



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

**Fakulta stavební
Katedra mechaniky**

Srovnání metod tvorby rovnoměrně rozprostřených návrhů

Comparison of space-filling design strategies

Soutěžní práce

Studijní program: Stavební inženýrství
Studijní obor: Materiálové inženýrství

Vedoucí práce: Ing. Matěj Lepš, Ph.D.

EVA MYŠÁKOVÁ

Praha 2011

Abstrakt: *Návrhové či plánované experimenty - design of experiments (DoE) tvoří základní část meta-modelování. Na výsledné návrhy jsou zpravidla kladeny dva hlavní požadavky - jejich vzájemná ortogonalita a rovnoměrné pokrytí daného prostoru (space-filling). Poslední dekáda byla svědkem vývoje několika metod kládoucích důraz na druhý z požadavků. Tyto metody jsou založeny na velice rozdílných myšlenkách a bývají značně komplexní. Jelikož se výpočetní náročnost stává v současné době hlavním omezením, zaměřili jsme se na porovnání výpočetního času a výkonnosti z pohledu rovnoměrnosti pokrytí prostoru. Náš příspěvek představuje a srovnává několik různých kvazi-náhodných generátorů využívajících Latine Hypercube Sampling (LHS) stejně jako Delaunayovu triangulaci (DT).*

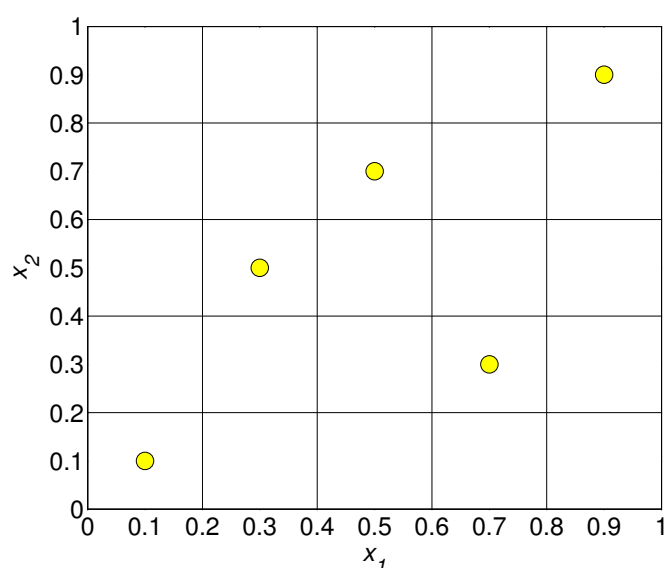
Abstract: *Space-Filling Design Strategies create an essential part of a surrogate modeling. Two main objectives are usually placed on the resulting designs - orthogonality and space-filling properties. The last decade has witnessed the development of several methods for the latter objective. These methods are based on totally different ideas and are characterized by distant complexities. Since the computing time can be the limiting constraint, we inspect the computing demands against the space-filling performances. In detail, our contribution presents and compares several different techniques of quazi-random numbers generators utilizing Latine Hypercube Sampling (LHS) methods as well as Delaunay triangulations.*

Obsah

1	Úvod	3
2	Metody tvorby rovnoměrně rozprostřeného návrhu	3
2.1	Latin Hypercube Sampling (lhsdesign-default)	3
2.2	Optimalizovaný Latin Hypercube Sampling - náhodná výměna (lhsdesign-výměna-náhodný+náhodný)	4
2.3	Optimalizovaný Latin Hypercube Sampling - heuristická procedura (lhsdesign-výměna-Cmin+náhodný)	4
2.4	Odebírání bodů z LHS návrhu (LHS-ubírání, LHS-ubírání-NEW)	6
2.5	Metody založené na Delaunayově triangulaci (Delaunay-stálá triangu- lace, Delaunay-rozdělování)	7
3	Výsledky a závěr	11
4	Poděkování	11
A	Výpočet objemu simplexu	18
B	Výpočet souřadnic těžiště simplexu	19
C	Přehled prvků (0-5)-dimenzionální hyperkrychle	20
D	Výpočet vzájemných vzdáleností bodů návrhu	21

1 Úvod

Návrhové či plánované experimenty - design of experiments (DoE) jsou základní částí vývoje každého meta-modelu [Simpson et al., 2001, Jin, 2005]. Cílem je získat maximum informací o daném systému při minimálním počtu vyčíslení. Výsledný meta-model je *a priori* neznámý, návrh by však měl pokrýt doménu co možná nejrovnoměrněji. Nejčastěji se takové DoE hodnotí metrikami zaměřenými na ortogonalitu a rovnoměrnost pokrytí. Odkazy na metriky zaměřené na ortogonalitu lze nalézt v [Cioppa and Lucas, 2007, Hofwing and Strömberg, 2010], reference na metriky zaměřené na rovnoměrnost pokrytí pak v článkách [Toropov et al., 2007] nebo [Crombecq et al., 2009]. Zde jsme zvolili metriku *maximin* (*Mm*) pro její jednoduchost a snadnost zobrazení. *Mm* je minimální ze vzdáleností mezi každými dvěma body návrhu a je maximalizována. Na základě této metriky porovnáme několik metod, které byly v minulých letech představeny k tvorbě DoE.



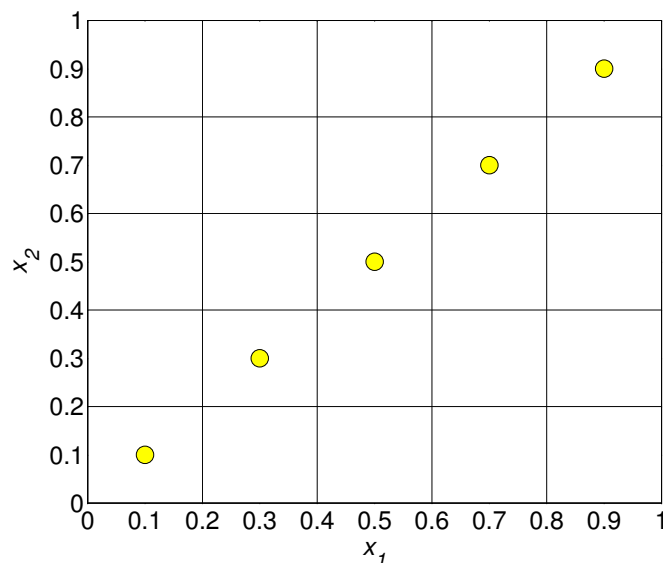
Obrázek 1: Příklad LHS návrhu pro dvě proměnné s 5 body umístěnými do středů intervalů.

2 Metody tvorby rovnoměrně rozprostřeného návrhu

2.1 Latin Hypercube Sampling (lhsdesign-default)

LHS je jedním z nejpopulárnějších algoritmů, zejména z důvodu jednoduchosti použití a rychlosti výpočtu, ačkoli jeho schopnosti rovnoměrného pokrytí prostoru (space-filling) mohou být špatné. V LHS je každá proměnná (dimenze) rozdělena na n intervalů, kde n odpovídá požadovanému počtu bodů návrhu. V rámci každého intervalu je bod umístěn náhodně¹ nezávisle pro každou proměnnou. V každém intervalu každé proměnné (dimenze) je umístěn pouze jeden bod. To vede k regulárnímu DoE, jak je patrné z Obrázku 1. Nejhorší případ LHS návrhu je ten, kdy body vytvoří diagonálu (Obrázek 2). Nejjednodušším řešením takového nedostatku

¹Nebo může být umístěn do středu intervalu - v prostředí programu MATLAB parametr "smooth".



