

EVO - Aplikované evoluční algoritmy

Simulované žihání - optimalizace řadicí sítě

Číslo zadania: 2

Meno a priezvisko: Richard Seipel

Login: xseipe00

1 Úvod

Cieľom projektu bolo optimalizovať radiacu sieť použitím algoritmu simulovaného žihania. Bolo potrebné zvoliť vhodnú reprezentáciu radiacej siete a simulovať jej použitie. Významnou časťou bolo experimentovanie s nastavením parametrov a výber najvhodnejších z nich. Výsledkom je program gerenujúci návrh radiacej siete, ktorá sa snaží dosiahnuť čo najvyššiu presnosť a efektívnosť.

2 Implementácia radiacej siete

Pre implementáciu radiacej siete bolo potrebné vhodne zvoliť jej reprezentáciu. Zvolil som pole trojíc v tvare (i_1, i_2, u) , kde i_1 a i_2 reprezentujú pozíciu vybraných hodnôt v sieti, ktoré majú byť porovnané komparátorom a prípadne vymenené. Parameter u obsahuje hodnotu **True** alebo **False**, ktorá reprezentuje, či dané prepojenie (komparátor) bude v sieti použité. K pridaniu tohto parametra som sa inšpiroval v článku [4], priloženom v zadaní. Možnosť ignorovania komparátoru dovoľí optimalizačnému algoritmu zostavovať a ohodnocovať radiace siete s rôznym počtom komparátorov. Následne použitím správnej fitness funkcie bude hľadanie riešenia smerované k menšiemu počtu komparátorov, a tým pádom menšej a jednoduchšej sieti. Samotný komparátor je implementovaný jednoduchou funkciou, ktorá prijme dva parametre s hodnotami na vybraných pozíciách v sieti, a ak je prvá väčšia ako druhá vykoná ich výmenu. Simulácia siete prebieha postupným použitím komparátora na všetkých pozíciách určených sieťou nad vstupnými dátami. Sieť je na začiatku výpočtu inicializovaná so všetkými možnými kombináciami hodnôt pre danú dĺžku vstupu, kde prvá hodnota (i_1) je vždy väčšia ako druhá (i_2).

3 Implementácia simulovaného žihania

Algoritmus v každej iterácii vyhodnotí hodnotu fitness funkcie pre danú inštanciu siete. Následne vymení poradie dvoch náhodne zvolených komparátorov a s určitou pravdepodobnosťou odstráni alebo vráti niektorý komparátor (zmena parametra u). Následne vyhodnotí, či takto zmenená sieť bude použitá, prípadne uloží najlepšiu a pokračuje ďalšou iteráciou.

4 Fitness funkcia

Program na začiatku výpočtu hodnoty fitness náhodne vygeneruje niekoľko permutácií vstupných hodnôt a pre každú z nich spočíta počet nesprávne zoradených hodnôt. Počty

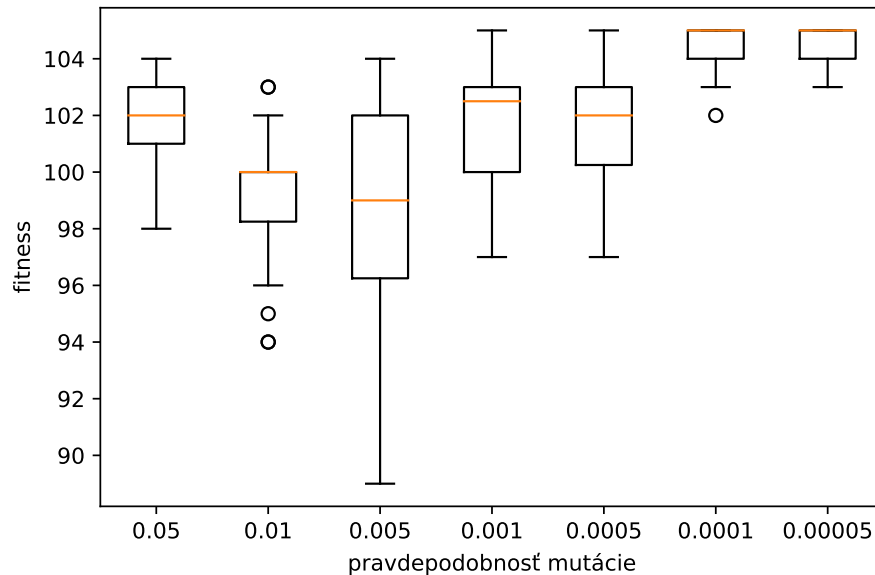
chýb sú následne sčítané. Cieľom tohto kroku je otestovať správnosť danej siete, keďže ohodnotenie všetkých možných kombinácií vstupov v každej iterácii by nebolo možné dosiahnuť v rozumnom čase už pre vyššie jednotky vstupov. Fitness funkcia umocní celkový počet nesprávne zoradených hodnôt a vynásobí ho konštantou. Následne k tejto hodnote pričíta počet komparátorov použitých v danej sieti. Týmto prístupom funkcia uprednostňuje siete s čo najnižším počtom komparátorov, no predovšetkým s veľmi nízkou až nulovou chybovosťou na testovacích vstupoch.

