Rozšíření programu distmesh pro vícedimenzionální problémy

Rektorysova soutěž
Katedra matematiky
Stavební fakulta ČVUT

Marek Tyburec
marek.tyburec@gmail.com

Vedoucí projektu: Doc. Ing. Matěj Lepš Ph.D.

říjen 2014

Obsah

1 Úvod **2**

2 Metoda `distmesh` **2**

3 Metoda `distmesh` založená na hledání nejbližších bodů **2**

4 Metody `shlukování`, `ubírání` a `ubírání_NEW` **3**

4.1 Metoda `shlukování` 4

4.2 Metoda `ubírání` 4

4.3 Metoda `ubírání_NEW` 4

5 Srovnání jednotlivých metod **4**

6 Závěr **5**

1 Úvod

Metody návrhu experimentů, taktéž uváděné jako *DoEs* (z anglického *design of experiments*), vychází z požadavku vhodného statistického zpracování dat tak, aby byly návrhové oblasti rovnoměrně pokryty a zpracovaná data tak reprezentovala celou návrhovou množinu (prostor).

Tato práce vychází z nástroje `distmesh` [4], který tato práce následně upravuje a výsledky porovnává s jeho původní verzí a metodami `ubírání`, `ubírání_NEW` a `shlukování` [2]. Návrhovým prostorem je uvažována hyperkoule o poloměru 1.

2 Metoda `distmesh`[1]

`Distmesh` je jednoduchý algoritmus, který je využíván pro generování sítí pro metodu konečných prvků (MKP).

Do algoritmu vstupuje sada návrhových bodů P, které leží uvnitř návrhového prostoru N. Jeho tvar je popsán pomocí distanční funkce $d(x_1, ..., x_n)$, která je definována jako:

$$\begin{aligned} d(x_1, ..., x_n) &\leq 0 \text{ pro } x \in N \\ d(x_1, ..., x_n) &> 0 \text{ pro } x \notin N, \end{aligned} \tag{1}$$

kde n je počet dimenzí návrhového prostoru.

Dále jsou využitím *Delaunayovy triangulace* jednotlivé body P navzájem propojeny trojúhelníkovou sítí. Vzniká tím analogická úloha k příhradové konstrukci, kdy jednotlivé body představují uzly konstrukce a vzniklá propojení mezi nimi pruty.

Jako u příhradové konstrukce, tak i zde musí být zachována podmínka silové rovnováhy. Je vypočtena optimální vzdálenost mezi pruty L_0 a skutečná délka prutů L. Z jejich rozdílu je určena velikost působící síly F:

$$F = L_0 - L \tag{2}$$

Výsledná síla je rozložena do všech směrů (dimenzí). Důsledkem působících sil jsou posuny. V závislosti na zvoleném časovém kroku Δt lze proto provést samotné posunutí jednotlivých bodů o vzdálenost ΔP:

$$P = P + \Delta P = P + \Delta t F \tag{3}$$

Pokud se některý z bodů posune mimo předepsaný návrhový prostor N, je generováno takové vnější silové zatížení, aby k posunu nedošlo. Toto vnější silové zatížení působí ve směru gradientu distanční funkce.

V případě, že se uzly posunou více, než je předepsaná tolerance, jsou aktualizované souřadnice vstupem pro opětovné vytvoření Delaunayovy triangulační sítě. V opačném případě získáváme optimální rozmístění uzlů. Cyklus je ukončen také při překročení předepsaného množství iterací.

3 Metoda `distmesh` založená na hledání nejbližších bodů

Časově a výpočetně nejnáročnějším krokem metody `distmesh` je Dalaunayova triangulace. Výpočetní nároky rostou exponencionálně s každou dimenzí návrhové oblasti. Je proto navržena úprava funkce `distmesh` tak, aby v ní byla Dalaunayova triangulace nahrazena.

Z analogie úlohy k příhradové konstrukci vyplývá, že je nutné jednotlivé body propojit vazbami, tj. pruty. Bylo vybráno napojení každého bodu k s nejbližším bodům (sousedům), tzn. každý bod vytváří s vazeb a je součástí alespoň s prutů konstrukce. K samotnému nalezení sousedních bodů byla použita v softwaru Matlab funkce `knnsearch`.

