

Implementacja algorytmu genetycznego. Prototyp 2 (z Circle Packing)

Algorytmy i Struktury Danych
Wydział Elektryczny, Politechnika Warszawska

Tomasz Sobutka Artur Skonecki
Prowadzący: Bartosz Chaber

Wygenerowano: 22 stycznia 2012

1 Specyfikacja funkcjonalna

1.1 Opis Problemu

Celem projektu jest zaimplementowanie algorytmu genetycznego służącego do optymalizacji wnętrza środka komunikacji miejskiej wraz z wizualizacją. Efektem końcowym działania algorytmu ma być znalezienie najlepszego “wnętrza środka komunikacji miejskiej”.

1.1.1 Źródła

- Algorytmy genetyczne i ich zastosowania, David E. Goldberg, WNT 2003
- Algorytmy genetyczne + struktury danych = programy ewolucyjne, Zbigniew Michalewicz, WNT 1999
- <http://www.obitko.com/tutorials/genetic-algorithms/ga-basic-description.php>
- <http://leonardo-m.livejournal.com/80721.html>
- <http://geneticalgorithms.ai-depot.com/Tutorial/Overview.html>
- http://en.wikipedia.org/wiki/Circle_packing

1.2 Tworzenie osobników

Pierwsza generacja osobników jest tworzona za pomocą losowania. Kolejne generacje są tworzone za pomocą następujących technik:

- Krzyżowanie - osobniki są losowo dobierane w pary i na tej podstawie obliczane jest ich potomstwo.
- Mutacja - dla osobników w określonej części populacji istnieje szansa, że zostaną w losowy sposób zmienione

1.3 Selekcja osobników

Funkcja używa algorytmu Circle Packing, w celu znalezienia najlepszych osobników. Podczas obliczania przystosowania brane są pod uwagę następujące cechy:

- wygodę wnętrza, związaną z liczbą miejsc siedzących
- koszt materiałów, większy dla miejsc siedzących niż dla wolnej przestrzeni

- ilość miejsc w pojeździe, każde miejsce siedzące to 1 miejsce w pojeździe, wolna przestrzeń jest optymalizowana pod względem liczby pasażerów

W programie wykorzystano następujące rozwiązania w selekcji osobników:

- Koło ruletki, osobniki o większym przystosowaniu mają większą szansę zostać wylosowane do tworzenia potomstwa
- Elityzm, osobniki o największym przystosowaniu zawsze przechodzą do następnej generacji

1.4 Kodowanie genotypu

Genotyp osobników jest rozpatrywany jako ciągi bajtów. Każdy ciąg koduje 1 cechę, jest określonej, stałej długości i podlega krzyżowaniu oraz ewentualnej mutacji. Wszystkie cechy są interpretowane jako liczby naturalne.

1.5 Cechy osobników

Każdy osobnik jest określony przez następujące cechy:

- długość
- szerokość
- maksymalna liczba miejsc siedzących
- liczba rzędów
- odległość między rzędami
- parametr określający co ile siedzeń w rzędzie występuje separator

1.5.1 Pierwsza Generacja

Wszystkie pierwsze osobniki są tworzone za pomocą losowania wartości cech osobników zawartych w dopuszczalnych wartościach .

1.6 Tworzenie kolejnych generacji

Kolejne generacje są tworzone z osobników wybranych przez koło ruletki i elityzm. Potomstwo wybranych osobników jest tworzone przez mutacje i krzyżowanie.

1.6.1 Operator mutacji

Mutacja polega na przypisaniu losowych wartości na losowych pozycjach w ciągach.

1.6.2 Operator Krzyżowania

Łączenie osobników przypomina “suwak”, który raz bierze wartość od jednego osobnika a raz od drugiego

1.7 Zakończenie symulacji

- Po określonej maksymalnej liczbie generacji.
- Po osiągnięciu określonej wartości funkcji określającej przystosowanie.
- Na żądanie użytkownika.

1.8 Interfejs użytkownika oraz parametry symulacji

W celu ułatwienia testowania implementacji algorytmu genetycznego, pakiet zawiera graficzny interfejs użytkownika pozwalający na obserwację postępu pracy za pomocą wizualizacji rozwiązania zadania.