Obrázek 2: Nevhodný LHS návrh.

je vytvoření zcela nového LHS návrhu, to jest užití “metody hrubé síly”. Takový postup je užit například v prostředí MATLAB ve funkci `lhsdesign`². Dále tuto metodu označujeme jako standardní LHS návrh.

2.2 Optimalizovaný Latin Hypercube Sampling - náhodná výměna (lhsdesign-výměna-náhodný+náhodný)

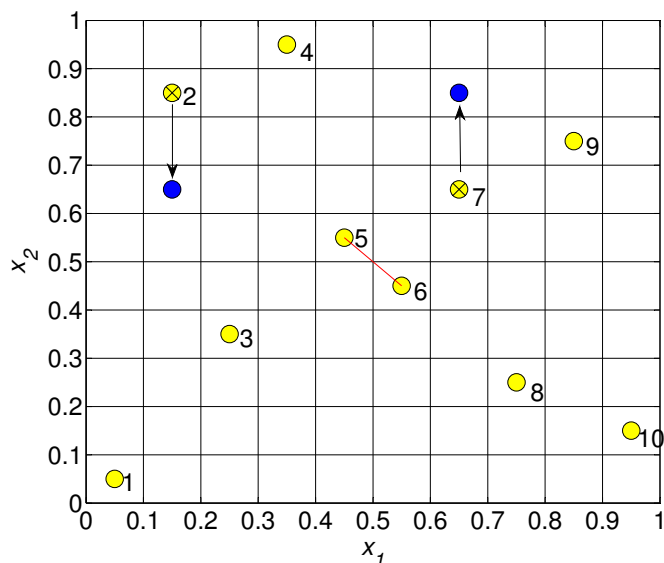
Špatný LHS návrh (z pohledu Mm metriky) může být vylepšen měněním pozic jednotlivých bodů (ovšem při zachování vlastností LHS - tj. ve 2D “pravidlo sudoku”), viz například odkazy [Novák and Lehký, 2006, Kučerová, 2007]. Jsou vybrány 2 body, jedna z dimenzí a prohozeny odpovídající souřadnice. Všechny tři volby jsou 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 Mm). Proto tuto metodu můžeme označit jako náhodný horolezecký (Hill Climbing) algoritmus. Jak je patrné i z Obrázku 3, úspěšná může být pouze výměna, do níž vstupuje jeden z bodů z dvojice s minimální vzájemnou vzdáleností - v následující metodě je toto zajištěno.

2.3 Optimalizovaný Latin Hypercube Sampling - heuristická procedura (lhsdesign-výměna-Cmin+náhodný)

Jelikož víme, které dvojici bodů odpovídá hodnota Mm , 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

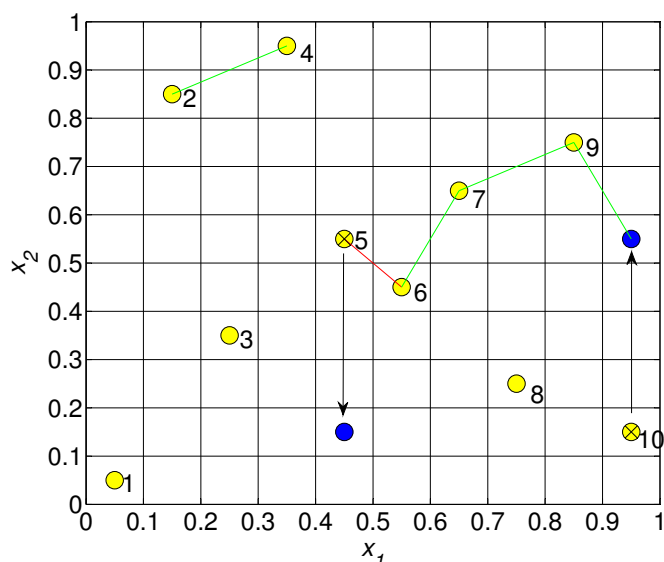
²Podle přednastavení je generováno 5 návrhů a nejlepší je představen uživateli.

2. METODY TVORBY ROVNOMĚRNĚ ROZPROSTŘENÉHO NÁVRHU



bod návrhu	souřadnice	
	x_1	x_2
1	0.05	0.05
2	0.15	0.85 → 0.65
3	0.25	0.35
4	0.35	0.95
5	0.45	0.55
6	0.55	0.45
7	0.65	0.65 → 0.85
8	0.75	0.25
9	0.85	0.75
10	0.95	0.15

Obrázek 3: Výměna dvou náhodných bodů návrhu (zde na souřadnici x_2). Červená úsečka vyznačuje Mm . Nedochozí ke zvětšení hodnoty Mm , tato výměna tedy nebude přijata.