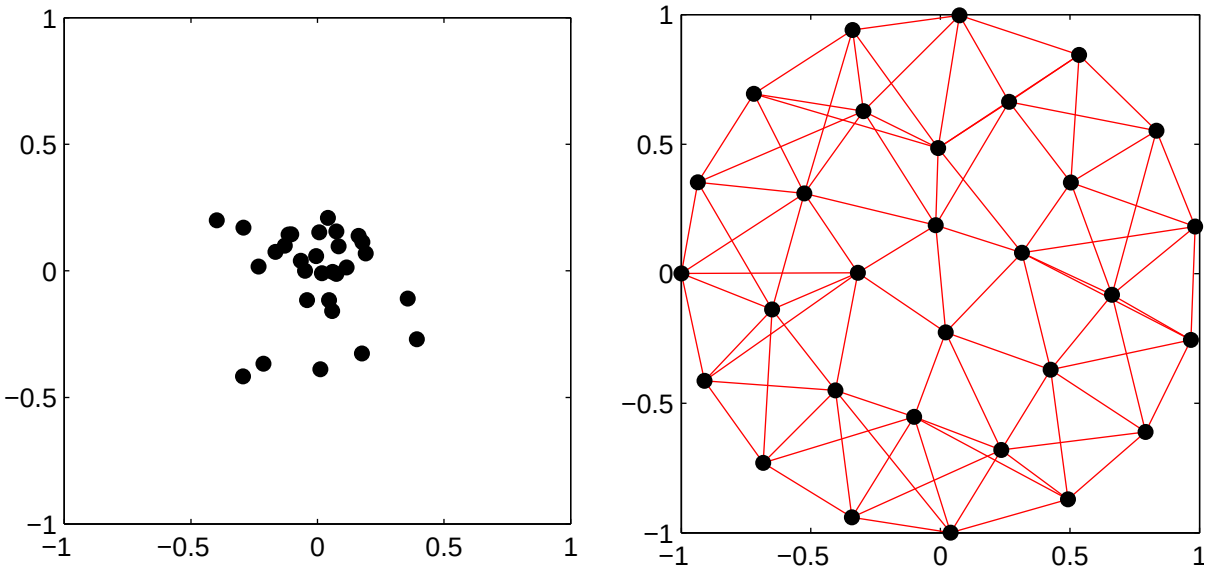
[1] Použita byla implementace z webových stránek http://persson.berkeley.edu/distmesh/distmesh.zip

Po vytvoření vazeb mezi body, tzn. prutů, pokračuje algoritmus podle původního schématu (viz Algoritmus 1).

Příklad takto řešené 2-dimenzionální úlohy je znázorněn na Obrázku 1.

Algoritmus 1 Zjednodušené schéma upraveného algoritmu distmesh

```
P : P ∈ N                                   ▷ Vstupním parametrem jsou návrhové body P ležící v návrhové oblasti N
ΔP = Inf                                    ▷ Pro vstup do while cyklu
while (ΔP > P_max) ∧ (iter ≤ iter_max) do   ▷ Pokud jsou posuny ΔP vetší než předepsaná tolerance P_max nebo byl překročen počet povolených iterací
    knnsearch(P,P,s)                        ▷ Nalezení s sousedních bodů pro každý bod P
    bars(P)                                 ▷ Vytvoření vazeb mezi sousedními body
    L_0,L ← bars(P)                         ▷ Dopočítání optimálních a nynějších délek prutů
    F = L_0 − L                             ▷ Spočítání velikostí působících sil
    F_vec ← F                               ▷ Rozložení sil F do jednotlivých směrů
    ΔP = F_vec Δt                           ▷ Vypočtení posunů bodů v čase Δt
    P = P + ΔP                              ▷ Nové souřadnice bodů
    if P ∉ N then                           ▷ Pokud bod opustí návrhovou oblast
        generateF(P ∉ N)                    ▷ Vytvoření takové síly, aby bod neopustil návrhovou oblast
    end if
    iter = iter+1                           ▷ Další iterace
end while
return P                                    ▷ Získány optimální pozice bodů P
```



Obrázek 1: Příklad řešené 2-dimenzionální úlohy upravenou metodou `distmesh`. Na levém obrázku je zobrazeno 30 vstupních bodů P. Pro každý bod bylo uvažováno 5 nejbližších sousedních bodů. Na pravém obrázku je stav po 100 iteračních cyklech. Návrhovou oblastí je kruh o poloměru 1.

4 Metody `shlukování`, `ubírání` a `ubírání_NEW`

Metody `shlukování`, `ubírání` a `ubírání_NEW` [2] slouží k zaplnění návrhového prostoru náhodně generovanými body. Výhodou těchto metod je jejich značná rychlost.

4.1 Metoda `shlukování`

Metoda **`shlukování`** je založená na rozdělení návrhové oblasti na přesně tolik podoblastí, kolik je návrhových bodů. Do každé podoblasti je následně náhodně umístěn právě jeden bod. Tyto body poté tvoří samotný návrh. Jedná se o velmi rychlou metodu, avšak i značně závislou na principu náhody.[2]

4.2 Metoda `ubírání`

U metody **`ubírání`** je vytvořen přehuštěný návrh náhodných bodů náležejících návrhovému prostoru. Následně je v každé iteraci náhodně odebrán jeden bod z dvojice bodů, které mají nejmenší vzájemnou vzdálenost, tzn. hodnotu EMM. Výsledný návrh má vyšší kvalitu než návrh metodou **`shlukování`**.[2]