5 Použité parametre

Finálne zvolené parametre a metódy použité v programe boli vybrané po vykonaní rôznych experimentov a ich následnom štatistickom vyhodnotení. Bola testovaná pravdepodobnosť mutácie, rôzne metódy klesania teploty, ich parametre, aj počet testov pri určovaní hodnoty fitness. Každý test bol vykonávaný na 30 behoch algoritmu.

5.1 Mutácie

Aby bolo možné zmenšovať počet použitých komparátorov, v každom kroku algoritmu je každý komparátor s určitou pravdepodobnosťou pridaný alebo odobraný z generovanej siete. Táto hodnota je označená ako pravdepodobnosť mutácie. Boli testované rôzne hodnoty pri základnom geometrickom chladnutí s $\alpha = 0.99$ (obr. 1). Ako najlepšia vyšla hodnota 0,005.



Obr. 1: Výsledky výberu pravdepodobnosti mutácie.

5.2 Teplota

Prístup ku klesaniu teploty určuje ako veľmi a ako dlho bude algoritmus prehľadávať aj horšie riešenia a znižovať tak šancu, že uviazne v lokálnom extréme. V každom z prístupov je prítomná konštanta α , ktorá určuje ako rýchlo bude teplota klesať.

Prvým bolo geometrické chladnutie (1), kde je teplota v každom kroku násobená konštantou α a tým sa pre $\alpha < 1$ postupne znižuje [2]. Pre vyššiu hodnotu blízku 1 bude klesať pomalšie, pre nižšiu napr. 0.9 rýchlejšie. Experimentoval som hodnotami α od 0,75 do 0,99 (obr. 3).

$$T_i = T_{i-1} * \alpha \quad (1)$$

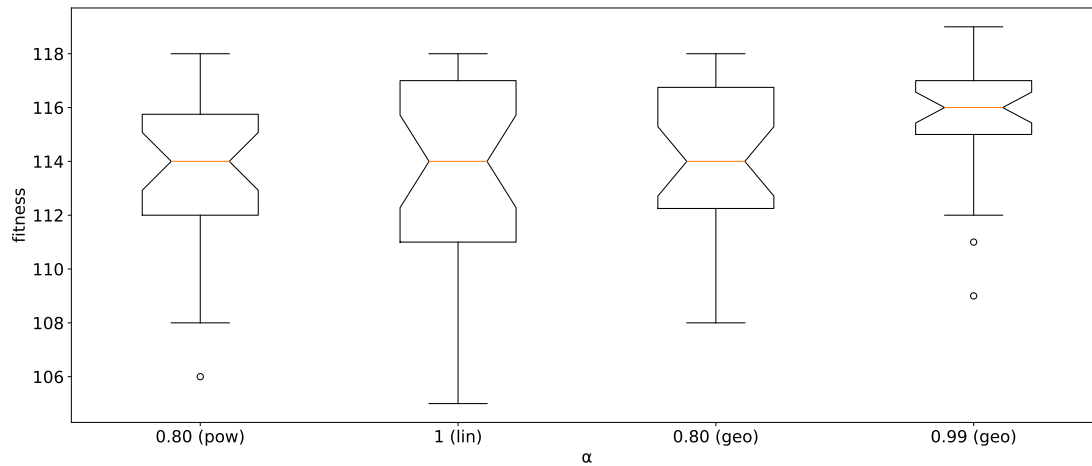
Ďalej som testoval polynomiálne chladnutie (2), ktoré má naviac parameter p , ktorý určuje akým číslom bude umocnený počet iterácií (obr. 4) [3].

