

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4

Zadanie č. 3b

Róbert Jačko

Študijný program: Informatika
Ročník: 2.
Predmet: Umelá inteligencia
Ak. rok: 2019/2020

Úloha

Majme hľadača pokladov, ktorý sa pohybuje vo svete definovanom dvojrozmernou mriežkou (viď. obrázok) a zbiera poklady, ktoré nájde po ceste. Začína na políčku označenom písmenom **S** a môže sa pohybovať štyrmi rôznymi smermi: hore **H**, dole **D**, doprava **P** a doľava **L**. K dispozícii má konečný počet krokov. Jeho úlohou je nazbierať čo najviac pokladov. Za nájdenie pokladu sa považuje len pozícia, pri ktorej je hľadač aj poklad na tom istom políčku. Susedné políčka sa neberú do úvahy.



Zadanie

Horeuvedenú úlohu riešte prostredníctvom evolučného programovania nad virtuálnym strojom.

Tento špecifický spôsob evolučného programovania využíva spoločnú pamäť pre údaje a inštrukcie. Pamäť je na začiatku vynulovaná a naplnená od prvej bunky inštrukciami. Za programom alebo od určeného miesta sú uložené inicializačné údaje (ak sú nejaké potrebné). Po inicializácii sa začne vykonávať program od prvej pamäťovej bunky. (Prvou je samozrejme bunka s adresou 000000.) Inštrukcie modifikujú pamäťové bunky, môžu realizovať vetvenie, programové skoky, čítať nejaké údaje zo vstupu a prípadne aj zapisovať na výstup. Program sa končí inštrukciou na zastavenie, po stanovenom počte krokov, pri chybnnej inštrukcii, po úplnom alebo nesprávnom výstupe. Kvalita programu sa ohodnotí na základe vyprodukovaného výstupu alebo, keď program nezapisuje na výstup, podľa výsledného stavu určených pamäťových buniek.

Objekty

World

Atribúty:

grid:	- mriežka, v ktorej sa pohybuje panáčik
y:	- aktuálna pozícia panáčka
x:	- aktuálna pozícia panáčka
num_of_gems:	- počet pokladov v mriežke

Subject

Atribúty:

gems_found:	- počet nájdených pokladov
data_cell:	- list buniek, resp. čísel z rozsahu <0, 255>
moves:	- string pohybov jedinca (písmená H, D, L, P)

Metódy:

get_fitness():	- na základe počtu pohybov a počtu nájdených pokladov vráti číslo reprezentujúce fitness jedinca
----------------	--

Opis algoritmu

1. Inicializácia počiatočných jedincov

- Vytvorí sa zadaný počet jedincov. Každému jedincovi je priradených 64 buniek s náhodnými číslami v rozsahu <0, 255>

2. Určenie fitness jedincov

- Každý jedinec v aktuálnej generácii sa vloží do virtuálneho stroja, kde sa skopírujú jeho bunky a stroj mu vyhodnotí jeho fitness

3. Ak nejaký jedinec našiel všetky poklady, tak skonči s úspechom

- Vyhodnotí sa podmienka počet nájdených pokladov jedincom == celkový počet pokladov

4. Výber rodičov

- Jedinci sa usporiadajú vzostupne podľa ich fitness a na základe zadaného vstupu sa zvolí spôsob výberu rodičov. Spôsoby sú ruleta a náhodný výber z lepšej polovice

4. Vytvorenie nových jedincov

- Každý dvaja rodičia budú mať dve deti
- Náhodne sa zvolí bod kríženia z rozsahu <0, 64>
- Prvé dieťa zdedí dátové bunky <0, bod kríženia> od prvého rodiča a <bod kríženia, 64> od druhého rodiča. Druhé dieťa zdedí bunky opačne (najprv od druhého, potom od prvého rodiča)
- Počet nových jedincov závisí od zadaného percenta elitárstva

5. Mutácia nových jedincov

- S pravdepodobnosťou, ktorú na vstupe zadá používateľ, nastane mutácia jedinca. Mutácia prebieha tak, že pre každú dátovú bunku jedinca je šanca 5%, že sa zmení na náhodnú hodnotu

6. Doplnenie nových jedincov najlepšími jedincami z predošlej generácie

- Do novej generácie sa doplnia najlepší jedinci z predošlej generácie tak aby sa rovnali počty jedincov pôvodnej a novej generácie

7. Prejdenie na krok 2

Porovnanie

Ruleta

Generácie:	50
Mutácia:	0%
Jedinci:	100
Elitárstvo:	0%
<hr/>	
Priemer:	58.33%

Generácie:	50
Mutácia:	6%
Jedinci:	100
Elitárstvo:	0%
<hr/>	
Priemer:	54.51%

Generácie:	50
Mutácia:	0%
Jedinci:	100
Elitárstvo:	5%
<hr/>	
Priemer:	73.28%

Generácie:	50
Mutácia:	0%
Jedinci:	100
Elitárstvo:	15%
<hr/>	
Priemer:	73.42%

Generácie:	50
Mutácia:	6%
Jedinci:	100
Elitárstvo:	15%
<hr/>	
Priemer:	75.01%

Generácie:	50
Mutácia:	3%
Jedinci:	100
Elitárstvo:	0%
<hr/>	
Priemer:	57.37%

Generácie:	50
Mutácia:	9%
Jedinci:	100
Elitárstvo:	0%
<hr/>	
Priemer:	54.99%

Generácie:	50
Mutácia:	0%
Jedinci:	100
Elitárstvo:	10%
<hr/>	
Priemer:	74.42%

Generácie:	50
Mutácia:	3%
Jedinci:	100
Elitárstvo:	10%
<hr/>	
Priemer:	74.62%

Generácie:	50
Mutácia:	3%
Jedinci:	100
Elitárstvo:	20%
<hr/>	
Priemer:	74.86%

Náhodný

Generácie: 50
Mutácia: 0%
Jedinci: 100
Elitárstvo: 0%

Priemer: 54.60%

Generácie: 50
Mutácia: 6%
Jedinci: 100
Elitárstvo: 0%

Priemer: 57.52%

Generácie: 50
Mutácia: 0%
Jedinci: 100
Elitárstvo: 5%

Priemer: 69.72%

Generácie: 50
Mutácia: 0%
Jedinci: 100
Elitárstvo: 15%

Priemer: 70.26%

Generácie: 50
Mutácia: 6%
Jedinci: 100
Elitárstvo: 15%

Priemer: 75.22%

Generácie: 50
Mutácia: 3%
Jedinci: 100
Elitárstvo: 0%

Priemer: 58.21%

Generácie: 50
Mutácia: 9%
Jedinci: 100
Elitárstvo: 0%

Priemer: 57.21%

Generácie: 50
Mutácia: 0%
Jedinci: 100
Elitárstvo: 10%

Priemer: 71.33%

Generácie: 50
Mutácia: 3%
Jedinci: 100
Elitárstvo: 10%

Priemer: 75.45%

Generácie: 50
Mutácia: 3%
Jedinci: 100
Elitárstvo: 20%

Priemer: 74.76%

Zhodnotenie

Test možností výberu rodiča vyšiel zhruba rovnako pre obe možnosti a nijak sa výrazne nelíšili.

Najvyššie priemerné fitness jedincov 50. generácie sa dosahovali pri 15% elitárstve a 3% až 6% šance mutácie.

Graf zobrazuje priemerné fitness jedincov v generácii.

Generácie: 50

Mutácia: 3%

Jedinci: 100

Elitárstvo: 15%