4.3 Metoda `ubírání_NEW`

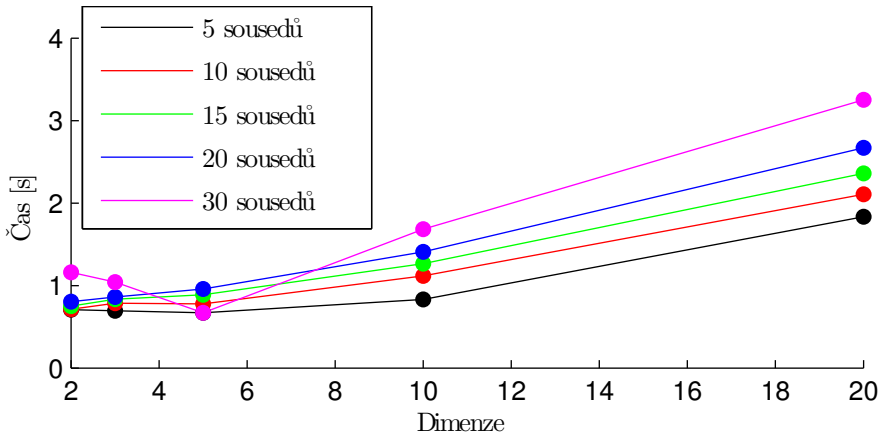
Metoda **`ubírání_NEW`** upravuje metodu **`ubírání`** tak, že odstranění jednoho ze dvou vzájemně si nejbližších bodů se již neděje náhodně, ale je odstraněn ten, který je blíže ke zbývajícím bodům. Výsledkem je opět o něco větší kvalita návrhu.[2]

5 Srovnání jednotlivých metod

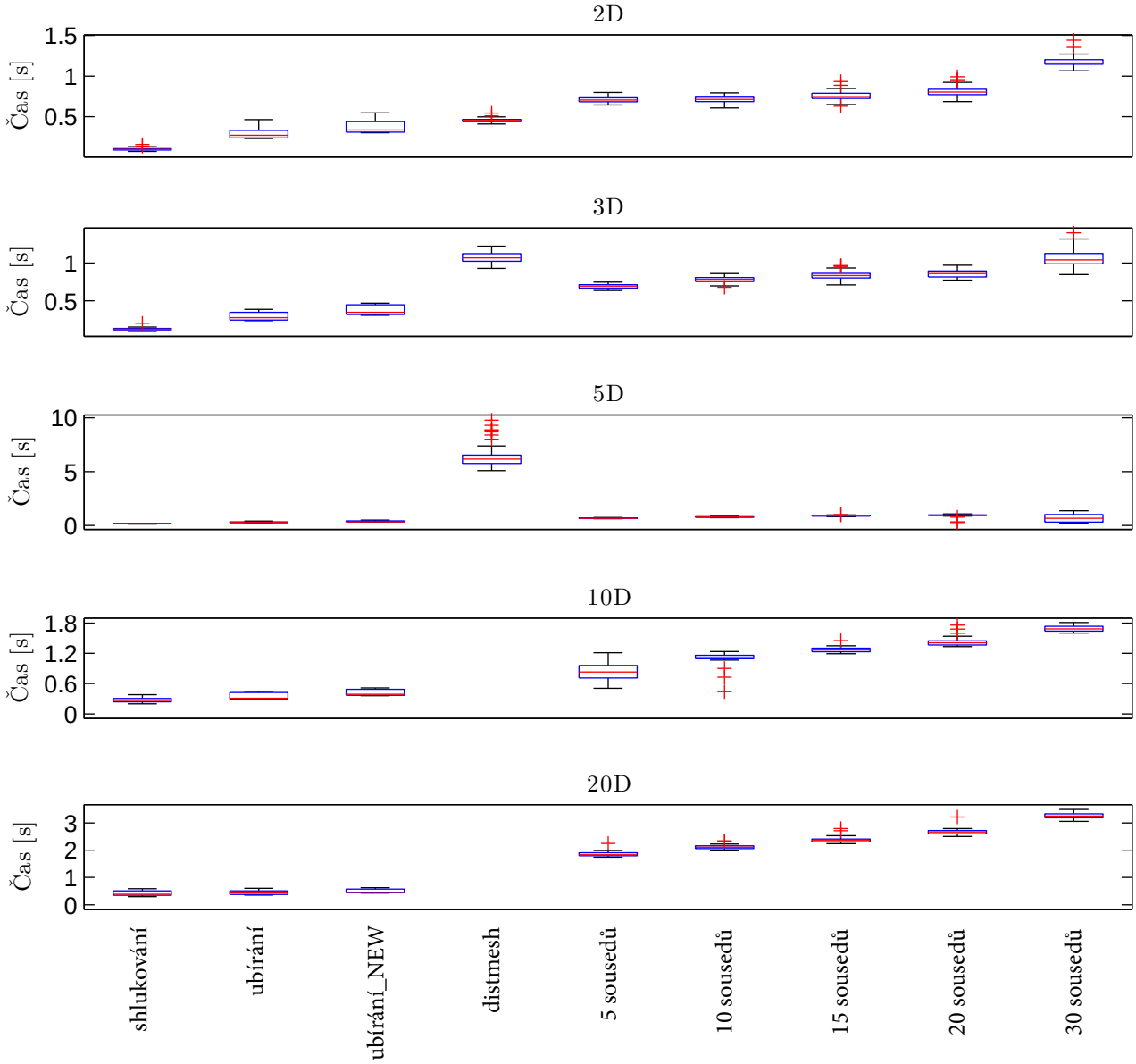
Výše popsané metody byly srovnány běžně používanými kritérii. Návrh byl proveden pro 100 bodů ve 2D, 3D, 5D, 10D a 20D. Pro každou kombinaci metody a dimenze bylo provedeno 100 výpočtů s maximálním počtem 100 iterací. Všechny výpočty byly provedeny za stejných podmínek. Pro metody **`ubírání`** a **`ubírání_NEW`** bylo uvažováno 500 bodů pro vytvoření přehuštěného návrhu. V případě původní metody **`distmesh`** bylo měření provedeno jen do 5-dimenzionálního návrhu, jelikož dostupný hardware nebyl dostatečný pro výpočet většího počtu dimenzí.

