Hľadanie pokladu

Matúš Tundér (74288)

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**, “dolu” **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.

Obsah

Hľadanie riešenia

prebieha podľa nasledovného postupu:

1. Zo súboru **settings.xml** si program načíta parametre, podľa ktorých si určí správanie sa algoritmu
2. Zo vstupu si načíta šírku, výšku a počet pokladov
3. Následne načíta X a Y súradnice pre každý poklad
4. A nakoniec načíta začiatočnú pozíciu hľadača pokladov
5. Program vygeneruje prvú generáciu podľa načítaných nastavení
6. Ak prešiel maximálny čas hľadania alebo maximálny počet generácií,
   tak vypíš doteraz nájdené najlepšie riešenie a čakaj na vstup od používateľa -
   ukončenie programu, nové hľadanie alebo aktualizovanie nastavení.
7. Paralelne spusti **program** každého **jedinca** pre aktuálnu generáciu
   7.1. Ak prešiel maximálny čas hľadania alebo maximálny počet generácií,
   tak skonči a zastav ďalšie hľadanie
   7.2. Spusti program jedinca, ktorý vráti cestu hľadača a aktualizuje fitness a počet nájdených pokladov jedinca
   7.3. Ak jedinec našiel všetky poklady,
   tak ulož jedinca a jeho cestu, skonči a zastav ďalšie hľadanie
8. Ak neprešli programy všetkých jedincov, tak
   8.1. Ak nie je uložený jedinec, tak skoč na krok 6 (prešiel čas)
   8.2. Inak vypíš informácie o úspešnom nájdení cesty
   8.3. Čakaj na vstup od používateľa - ukončenie programu, nové hľadanie alebo aktualizovanie nastavení.
9. Zoraď jedincov podľa fitness (Legacy: Zistenie elít a 3tatistiky)
10. Ak je povolený elitarizmus, tak prekopíruj elitu do novej generácie

11. Vyber dvoch jedincov pomocou rulety alebo Turnaja a skríž ich
12. Vytvorený jedinec s istou pravdepodobnosťou mutuje a následne vstupuje do novej populácie
13. Zameň novú generáciu s aktuálnou generáciou
14. Prejdi na krok vykonávania 6

Jedinec

Jedinec obsahuje pole 64 buniek, ktoré obsahuje inštrukcie s hodnotami, fitness po spustení jeho programu a počet nájdených pokladov.

Generovanie jedinca

Prvá generácia jedincov má náhodne vygenerované hodnoty pre prvých N buniek. Počet buniek, ktoré sa vygenerujú sa dá nastaviť. Štandardná hodnota je 16.

Kríženie jedincov

Vyber bodu kríženia je obmedzený minimálnym a maximálnym indexom, ktorý sa môže náhodne vygenerovať. Štandardný rozsah je 24 až 40, takže vždy sa minimálne zachová 24 buniek aspoň s jedného rodiča po skrížení.

Mutácia

Jedincov je možné mutovať tromi spôsobmi a ich pravdepodobnosť je nastaviteľlná. Štandardná pravdepodobnosť týchto mutácií je 2%, 3% a 5%, tým pádom mutácia prebehne s pravdepodobnosťou 10%.

1. Naplnenie náhodnej bunky náhodným obsahom
2. Invertovanie náhodnej bunky
3. Invertovanie náhodného bitu

Nová generácia

Používateľ si vie nastaviť či metódu selekcie novej generácie (ruleta alebo turnaj) a elitarizmus. Vytváranie je popísané v krokoch 9 až 13, ktoré sú upresnené v jednotlivých kapitolách ako je Generovanie jedinca, Kríženie jedincov a Mutácia. Spomenuté nebolo, že pri výbere pomocou rulety je ešte nutné zarovnať všetky hodnoty fitness na 1 (v prípade nulových alebo záporných hodnôt) aby sme mali možnosť vyberať naozaj zo všetkých jedincov.

Virtuálny stroj

Inštrukcie

Virtuálny stroj pozná 4 inštrukcie podľa tvaru uvedeného v tabuľke. Najvyššie 2 bity znázorňujú inštrukciu a nižších 6 bitov znázorňuje adresu s ktorou daná inštrukcia pracuje.

Inkrementácia	00XXXXXX
Dekrementácia	01XXXXXX
Skok	10XXXXXX
Výpis	11XXXXXX

Inkrementácia a Dekrementácia

Tieto 2 inštrukcie menia hodnotu bunky na danej adrese pripočítaním alebo odčítaním jednotky. Inštrukcie vedia zmeniť všetkých 8 bitov bunky. Tieto inštrukcie majú povolené pretečenie (overflow).

Skok

Inštrukcia skoku zmení nasledujúcu adresu pre vykonanie príkazu na adresu nachádzajúcu pri tejto inštrukcii. Z toho vyplýva, že inštrukcia po skoku sa zavolá až v ďalšej iterácií.

Výpis

Inštrukcia výpisu určuje smer na základe najnižších dvoch bitov v bunke získanej podľa adresy ktorú dostal, kde smer pre danú kombináciu môžeme vidieť v tabuľke.

