

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Fakulta informatiky a informačných technológií

ZADANIE 3

Úloha b) – evolučné programovanie

Dávid Kromka
Cvičenie: utorok 16:00
22.11.2021

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Slíž, utorok 16:00

Obsah

Obsah.....	2
1. Riešený problém	3
2. Opis riešenia.....	3
3. Použitý algoritmus	5
4. Testovanie	5
5. Zhodnotenie	11

1. Riešený problém

Zadaním úlohy je na definovanej dvojrozmiernej mriežke nájsť všetky rozmiestnené poklady prostredníctvom hľadača pokladov, ktorý sa môže hýbať štyrmi rôznymi smermi: hore **H**, dole **D**, doprava **P** a doľava **L**. Za nájdenie pokladu sa považuje pozícia, kedy hľadač má rovnakú polohu na mriežke ako poklad.

Tento problém je potrebné riešiť evolučným programovaním nad virtuálnym strojom, čo je špeciálny spôsob evolučného algoritmu využívajúci spoločnú pamäť pre údaje a inštrukcie.

2. Opis riešenia

Základným prvkom riešenia je reprezentácia virtuálneho stroja. Náš virtuálny stroj pozostáva zo 64 pamäťových buniek, ktorých veľkosť je 8 bitov (integer od 0 do 255). Prvé 2 bity v bunke predstavujú inštrukciu.

inštrukcia	tvar
inkrementácia	00XXXXXX
dekrementácia	01XXXXXX
skok	10XXXXXX
výpis	11XXXXXX

Nasledujúcich 6 bitov predstavuje adresu, s ktorou inštrukcia pracuje. Pri inštrukcii výpis sa pracuje s poslednými 2 bitmi, kde ,00' znamená pohyb hore, ,01' pohyb dole, ,10' pohyb vpravo a ,11' pohyb vľavo hľadača pokladov v mriežke.

Vizualizácia virtuálneho stroja:

Adresa:	000000	000001	000010	000011	000100	000101	000110	...
Hodnota:	00001000	10011001	00011010	01010110	10000101	11001110	10110000	...

Reprezentácia v kóde:

```
Population = []
for q in range(self.n):
    population.append([format(random.randint(0, 255), '08b') for _
in range(40)])
    for i in range(24):
        population[q].append(format(0, '08b'))
```

V funkcii main() je v cykle vytváraná populácia virtuálnych strojov v počte podľa zadanej hodnoty. V každom virtuálnom stroji je inicializovaných prvých 40 buniek na náhodnú hodnotu, zvyšných 20 buniek má hodnotu 0.

Následne je nad každým virtuálnym strojom vytvorená inštancia triedy Chromosome, ktorej premennými sú virtuálny stroj a fitness, na začiatku nastavená na hodnotu 0.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Slíž, utorok 16:00

class Chromosome:

```
def __init__(self, chromosome):  
    self.chromosome = chromosome  
    self.treasures_found = 0
```

Inštancia trieda Area predstavuje dvojrozmernú mriežku so zadanými rozmermi, polohou a množstvom pokladov a polohou hľadača pokladu.

V cykle sa nad každym prvkom populácie virtuálnych strojov vykoná funkcia

`def machine_work(self, surface, draw, chromosome)` v class `VirtualMachine`, ktorá vykoná inštrukcie zapísané vo virtuálnom stroji. Ukončí sa, ak je počet iterácii 500, sú nájdené všetky poklady alebo hľadač vybočil z mriežky. Táto funkcia vracia fitness hodnotu každého virtuálneho stroja. Fitness hodnota je počet nájdených pokladov hľadača v mriežke. Ak je argument `draw` vo funkcii nastavený na 1, funkcia vypíše postupnosť krokov hľadača.

Nasleduje výber rodičov na kríženie, v programe sú použité metódy ruleta a duel.

Pri rulete sa do premennej `pop_sum` uloží súčet všetkých fitness ohodnotení danej populácie. Premenná `r` predstavuje náhodnú hodnotu od 0 po `pop_sum`. V cykle sa prechádza cez populáciu a do premennej `s` sa postupne pridávajú hodnoty jednotlivých fitness ohodnotení. Keď bude premenná `s > r`, cyklus sa zastaví a daný virtuálny stroj bude jedným z rodičov. Takto sa v cykle za sebou vyberú rodičia na kríženie, dvojica nemôže byť identická a počet vybraných jedincov je totožný s počtom jedincov v počiatočnej populácii.