Hodnocenými kritériemi byla doba výpočtu t (viz Obrázek 2 a 3), kritéria na ortogonalitu: *Pearsonův korelační koeficient* ($PMCC$), *Kendallův korelační koeficient* ($KRCC$), *Spearmanův korelační koeficient* ($SRCC$), *číslo podmíněnosti* (CN) a *největší hodnota korelační matice* (cc). Dále byla posuzována kriteria hodnotící rovnoměrnost pokrytí: *Audze-Eglais* (AE), *Eukleidovská maximin vzálenost* (EMM) a *modifikovaná L_2 diskrepance* (ML_2). Porovnání je zobrazeno na Obrázkách 4-8.

Hodnota AE udává potenciální energii. Nejlepšího rozmístění bodů je dosaženo při minimální hodnotě AE. Hodnota EMM udává minimální vzdálenost mezi všemi body. Její hodnota má být maximální. Hodnota ML_2 má být minimální. Číslo podmíněnosti (CN) se minimalizuje, a pokud se blíží 1, je ortogonalita návrhu největší. V případě $PMCC$, $KRCC$ a $SRCC$ se minimalizuje odmocnina sumy druhých mocnin příslušných korelačních koeficientů. Hodnota cc se také minimalizuje.[1]



Obrázek 2: Časová závislost upravené metody **`distmesh`** na počtu dimenzí. Zobrazené jsou hodnoty medianu ze 100 měření.

