovaných návrhů. Ty jsou zpravidla vytvořeny pro velký počet návrhových bodů v různých dimenzích a existují metody pro jejich „ořezání“ pro požadovaný počet bodů a dimenzí; více např. v [Cioppa and Lucas, 2007].

Tato práce se zabývala pouze tvorbou návrhů v regulárních návrhových prostorech (v prostorech tvaru hyperkrychle). Náš další výzkum bude vyšetřovat schopnosti prezentovaných metod pro návrh na neregulárních doménách. Na první pohled se zde metody založené na LHS zdají nevhodné a lepších výsledků zde mohou dosáhnout metody využívající triangulaci, neboť pomocí ní můžeme získat dobrou představu o uspořádání i nepravidelných prostorů.

Jedna z metod pro tvorbu návrhů s omezujícími podmínkami ve dvou dimenzích již byla představena v [Myšáková and Lepš, 2012a]. Kombinuje použití Delaunayovy triangulace a nástroje `Distmesh` (viz Obrázek 6.1).



Obrázek 6.1: Metoda tvorby DoE v neregulárních prostorech.

Literatura

- [Del, 2001] (2001). Delaunay triangulation. www stránky: http://en.wikipedia.org/wiki/Delaunay_triangulation/. Poslední změna 15. března 2011.
- [Hyp, 2002] (2002). Hypercube. www stránky: <http://en.wikipedia.org/wiki/Hypercube/>. Poslední změna 7. března 2011.
- [Cen, 2003] (2003). Centroid. www stránky: <http://en.wikipedia.org/wiki/Centroid/>. Poslední změna 4. února 2011.
- [BCS, 2004] (2004). Barycentric coordinate system. www stránky: [http://en.wikipedia.org/wiki/Barycentric_coordinate_system_\(mathematics\)/](http://en.wikipedia.org/wiki/Barycentric_coordinate_system_(mathematics)). Poslední změna 24. února 2011.
- [Vol, 2011] (2011). Simplex volumes and the cayley-menger determinant. www stránky: <http://www.mathpages.com/home/kmath664/kmath664.htm/>.
- [Box and Draper, 1987] Box, G. and Draper, N. (1987). *Empirical model-building and response surfaces*. Wiley series in probability and mathematical statistics: Applied probability and statistics. Wiley.
- [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.
- [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.
- [Fang et al., 2006] Fang, K.-T., Li, R., and Sudjianto, A. (2006). *Design and modeling for computer experiments*. Chapman & Hall/CRC.
- [Fisher, 1935] Fisher, R. (1935). *The design of experiments*. Oliver and Boyd.
- [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*, 42:73–88.

- [Husslage, 2006] Husslage, B. G. M. (2006). *Maximin designs for computer experiments*. PhD thesis, Universiteit van Tilburg.
- [Janouchová and Kučerová, 2011] Janouchová, E. and Kučerová, A. (Sent for publication, 2011). Competitive comparison of optimal designs of experiments for sampling-based sensitivity analysis. *Computers & Structures*.
- [Klvaňa, 2005] Klvaňa, J. (2005). *Modelování 20: operační výzkum 2*. Vydavatelství ČVUT.
- [Kučerová, 2007] Kučerová, A. (2007). *Identification of nonlinear mechanical model parameters based on softcomputing methods*. PhD thesis, Ecole Normale Supérieure de Cachan, Laboratoire de Mécanique et Technologie.
- [Maaranen et al., 2007] Maaranen, H., Miettinen, K., and Penttinen, A. (2007). On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, 37(3):405–436.
- [Montgomery, 2000] Montgomery, D. C. (2000). *Design and Analysis of Experiments, 5th Edition*. Wiley.
- [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.
- [Myšáková and Lepš, 2012a] Myšáková, E. and Lepš, M. (2012a). Method for constrained designs of experiments in two dimensions. In *Proceedings of the 18th International Conference Engineering Mechanics*, pages 222–224.
- [Myšáková and Lepš, 2012b] Myšáková, E. and Lepš, M. (2012b). Sequential LHS Design Strategy for Reliability Analysis and Robust Optimization. In *2nd Workshop on structural analysis of lightweight structures*.
- [Novák and Lehký, 2006] Novák, D. and Lehký, D. (2006). ANN inverse analysis based on stochastic small-sample training set simulation. *Engineering Applications of Artificial Intelligence*, 19(7):731–740.
- [Persson and Strang, 2004] Persson, P.-O. and Strang, G. (2004). A simple mesh generator in MATLAB. *SIAM Review*, 46(2):329–345.
- [Press et al., 2007] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition.
- [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.
- [Szabó et al., 2007] Szabó, P. G., Markót, M. C., Csendes, T., Specht, E., Casado, L. G., and Garcãa, I. (2007). *New Approaches to Circle Packing in a Square: With Program Codes*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

- [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.
- [van Dam et al., 2009] van Dam, E. R., Rennen, G., and Husslage, B. (2009). Bounds for maximin latin hypercube designs. *Oper. Res.*, 57(3):595–608.
- [Vořechovský, 2009] Vořechovský, M. (2009). Hierarchical Subset Latin Hypercube Sampling for correlated random vectors. In Topping, B. and Tsompanakis, Y., editors, *Proceedings of the First International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, held in Madeira, Portugal*. Civil-Comp Press, Stirlingshire, UK.