Komendy

Interfejs umożliwia wykonanie poleceń:

Komenda	Argument	Opis
start	-	Rozpoczęcie / kontynuacja symulacji
stop	-	Wstrzymanie symulacji
reset	-	Usunięcie bieżącej symulacji
show	[int]	Ustawienie wizualizacji na osobnika $(1 - \infty)$ (od najlepszego do najgorszego), wartość 0 oznacza osobnika obecnie ocenianego
[integer]	-	skrót do komendy show

Parametry symulacji - część 1

Interfjes pozwala ustawić następujące parametry symulacji (niektóre parametry wymagają resetu przed zadziałaniem):

Parametr	Argument	Opis
Fcircle-radius	[float]	Promień zajmowany przez pojedynczego pasażera
Fcost-seat	[float]	Koszt siedzenia
Fcost-space	[float]	Koszt przestrzeni
Felitism	[float]	Procent populacji do której stosuje się elityzm
Ffactor-comfort	[float]	waga komfortu w funkcji oceniającej
Ffactor-cost	[float]	waga kosztu w funkcji oceniającej
Ffactor-	[float]	waga dostępnej przestrzeni w funkcji oceniającej
Fmax-adaptation	[float]	Wartość funkcji określającej przystosowanie po przekroczeniu której symulacja zakończy działanie
Fmax-cost	[float]	maksymalny koszt, jeżeli koszt osobnika jest większy, to wartość funkcji oceniającej przystosowanie będzie wynosić zero
Fmin-comfort	[float]	minimalny komfort, jeżeli komfort osobnika jest mniejszy, to wartość funkcji oceniającej przystosowanie będzie wynosić zero
Fmin-space	[float]	minimalna przestrzeń, jeżeli przestrzeń osobnika jest mniejsza, to wartość funkcji oceniającej przystosowanie będzie wynosić zero
Fmutation-rate	[float]	Współczynnik mutacji
Froulette	[float]	Procent osobników losowanych przez ruletkę podczas wyboru osobników do tworzenia nowej generacji
Fseat-offset	[float]	Wolna przestrzeń pozostawiana za każdym siedzeniem w rzędzie
Fseat-radius	[float]	Promień siedzenia brany pod uwagę przez circle packing
Fseat-size	[float]	Długość boku rysowanego siedzenia

Parametry symulacji - część 2

Parametr	Argument	Opis
Icircles	[int]	Liczba kółek symulujących pasażerów
Igenome-bits	[int]	liczba bitów w na które konwertowane są cechy osobników
Imax-adaptation	[int]	Maksymalna wartość funkcji określającej adaptację, w przypadku przekroczenia symulacja jest wstrzymywana
Imax-bus-h Imin-bus-h	[int]	Maksymalna i minimalna szerokość autobusu
Imax-bus-w, Imin-bus-w	[int]	Maksymalna i minimalna długość autobusu
Imax-generations	[int]	Maksymalna liczba generacji przy której symulacja zakończy działanie, 0 to brak ograniczenia
Imax-rows, Imin-rows	[int]	Maksymalna i minimalna liczba rzędów siedzeń w autobusie
Imax-rows-spacing, Imin-rows-spacing	[int]	Maksymalny i minimalny odstęp między rzędami siedzeń w autobusie
Imax-seats, Imin-seats	[int]	Maksymalna i minimalna liczba siedzeń w autobusie
Imax-seat-separator , Imin-seat-separator	[int]	Maksymalna i minimalna wartość, która określa co ile siedzeń w rzędzie jest separator
Ipacking-iter	[int]	liczba iteracji circle packing dla każdej ewaluacji osobnika
Ipopulation	[int]	Wielkość populacji

Dodatkowe funkcje interfejsu użytkownika

- Wyświetla liczbę generacji
- Wyświetla numer obecnie ocenianego osobnika w populacji
- Wyświetla czas trwania symulacji
- Okno wizualizacji pozwala wyświetlić dowolnego osobnika (domyślnie wyświetlany jest aktualnie oceniany)
- Pokazuje statystyki wyświetlanego osobnika
- Pokazuje parametry wyświetlanego osobnika

1.9 Opis interfejsu użytkownika

- Na samej górze okna programu znajduje się “linia poleceń”, przez którą odbywa się cały dialog z programem.
- Na środku znajduje się wizualizacja autobusu. Pasażerowie są reprezentowani kółka a fotele przez kwadraty.
- Po lewej stronie umieszczone są statystyka symulacji oraz statystyka osobnika.
- Po prawej stronie umieszczone są status symulacji i wizualizacji oraz parametry symulacji.

prompt>

Genetic algorithm evaluator and viewer.