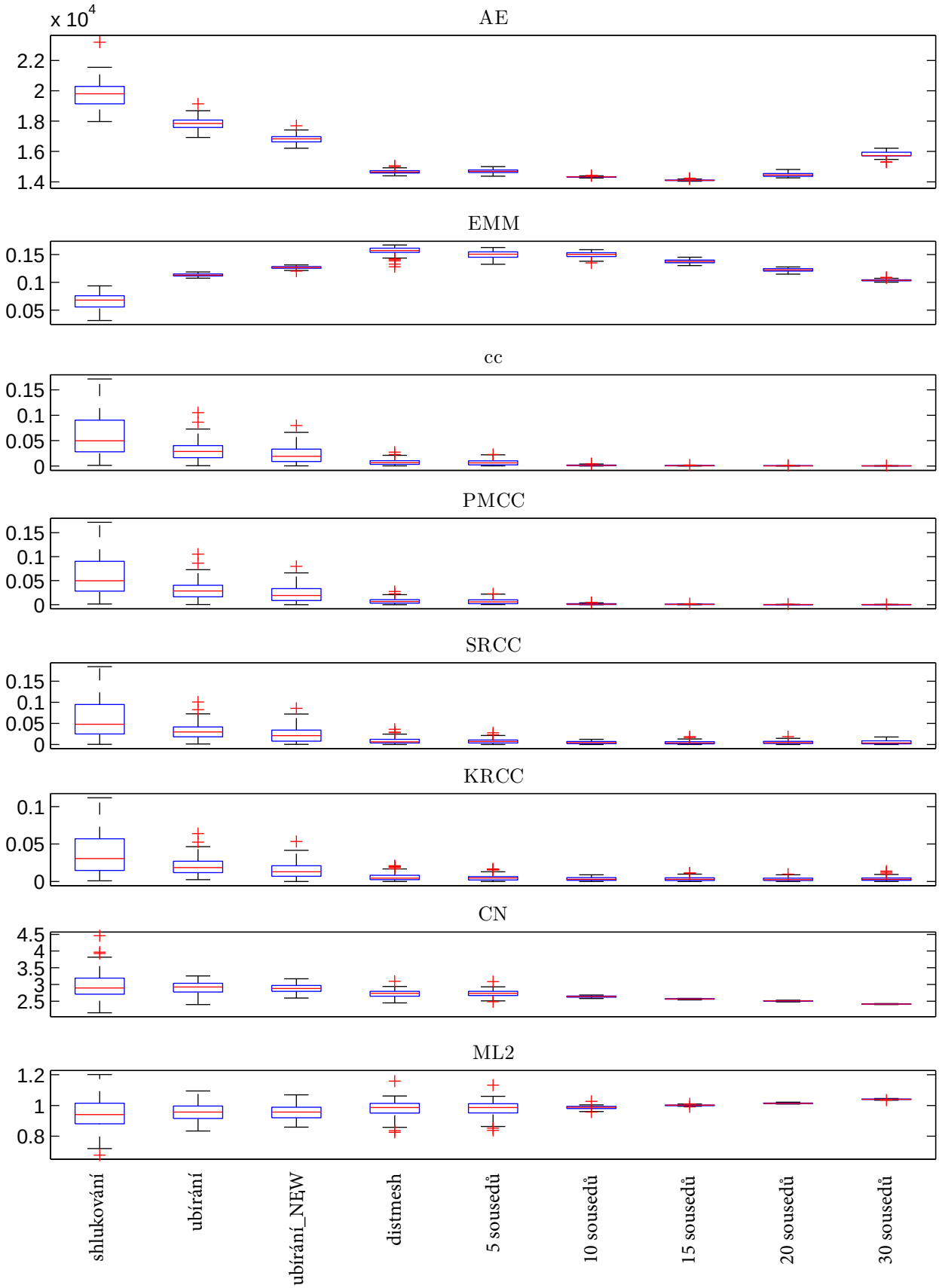


Obrázek 3: Porovnání časové náročnosti provedených výpočtů. Výsledky zobrazují statistické údaje ze 100 spuštění.