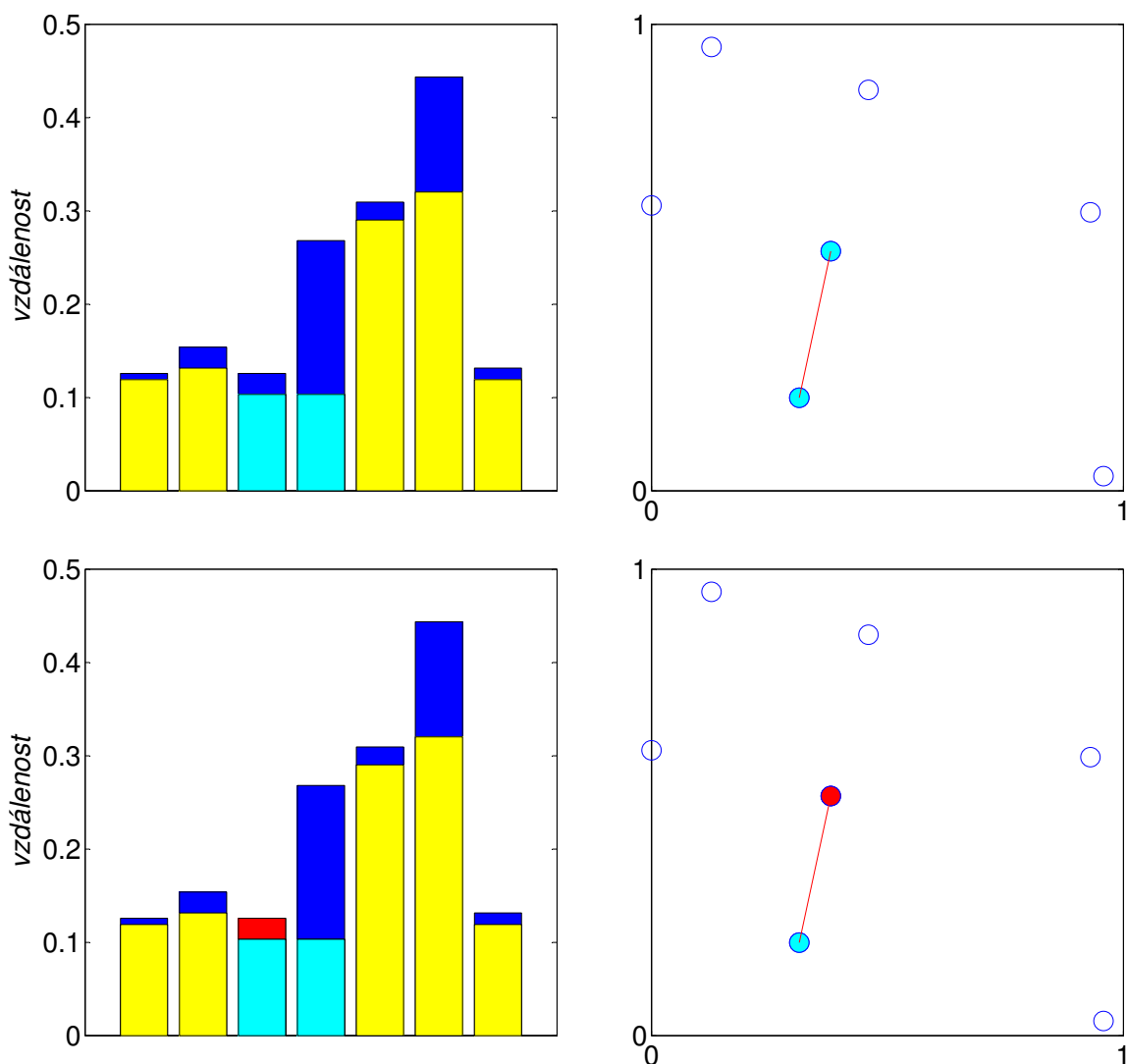


bod návrhu	souřadnice	
	x_1	x_2
1	0.05	0.05
2	0.15	0.85
3	0.25	0.35
4	0.35	0.95
5	0.45	0.55 → 0.15
6	0.55	0.45
7	0.65	0.65
8	0.75	0.25
9	0.85	0.75
10	0.95	0.15 → 0.55

Obrázek 4: Výměna bodu z dvojice s Mm a náhodně vybraného bodu (zde na souřadnici x_2). Červená úsečka vyznačuje Mm před výměnou, zelená po výměně. Dochází ke zvětšení hodnoty Mm , tato výměna tedy bude přijata.

s Mm (který z dvojice to bude je voleno náhodně³) a jiného, náhodně zvoleného bodu, dokud nedojde ke zlepšení - ke zvýšení hodnoty Mm . Poté algoritmus najde novou dvojici s minimální vzdáleností a pokračuje stejným způsobem. Metodu přibližuje Obrázek 4.

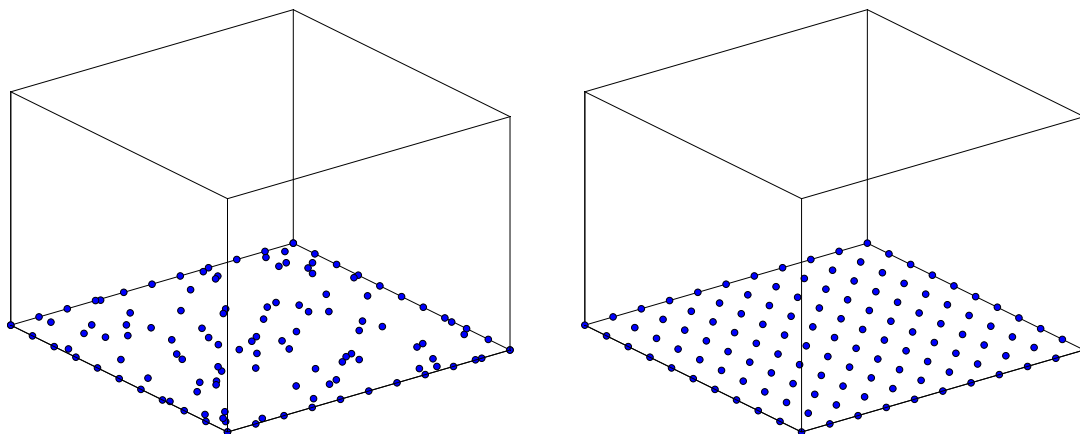
³Zde je prostor pro vylepšení metody tak, aby byl vždy do výměny poslán "horší" z dvojice s Mm , tj. bod, jehož druhá nejkratší vzdálenost k jinému bodu je menší než u druhého z dvojice s Mm ; popsaná metoda by byla časově náročnější z důvodu nutnosti obsáhlejší analýzy vektoru vzájemných vzdáleností; u použité verze stačí nalézt minimální prvek tohoto vektoru.



Obrázek 5: Odebrání “horšího” bodu z dvojice s Mm . Grafy vlevo znázorňují pro každý bod jeho první a druhou nejkratší vzdálenost k jinému bodu. Azurové sloupce značí minimální vzájemnou vzdálenost (Mm), červený sloupec označuje kratší z druhých nejkratších vzdáleností dvojice bodů s Mm - odpovídající bod bude odebrán. Klíč: Žluté sloupce = první nejkratší vzdálenosti k jinému bodu, Modré sloupce = druhé nejkratší vzdálenosti k jinému bodu, Azurové sloupce = Mm , Červený sloupec = kratší z druhých nejkratších vzdáleností dvojice bodů s Mm , Azurové body = dvojice s Mm , Červený bod = “horší” z dvojice s Mm , Červená úsečka = Mm .

2.4 Odebírání bodů z LHS návrhu (LHS-ubírání, LHS-ubírání-NEW)

Poslední metoda založená na LHS využívá rychlosti generování LHS návrhu. Vytvoření LHS návrhu s více body nezabírá příliš času. Proto vytvoříme návrh s více body, než je třeba, a poté opakovaně odebíráme bod z dvojice s minimální vzájemnou vzdáleností, dokud není docíleno požadovaného počtu bodů. Tato metoda má dvě varianty - v první je odebraný bod z dvojice s Mm vybrán náhodně, v druhé je odebrán ten, jehož druhá nejkratší vzdálenost k ostatním bodům je menší (viz Obrázek 5).



Obrázek 6: Dělení stěny 3D domény. Na prvním obrázku je LHS návrh, na druhém plně faktoriální návrh.

2.5 Metody založené na Delaunayově triangulaci (Delaunay-stálá triangulace, Delaunay-rozdělování)

Poslední skupina metod je založena na triangulaci přípustné domény (“triangulace” je termín vhodný pro 2D, obecně jde o rozdělení domény na simplexu) [Crombecq et al., 2009]. Jelikož je relativně jednoduché spočítat objem simplexu, můžeme tak získat hrubý odhad toho, kde je největší nevyplněný prostor. Postup výpočtu objemu simplexu uvádíme v Příloze A. Začínáme neoptimalizovaným LHS návrhem s několika body. Následně je provedena Delaunayova triangulace [Del, 2001]. Poté naše procedura opakovaně přidává další bod návrhu do těžiště největšího simplexu (viz Příloha B o výpočtu souřadnic těžiště simplexu). I zde jsou dvě varianty algoritmu.