$$T_i = \frac{T_0}{1 + (1 - \alpha) * (i - 1)^p} \quad (2)$$

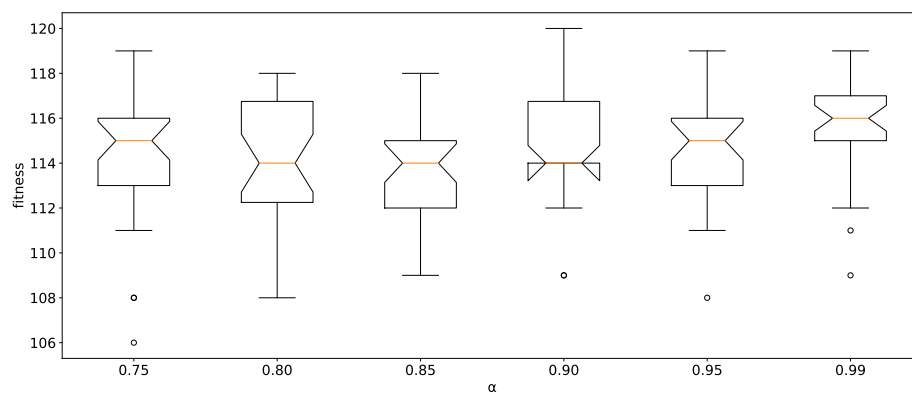
Posledným testovaným bolo lineárne chladnutie (3), pri ktorom klesanie teploty prebieha postupným odčítaním konštanty α od počiatočnej teploty [2]. Testované hodnoty v tomto prípade boli: 1,2,3,4,5 a 10, kde vyššia hodnota zaručuje rýchlejšie klesanie teploty (obr. 5).

$$T_i = \begin{cases} T_{i-1} - \alpha, & \text{ak } T_{i-1} > \alpha. \\ 0, & \text{inak} \end{cases} \quad (3)$$

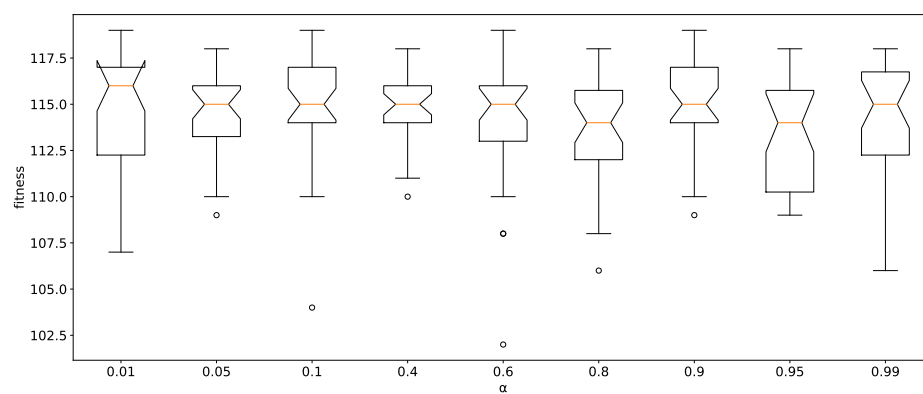
Výsledky testovania nakoniec neboli až tak významne odlišné aj pri použitom počte testovaných hodnôt a prístupov. Pri porovaní najlepších hodnôt α z každého prístupu k chladnutiu sa priemerné hodnoty fitness pri 16 prvkoch na vstupe a 10000 iteráciach pohybovali medzi 114 a 116 (obr. 2).



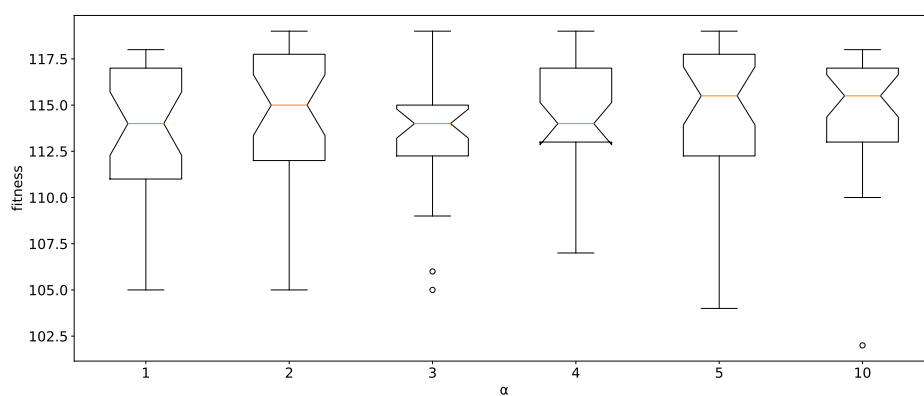
Obr. 2: Porovnanie najlepších α z každej skupiny.



Obr. 3: Geometrické klesanie teploty s rôznymi nastaveniami α .