00	H – hore
01	D – dole
10	P -vpravo
11	L - vľavo

Zastavenie programu

Program sa môže zastaviť v 3 nasledovných prípadoch:

- Po 500 vykonaných inštrukciách
- Keď hľadač pokladu vyjde mimo mriežky
- Keď hľadač pokladu nájde všetky poklady

Nastavenia

Nastavenia sa nachádzajú v súbore ./settings.xml

Príklad nastavení:

```
<?xml version="1.0"?>
<Settings InitRadnom="16" MaxJedincov="250"  SelectionType="Turnaj" Output="Result"
Plocha="vstup.txt" Stats="stats3.txt">
  <StopAfter Hodnota="20" Typ="Secs" />
  <Elitarizmus Hodnota="10" Typ="Percenta" />
  <BodKrizenia Min="24" Max="40" />
  <Fitness Poklad="100" Krok="1" VyjdenieMimoMriezky="5" />
  <PomerMutacie BezMutacie="90" NahodnaBunka="2" XorNahodnaBunka="3" XorNahodnyBit="5" />
</Settings>
```

Settings

InitRadnom

Počet buniek, ktoré sa náhodne inicializujú pre prvú generáciu.

MaxJedincov

Počet jedincov v jednej generácií.

SelectionType

Typ selekcie pre vyberanie jedincov z populácie

- Ruleta
- Turnaj

Output
Typ výpisu informácií na obrazovku.

- Result – iba výsledok
- All – Všetkých vygenerovaných jedincov
- Top – Najlepšieho jedinca v každej generácií

Plocha
Súbor z ktorého sa načíta veľkosť plochy, rozloženie pokladov a začiatočná pozícia hľadača pkladu.

Stats
Cesta k súboru kam sa uloží štatistika vývoja fitness.

StopAfter
Určuje pokiaľ bude prebiehať evolúcia.

Typ
- Secs – maximálny počet sekúnd
- Gens – Maximálny počet generácií

Hodnota
Určuje hraničnú hodnotu pre daný typ.

Elitarizmus
Pokiaľ tento element existuje tak je povolený elitarizmus.

Typ
- Percenta
- Pocet

Hodnota
Počet elít.

BodKrizenia
Určuje hranice pre zvolenie bodu kríženia.

Fitness
Poklad
Koľko bodov sa pripočíta po nájdení pokladu.

Krok
Koľko bodov sa odčíta po každom kroku.

VyjdenieMimoMriezky
Koľko bodov sa odčíta keď program skončí tým, že vyšiel mimo mriežku.

PomerMutacie
Určuje v akom pomere sa budú jedinci mutovať pred vstupom do novej generácie.

BezMutacie
Jedinec sa vloží do novej generácie bez mutácie.

NahodnaBunka
Náhodnej bunke v jedincovi sa pridá nová náhodná hodnota.

XorNahodnaBunka
Invertuje sa náhodná bunka.

XorNahodnyBit
Invertuje sa náhodný bit.

Zhodnotenie
Počas tvorby programu som skúšal viacero typov pripočítavania a odpočítavania Fitness. Taktiež som sa rozhodol uchovávať fitness ako integer (a vynásobiť všetky hodnoty) namiesto desatinného čísla aby som mohol jednoduchšie vyberať jedincov v rulete.

Pri príliš nízkej pravdepodobnosti mutácií mu trvalo dlhšie nájdenie jedinca, ktorý našiel všetky poklady. Pri malom počtom jedincoch sa postupne všetci jedinci začali podobať a kríženie prestavalo mať zmysel (fungovať).

Najlepším zrýchlením bolo paralelizovať spúšťanie programov jedincov a obmedzenie výpisu na obrazovku.

Pre lepšiu prehľadnosť som k výpisu cesty, ktorou prešiel hľadač pokladu ešte pridal symbol **$,** čo znázorňuje že po danom pohybe hľadač našiel poklad, čo môžeme vidieť na nasledujúcom obrázku (Výsledná cesta je zeleným textom).

```
<Sirka> <Vyska> <PocetPokladov>
<x> <y> // pokladu 1
<x> <y> // pokladu 2
...
<x> <y> // pokladu n
<x> <y> // Zaciatocna pozicia

7 7 5
4 1
2 2
1 4
4 5
6 3
3 6
gen: 413
Nasiel som riesenie:
Gen: 414 | Kroky: 22 | Cesta: HP$HDHDHLLL$HHP$HDPPH$DPPD$
```

Porovnanie

Vytvárania nových generácií s a bez elitarizmu. Z nasledujúcich grafoch, ktoré ukazujú vývoj fitness si môžeme hneď všimnúť, že najväčším rozdielom bez elitarizmu je klesanie najlepších nájdených hodnôt. Pri elitarizme sa zvyšný jedinci postupne prispôsobia k elitám čím sa na druhú stranu môže znížiť variabilita, keďže čím ďalej sa bude viac a viac jedincov podobať elitám.