V první variantě je po každém kroku (po přidání nového bodu) provedena nová triangulace, nalezen simplex s největším objemem a do jeho těžiště opět přidán bod. Postup dokumentuje Obrázek 7.

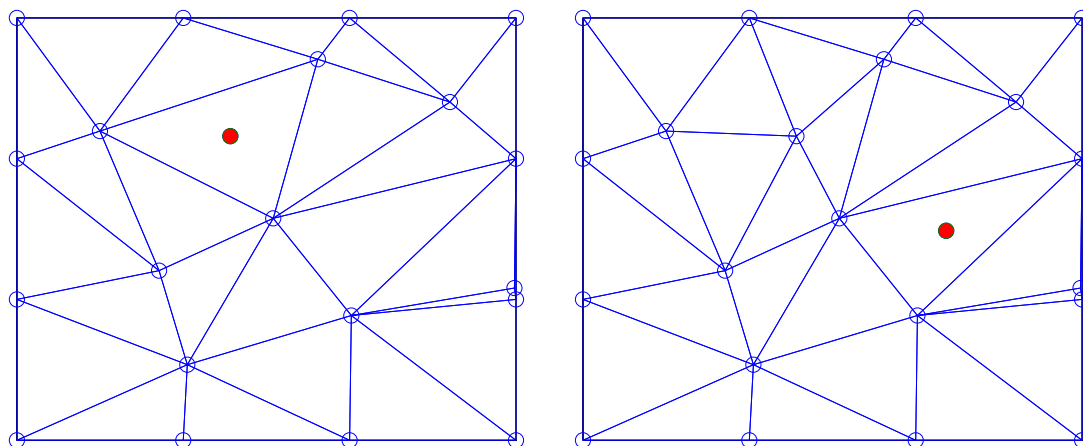
Ve druhé variantě je triangulace prováděna jen po určitém počtu iterací. V ostatních krocích je simplex, do jehož těžiště byl přidán nový bod, pouze rozdělen na $(dim+1)$ simplexů, které jsou považovány za plnohodnotné nové simplexu pro následný výpočet objemu⁴; dim značí počet proměnných, tj. dimenzí. Tato varianta algoritmu je znázorněna na Obrázku 8.

Zároveň je řešena otázka dělení hran (ve 2D), stěn (ve 3D), respektive $(dim-1)$ -facet (viz Příloha C) objektů v řešené doméně. Část tohoto problému dokumentuje Obrázek 9⁵. Porovnali jsem dvě varianty - v jedné byly na hranách generovány LHS návrhy, v druhé byl použit tzv. plně faktoriální (*fullfact*) návrh (viz Obrázek 6). Tento se ukázal jako výhodnější. Rozdělení hran není provedeno najednou, nýbrž dynamicky během výpočtu v závislosti na aktuálním počtu bodů v generovaném návrhu.

Tyto dvě DT-metody jsou v podstatě nenáhodné, kromě původních bodů vygenerovaných LHS metodou (v našem případě je takto vytvořena 1/10 z požadovaného počtu bodů návrhu)

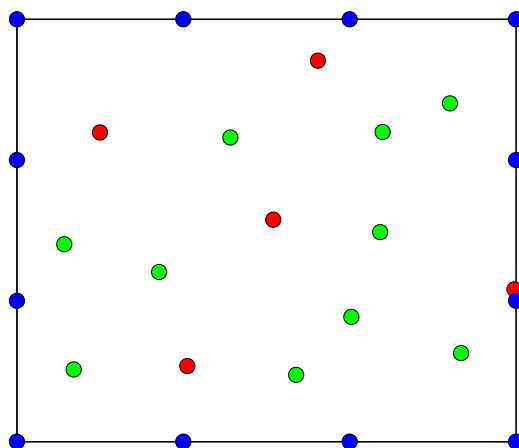
⁴Může dojít k situaci, kdy je simplex s největším objemem tak velký, že po jeho rozdělení jsou i jeho části při hledání dalšího simplexu pro přidání bodu vyhodnoceny jako největší v doméně.

⁵Zde ve 2D - jde tedy skutečně o dělení hran, ve více dimenzích je nutno dělit vícedimenzionální (o dimenzi nižší než řešená doména) hraniční objekty. Viz Příloha C.



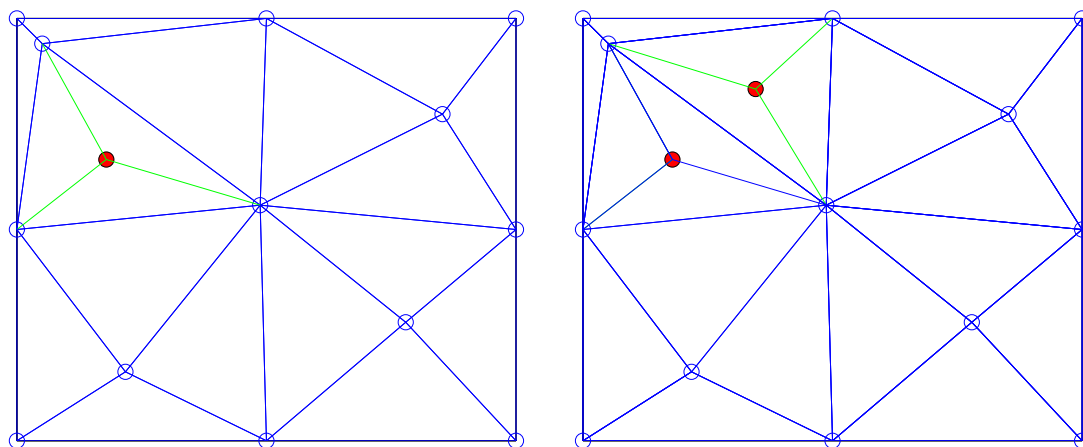
(a) Červený bod znázorňuje těžiště největšího simplexu - bude přidán jako nový bod návrhu.

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



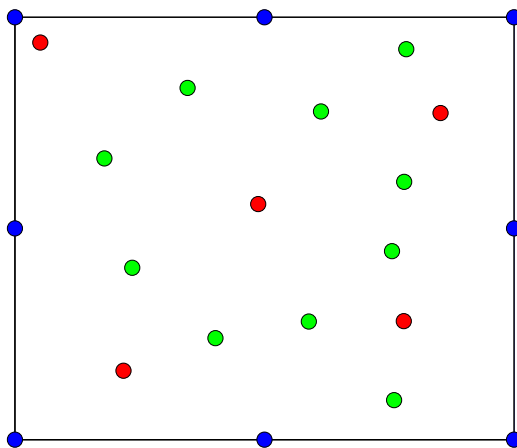
(c) Výsledný obrázek po vytvoření celého návrhu. Modré body označují hranice domény (jsou jen pomocné během výpočtu, nejsou součástí výsledného návrhu), červené body byly vytvořeny LHS metodou, zelené přidány v každé iteraci do těžiště simplexu s největším objemem.

Obrázek 7: První varianta metody založené na Delaunayově triangulaci (ve 2D) - po přidání bodu je provedena nová triangulace celé přípustné domény.



