

Sprawozdanie

Projekt 2

Problem komiwojażera z dwoma komiwojażerami

Metoda algorytmu genetycznego (1A)

Jacek Hojczak

Wstęp

Celem projektu było zaimplementowanie algorytmu genetycznego w celu rozwiązywania problemu TSP oraz dostrójenie parametrów algorytmu tak żeby uzyskać najlepsze wyniki.

Program implementujący algorytm został napisany w języku programowania Java.

Do celów testowych i strojenia parametrów algorytm był stosowany na dwóch różnych problemach za czerpanych z portalu internetowego <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>.

Jeden problem o wielkości 48 wierzchołkach do celów sprawdzania poprawności działania algorytmu oraz drugi problem od 280 wierzchołkach na którym parametry były strojone.

Autor projektu zdecydował się na strojenie 3 parametrów:

- Prawdopodobieństwo mutacji
- Prawdopodobieństwo krzyżowania
- Wielkość populacji startowej

Wybrany operator mutacji to inwersja.

Wybrany operator krzyżowania to PMX.

Opis programu

Program został napisany w języku *Java*, do kompilacji wymagane jest narzędzie *maven*. Aby skompilować program należy go rozpakować do katalogu po czym z tego katalogu uruchomić komendę z po przez polecenie:

```
mvn clean install
```

Uruchomić programu można za pomocą polecenia:

```
java -jar target/meh-jar-with-dependencies.jar src/main/resources/p2.properties
```

target/meh-jar-with-dependencies.jar – ścieżka do programu wykonywalnego

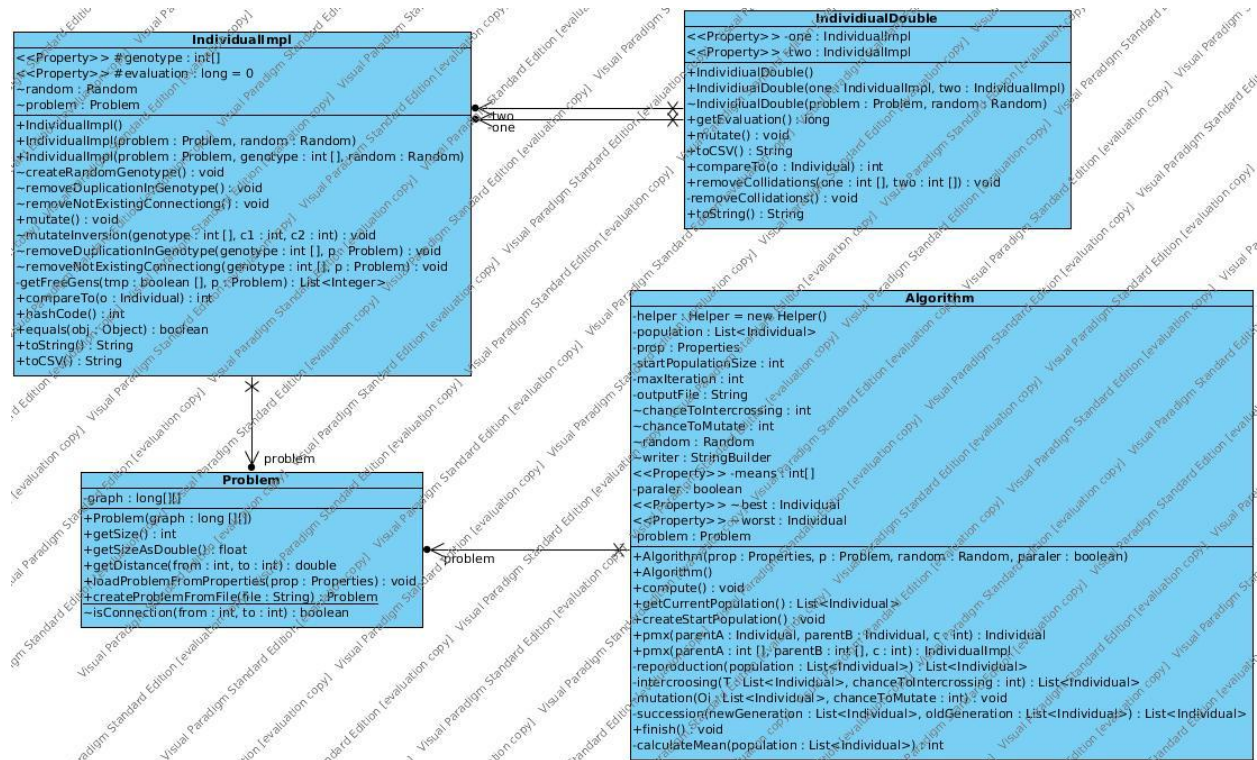
src/main/resources/p1.properties – ścieżka do pliku konfiguracyjnego