Pri metóde `duel` sú z populácie náhodne zvolené 2 virtuálne stroje, pri ktorých sa porovná ich fitness ohodnotenie. Ten, ktorého ohodnotenie je vyššie, je vybraný za rodiča. Takto sa postupne vytvoria dvojice rodičov, v dvojici nemôžu byť jedinci totožní. Ich počet je zhodný s počiatočnou populáciou.

Následne nastáva kríženie vo funkcii `def cross(self, chromosomes)`, kde sa zvolí index rozdelenie v intervale od 20 do 45. Z dvoch rodičov vzniknú 2 nové deti, ktoré podľa indexu rozdelenia budú inverzné. Prvý bude mať bunky od 0 do hodnoty indexu zhodné s prvým rodičom a zvyšné zhodné s druhým rodičom. Druhé dieťa bude presne naopak. Po krížení nastáva mutácia 0 až 5 krát na jednom dieťati, kde je náhodná bunka zmenená na náhodnú hodnotu v intervale 0 až 255.

Takto vznikla nová populácia, na ktorej sa celý cyklus opakuje až kým sa nenájde virtuálny stroj s takými inštrukciami, ktoré nájde všetky poklady alebo počet iterácii cyklu dosiahne stanovený počet.

Vzhľad a ovládanie programu – program sa spúšťa v termináli a ovláda sa zadávaním hodnôt podľa požiadavky.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Slíž, utorok 16:00

```
Zadajte x,y veľkosť mriežky, "k" pre skončí: 4,4
Počet jedincov: 25
Zadajte x,y súradnicu hľadača: 2,2
Zadajte x,y súradnicu pokladu, "k" pre skončí: 1,2
Zadajte x,y súradnicu pokladu, "k" pre skončí: 2,2
Zadajte x,y súradnicu pokladu, "k" pre skončí: 3,1
Zadajte x,y súradnicu pokladu, "k" pre skončí: k
0: ruleta, 1: duel: 0
Mriežka na začiatku:
[0, 0, 0, 0]
[0, 0, 0, 1]
[0, 1, 1, 0]
[0, 0, 0, 0]
Nájdené poklady, virtuálny stroj:
3 ['10011010', '10111010', '10010000', '10101100', '01011100', '10010000', '10001010', '10010000', '01011000', '10011011', '01010011',
Mriežka po hľadaní:
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
[0, 0, 0, 0]
Postupnosť krokov:
['h', 'd', 'l', 'p', 'l', 'p', 'p', 'h', 'h']
Zadajte x,y veľkosť mriežky, "k" pre skončí: |
```

3. Použitý algoritmus

1. Zo vstupu sa načíta rozmer mriežky, rozloženie a počet pokladov, počiatočná poloha hľadača, počet jedincov populácie, spôsob selekcie.
2. Počiatočná populácia jedincov sa nainicializuje náhodnými hodnotami v prvých 40 bunkách a vo zvyšných 24 bunkách bude hodnota 0.
3. Každému jedincovi sa určí fitness ohodnotenie, ktoré predstavuje počet nájdených pokladov do zastavenia jeho programu.
4. Ak je nájdený jedinec, ktorý našiel všetky poklady, vypíše sa postupnosť krokov hľadača a program končí.
5. Metódou zvolenej selekcie ruleta alebo duel sa určia dvojice rodičov na kríženie.
6. Nastane kríženie a nové jedince s istou pravdepodobnosťou mutujú.
7. Vznikla nová populácia, pokračuje sa znova na bod 3.

4. Testovanie

Testovanie prebieha na rôznych veľkostiach mriežky, s rôznymi počtami a rozloženiami pokladov a metódou selekcie ruleta alebo duel. Pri každom teste je uvedený počet vytvorených generácií, kým sa našiel jedinec, ktorého inštrukciami boli nájdené všetky poklady.

