

Protokol semestrální práce

Markéta Minářová

FIT ČVUT

BI-ZUM 2020/2021

Téma: Sudoku - klikař

Zadání: SUDOKU, ve které si vsadíme na štěstíčko tedy kliku. Systematické algoritmy nejsou vždy snadné na naprogramování, zato lokální prohledávání/optimalizaci zvládne každý. Platíme za to neúplností lokálních algoritmů, ale jejich jednoduchost to vyvažuje.

- Formulujte SUDOKU jako úlohu lokálního prohledávání, čili ve smyslu algoritmu stoupání do kopce.
- Vyzkoušejte různé varianty, jak lokálně měnit kandidáta na řešení a porovnejte je.
- Pole velikosti 9x9 je pro dnešní počítače hračka (samozřejmě, jak kdy), proto vyzkoušejte zobecněné SUDOKU na větší hrací ploše.

Řešení: Jako algoritmus lokálního prohledávání jsem implementovala klasický hill-climbing, steepest hill-climbing a vylepšené simulované žíhání, abych porovnala uvážnutí v lokálním minimu.

100% správné řešení dává nejčastěji simulované žíhání a velmi zřídka i zbylé 2 algoritmy.

Pro tyto algoritmy je potřeba implementovat fitness/cost funkci, funkci vygenerování počátečního řešení a náhodný výběr souseda.

Program nejprve přečte vstup ze souboru a do svého pole si uloží zadání, tedy fixní čísla SUDOKU, které se již měnit nebudou. Poté

program projede všechny menší čtverce dané SUDOKU (tedy v případě rozměrů 9x9 bude procházet čtverce 3x3, pozn. dále označováno jako *subsquares*) a vyplní je náhodně hodnotami od 1 do velikosti sudoku (9x9: číslicemi od 1 do 9).

Tedy v jednotlivých subsquares nebudou žádné kolize.

Fitness funkce počítá celkový počet kolizí v sudoku, tedy čím nižší fitness, tím lepší řešení. Ideálně bychom se chtěli dostat do stavu, kdy je fitness rovno 0 - žádná kolize, tudíž jsme našli kýžené řešení. Fitness funkce počítá kolize pouze v řádcích a sloupcích, protože v jednotlivých subsquares už máme garanci toho, že prvky se nebudou opakovat.

Generování náhodného souseda jsem implementovala tak, že vyberu náhodný subsquare z celé SUDOKU a v něm vyberu 2 čtverečky, které se prohodí, pokud:

- 1.) neobsahují fixní číslo ze zadání
- 2.) nemají totožné souřadnice

Při implementaci hill-climbing i steepest ascending jsem narazila na problémy s lokálním minimem téměř při každém spuštění, tedy ambice, že se to někdy podaří, výrazně klesly, nicméně párkrát se to podařilo.

Simulované žíhání na tom bylo lépe, nicméně se zde objevil problém s nastavením tzv. annealing schedule, tedy počáteční teploty a počet iterací před snížením této teploty. Po prozkoumání různých vědeckých prací, které jsou vyčteny na konci dokumentu, jsem zkusila některé z doporučených metod pro tyto parametry. Počet iterací jsem stanovila jako počet volných políček, které se mají vyplnit a počáteční teplota se počítá jako směrodatná odchylka několika málo fitness hodnot vygenerovaných sousedů.

Tyto přístupy se projeví jako nejlepší i když správné řešení také nedávají pokaždé, nicméně se k němu přibližují více, než mnou zkoušené jiné parametry.

Tento program je implementován i pro SUDOKU jiné velikosti a to konkrétně jakékoliv druhé mocniny. V ukázkových příkladech mám ukázkou SUDOKU 16x16, nicméně výpočet této SUDOKU trvá mnohem déle.

Shrnutí: Ve svém programu jsem experimentovala jak s kandidáty na souseda, tak zejména s teplotním nastavením v simulovaném žíhání. První srovnání je se sousedy - program dává znatelně lepší výsledky, pokud se prohazují prvky v subsquare, nikoliv v řádku či sloupci (příčemž by se pořáteční řešení generovalo v řádcích, resp. sloupcích).

Další poznatek je, že pokud se teplota počítá jako směrodatná odchylka a počet iterací jako počet volných políček, tak tato kombinace se osvědčila nejlépe.

Ačkoliv simulované žíhání optimalizuje řešení, do lokálního minima padá více, než jsem předpokládala.

Zdroje:

Walid Ben-Ameur, *Computing the Initial Temperature of Simulated Annealing*, Computational Optimization and Applications 29:369-385, 2004

Rhyd Lewis, *Metaheuristics can solve sudoku puzzles*, J Heuristics 13:387-401, 2007

M. A. Al-Betar, M. A. Awadallah, A. L. Bolaji, B. O. Alijla, *β -Hill Climbing algorithm for sudoku game*, Palestinian International Conference on Information and Communication Technology, 2017