

Implementacja algorytmu genetycznego

Prototyp 1

Algorytmy i Struktury Danych

Wydział Elektryczny, Politechnika Warszawska

Tomasz Sobutka

Artur Skonecki

Prowadzący: Bartosz Chaber

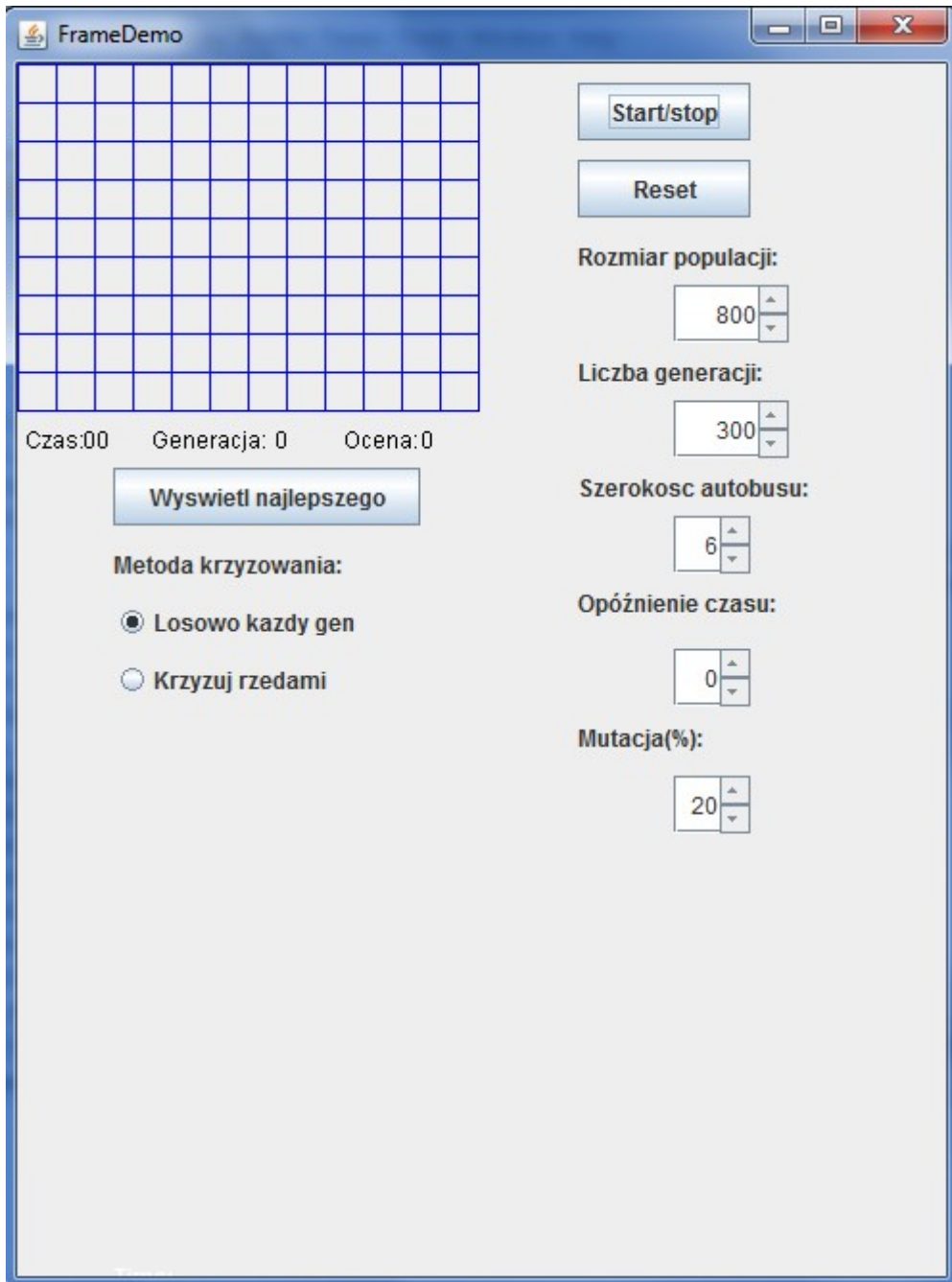
Wygenerowano: 21 stycznia 2012

1. Specyfikacja funkcjonalna

1.1 Cel programu

Celem programu było znalezienie jak najlepszego wnętrza środka komunikacji miejskiej - autobusu, przy pomocy algorytmów genetycznych.