Pre grafické znázornenie zvyšovania a znižovania súčtu fitness ohodnotení populácie je pri každom behu programu vygenerovaný ich graf pomocou knižnice matplotlib.pyplot.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Slíž, utorok 16:00

Test 1

100 opakovaní programu s nastavením:

Veľkosť mriežky:	7x7
Počiatočná poloha hľadača:	[3, 6]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 4], [4, 5]]
Počet jedincov populácia:	35
Metóda selekcie:	ruleta

Výsledky:

Celkový počet generácii: 58096
Priemerný počet generácii: 580.96
Minimálny počet generácii: 2
Maximálny počet generácii: 5031

Test 2

100 opakovaní programu s nastavením:

Veľkosť mriežky:	7x7
Počiatočná poloha hľadača:	[3, 6]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 4], [4, 5]]
Počet jedincov populácia:	35
Metóda selekcie:	duel

Výsledky:

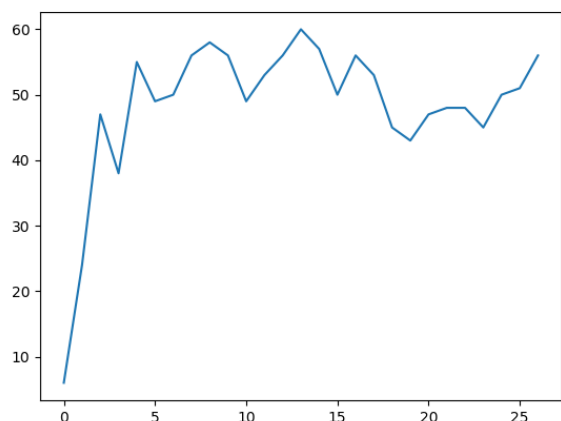
Celkový počet generácii: 74230
Priemerný počet generácii: 742,3
Minimálny počet generácii: 3
Maximálny počet generácii: 8759

Test 1 a Test 2 majú rovnaké počiatočné nastavenie, rozdielom je metóda selekcie. Pri viacerých behoch týchto testov boli výsledky veľmi podobné a to napriek tomu, že na začiatku sú virtuálne stroje inicializované na náhodné hodnoty. Z testov je taktiež jasné, že pri metóde selekcie duel sú výsledky efektivity horšie než pri metóde ruleta, vo všetkých prípadoch však bol nájdený jedinec s inštrukciami, podľa ktorých hľadač našiel všetky poklady.

Test 3

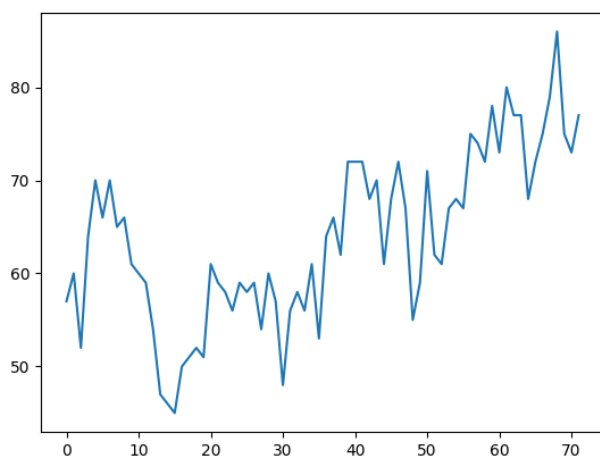
Veľkosť mriežky:	7x7
Počiatočná poloha hľadača:	[3, 6]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 4], [4, 5]]
Počet jedincov populácia:	35
Metóda selekcie:	ruleta

Meno: Dávid Kromka
 Ič: 110834
 Cvičenie: Slíž, utorok 16:00



Test 4

Veľkosť mriežky:	7x7
Počiatková poloha hľadača:	[3, 6]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 4], [4, 5]]
Počet jedincov populácia:	35
Metóda selekcie:	duel

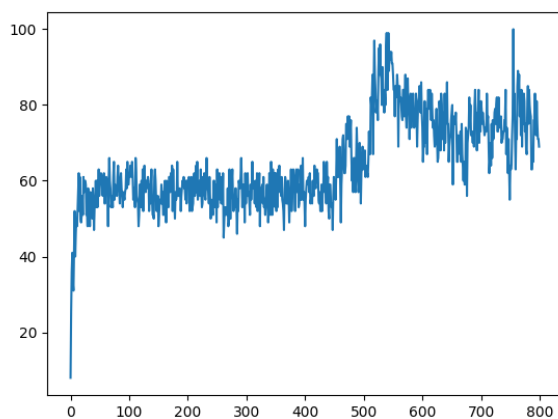


Test 3 a Test 4 má rovnakú počiatočnú populáciu, rozdiel je iba v metóde selekcie. Obidva testy našli všetky poklady.