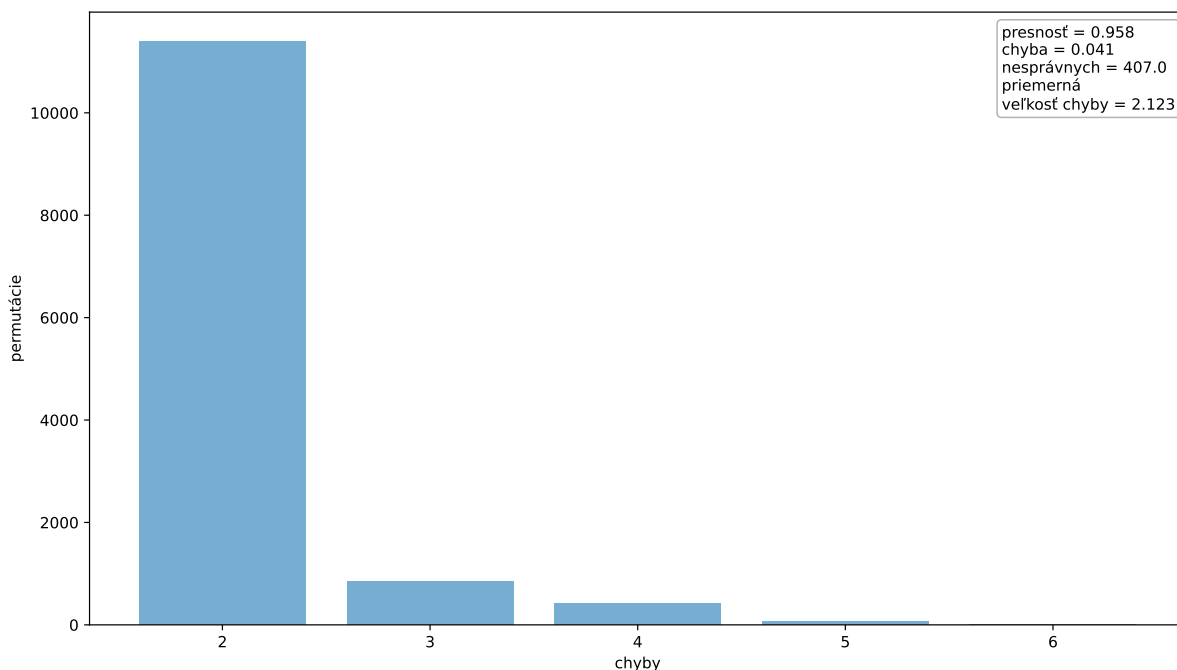
Obr. 4: Polynomiálne klesanie teploty s rôznymi nastaveniami α .



Obr. 5: Lineárne klesanie teploty s rôznymi nastaveniami α .