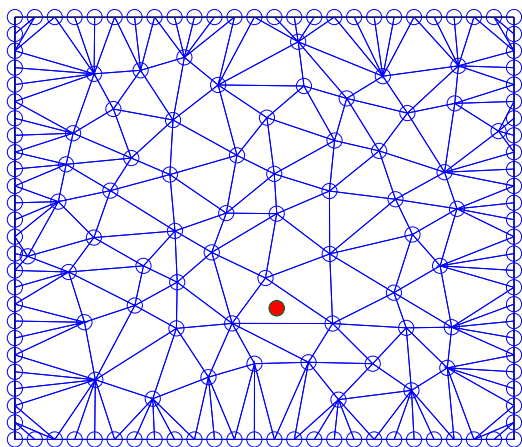
(a) Červený bod znázorňuje těžiště největšího simplexu - bude přidán jako nový bod návrhu. Zelené úsečky vyznačují rozdělení simplexu na 3 nové.

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

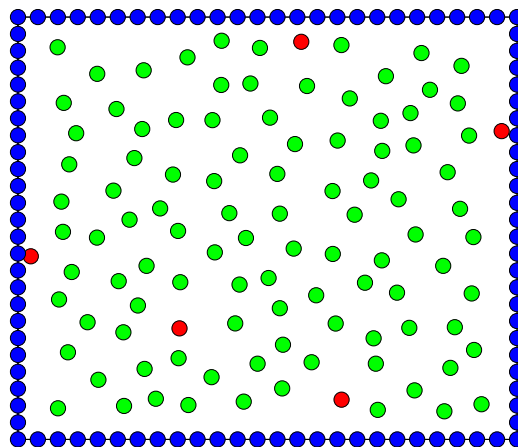


(c) Výsledný obrázek po vytvoření celého návrhu. Modré body označují hranice domény (jsou jen pomocné během výpočtu, nejsou součástí výsledného návrhu), červené body byly vytvořeny LHS metodou, zelené přidány v každé iteraci do těžiště simplexu s největším objemem.

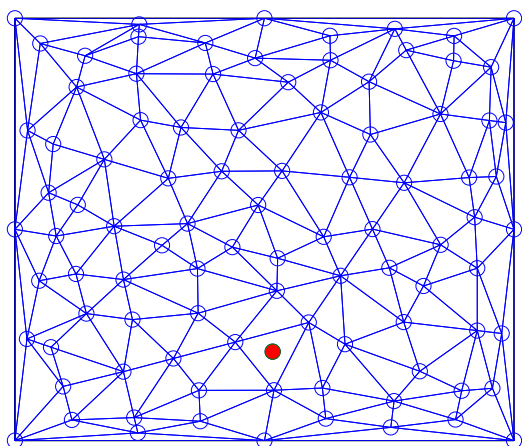
Obrázek 8: Druhá varianta metody založené na Delaunayově triangulaci (ve 2D) - po přidání bodu není provedena nová triangulace celé přípustné domény, největší simplex je pouze rozdělen.



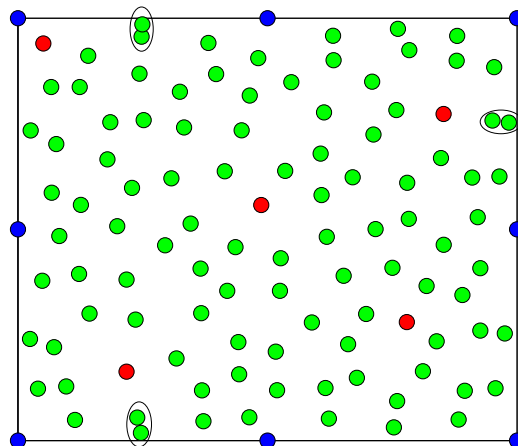
(a) Použití hustého dělení hrany. Vznikají typické úzké, do středu domény orientované simplex.



(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 simplex podél hran domény.



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

Obrázek 9: Rozdíl v hustotě dělení hran. Použita varianta plně faktoriálního návrhu - hrana je pravidelně rozdělena. Pro Obrázky 9b, 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.

jsou všechny ostatní přidány deterministickými postupy, tj. Delaunayova triangulace, výpočet objemů a souřadnice těžiště.

První varianta poskytuje lepší představu o dosud nevyplněném prostoru; druhá varianta je méně přesná, ale (zvláště ve vyšších dimenzích) rychlejší.

3 Výsledky a závěr

Algoritmy byly implementovány v programu MATLAB a uváděné výsledky jsou statistikou ze 100 spuštění každého z nich. Používáme funkce `lhsdesign` a `delaunayn`, které program MATLAB obsahuje⁶. K výpočtu vzájemných vzdáleností bodů a tedy i hodnoty Mm byl použit algoritmus, který je součástí funkce `lhsdesign` a je uveden v Příloze D.

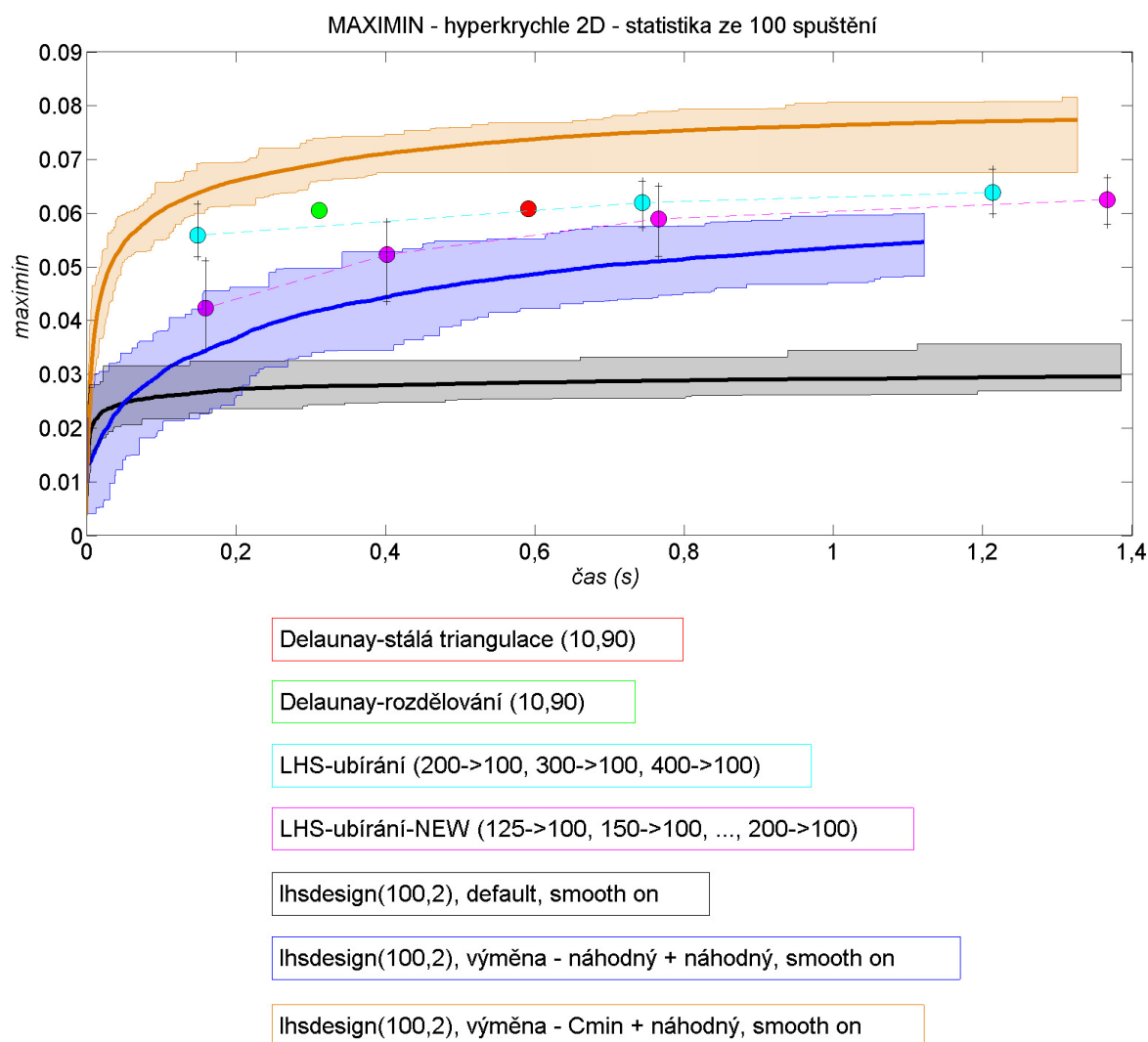
Z Obrázků 10, 11, 12, 13 je zřejmé, že standardní LHS a LHS s náhodnými výměnami zůstává za ostatními metodami. Taktéž je evidentní, že metody založené na Delaunayově triangulaci neobstojí ve srovnání s metodami heuristickými. Nadto, s rostoucím počtem dimenzí jsou DT-metody časově výrazně náročnější než LHS-metody. Naše výsledky ukazují, že ve čtyřech dimenzích dosahují heuristické metody, ve stejném čase, více než dvakrát vyšší hodnoty *maximin*.

