

SUDOKU - Hill climbing

Základy umělé inteligence

LS 2024/25

Jana Římalová

Zadání a cíl práce

Cílem této práce je navrhnout a implementovat heuristický algoritmus Hill Climbing, který se pokusí vyřešit zadané sudoku různé velikosti a obtížnosti. Program je schopen pracovat s více předdefinovanými vstupy a postupně hledat řešení pomocí náhodně generovaných sousedních stavů.

Popis problému

Pravidla sudoku jsou všem známa. V této práci mám předdefinovaná sudoku 3 velikostí - 4x4, 9x9 a 16x16. Program je ale teoreticky schopen pracovat s mřížkami o rozměrech jakékoliv druhé mocniny celého čísla.

Pro větší velikosti (9x9, 16x16) jsou připraveny 3 úrovně obtížnosti:

- Light – 2/3 čísel jsou předvyplněny správně
- Medium – polovina čísel je předvyplněna
- Hard – 1/3 čísel je předvyplněna

Použitý algoritmus

Zvolený algoritmus je **hill climbing se zpětným restartem**, aby nedošlo k zaseknutí v lokálním minimu. Program v každé iteraci:

1. Vygeneruje počáteční řešení doplněním prázdných míst náhodně (při dodržení unikátnosti hodnot v rámci řádku, sloupce i podčtverce).
2. Vyhodnotí fitness pro toto řešení (viz dále).
3. Náhodně vygeneruje sousední stav (viz dále).
4. Pokud má nový stav nižší fitness než předchozí, pokračuje z něj.

5. Pokud dojde k lokálnímu minimu nebo vyčerpá pokusy, provede restart s novým náhodným výchozím stavem.

Maximální počet pokusů i restartů je omezen.

Výpočet fitness

Fitness (v programu označeno jako *antiFitness*) je navrženo tak, aby penalizovalo porušení pravidel sudoku. Hodnotí se:

- **Rozdíl součtu** čísel v každém řádku, sloupci a podčtverci od ideálního součtu.
- **Duplicitní čísla** – každá duplicita je penalizována konstantou +1000 za každou opakující se hodnotu.

Generování sousedů

Původně jsem uvažovala o strategii **prohazování dvou náhodných řádků**, nicméně tato metoda nevedla ke zlepšení výsledků, protože došlo k výraznému narušení původní struktury sudoku. Z tohoto důvodu jsem zvolila jiný přístup:

- V každém sousedovi se prohodí dvě náhodné hodnoty v náhodném řádku
- S pravděpodobností 50 % se poté zamíchá jeden řádek

Prohazují se pouze ta čísla, která nebyla vyplněna na začátku, ke kontrole slouží dvourozměrné pole *movable*.

Výsledky a pozorování

- U velikosti 4×4 program vždy najde správné řešení během několika milisekund.
- U 9×9 je výpočet výrazně pomalejší – trvá několik minut a ne vždy se podaří najít správné řešení.
- U 16×16 nebylo řešení nikdy nalezeno v rozumném a přijatelném čase.

Shrnutí nejčastěji dosaženého fitness pro každou počáteční mřížku jsem zanesla do následující tabulky:

Rozměr	Obtížnost	Dosažené fitness
9x9	light	2000
9x9	medium	10 000
9x9	hard	20 000
16x16	light	60 000
16x16	medium	80 000
16x16	hard	100 000

Fitness 10 000 u sudoku 9x9 medium znamená, že přibližně 10 čísel je špatně vyplněných (za každou duplicitní hodnotu v řádku/sloupci/podčtverci přičítám k fitness 1000), což je z 81 čísel 82 % správně vyplněných. Analogickým způsobem následující tabulka uvádí, jak přesná řešení byl můj algoritmus schopen najít.

Rozměr	Obtížnost	Přesnost
9x9	light	97 %
9x9	medium	82 %
9x9	hard	75 %
16x16	light	76 %
16x16	medium	68 %
16x16	hard	60 %

Závěr

Navržený hill climbing algoritmus funguje spolehlivě a rychle pro malá sudoku a pro větší případy dokáže nalézt částečně správná řešení, která dosahují středně vyšší až vysoké přesnosti. K vylepšení by bylo vhodné optimalizovat výběr sousedů v momentě, kdy už máme skoro správné řešení, například detekovat, které buňky narušují strukturu sudoku a ty upravit, místo náhodného generování nových sousedů