Program posiada plik konfiguracyjny w którym znajdują się parametry dla algorytmu genetycznego oraz parametry programu

```
# startowa wielkość populacji
population.start.size=3000
# maksymalna ilość iteracji
max.iteration=1000
# prawdopodobieństwo zajścia krzyżowania procentowo tzn.
# losowna jest liczba z rozkładu jednostajnego z przedziału 0-99, a następnie
# liczba ta jest przyrównana do poniższego parametru w poniższy sposób:
# 0-99 < {hance.to.intercroosing}
chance.to.intercroosing=50
# prawdopodobieństwo zajścia mutacji procentowo
# w takiej samej formie jak parametr chance.to.intercroosing
chance.to.mutate=15
# ścieżka do pliku z zdefiniowanym problemem zgodnym z formatem tsp1ib
problem.file=./src/main/resources/a280.xml
# nazwa pliku wyjściowego, do nazwy podanej w parametrze nazwy zostanie dodane
# rozszerzenie .csv
output.file=output
# przełącza program w specjalny tryb działania wykorzystany do sprawdzenia różnych
# zestawów parametrów zapisanych w kodzie źródłowym. Parametr wykorzystywany był do
# kalibracji parametrów na potrzeby sprawozdania.
# false - tryb standardowy
# true - tryb specjalny
all=false
# Parametr wykorzystywany w trybie specjalnym. Oznacza że każdy zestaw parametrów
# będzie powtórzony tyle razy ile wynosi wartość parametru.
iteration.per.param=10
```

Architektura programu

Głównymi klasami programu są 4 klasy przedstawione na poniższym diagramie:



Individual

Klasa odpowiedzialna za trzymanie pojedynczego rozwiązania. W klasie tej przechowywana jest jedna trasa w postaci tablicy liczb całkowitych w zmiennej `genotyp`. Indeks tablicy w tej tabeli oznacza jednostkę czasu. Wartość pod danym indeksem oznacza odwiedzany wierzchołek w czasie wskazującym przez indeks. Na każdej może znajdować się dowolne miasto o ile spełnia warunki zawarte w grafie. Przykład:

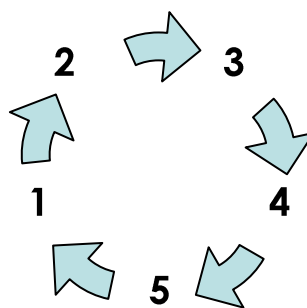


Figure 1 Przykładowa ścieżka komiwojażera

Powyższy cykl przedstawiony jest w tablicy w ten sposób:

[1,2,3,4,5]

Klasa `pond` to zawiera 3 ważne metody. Metoda

- `void mutate()` - mutująca swój genotyp zgodnie z operatorem mutacji inwersji.
- `void removeDuplicationInGenotype` – usuwa zduplikowane wierzchołki z genotypu i zastępuje je brakującymi
- `void removeNotExistingConnections()` – naprawia genotyp pod względem nie istniejących w grafie przejść tzn. metoda eliminuje nie poprawne przejścia między wierzchołkami i stara się je zastąpić istniejącymi.
- `long getEvaluation()` – zwraca długość drogi zapisanej w genotypie

DoubleIndividual

Klasa zawierająca dwa obiekty klasy `Individual`. Przechowuje rozwiązanie podwójnego problemu TSP tzn. z dwoma komiwojażerami. Ważne metody:

- `void removeCollidations(int[] one, int[] two)` - metoda usuwa kolizje pomiędzy dwoma genotypami i naprawia w ten sposób oba genotypy żeby wprowadzając jak najmniej zmian dla długości drogi zapewnić ograniczenia zadania tzn. Żeby obaj komiwojażerowie zaczynali i kończyli w tym samym wierzchołku i żeby w dany momencie nie odwiedzali tego samego wierzchołka.