Náš příspěvek ukázal výsledky na regulárních doménách (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 ve srovnání s těmi založenými na triangulaci. Ty jsou triangulací schopné popsat téměř jakýkoli tvar, a proto se tu jejich aplikace zdá logická.

4 Poděkování

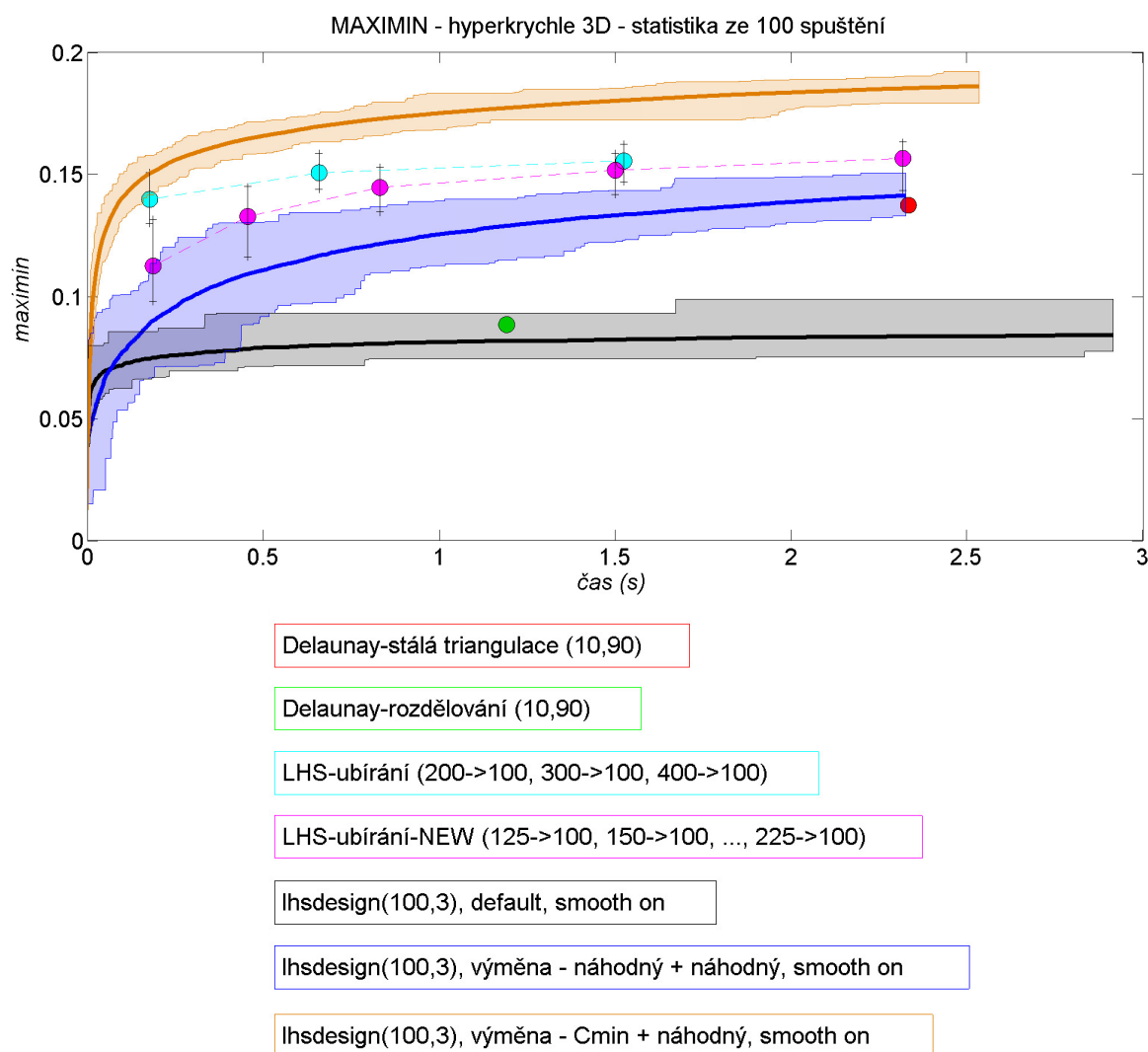
Autoři by rádi poděkovali za finanční podporu Ministerstva školství, mládeže a tělovýchovy v rámci projektu MSM 6840770003 (Algorithms for computer simulation and application in engineering).

⁶Funkci `lhsdesign` obsahuje statistický toolbox programu MATLAB.