generation = 10

specimen = 0

time = 50.62

bus height = 95

bus width = 359

floor space = 118306.0

rows spacing = 32

seat separator = 10

seats = 47

seats space = 10575.0

standing = 33

stat = 1046.125

stat com fort = 920.0

stat cost effect = 47.125

stat space = 79.0

running : True

show : 1

Fcircle-radius : 10.5

Fcost-seat : 4.0

Fcost-space : 1.0

Felitism : 0.1

Ffactor-comfort : 20.0

Ffactor-cost : 0.001

Ffactor-space : 1.0

Fmax-cost : 100000.0

Fmin-comfort : 1.0

Fmin-space : 1.0

Froulette : 0.5

Fseat-offset : 3.0

Fseat-radius : 10.6066017178

Fseat-size : 15.0

Icircles : 250

Igenome-bits : 32

Imax-adaptation : 0

Imax-bus-h : 100

Imax-bus-w : 400

Imax-generations : 100

Imax-rows : 10

Imax-rows-spacing : 100

Imax-seat-separator : 10

Imax-seats : 200

Imin-bus-h : 10

Imin-bus-w : 10

Imin-rows : 1

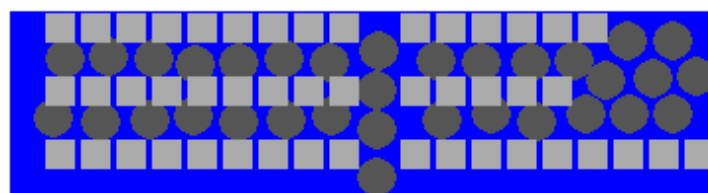
Imin-rows-spacing : 32

Imin-seat-separator : 1

Imin-seats : 0

Ipacking-iter : 100

Ipopulation : 20



2 Specyfikacja implementacyjna

2.1 Narzędzia

Program został zaimplementowany w języku **Python 2.5**, przy wykorzystaniu biblioteki **PyGame** wersja **1.7.1**. Program wykorzystuje moduł **eztext**, który implementuje prostą linię komend. Dokumentacja jest generowana za pomocą *LaTeX*.

2.2 Pliki i Klasy

Klasa	Plik	Opis
-	run.sh	startuje program
*	src/eztext.py	Moduł (zewnętrzny) implementujący prostą linię komend w PyGame.
-	src/Util.py	Zawiera funkcje pomocnicze.
-	src/Main.py	Punkt wejściowy. Znajduje się tu również główna pętla programu.
Breeder	src/Genetic.py	Klasa implemująca tworzenie nowej populacji na podstawie starej.
GenerationEvaluator	src/Genetic.py	Klasa wspomagająca wybieranie osobników z generacji.
Generation	src/Genetic.py	Klasa reprezentująca generację osobników.
Config	src/Config.py	Służy do konfiguracji symulacji i parametrów programu.
Specimen	src/Objects.py	Reprezentuje pojedynczego osobnika.
Seat	src/Objects.py	Reprezentuje pojedynczy fotel zawarty w osobniku.
Circle	src/Objects.py	Reprezentuje pojedynczego pasażera.
Overlay	src/Overlay.py	Wyświetla interfejs użytkownika.

2.3 Opis implementacji algorytmu genetycznego

2.3.1 Kodowanie

Wszystkie cechy osobników są określone przez liczby całkowite. Są one konwertowane do reprezentacji bitowej (o długości określonej parametrem Igenome-bits) za pomocą funkcji Util.int2bin(). Po wykonaniu operacji krzyżowania i mutacji wartości są konwertowane z powrotem to liczb całkowitych za pomocą funkcji Util.bin2int().

2.3.2 Krzyżowanie i mutacja

Łączenie osobników przypomina “suwak”, który raz bierze wartość od jednego osobnika a raz od drugiego (`Breeder._crossover()`). Przed wykonaniem krzyżowania, wszystkie cechy są konwertowane do reprezentacji binarnej i łączone w jeden długi ciąg. Mutacja jest wykonywana na losowych osobnikach przez funkcję `Breeder._mutate()`.

2.3.3 Określanie przystosowania

Funkcja określająca przystosowanie (`process_specimen()` w module `Genetic`) bierze pod uwagę 3 parametry: komfort (liczba siedzeń), koszt (przesteń zajmowana przez siedzeń i wolna przestrzeń), oraz całkowita liczba pasażerów w autobusie. Ponadto program bierze pod uwagę wagi poszczególnych parametrów (`Ffactor-comfort`, `Ffactor-cost`, `Ffactor-capacity`). Liczba pasażerów jest określana przy pomocy funkcji implementującej algorytm Circle Packing (`Util.pack()`)

2.3.4 Selekcja

Selekcja osobników do nowego tworzenia nowego pokolenia odbywa się za pomocą koła ruletki (`GenerationEvaluator.roulette()`). Osobniki o większym przystosowaniu mają większą szansę zostać wybrane przez koło ruletki.

2.3.5 Elityzm

Osobniki, które zostały określone jako najlepsze są przekopiiowywane do nowej populacji (`GenerationEvaluator.get_elite()`).