Algorithm

Klasa stanowiąca główną jednostkę wykonywanego algorytmu. Implementuje operator mutacji `pmx`, metodę reprodukcji, selekcji oraz główną pętlę algorytmu, umożliwia manipulację parametrami algorytmu.

- `compute()` – rozpoczyna wykonywanie głównej pętli algorytmu
- `List<Individual> succession(List<Individual> newGeneration, List<Individual> oldGeneration)` – zwraca listę rozwiązań przechodzących do nowej iteracji wybierając ją z starego i nowo utworzonego pokolenia. Algorytm sukcesji opiera się na wyborze 25% rozwiązań z starego pokolenia oraz 75% z nowego, przy czym najlepsze rozwiązania mają największe szanse przejścia.
- `List<Individual> intercrossing(List<Individual> T, int chanceToInter crossing)` – metoda implementuje krzyżowanie rozwiązań za pomocą operatora `pmx` oraz umożliwia wprowadzenie parametru prawdopodobieństwa zajścia krzyżowania.
- `private List<Individual> reproduction(List<Individual> population)` – metoda implementująca metodę reprodukcji.

Założenia doświadczenia

Doświadczenie zakładało strojenie parametrów:

- Prawdopodobieństwo mutacji
- Prawdopodobieństwo krzyżowania
- Wielkość populacji

Żeby móc wykonać strojenie parametrów założono następujące wartości innych parametrów algorytmu:

Wielkość populacji w każdej iteracji jest stała.

Sukcesja do nowej iteracji realizowana jest tak: 25% rozwiązań z starych rozwiązań i 75% nowych rozwiązań przechodzi. Przy czym rozwiązania najlepsze mają największe prawdopodobieństwo przejścia.

Ilość iteracji wynosi 1000 powtórzeń pętli głównej,

Wielkość zadania: graf o 280 wierzchołkach o symetrycznych krawędziach.

W celu znalezienia najlepszych wartości parametrów przeprowadzenie następujące doświadczenia. Dla każdego ustalonego zestawu parametrów wykonano 10 powtórzeń pełnego przebiegu algorytmu. Mierzone były średnia rozwiązania uzyskanych w każdej iteracji, średnia z tych średnich, najlepsze i najgorsze rozwiązania oraz średnia z najlepszych i najgorszych rozwiązań. Poniżej zostaną przedstawione summaryczne wyniki poszczególnych doświadczeń. Dla każdej serii w której zmieniany był jeden z trzech szukanych parametrów inne z tych trzech miały wartości domyślne czyli:

szansa mutacji = 10

szansa krzyżowania = 90

wielkość populacji = 1000

Wyniki doświadczeń

Poniżej zostaną przedstawione w postaci tabel i wykresów wyniki doświadczeń.