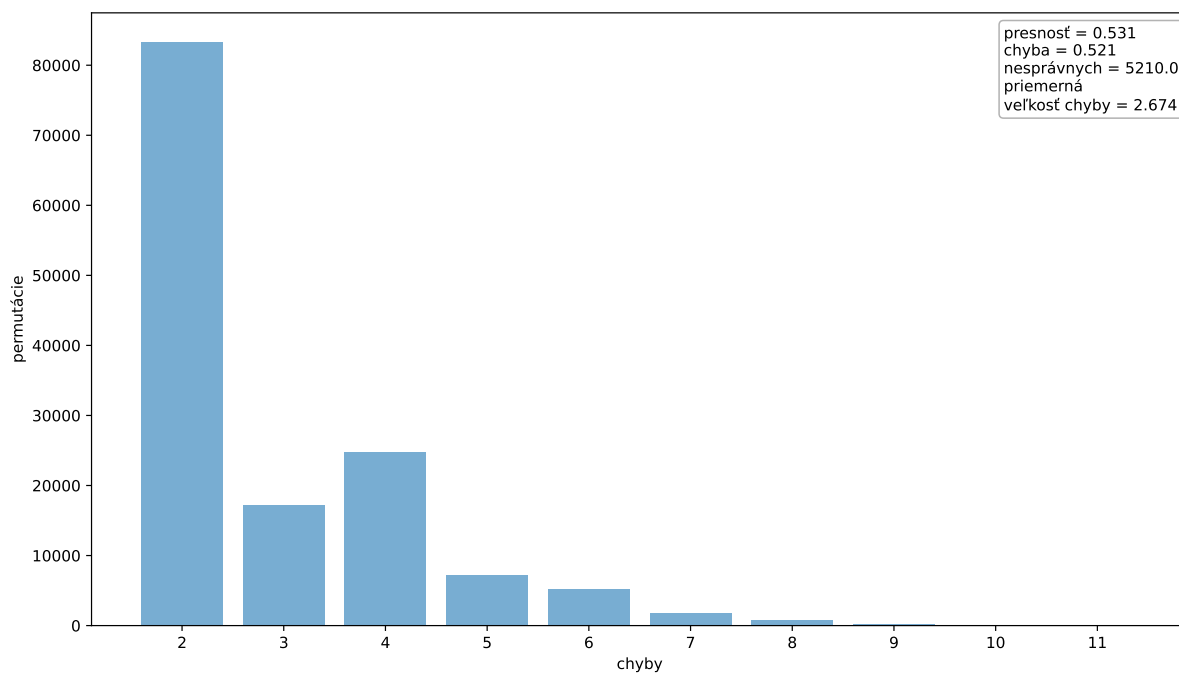
5.3 Chybovosť

Z dôvodu postupu vypočítavania fitness funkcie nie je možné zaručiť 100% presnosť vygenerovaných radiacích sietí. Preto bola pri testovaní sledovaná aj táto hodnota. Pre každú skupinu behov bola vyhodnotená chybovosť najlepšej siete z nej. Chybovosť tak bola sledovaná priebežne pri všetkých prebiehajúcich testoch a tiež bol samostatne testovaný vplyv počtu testov vo fitness funkcii na jej priemernú výšku vo všetkých behoch v skupine.

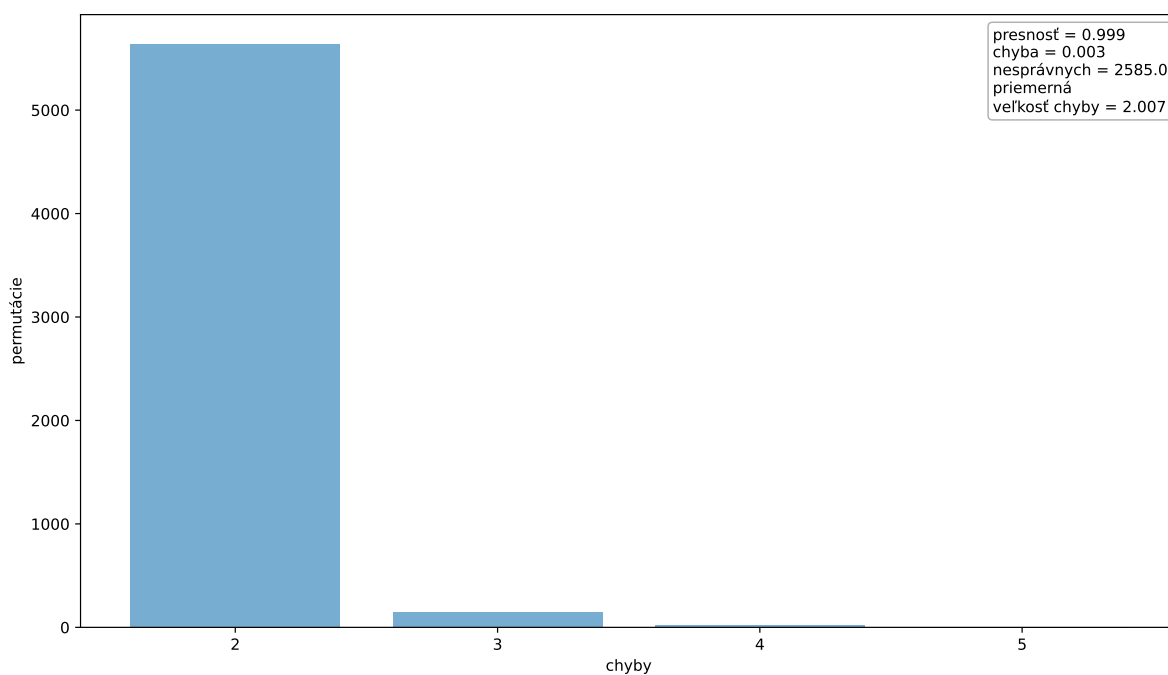


Obr. 6: 100 testov: Chybovosť pri geometrickom chladnutí s $\alpha = 0.99$, pravdepodobnosti mutácie 0.005, počte vstupov 16 a 100 testoch pri počítaní fitness funkcie.

Chybovosť pri finálne zvolených parametroch, počte vstupov 16 a 100 testoch (obr. 6) pri počítaní fitness funkcie, bola výrazne menšia ako pri nastavení počtu testov na hodnotu 10 (obr. 7). Pri skúške 1000 testov na 5 behoch programu sa priemerná presnosť dostala až na 99,9%, no výpočet pri 10000 iteráciách bol už značne časovo náročný (obr. 8).