1.2 Interfejs użytkownika



2. Specyfikacja implementacyjna

2.1 Cechy osobników

Autobus jest opisany przez następujące parametry:

- szerokość w liczbie mieszczących się w kolumnie ludzi
- długość w liczbie mieszczących się w rzędzie ludzi
- rozmieszczenie siedzeń i miejsc stojących

2.2 Populacja początkowa

Populacja początkowa jest losowana. Prawdopodobieństwo wystąpienia siedzenia lub miejsca stojącego na danym miejscu wynosi 50%.

2.3 Algorytm BFS

Do zbudowania funkcji oceny służy nam algorytm przeszukiwania wszerz w grafie. W projekcie graf składa się z poszczególnych miejsc jako wierzchołki grafu oraz krawędzie, które budowane są na podstawie, że z danego miejsca można przejść do miejsca znajdującego się obok, czyli wyżej, niżej, na lewo i na prawo - patrząc na wizualizację. To oznacza że wierzchołek ma 4 krawędzie, oprócz miejsc skrajnych autobusu. Jednak w projekcie, przeszukiwanie kończy się na miejscu siedzącym z którego nie da się pójść dalej. Innymi słowy taki wierzchołek nie jest dodawany do kolejki. Wynikiem działania algorytmu jest liczba odwiedzonych wierzchołków grafu czyli miejsc w autobusie.

2.4 Funkcja oceny

Miejsce	siedzące	stojące
---------	----------	---------

Wygoda	7	3
--------	---	---

Koszt	3	1
-------	---	---

Wartość funkcji oceny $(wygoda - koszt) * bfs / dlugosc$

2.5 Selekcja osobników

Selekcja osobników jest dokonywana na podstawie wartości funkcji oceny. Najlepsze 25% osobników zostaje do następnego pokolenia. Pozostałe 75% osobników są budowane za pomocą krzyżowania. ze sobą tych najlepszych osobników.

2.6 Krzyżowanie

W programie są zaimplementowane dwa sposoby krzyżowania. Pierwsze to losowanie dla każdego miejsca w autobusie, z którego autobusu ma pochodzić dane miejsce. Gdy długość autobusu wzięta jest z większego autobusu, to wtedy dla końcowego fragmentu nie ma wyboru i trzeba przepisać genotyp.

Drugi sposób to podzielenie autobusu na trzy fragmenty po mniej więcej równej liczbie kolumn. Tworząc autobusy potomne mieszamy różne części autobusu. Ten sposób jest jednak dużo mniej

wydajny.

2.7 Mutacja

Z prawdopodobieństwem określonym w programie jedno z miejsc w każdym autobusie po operacji krzyżowania ulega transformacji w inne.

Wartości przedstawione na rysunku są wartościami domyślnymi.

2.8 Kodowanie genotypu

Kodowanie genotypu jest zaimplementowane za pomocą:

- Dwóch liczb całkowitych typu int określających długość i szerokość
- Tablicy dwuwymiarowej typu int. Wartość w tablicy określa czy w danym miejscu jest siedzenie, miejsce stojące, czy też wejście do autobusu.

2.9 Podział na klasy

Klasa Populacja - zawiera metody służące do inicjalizacji populacji, selekcji, krzyżowania, mutowania

Klasa Autobus - zawiera metody służące do inicjalizacji autobusu, ustawiania miejsc, obliczania funkcji oceny, implementacji algorytmu bfs

Klasa FrameDemo - dziedziczy z klasy JFrame, zawiera metody do wizualizacji autobusu i interfejsu użytkownika

Klasa Panel - dziedziczy z JPanel służy do lepszego oprogramowania panelu z wizualizacją

Klasa Figura - służy do rysowania miejsc siedzących i stojących w wizualizacji

Klasa AlgorytmyGenetyczne - zawiera main()

Klasa ListySasiedztwa - przechowuje informacje, który wierzchołek grafu został odwiedzony, potrzebna do implementacji bfs