Prawdopodobieństwo krzyżowania

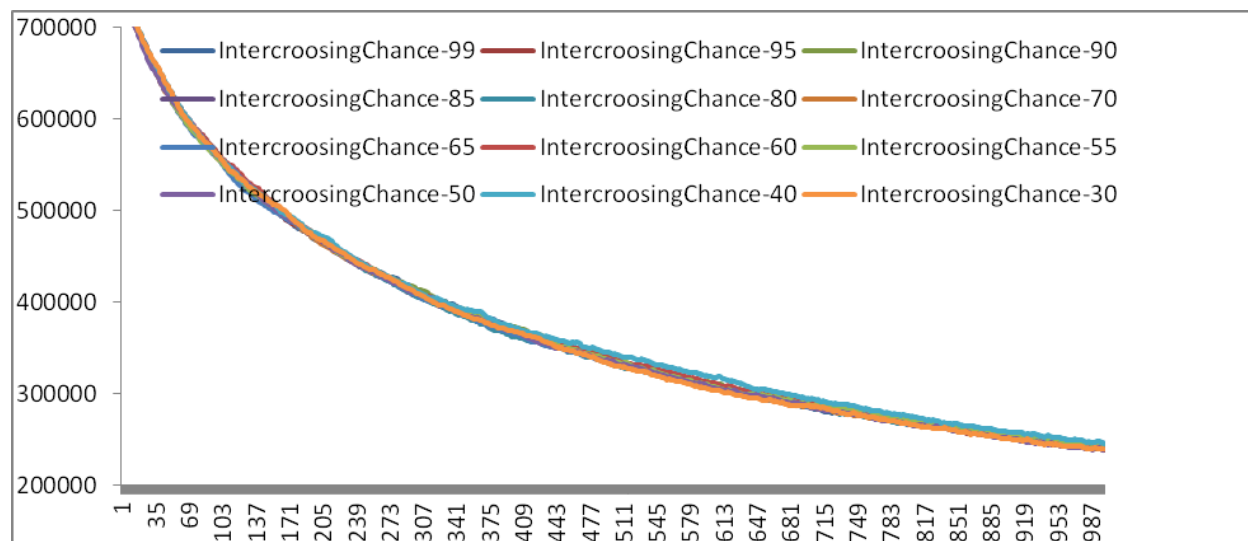


Figure 2 Średnie wartości w każdej iteracji z 10 potworzeń dla każdego zestawu z prawdopodobieństwem krzyżowania

Wartość	99	95	90	85	80	70	65	60	55	50	40	30
Czas	581	576	579	575	576	575	580	586	585	586	579	568
Najlepszy	21852	21460	21640	21612	21706	21644	21927	21806	21724	21383	21951	21558
Najgorszy	72834	72602	72767	72408	72825	72652	72689	72717	72637	72652	72696	72938

Figure 3 Tabel z summarycznymi wartościami dla strojenia prawdopodobieństwa krzyżowania

Prawdopodobieństwo mutacji

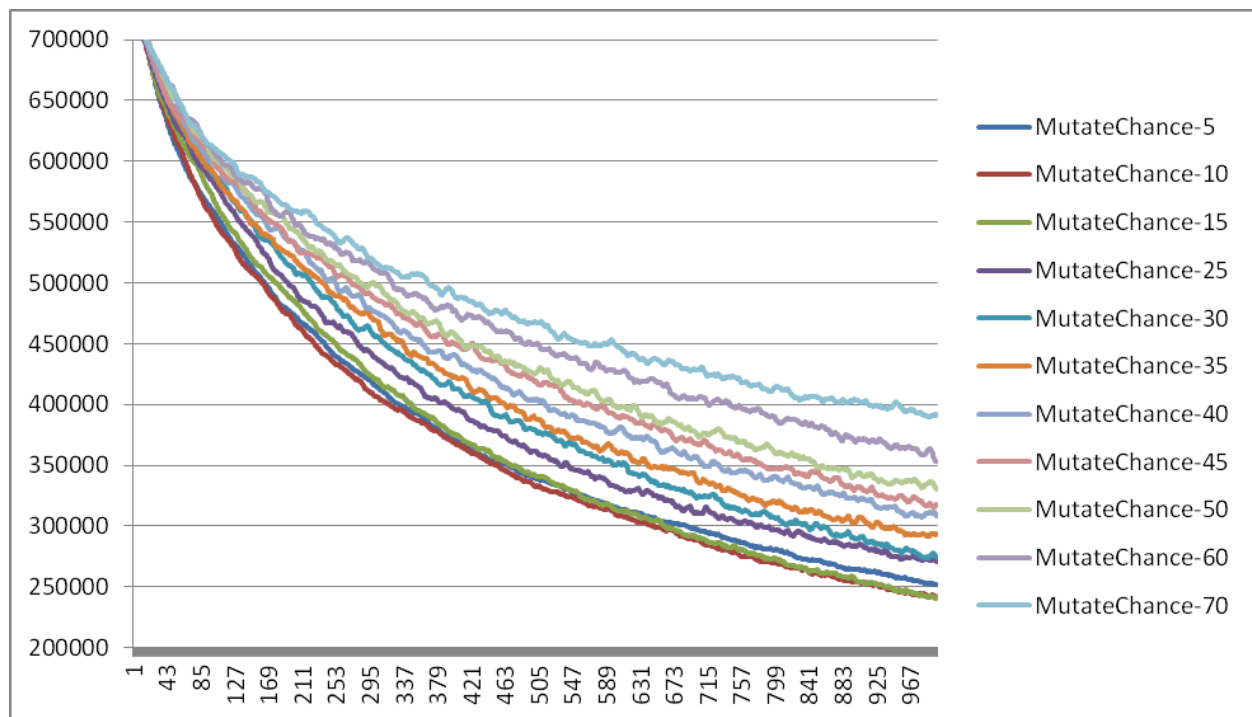


Figure 4 Średnie wartości w każdej iteracji z 10 potworzeń dla każdego zestawu z prawdopodobieństwem mutacji

