

Slovenská technická univerzita

Fakulta informatiky a informačných technológií
Ilkovičova 3, 812 19 Bratislava

Umelá inteligencia

Zadanie č. 3

Zenová záhrada

Adam Žúrek

Cvičiaci: Mgr. Peter Laurinec
Študijný odbor: Informatika
Ročník: 2. Bc
Akademický rok: 2016/2017

Zadanie

Zenová záhradka je plocha vysypaná hrubším pieskom (drobnými kamienkami). Obsahuje však aj nepohyblivé väčšie objekty, ako napríklad kamene, sochy, konštrukcie, samorasty. Mních má upraviť piesok v záhradke pomocou hrablí tak, že vzniknú pásy ako na nasledujúcom obrázku.

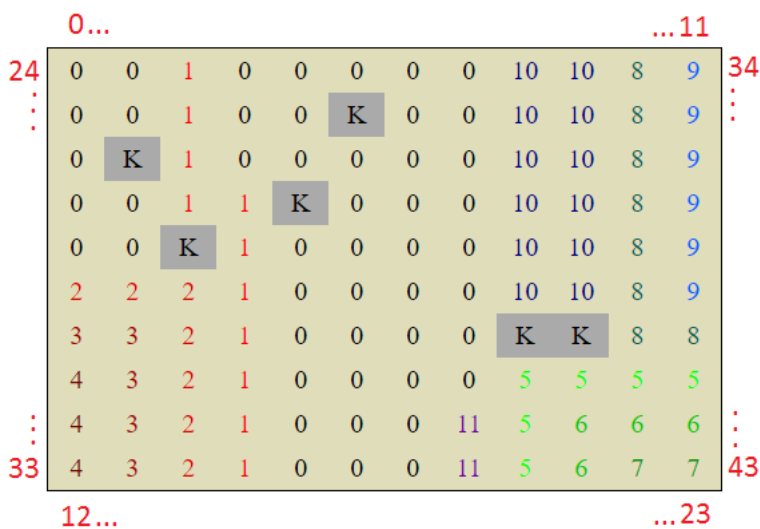
Pásy môžu ísť len vodorovne alebo zvislo, nikdy nie šikmo. Začína vždy na okraji záhradky a ťahá rovný pás až po druhý okraj alebo po prekážku. Na okraji - mimo záhradky môže chodiť ako chce. Ak však príde k prekážke - kameňu alebo už pohrabanému piesku - musí sa otočiť, ak má kam. Ak má voľné smery vľavo aj vpravo, je jeho vec, kam sa otočí. Ak má voľný len jeden smer, otočí sa tam. Ak sa nemá kam otočiť, je koniec hry. Úspešná hra je taká, v ktorej mních dokáže za daných pravidiel pohrabať celú záhradu, prípadne maximálny možný počet políčok. Výstupom je pokrytie danej záhrady prechodmi mnícha.

Úlohu je možné rozšíriť tak, že mních navyše zbiera popadané lístie. Listy musí zbierať v poradí: najprv žlté, potom pomarančové a nakoniec červené. Príklad vidno na obrázku nižšie. Je potrebné primerane upraviť fitness funkciu. Za takto rozšírenú úlohu je možné získať navyše jeden bonusový bod.

Uvedenú úlohu riešte pomocou evolučného algoritmu. Maximálny počet génov nesmie presiahnuť polovicu obvodu záhrady plus počet kameňov, v našom príklade podľa prvého obrázku $12+10+6=28$. Fitness je určená počtom pohrabaných políčok. Výstupom je matica, znázorňujúca cesty mnícha. Je potrebné, aby program zvládol aspoň záhradku podľa prvého obrázku, ale vstupom môže byť v princípe ľubovoľná mapa.

Vlastnosti použitých génov

Dátovú štruktúru, do ktorej som uložil gény je typu tuple, ktorý má dva parametre. Prvý je celočíselná premenná väčšia alebo rovná nule a menšia ako obvod záhradky. Reprezentuje začiatočnú pozíciu mnícha v záhradke.



$$\text{Circle (Obvod)} = (12+10)*2-1 = 43$$

Obr.1 - Šifrovanie začiatočnej pozície mnícha v jednom čísle.