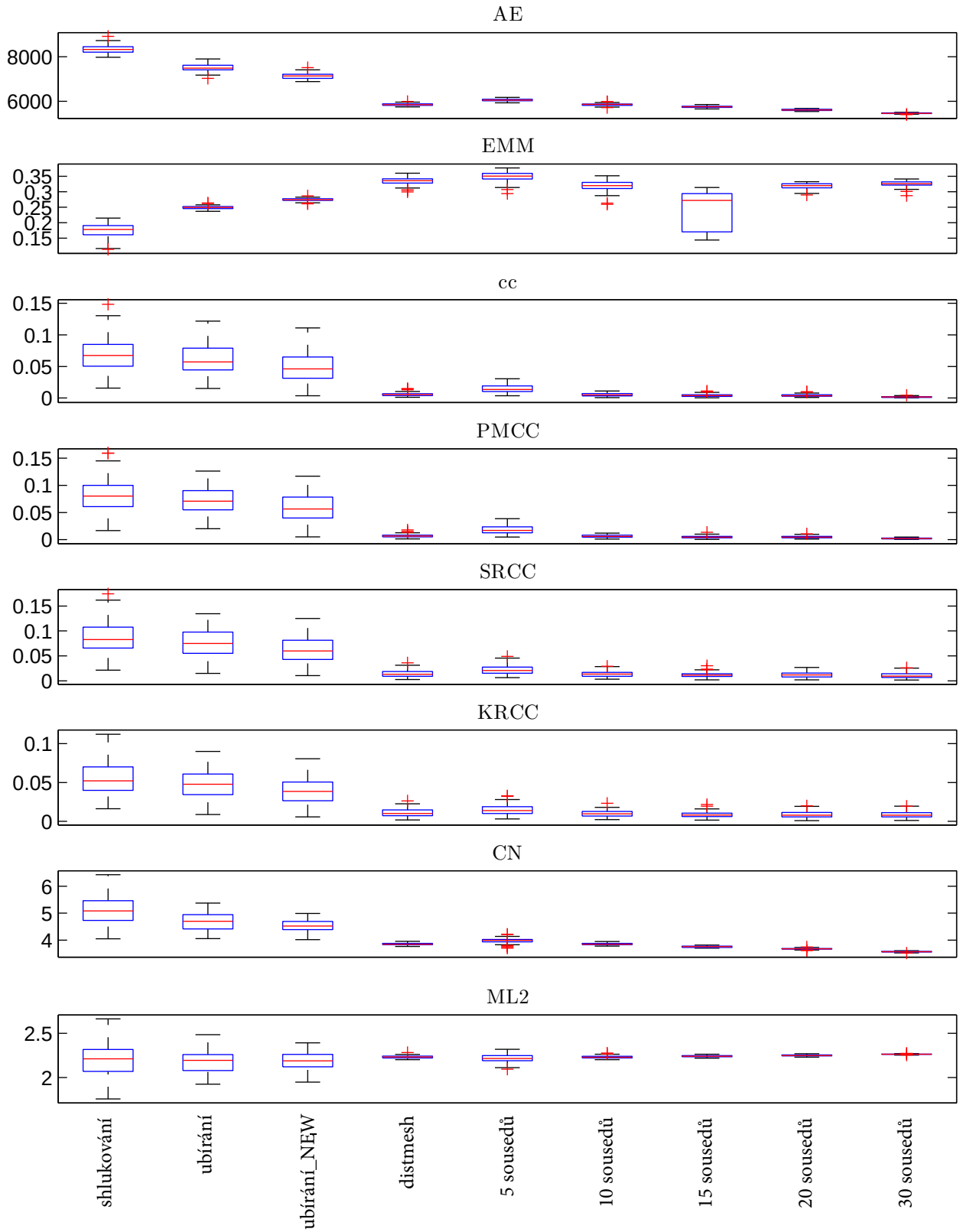
6 Závěr

Dle naměřených hodnot vyplývá, že použitá úprava algoritmu `distmesh` úlohu značně zrychlila, oproti původní metodě již není časová závislost na počtu dimenzích exponencionální, ale přibližně lineární, viz Obrázek 2. Kvalitativně dosahuje podobných výsledků, viz Obrázek 4-8.

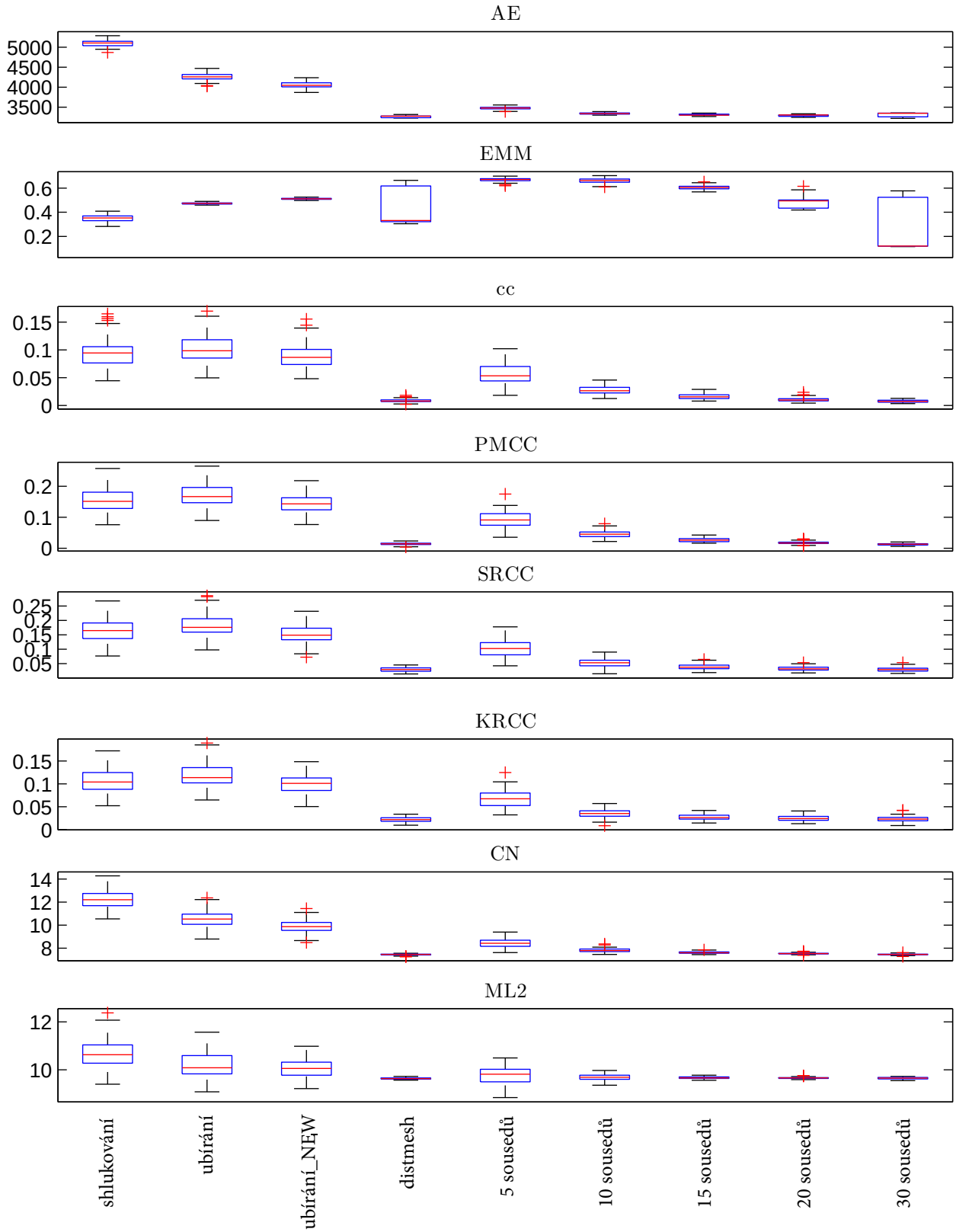
Porovnáme-li upravenou metodu `distmesh` s metodami `shlukování`, `ubírání` a `ubírání_NEW`, je patrné, že se jedná o metodu několikanásobně pomalejší, nicméně při správné volbě počtu nejbližších bodů dosahuje lepšího výsledného návrhu.