Meno: Dávid Kromka
Ič: 110834
Cvičenie: Slíž, utorok 16:00

Test 5

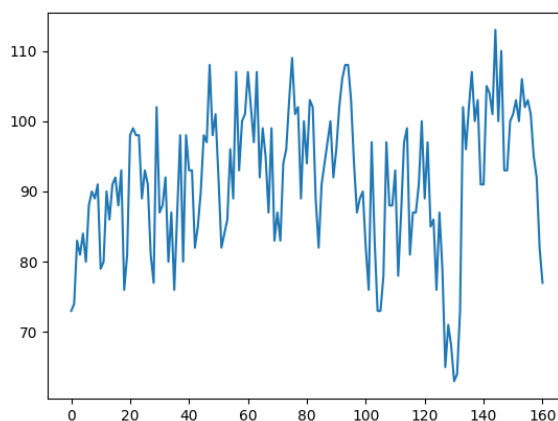
Veľkosť mriežky:	7x7
Počiatočná poloha hľadača:	[3, 6]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 4], [4, 5]]
Počet jedincov populácia:	35
Metóda selekcie:	ruleta



Postupnosť krokov: ['p', 'h', 'l', 'l', 'h', 'l', 'p', 'p', 'p', 'h', 'l', 'l', 'h', 'p', 'd', 'p', 'p', 'p', 'h', 'l', 'l', 'h', 'd']

Test 6

Veľkosť mriežky:	7x7
Počiatočná poloha hľadača:	[3, 6]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 4], [4, 5]]
Počet jedincov populácia:	35
Metóda selekcie:	duel



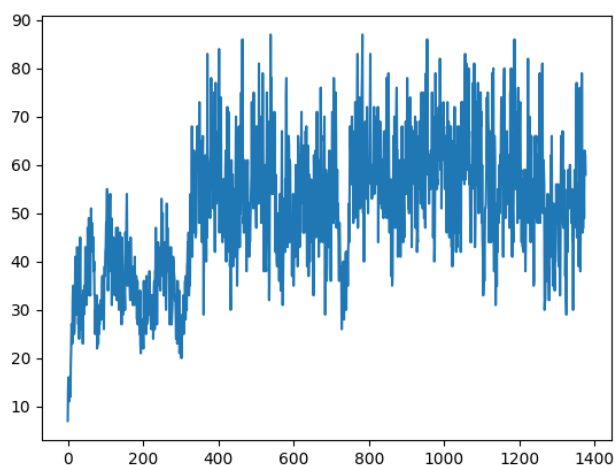
Meno: Dávid Kromka
 Ič: 110834
 Cvičenie: Slíž, utorok 16:00

Postupnosť krokov ['p', 'h', 'l', 'l', 'h', 'l', 'p', 'h', 'l', 'h', 'l', 'p', 'h', 'd', 'p', 'h', 'p', 'p', 'l', 'd', 'p', 'h', 'p', 'd', 'l', 'd', 'p', 'h', 'p', 'd', 'l']

V testoch 5 a 6 je možné vidieť, že metóda selekcie duel nie je vždy horšia ako ruleta. Pri oboch testoch je rovnaká počiatočná populácia. Počet vytvorených generácií pri metóde selekcie duel je výrazne nižší než počet pri metóde ruleta.

Test 7

Veľkosť mriežky:	10x10
Počiatočná poloha hľadača:	[5, 5]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 3], [2, 8], [3, 3]]
Počet jedincov populácia:	35
Metóda selekcie:	ruleta