Obr. 7: 10 testov: Chybovosť pri geometrickom chladnutí s $\alpha = 0.99$, pravdepodobnosti mutácie 0.005, počte vstupov 16 a 10 testoch pri počítaní fitness funkcie.

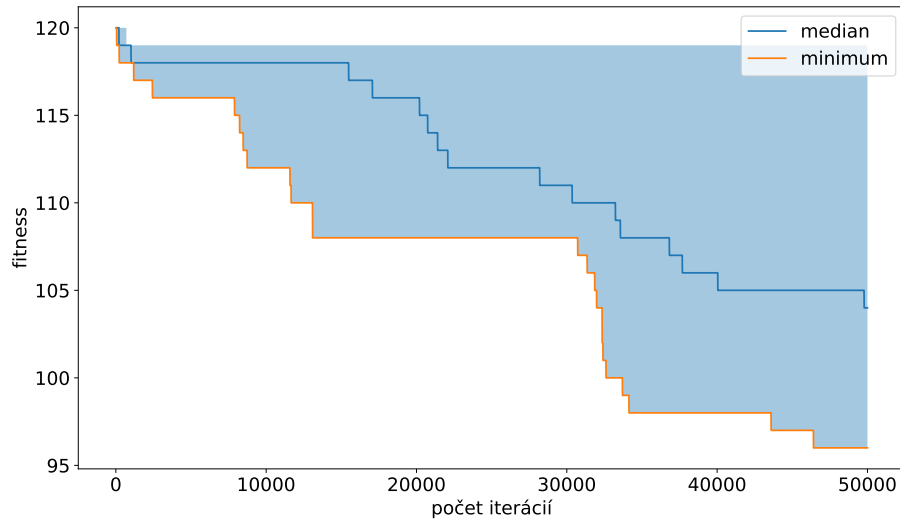


Obr. 8: 1000 testov: Chybovosť pri geometrickom chladnutí s $\alpha = 0.99$, pravdepodobnosti mutácie 0.005, počte vstupov 16 a 1000 testoch pri počítaní fitness funkcie.

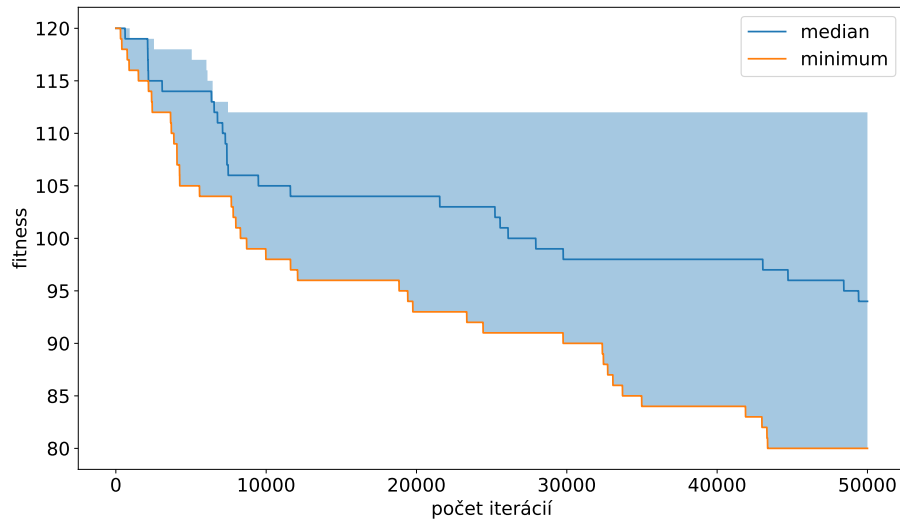
6 Výsledky

Výstupy z vyššie popísaných experimentov boli aplikované pri finálnych testoch algoritmu pri 50000 iteráciách a maximálne 500 testoch pri vyhodnocovaní fitness v každej iterácii.

Testované boli veľkosti vstupu 8, 12 a 16. Bola porovaná pôvodná implemetácia s geometrickým klesaním teploty a $\alpha = 0,99$ s implemenáciou využívajúcou polynomiálne klesanie teploty a hodnotu $\alpha = 0,8$. Podľa dostupných informácií [1] pre veľkosti vstupov 8 a 12 obe implementácie dosahovali výsledky blízke najlepším existujúcim riešeniam. Pre 12 to bolo 43 a 44 oproti najlepšiemu riešeniu 39 (obr. 13 a 14) a pre 8 obe našli najlepšie možné riešenie 19 (obr. 11 a 12). Pre veľkosť vstupu 16 bolo jasne vidieť rozdiel v prospech implementácie zvolenej podľa výsledkov experimentov (obr. 9 a 10).