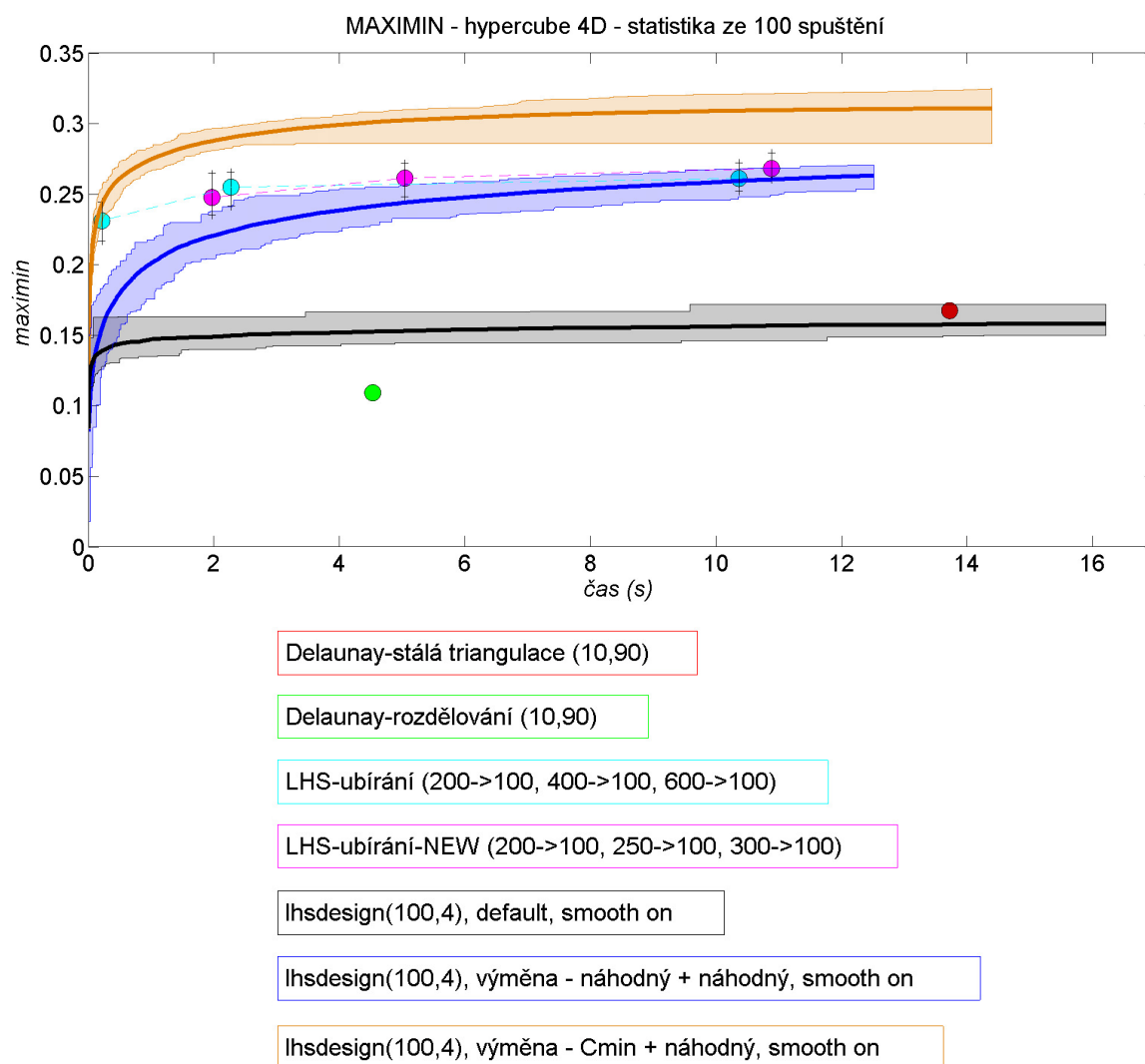


Obrázek 10: Porovnání rychlosti (*čas*) a kvality (*maximin* - vyšší hodnota je lepší) algoritmů na 2D doméně. Klíč: Černé křivky = standardní LHS, Modré křivky = náhodná výměna, Okrové křivky = heuristická metoda, Azurové body = odebrání bodů (odebrán náhodný z dvojice), Purpurové body = odebrání bodů (odebrán "horší" z dvojice), Červený bod = Delaunayova triangulace s re-triangulováním, Zelený bod = Delaunayova triangulace jen v určitých iteracích; *smooth on* generuje body náhodně v rámci intervalu; (10,90) znamená 10 náhodných počátečních bodů a 90 bodů přidaných triangulací, ($n - > 100$) označuje 100 bodů zbylých po odebrání z n původně vygenerovaných a (100,2) značí 100 bodů ve 2D; tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty.

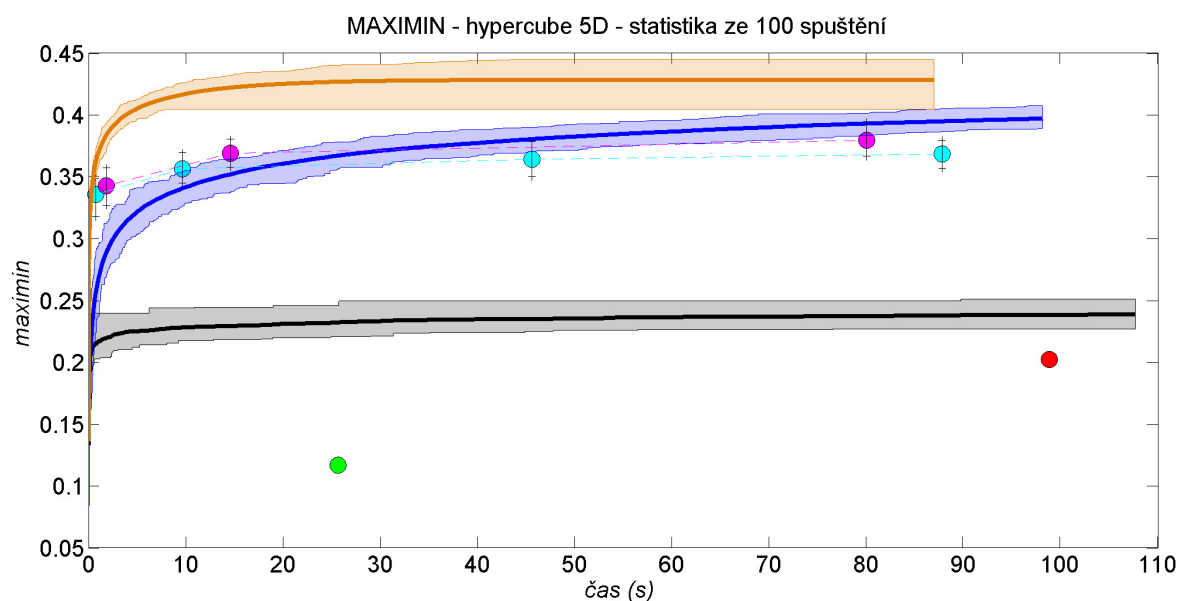
4. PODĚKOVÁNÍ



Obrázek 11: Porovnání rychlosti (*čas*) a kvality (*maximin* - vyšší hodnota je lepší) algoritmů na 3D doméně. Klíč: Černé křivky = standardní LHS, Modré křivky = náhodná výměna, Okrové křivky = heuristická metoda, Azurové body = odebírání bodů (odebrán náhodný z dvojice), Purpurové body = odebírání bodů (odebrán “horší” z dvojice), Červený bod = Delaunayova triangulace s re-triangulováním, Zelený bod = Delaunayova triangulace jen v určitých iteracích; *smooth on* generuje body náhodně v rámci intervalu; (10,90) znamená 10 náhodných počátečních bodů a 90 bodů přidanych triangulací, ($n - > 100$) označuje 100 bodů zbylých po odebírání z n původně vygenerovaných a (100,3) značí 100 bodů ve 3D; tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty.