Příloha A

Výpočet objemu simplexu

Jelikož známe souřadnice vrcholů simplexu, používáme k výpočtu jeho objemu vztah využívající právě (a pouze) souřadnice vrcholů [Vol, 2011].

Výpočet objemu simplexu ve 2D (3 vrcholy):

$$V_2 = \frac{1}{2!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} \\ 1 & x_{1(2)} & x_{2(2)} \\ 1 & x_{1(3)} & x_{2(3)} \end{vmatrix}$$

Výpočet objemu simplexu ve 3D (4 vrcholy):

$$V_3 = \frac{1}{3!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & x_{3(1)} \\ 1 & x_{1(2)} & x_{2(2)} & x_{3(2)} \\ 1 & x_{1(3)} & x_{2(3)} & x_{3(3)} \\ 1 & x_{1(4)} & x_{2(4)} & x_{3(4)} \end{vmatrix}$$

Výpočet objemu simplexu v n D ($n + 1$ vrcholů):

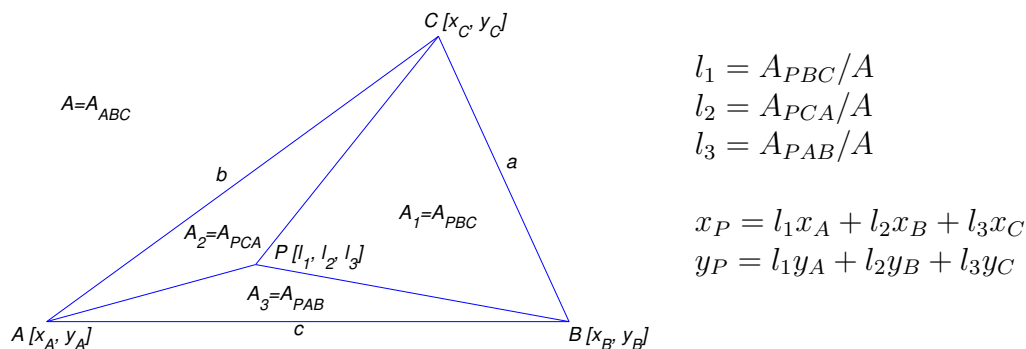
$$V_n = \frac{1}{n!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & \dots & \dots & x_{n(1)} \\ 1 & x_{1(2)} & x_{2(2)} & \dots & \dots & x_{n(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1(n+1)} & x_{2(n+1)} & \dots & \dots & x_{n(n+1)} \end{vmatrix}$$

V zápisu $x_{a(b)}$ a značí proměnnou (dimenzi), b označuje bod návrhu.

Příloha B

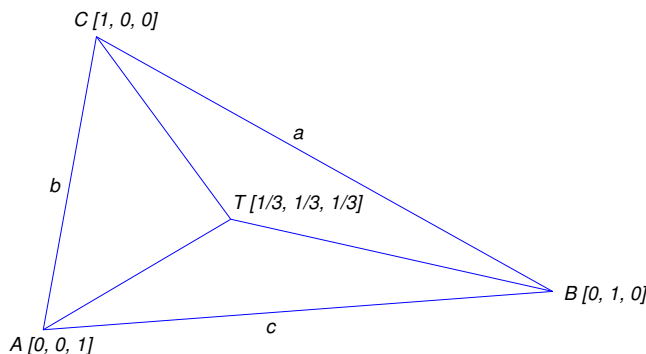
Výpočet souřadnic těžiště simplexu

K výpočtu souřadnic těžiště jsou použity plošné (trojúhelníkové, barycentrické) souřadnice [BCS, 2004]. Pro daný bod P udávají poměr plochy trojúhelníku tvořeného bodem P a stranami trojúhelníku ABC a plochy trojúhelníku ABC.



Obrázek B.1: Plošné (trojúhelníkové) souřadnice.

Těžiště trojúhelníku má plošné souřadnice $[1/3, 1/3, 1/3]$, těžiště čtyřstěnu $[1/4, 1/4, 1/4, 1/4]$, těžiště n -dimenzionálního simplexu potom $[1/(n+1), 1/(n+1), \dots, 1/(n+1)]$ [Cen, 2003].



Obrázek B.2: Těžiště trojúhelníku s použitím plošných souřadnic.

Příloha C

Přehled prvků (0-5)-dimenzionální hyperkrychle

dimenze	název	vrcholy	hrany	stěny	buňky	4-facety	5-facety
0	Bod	1					
1	Přímka	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 C.1: Přehled prvků hyperkrychle pro 0-5 dimenzí.

V Tabulce C.1 uvádíme přehled prvků hyperkrychlí do pěti dimenzí. Více informací na [Hyp, 2002].

Příloha D

Parametry použitého počítače a softwaru

ASUS PRO5DID-SX128V	
Procesor	Intel Core 2 Duo T6570 s frekvencí 2,1 GHz, 2 MB L2 Cache, 800 MHz FSB, 2 jádra, 2 thready, Max TDP: 35 W
Operační paměť	4 (2+2) GB DDR3 1066 MHz
Grafická karta	NVIDIA GeForce GT320M s 1024 MB vlastní paměti
Operační systém	Microsoft Windows 7 Home Premium 64-bit
Verze MATLABu	R2010a
Překladač \LaTeX u	MiKTeX 2.9

Tabulka D.1: Použitá počítačová sestava.

Příloha E

Obsah přiloženého CD

K této práci je přiloženo CD se zdrojovými kódy (soubory typu .m programu MATLAB) pro jednotlivé metody uvedené v této práci.