Wartość	5	10	15	25	30	35	40	45	50	60	70
Czas	570	579	594	621	634	648	659	675	685	707	728
Najlepszy	22792	21725	21624	23757	24157	25408	26702	27475	28554	30736	33284
Najgorszy	72574	72677	72816	72819	72728	72667	72610	72606	72565	72644	72653

Figure 5 Tabel z summarycznymi wartościami dla strojenia prawdopodobieństwa krzyżowania

Rozmiar populacji

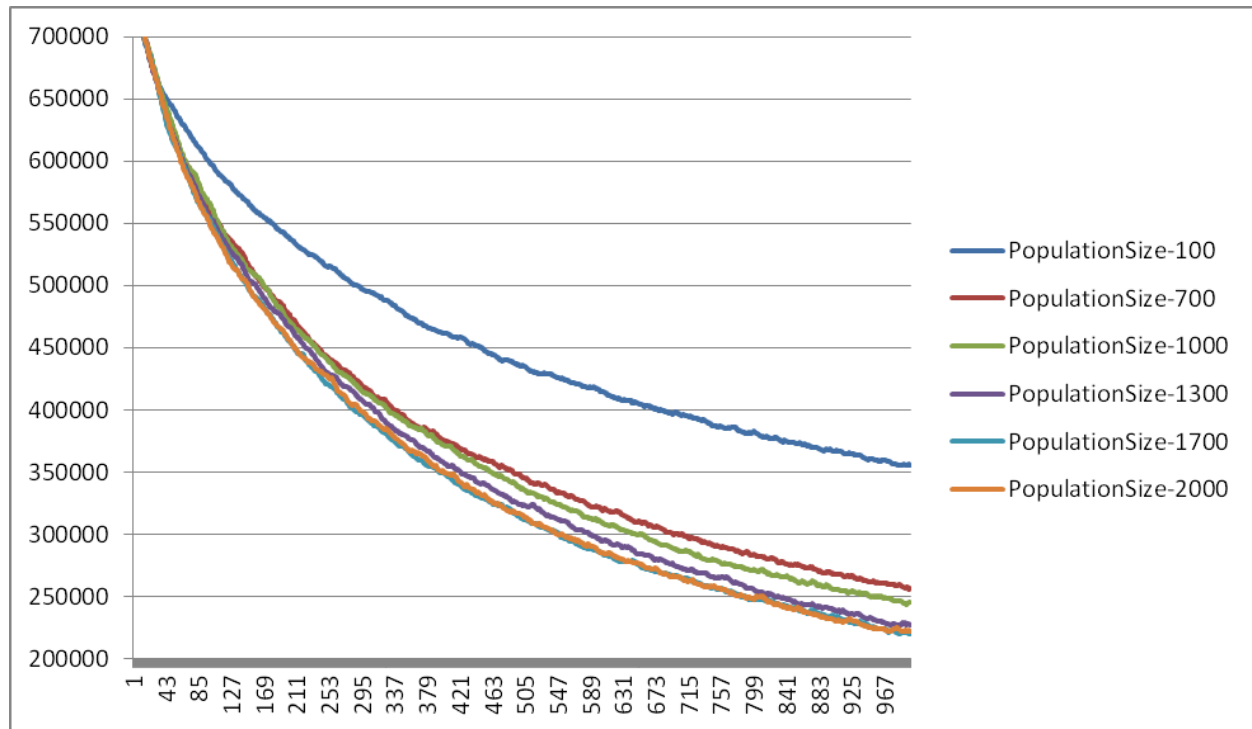


Figure 6 Średnie wartości w każdej iteracji z 10 pot wórzeń dla każdego zestawu z rozmiarem populacji

