

Peter Kúdela

UI

Zadanie 3 (Genetický algoritmus)

Fiit

22.11.2020

1. Algoritmus

Mapa:

Mapa je vložená priamo v kóde ako 2D array no program si všetky potrebné informácie počíta sám takže je možné ju modifikovať. Na začiatku programu si vypočítame začiatočnú pozíciu označenú písmenom s, taktiež spočítame písmená „g“ reprezentujúce poklady.

Class Individual:

Táto trieda reprezentuje našich jedincov, udržiava si hodnotu fitness ktorá je automaticky nastavená na 100, a 64 pamäťových buniek.

Individual.randomize():

Funkcia naplní všetky bunky jedinca náhodnou hodnotou.

Virtuálny stroj:

Do stroja vkladáme vždy kópiu jedinca, stroj si drží adresu na ktorej sa práve nachádza, každým behom slučky ju zvyšuje ak sme nenašli bunku skok.

Príkazy buniek získavame bitwise operáciou and s maskou 192 a nasledovným porovnávaním. Adresy buniek nad ktorými sa dané operácie vykonávajú sa získavajú rovnakým spôsobom no s inou maskou.

Funkcia vracia hodnotu fitness jedinca.

Selekcia:

V programe sú implementované dva typy selekcie a to tournament a roulette.

Tournament():

Funkcia vyberie náhodných jedincov z generácie, porovná ich a vráti jedinca s lepšou hodnotou fitness.

Roulete():

Funkcia náhodne vyberie jedinca z generácie, jedinci majú väčšiu šancu byť vybraný ak majú vyššiu hodnotu fitness

Kríženie:

V programe sú implementované dva typy kríženia a to halfnhalf() a mixChild().

Halfnhalf():

Vygenerujeme náhodné číslo od 0 po 63, bunky po index vygenerovaného čísla berieme z prvého rodiča, ostatné z druhého rodiča.

MixChild():

Každá bunka má 50% šancu na to byť braná od rodiča 1, inak od rodiča 2.

Mutácia:

Každý novovytvorený jedinec prejde touto funkciou, funkcia si najprv generuje číslo od 0 po 1, pokiaľ je náhodné číslo < 0.1 , jedinec je nezmenený, pokiaľ je hodnota vyššia ako 0.9, celý jedinec je zmenený, inak má každá bunka $1/40$ šancu na zmenu.

Elitizmus:

V programe je tiež implementovaná forma elitizmu, program si z každej generácie uchováva práve jedného jedinca ktorý má najvyšší fitness.

2. Výsledky

Všetky testy sú vykonané pri generáciách tvorených 250 individuálmi

Tournament with mixedChild():Run1:

Prvých 140 generácií malo problém dostať priemerný fitness vyššie ako 120. Priemerný fitness sa ďalej postupne zvyšoval až po 189 na generácií 183, v generácií 184 bola nájdená cesta s všetkými pokladmi.

Run2:

Cesta bola nájdená v generácií 86, priemerný fitness sa na rozdiel od prvého behu zvyšoval a nestagnoval na žiadnej hodnote, dosiahol výšku 203 pred nájdením výsledku.

Run3:

Prvých 200 generácií stagnovalo s priemerným fitness 100-110, až po generácií 200 sa fitness začal zlepšovať dosahujúc hodnotu 277 v generácií 323 a nájdením cesty v nasledujúcej generácií.

Tournament with halfnhalf():Run1:

Priemerný fitness sa takmer každou generáciou zvýšil, výsledok sa našiel už v generácií 15. Priemerný fitness v tom čase bol 146.

Run2:

Cesta bola nájdená v generácií 93, fitness sa v generáciách 1-30 zvyšoval, po generácií 30 až po nájdenie výsledku sa zdržiaval v hodnotách 250-270.

Run3:

Podobne ako v behu 2, program začal stagnováť s hodnotami 250-270 od generácie 30, toto trvalo až do generácie 185 kde sa fitness znova začal zvyšovať až po generáciu 212 kde opäť stagnuje tentoraz na hodnote 320, od generácie 250 sa znovu začne zvyšovať až po nájdenie výsledku v generácií 280 s priemernou fitness hodnotou 343.

Roulette with mixedChild():

Run1:

Hodnota fitness sa rovnomerne zvyšovala bez stagnácie až do nájdania výsledku v generácií 74 (priemerná fitness 185)

Run2:

Beh veľmi podobný prvému, výsledok v gen 66 (priemerná fitness 193)

Run3:

Výsledok nájdený v generácií 26 (priemerná fitness 151)

Roulette with halfnhalf():

Run1:

Cieľ bol nájdený v generácií 20, priemerný fitness dosiahol 178 bez stagnovania.

Run2:

Cieľ bol nájdený v generácií 35, priemerný fitness dosiahol 289 bez stagnovania.

Run3:

Cieľ nájdený v generácií 199, priemerný fitness sa zvyšoval po generáciu 37, od tohto bodu až po nájdenie výsledku sa zdržiaval pri hodnotách fitness 250-260

2.1 Zhodnotenie

Najlepšie výsledky sme dosiahli použitím kombinácie rulety a mixedChild(), ruleta celkovo v našich výsledkoch dosahovala cieľ pri menšom počte generácií, dôvodom je pravdepodobne pomerne vysoký počet individuálov v generácií, ruleta vyberá jedincov na základe fitness, kde turnaj len porovnáva dvoch náhodných a berie lepšieho.

3. Možnosti zlepšenia

Program má tendenciu stagnovať pri určitých hodnotách fitness, elitizmus jedného jedinca tomuto problému pomohol, preto je možné že implementácia elitizmu väčšieho množstva jedincov z generácií by mohlo viesť ku lepším výsledkom, taktiež by bolo vhodné doladiť pravdepodobnosti mutovania jedincov.