Druhý parameter je typu Boolean a hodnota True reprezentuje uprednostnené otáčanie doprava a False doľava.

Opis pohybu mnícha

Podľa čísla v začiatočnom géne program vie, akým smerom sa mních vydá - viz obr.1. Ak je hodnota v géne v prvom parametri medzi 0 a 11 pôjde smerom nadol, ak medzi 12 a 23 pôjde smerom nahor atď. Na mapovanie pohybu nám stačia dva mníchove stavy a to, či sa pohybuje vertikálne alebo horizontálne. Mních najskôr pozrie, či na miesto, kde si má stúpiť hneď nie je kameň. Ak nie je pokračuje stále rovno a kontroluje si, či pred ním nie je prekážka alebo koniec mapy. Ak je pred ním prekážka, jeho pohyb sa z vertikálneho zmení automaticky na horizontálny alebo naopak. To znamená, že našu premennú kontrolujúcu smer pohybu je boolean a pri otáčaní ju stačí iba znegovať. Mních sa pozrie do génu, aké otočenie má uprednostnené a zistí si druh pohybu a podľa toho sa otočí. Tento úsek kódu znázorňuje jednoduché menenie súradníc pri otáčaní doprava, doľava, vertikálne, horizontálne:

```
while True:
    if not check(x + x_inc, y + y_inc, height, width):
        break
    elif garden[x + x_inc][y + y_inc]:
        x_inc, y_inc = y_inc, x_inc
        if direction:          # VERTICAL
            if genes[1]:       # RIGHT
                x_inc, y_inc = -x_inc, -y_inc
            else:              # HORIZONTAL
                if not genes[1]: # LEFT
                    x_inc, y_inc = -x_inc, -y_inc
```

Obr.2 – Otáčanie a rozhodovanie sa mnícha pri prekážkach.

Ak mních dôjde na okraj mapky, zoberie si ďalší gén a pokračuje zasa odznova.

Spôsob tvorby novej generácie

Novú generáciu vytvorím istým percentom najlepších mníchov zo starej a zvyšný počet domnožím náhodne vybratými dvomi mníchmi a ich génmi zo starej generácie, pričom vyššie hodnotenie génov sa berie do úvahy ako vyššia pravdepodobnosť výberu.

```
for i in range(monk_numb):
    sum += package[i][0]

weights = [package[i][0]/sum for i in range(monk_numb)]
```

Obr.3 – Ukážka algoritmu vyberania váhy pravdepodobnosti jednotlivých mníchov zo starej generácie.

Obom vybraným mníchom sa náhodne vyberie polka génov a zlúči a vytvorí sa nový jedinec do novej generácie. Pri vytváraní jedinca do novej generácie je 2% šanca, že jedinec zmutuje. Mutácia prebieha veľmi jednoducho. Mníchovi zoberiem náhodne jeden gén a s 80% pravdepodobnosťou mu zmením náhodne začiatočnú polohu a s 20% mu zmením uprednostnený smer.

Možnosti nastavenia parametrov

Používateľ si môže nastaviť rozmery záhrady, počet aj polohu kameňov v nej, počet mníchov v jednej generácii, počet génov, koľko bude mních vlastniť a maximálny počet generácií.

Zhodnotenie vlastností vytvoreného systému

Systém má ošetrené chyby vstupu pre používateľa, primeranú optimalizáciu a výstup do konzole a takisto grafické znázornenie postupu najlepšieho jedinca. Systém je naprogramovaný v programovacom jazyku Python za pomoci knižníc numpy a random. Systému sa podarí nájsť najlepší výsledok a je bezchybný. Taktiež je plnohodnotne optimalizovateľný príp. zmenou hodnôt konštánt, pretože všetky premenné mám adekvátne a zrozumiteľne pomenované.

Opis vylepšovania, doladovania riešenia

Riešenie je veľmi dobre navrhnuté a algoritmu hľadania cesty nemám čo vytknúť. Optimalizovať by sa dal program nasadením quick select algoritmu na triedenie najlepších génov a optimalizácia taktiež pripadá do úvahy s doladením hodnôt premenných ako som spomenul vyššie a priblížiť sa tým viac k lepšej pravdepodobnosti nájdenia ideálnych ciest.