

Sokoban evolučným spôsobom

Projekt z umelej inteligencie

Matej Pavla

Zhodnotenie

- Podarilo sa mi implementovať evolučné riešenie na problém sokobana
- Riešenie nefunguje na všetkých možných mapkách, no zvláda niektoré jednoduchšie
- Existujúce riešenia spôsobom prehľadávania stavového priestoru, napr. IDA, dokážu riešiť aj takéto riešenia
- Moje riešenie je preto najvhodnejšie porovnávať s iným evolučným riešením problému

Porovnanie

- Takým riešením je napríklad diplomová práca od Tom Schaul.
- <http://www.whatisthought.com/schaulthesis.pdf>
- Iné evolučné riešenia sokobana som nenašiel.
- On ale volil úplne iný evolučný prístup:
 - Reprezentácia jedinca je na úrovni inštrukcií v pseudojazyku
 - Agenti vsádzajú na rozhodnutia o ďalšom pohybe v mapke
 - Každé kolo začína aukciou, agenti navzájom o svojich vsádzkach nevedia
 - Vyhráva agent, ktorý vsadil najviac
 - Okrem agentov sa vyskytuje v riešení entita Banka, ktorej agenti platia dane (napr. za každú vsádzku)
 - Agenti sa najprv trénujú na jednoduchých riešeniach (napr. jedna krabica v jednej izbe), následne sa snažia vyriešiť komplikovanejšie.
 - Spôsob evolúcie ostáva po istý bod (trénovanie), kedy sa evolúcia zastaví a výsledný najlepší agenti sa pokúsia prísť s riešením
 - Využíval existujúci framework, ktorý toto všetko implementuje, špecifické boli len niektoré inštrukcie v géne jedinca

Porovnanie

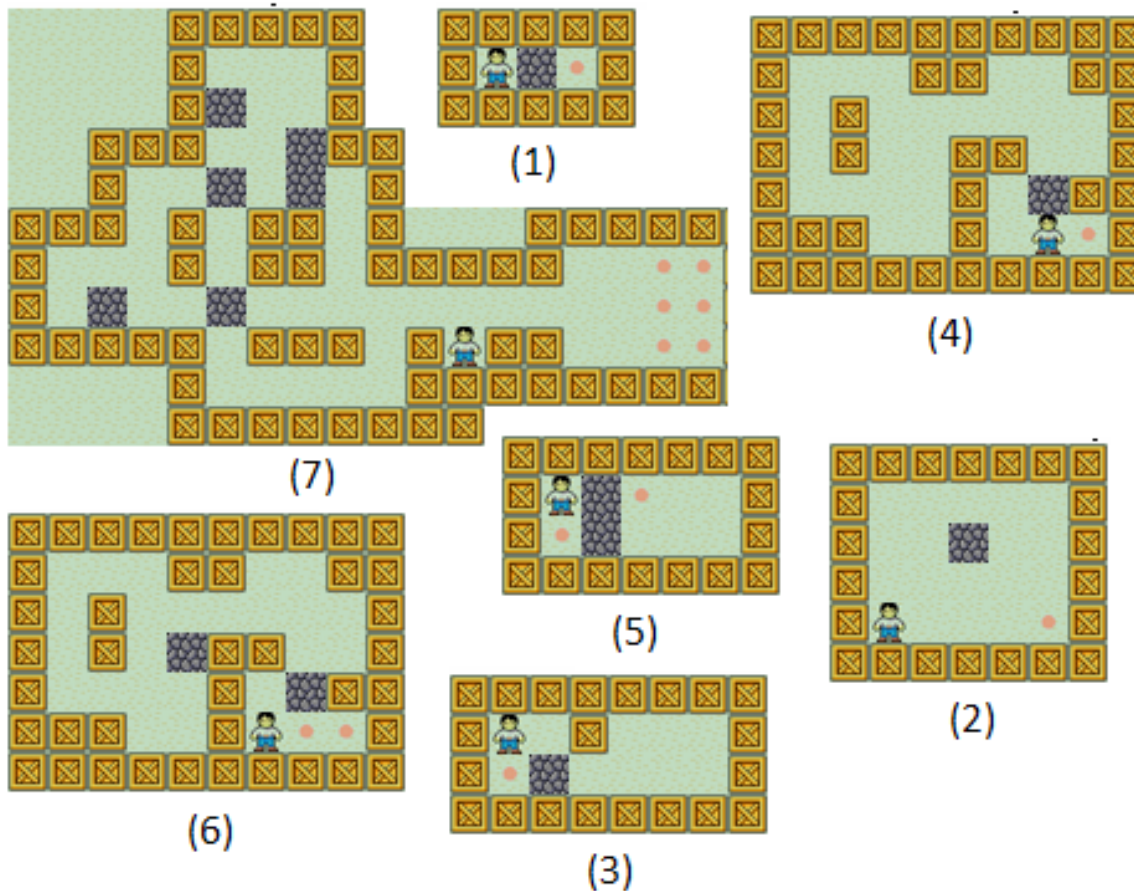
- Autor testoval svoje riešenie na 7 mapách
- Na 6 fungovalo, na siedmom našlo riešenie len niekedy náhodne, tzn. nedalo sa povedať že by sa agent „naučil“ vyriešiť problém na danej mapke.
- Preto autor už ani viac riešení neskúšal
- Porovnanie úspešnosti s mojím riešením:
 - Na 6 mapkách na ktorých sa porovnávanému riešeniu podarilo uspieť sa podarilo uspieť aj môjmu riešeniu
 - Siedmu mapku sa mi nepodarilo vyriešiť vôbec
 - Úspešnosť porovnávaného riešenia máme len z tabuliek a meraní autora, binárka či samotný zdrojový kód riešenia sa mi nepodarilo nájsť, autor ho neuvádza

Porovnanie a vyhodnotenie, kam sa uberať

- Moje aj porovnávané riešenia vyriešili rovnaký počet mapiek
- Zo svojho pozorovania vidím, že problém, ktorý môjmu riešeniu zabraňoval v úspešnom vyriešení, bol netriviálne detekovateľný deadlock
- O podobných problémoch hovorí aj autor v porovnávanom projekte
- Navrhol niekoľko funkcií, ktoré ale nestihol implementovať, ktoré by mali lepšie usmerniť evolúciu
- Napríklad navrhol účinnejší detektor deadlock situácií, čo považujem aj ja za cestu k lepším výsledkom v mojom riešení
- Najväčší problém nedostatok času, pretože existuje veľké množstvo rôznych deadlock situácií
 - Navrhoval by som detekciu deadlockou ako samostatnú tému, nejaký modul, ktorý by sa tomu venoval, tiež napríklad evolučným spôsobom, ale celkovým prehľadávaním priestoru

Testovanie

- Testované mapky sú v textovej podobe v mojom projekte v súbore: `source/SokobanEvo/assets/schaul.txt`
- Grafická reprezentácia mapiek:



Tabuľkové porovnanie

Id mapky	Úspešnosť ja	Úspešnosť porovnávaný	Čas ja	Čas porovnávaný
1	vyriešené	vyriešené	< 1s	neuvádza
2	vyriešené	vyriešené	< 1s	neuvádza
3	vyriešené	vyriešené	< 1s	neuvádza
4	vyriešené	vyriešené	< 1s	neuvádza
5	vyriešené	vyriešené	< 1s	neuvádza
6	vyriešené	vyriešené	~ 2 – 5 s	neuvádza
7	nevyliešené	vyriešené nenaučené	--	neuvádza