Obrázek 12: Porovnání rychlosti (*čas*) a kvality (*maximin* - vyšší hodnota je lepší) algoritmů na 4D doméně. Klíč: Černé křivky = standardní LHS, Modré křivky = náhodná výměna, Okrové křivky = heuristická metoda, Azurové body = odebrání bodů (odebrán náhodný z dvojice), Purpurové body = odebrání bodů (odebrán "horší" z dvojice), Červený bod = Delaunayova triangulace s re-triangulováním, Zelený bod = Delaunayova triangulace jen v určitých iteracích; *smooth on* generuje body náhodně v rámci intervalu; (10,90) znamená 10 náhodných počátečních bodů a 90 bodů přidanych triangulací, ($n - > 100$) označuje 100 bodů zbylých po odebrání z n původně vygenerovaných a (100,4) značí 100 bodů ve 4D; tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty.



Delaunay-stálá triangulace (10,90)

Delaunay-rozdělování (10,90)

LHS-ubírání (250->100, 500->100, 750->100, 1000->100)

LHS-ubírání-NEW (200->100, 300->100, 400->100)

lhsdesign(100,5), default, smooth on

lhsdesign(100,5), výměna - náhodný + náhodný, smooth on

lhsdesign(100,5), výměna - Cmin + náhodný, smooth on

Obrázek 13: Porovnání rychlosti (*čas*) a kvality (*maximin* - vyšší hodnota je lepší) algoritmů na 5D doméně. Klíč: Černé křivky = standardní LHS, Modré křivky = náhodná výměna, Okrové křivky = heuristická metoda, Azurové body = odebírání bodů (odebrán náhodný z dvojice), Purpurové body = odebírání bodů (odebrán "horší" z dvojice), Červený bod = Delaunayova triangulace s re-triangulováním, Zelený bod = Delaunayova triangulace jen v určitých iteracích; *smooth on* generuje body náhodně v rámci intervalu; (10,90) znamená 10 náhodných počátečních bodů a 90 bodů přidanych triangulací, ($n - > 100$) označuje 100 bodů zbylých po odebírání z n původně vygenerovaných a (100, 5) značí 100 bodů v 5D; tenké barevné křivky a znaménka + znázorňují minimální a maximální hodnoty, tučná křivka značí průměrné hodnoty.

Literatura

- [Del, 2001] (2001). Delaunay triangulation. www stránky: http://en.wikipedia.org/wiki/Delaunay_triangulation/. Last modified on 15 March 2011.
- [Hyp, 2002] (2002). Hypercube. www stránky: <http://en.wikipedia.org/wiki/Hypercube/>. Last modified on 7 March 2011.
- [Cen, 2003] (2003). Centroid. www stránky: <http://en.wikipedia.org/wiki/Centroid/>. Last modified on 4 February 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)/). Last modified on 24 February 2011.
- [Vol, 2011] (2011). Simplex volumes and the cayley-menger determinant. www stránky: <http://www.mathpages.com/home/kmath664/kmath664.htm/>.
- [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.
- [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.
- [Jin, 2005] Jin, Y. (2005). A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9:3–12.
- [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.
- [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.

- [Simpson et al., 2001] Simpson, T. W., Peplinski, J. D., Koch, P. N., and Allen, J. K. (2001). Metamodels for computer-based engineering design: Survey and recommendations. *Engineering with Computers*, 17:129–150.
- [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.

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):

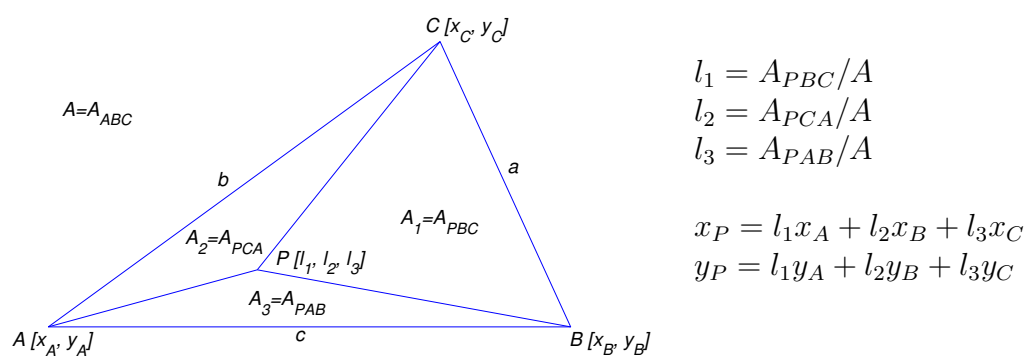
$$V_n = \frac{1}{n!} \begin{vmatrix} 1 & x_{1(1)} & x_{2(1)} & \dots & \dots & x_{n(1)} \\ 1 & x_{1(2)} & x_{2(2)} & \dots & \dots & x_{n(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1(n+1)} & x_{2(n+1)} & \dots & \dots & x_{n(n+1)} \end{vmatrix}$$

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

Příloha B

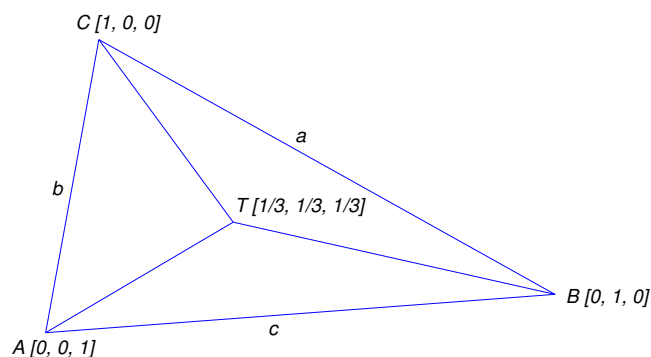
Výpočet souřadnic těžiště simplexu

K výpočtu souřadnic těžiště jsou použity plošné (trojúhelníkové) 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 čtyřřetěnu $[1/4, 1/4, 1/4, 1/4]$, těžiště n-dimenzionálního simplexu potom $[1/n, 1/n, \dots, 1/n]$ [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

Výpočet vzájemných vzdáleností bodů návrhu

```
1 function s=maximin_full(x)
2                                     % x je matice souřadnic
3                                     % bodů návrhu (počet řádků
4                                     % odpovídá aktuálnímu počtu
5                                     % bodů návrhu)
6     [m,p] = size(x);
7     pp = (m-1):-1:2;
8     I = zeros(m*(m-1)/2,1);
9     I(cumsum([1 pp])) = 1;
10    I = cumsum(I);
11    J = ones(m*(m-1)/2,1);
12    J(cumsum(pp)+1) = 2-pp;
13    J(1)=2;
14    J = cumsum(J);
15
16    d = zeros(size(I));
17    for j=1:p
18        d = d + (x(I,j)-x(J,j)).^2;
19                                     % d je vektor vzájemných
20                                     % vzdáleností bodů
21    end
22
23    s = sqrt(min(d));
24                                     % s značí minimální
25                                     % ze vzájemných vzdáleností
26                                     % bodů (maximin)
```