Obr. 9: Vývoj fitness funkcie s pôvodným nastavením, geometrické chladnutie s $\alpha = 0,99$ pri veľkosti 16.

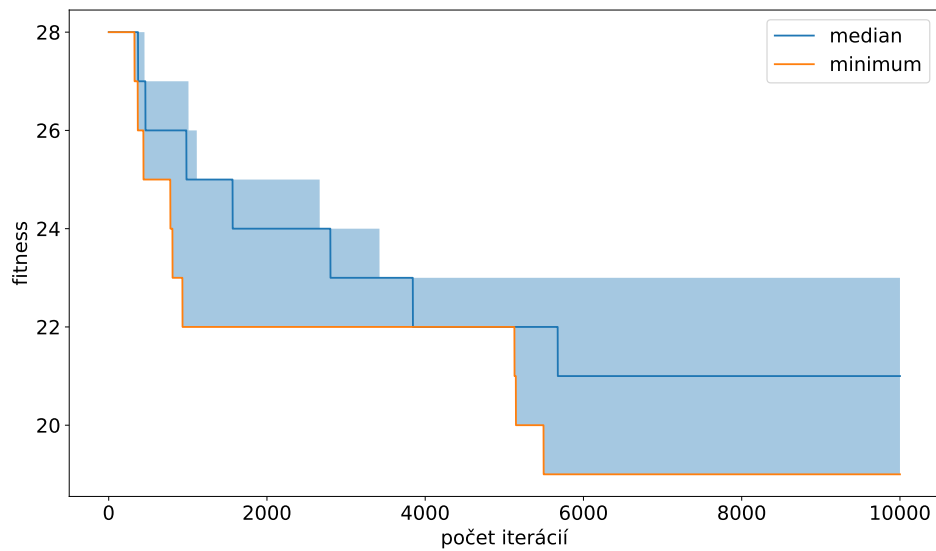


Obr. 10: Vývoj fitness funkcie s najlepším nastavením, polynomiálne chladnutie s $\alpha = 0,8$ pri veľkosti 16.

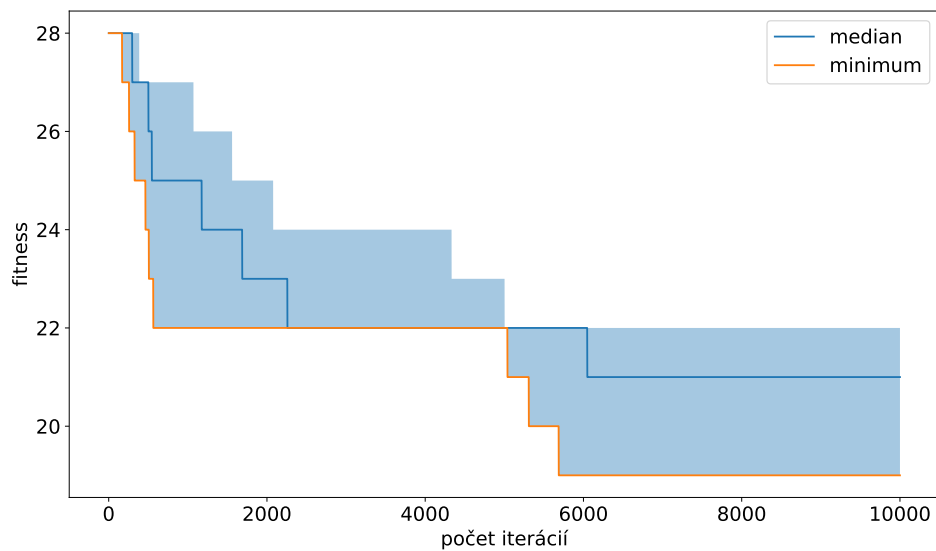
Literatúra

- [1] CODISH, M., CRUZ FILIPE, L., EHLERS, T., MÜLLER, M. a SCHNEIDER KAMP, P. Sorting networks: To the end and back again. *Journal of Computer and System Sciences*. Elsevier. 2019, zv. 104, s. 184–201.
- [2] MAHDI, W., MEDJAHED, S. A. a OUALI, M. Performance analysis of simulated annealing cooling schedules in the context of dense image matching. *Computación y Sistemas*. Instituto Politécnico Nacional, Centro de Investigación en Computación. 2017, zv. 21, č. 3, s. 493–501.
- [3] MARTÍN, J. F. D. a SIERRA, J. M. R. A comparison of cooling schedules for simulated annealing. In: *Encyclopedia of Artificial Intelligence*. IGI Global, 2009, s. 344–352. Dostupné z: <http://what-when-how.com/artificial-intelligence/a-comparison-of-cooling-schedules-for-simulated-annealing-artificial-intelligence/>.
- [4] MRAZEK, V. a VASICEK, Z. Automatic design of arbitrary-size approximate sorting networks with error guarantee. In: IEEE. *2016 26th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*. 2016, s. 221–228.