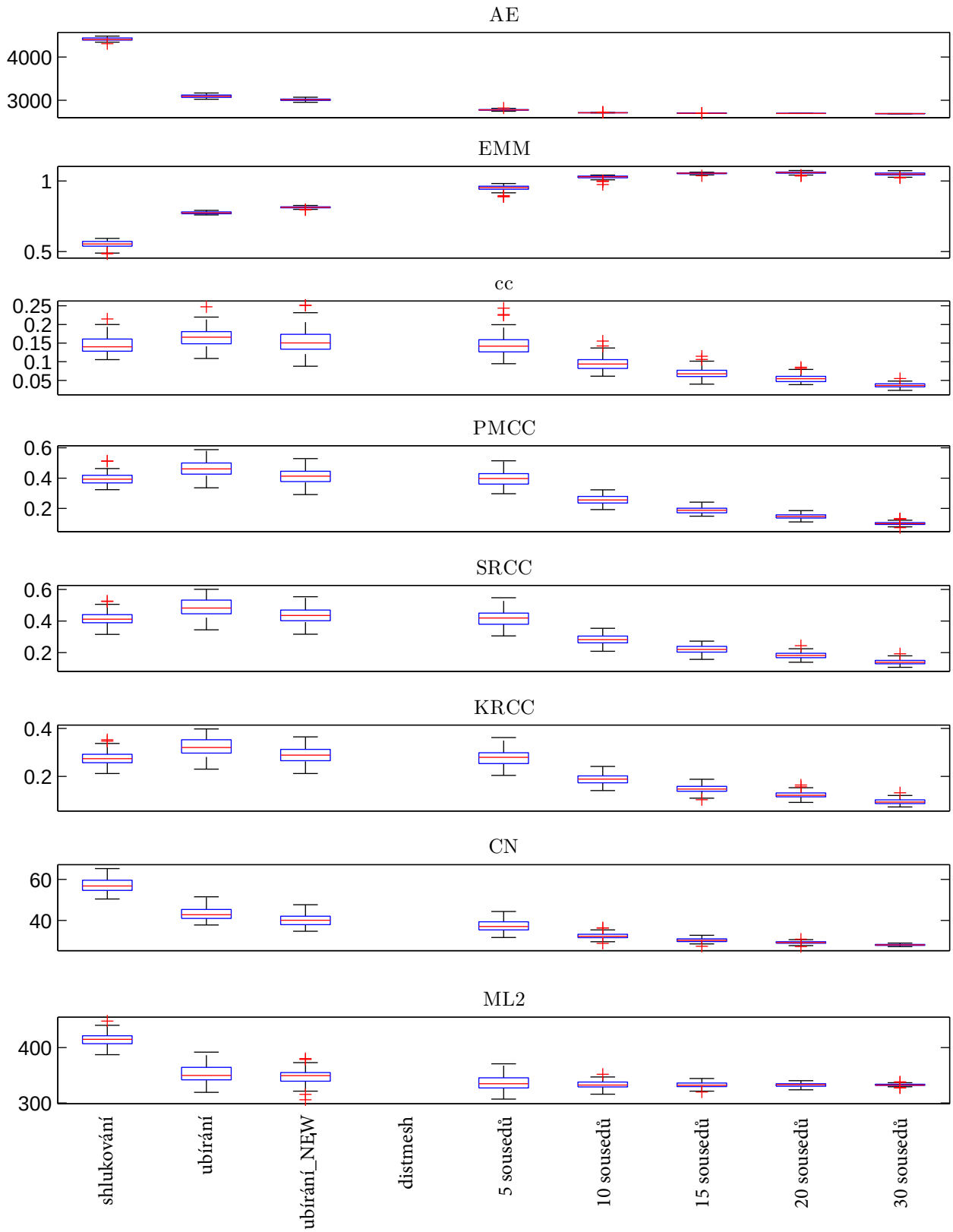
Obrázek 4: Srovnání jednotlivých metod pro 2D návrhovou oblast. Metody s uvedenými počty sousedů jsou řešeny pomocí upravené verze algoritmu `distmesh`.



