

Simulované žíhání pro řešení sudoku

BI-ZUM

Semestrální projekt

Petr Motl

27. dubna 2024

Abstrakt

Tento protokol popisuje použití algoritmu simulovaného žíhání pro řešení problému sudoku. Cílem je vyzkoušet různé varianty, jak lokálně měnit kandidáta na řešení, a porovnat jejich účinnost.

1 Úvod

Sudoku je populární logická hra vyžadující umístění čísel do mřížky tak, aby se každé číslo vyskytlo právě jednou v každém řádku, sloupci a bloku. Simulované žíhání (SA) je heuristická optimalizační metoda, která je často využívána pro řešení složitých optimalizačních problémů, inspirovaná procesem žíhání kovů.

2 Kandidátní funkce

Pro generování kandidátních řešení byly implementovány následující funkce:

- **swapCellsInRow**: Prohození dvou náhodných buněk v náhodně vybraném řádku.
- **swapCellsInSubGrid**: Prohození dvou náhodných buněk v náhodně vybraném bloku.
- **swapCellsInSubGridRowCols**: Prohození dvou náhodných buněk v buď řádku nebo sloupci, vybraném v rámci náhodného bloku.

3 Experimentální nastavení

Vhodné nastavení bylo určeno s počáteční teplotou 0.5 a rychlostí ochlazování 0.99999, což umožnilo efektivní průzkum prostoru řešení. Provedeno bylo deset běhů pro každou kandidátní funkci, pro sudoku velikosti 9x9 a 16x16.

4 Vyhodnocení

K vyhodnocení výsledků byla použita funkce `calculateScore`, která hodnotí řešení na základě počtu unikátních čísel v řádcích a sloupcích. Za každé unikátní číslo v rámci řádku nebo sloupce je k aktuálnímu skóre přičteno -1. Optimální skóre pro 9x9 sudoku je tedy -162 a pro 16x16 sudoku -512, atd. Data z těchto běhů byla následně uložena a vykreslena do grafů v Jupyter notebooku, který je dostupný v příloze.

5 Výsledky

Experimenty ukázaly, že jako nejvhodnější se ukazuje funkce **swapCellsInSubGrid**, což bylo patrné z pozorování rychlejší konvergence a lepšího skóre v porovnání s ostatními testovanými funkcemi v jednotlivých grafech.

6 Závěr

Simulované žíhání se ukázalo být dobrým řešením pro sudoku menších velikostí. Výsledky naznačují, že další výzkum by se měl zaměřit na další optimalizace kandidátních funkcí, experimentování s jinými parametry ochlazování a testování na větších Sudoku mřížkách.

7 Reference

Reference

- [1] Přednášky z kurzu BI-ZUM,
<https://courses.fit.cvut.cz/BI-ZUM/lectures/index.html>