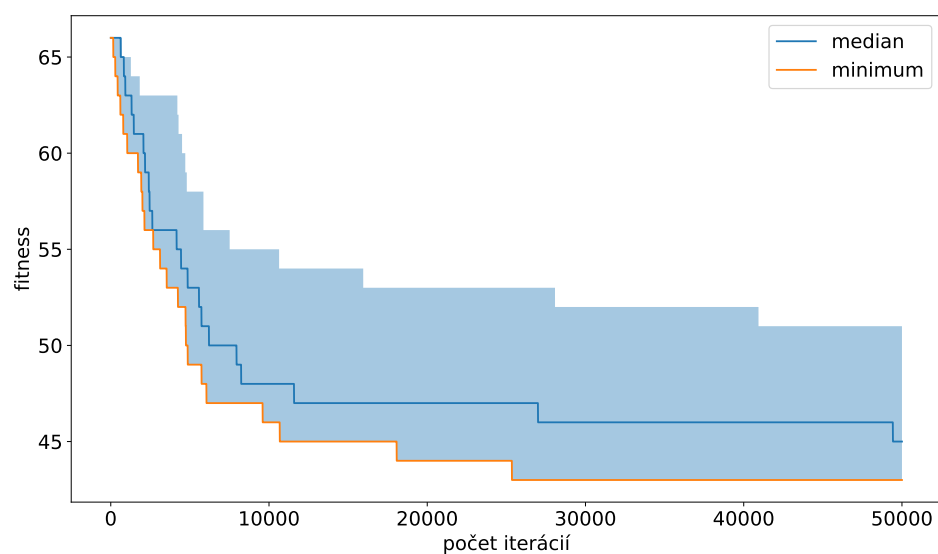
A Ďalšie výsledky



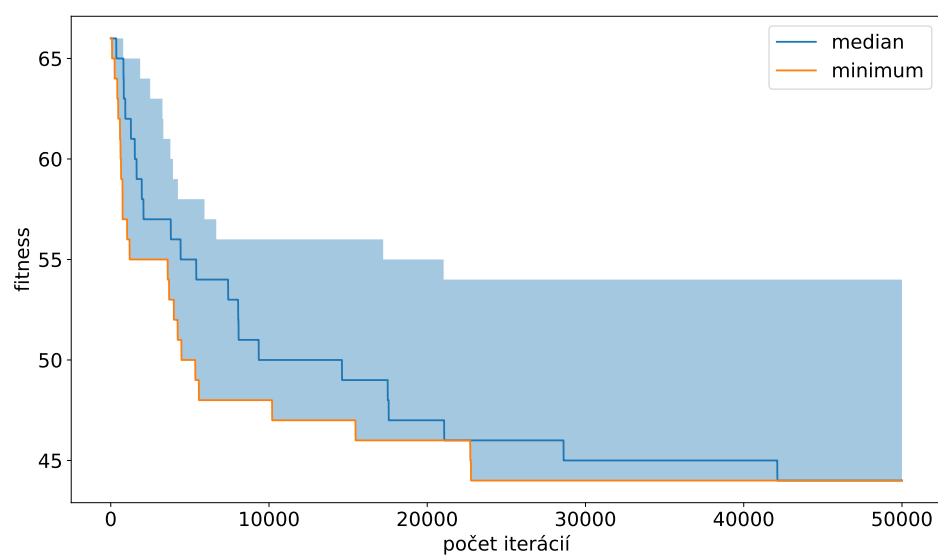
Obr. 11: Vývoj fitness funkcie s pôvodným nastavením, geometrické chladnutie s $\alpha = 0,99$ pri veľkosti 8.



Obr. 12: Vývoj fitness funkcie s najlepším nastavením, polynomiálne chladnutie s $\alpha = 0,8$ pri veľkosti 8.



Obr. 13: Vývoj fitness funkcie s pôvodným nastavením, geometrické chladnutie s $\alpha = 0,99$ pri veľkosti 12.



Obr. 14: Vývoj fitness funkcie s najlepším nastavením, polynomiálne chladnutie s $\alpha = 0,8$ pri veľkosti 12.