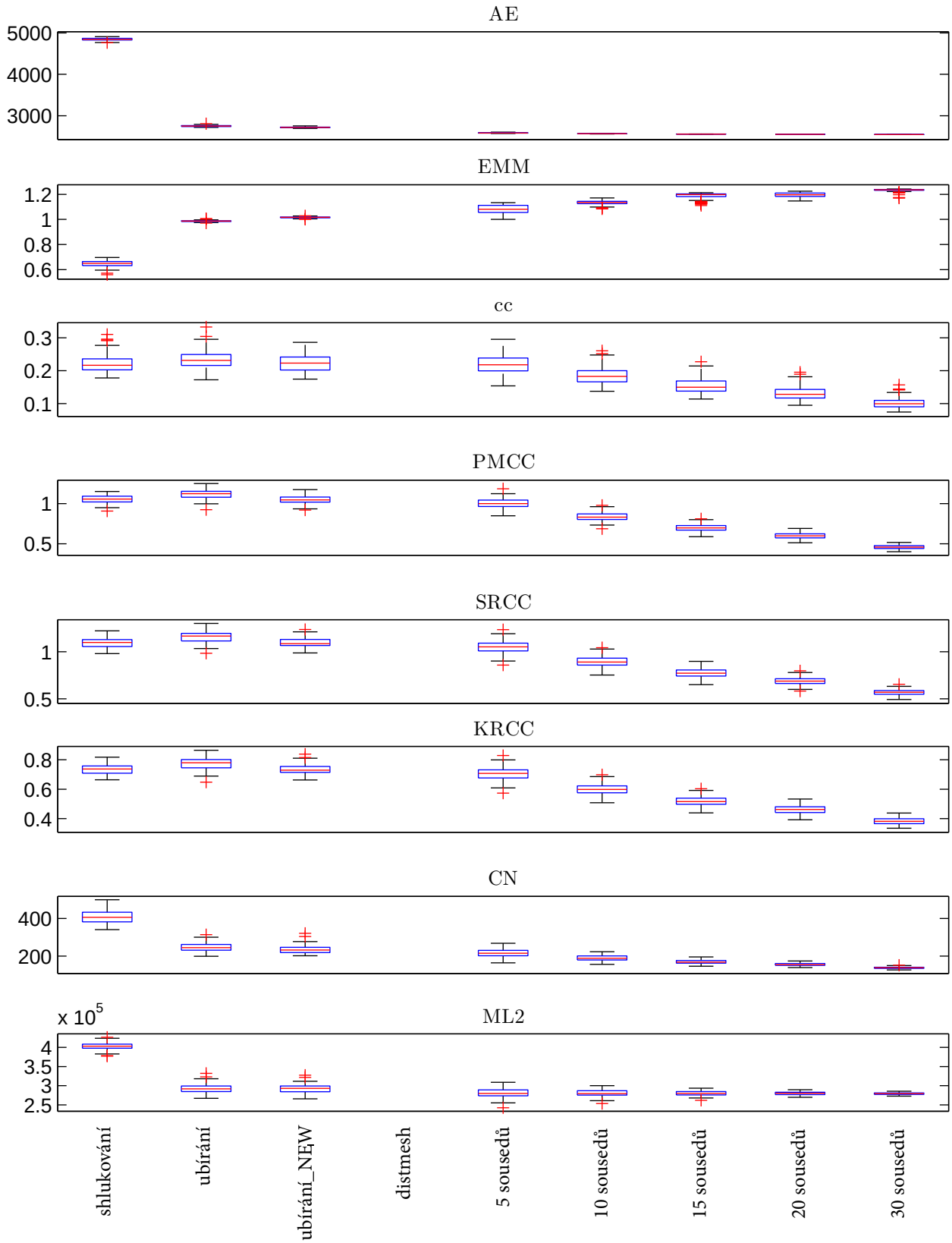
Obrázek 5: Srovnání jednotlivých metod pro 3D návrhovou oblast. Metody s uvedenými počty sousedů jsou řešeny pomocí upravené verze algoritmu `distmesh`.



Obrázek 6: Srovnání jednotlivých metod pro 5D návrhovou oblast. Metody s uvedenými počty sousedů jsou řešeny pomocí upravené verze algoritmu `distmesh`.



Obrázek 7: Srovnání jednotlivých metod pro 10D návrhovou oblast. Metody s uvedenými počty sousedů jsou řešeny pomocí upravené verze algoritmu `distmesh`.



Obrázek 8: Srovnání jednotlivých metod pro 20D návrhovou oblast. Metody s uvedenými počty sousedů jsou řešeny pomocí upravené verze algoritmu `distmesh`.

Reference

[1] E. Janouchová and A. Kučerová, *Competitive comparison of optimal designs of experiments for sampling-based sensitivity analysis*, Computers & Structures, 124 (2013), pp. 47–60.

[2] E. Myšáková, *Optimalizace uniformity počítačových návrhů pro omezené návrhové prostory.*, Master's thesis, ČVUT, 2014.

[3] E. Myšáková and M. Lepš, *Uniform space-filling design of experiments in hypersphere*, Nano and Macro Mechanics, (2014).

[4] P.-O. Persson and G. Strang, *A simple mesh generator in matlab*, SIAM review, 46 (2004), pp. 329–345.