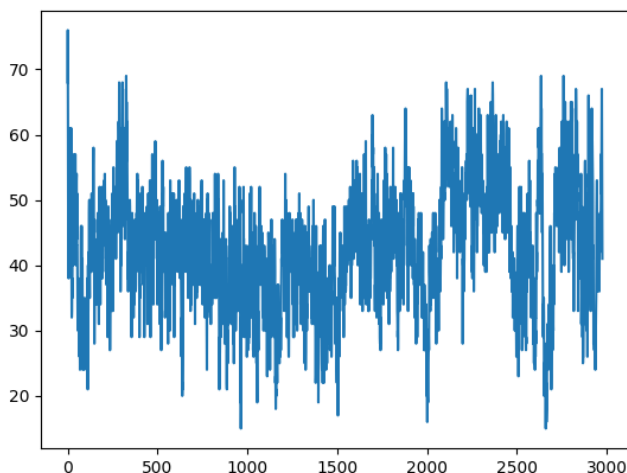


Postupnosť krokov ['h', 'h', 'p', 'd', 'p', 'h', 'h', 'l', 'p', 'd', 'l', 'h', 'l', 'l', 'p', 'd', 'h', 'h', 'l', 'l', 'p', 'd', 'd', 'h', 'l', 'l', 'p', 'd', 'd', 'h', 'l', 'l', 'p', 'd', 'p', 'h', 'l', 'l', 'p', 'd', 'p', 'd', 'l', 'l', 'p', 'd', 'l', 'p', 'p', 'l', 'l', 'p', 'd', 'p', 'p']

Test 8

Veľkosť mriežky:	10x10
Počiatočná poloha hľadača:	[5, 5]
Rozloženie pokladov:	[[4, 1], [2, 2], [6, 3], [1, 3], [2, 8], [3, 3]]
Počet jedincov populácia:	35
Metóda selekcie:	duel

Meno: Dávid Kromka
 Ič: 110834
 Cvičenie: Slíž, utorok 16:00



Postupnosť krokov ['h', 'h', 'p', 'h', 'h', 'l', 'l', 'd', 'd', 'd', 'l', 'd', 'p', 'l', 'h', 'l', 'p', 'h', 'h', 'p', 'l', 'd', 'd', 'd', 'l', 'd', 'p', 'l', 'h', 'l', 'p', 'h', 'h', 'd', 'l', 'd', 'd', 'd', 'p', 'd', 'p', 'l', 'h', 'l', 'p', 'h', 'h', 'h', 'l', 'd', 'd', 'h', 'p', 'd', 'p', 'l', 'h', 'l', 'p', 'h', 'h', 'l', 'l', 'd', 'd', 'h', 'p', 'd', 'p', 'l', 'h', 'l', 'p', 'h', 'h', 'p', 'l']

Test 7 a Test 8 je s rovnakou počiatočnou populáciou a rovnakým počiatočným nastavením. Riešenie s metódou ruleta je lepšie v počte generácii aj v počte krokov.

Test 9

Veľkosť mriežky:	4x4
Počiatočná poloha hľadača:	[2, 2]
Rozloženie pokladov:	[[1, 1], [3, 2]]
Počet jedincov populácia:	20
Metóda selekcie:	ruleta

Postupnosť krokov ['p', 'd', 'l', 'h', 'h', 'l', 'd']

Test 10

Veľkosť mriežky:	4x4
Počiatočná poloha hľadača:	[2, 2]
Rozloženie pokladov:	[[1, 1], [3, 2]]
Počet jedincov populácia:	20
Metóda selekcie:	duel

Postupnosť krokov ['p', 'd', 'l', 'h', 'h', 'l', 'd']

V Teste 9 a Teste 10 je počiatočná populácia rovnaká. Mriežka je malá a sú v nej iba 2 poklady. Obidve metódy, ruleta aj duel našli všetky poklady hneď v prvej generácii.

5. Zhodnotenie

Pre čo najviac optimálne výsledky programu boli testované viaceré prístupy ku kríženiu, mutácii počtu jedincov či inicializácii jedincov a podľa toho implementované v programe. Z testov je možné vyvodiť záver, že program funguje správne a vo väčšine prípadov, najmä pri menších rozmeroch mriežky je schopný nájsť všetky poklady. Metóda selekcie ruleta má v priemere lepšie výsledky v počte vytvorených generácií ako metóda duel. Na grafoch je možné vidieť, že krivka vo väčšine prípadov s počtom vytvorených generácií rastie, no sú vyberané aj menej dobré jedince.

Program je ľahko rozširiteľný o ďalšie metódy selekcie, ktoré význačne ovplyvňujú počet nájdených generácií. Riešenie je realizované v programovacom jazyku Python s využitím modelov random, matplotlib a numpy.