

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. Po restartu se vygeneruje nové náhodné počáteční řešení a od něj se potom odvíjí. Program nejprve doplní zadanou mřížku tak, aby neobsahovala prázdné hodnoty a poté v každé iteraci

1. Náhodně vygeneruje 4 sousední stavy a vybere z nich ten s nejmenším fitness (viz dále).
2. Pokud má nový stav nižší fitness než dosavadní nejlepší, pokračuje z něj.
3. 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 novém 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 jemně pomalejší – pro light verzi vždy najde řešení během milisekund, ale pro obtížnější varianty nejen trvá déle, ale ne vždy skončí úspěšně, u medium je úspěšnost nalezení řešení asi 30 %.
- U 16×16 nebylo zatím nalezeno řešení v rozumném čase

Program jsem pro každou mřížku spustila třikrát.

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

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, kolik k tomu potřeboval pokusů a jak dlouho hledání trvalo (pokud se dlouhou dobu nejlepší řešení nezměnilo, běh algoritmu jsem ukončila).

Rozměr	Obtížnost	Fitness	Průměrně pokusů	Doba trvání	Přesnost
9x9	light	0	500	milisekundy	100 %
9x9	medium	4000	150 000	3 minuty	95 %
9x9	hard	10 000	400 000	4 minuty	87 %
16x16	light	25 000	170 000	4 minuty	90 %
16x16	medium	62 000	666 000	4 minuty	75 %
16x16	hard	75 000	350 000	6 minut	70 %

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