Wartość	100	700	1000	1300	1700	2000	3000
Czas	51	402	584	763	1001	1179	1790
Najelpszy	32090	23028	21954	20351	19708	19811	18848
Najgroszy	71611	72483	72641	72623	73131	72786	72995

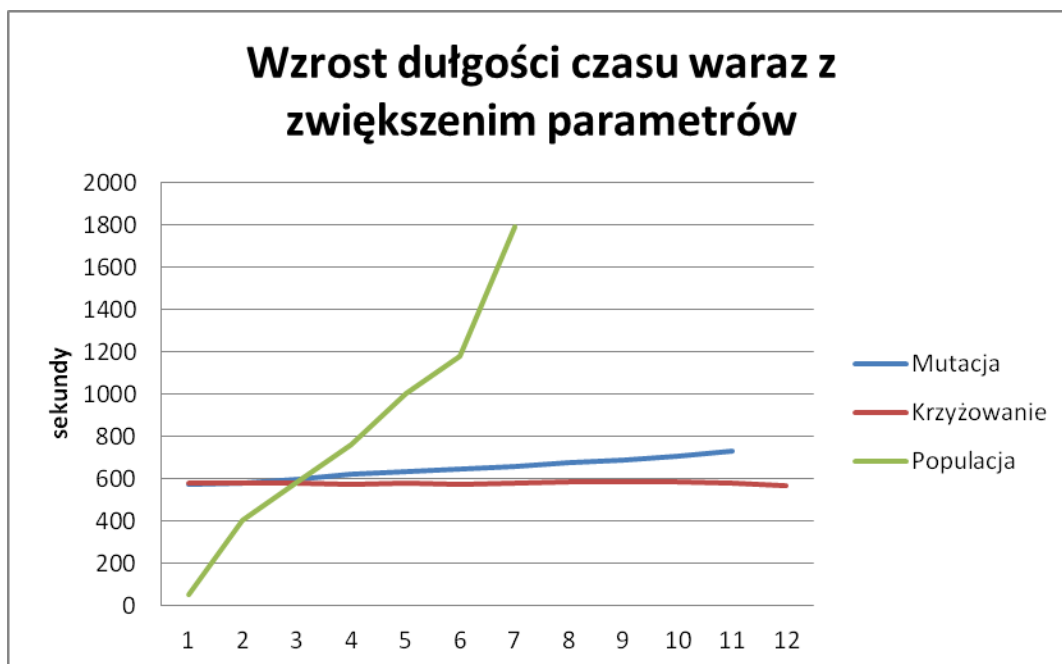
Figure 7 Tabel z summarycznymi wartościami dla strojenia rozmiaru populacji

Analiza doświadczeń

Zmieniając prawdopodobieństwo krzyżowania uzyskano najmniejsze zmiany poprawy jakości rozwiązań. Jest to zaskakujący efekt biorąc pod uwagę że literatura wskazuje że powinno mieć ono znaczny wpływ na działanie algorytmu.

Zmieniając prawdopodobieństwo mutacji w uzyskano najlepszą jakość rozwiązań stosując parametr o wartości 15 co jest równoznacznie z prawdopodobieństwem zajścia mutacji równą 0.15.

Najlepszą poprawę jakości parametrów uzyskano zwiększając parametr wielkości populacji. Im większa populacja tym lepszy wynik dawał algorytm. Problem z zwiększaniem tego parametru jest bardzo szybki czas potrzebny na obliczenia gdyż drastycznie zwiększa się ilość operacji w jednej iteracji pętli głównej. Zależność tego czasu pokazuje poniższy wykres.



Najlepszym wynikiem osiągniętym podczas strojenia parametrów był wynik:

Genotype1=[45, 59, 84, 110, 109, 174, 75, 85, 81, 94, 91, 78, 82, 101, 168, 100, 165, 158, 172, 169, 187, 181, 179, 182, 200, 270, 271, 274, 273, 259, 275, 276, 8, 249, 243, 244, 239, 246, 207, 211, 204, 216, 226, 228, 227, 212, 215, 203, 266, 23, 129, 154, 120, 177, 178, 159, 157, 99, 98, 96, 76, 77, 111, 83, 73, 70, 72, 103, 113, 60, 62, 43, 47, 40, 32, 31, 28, 20, 127, 152, 262, 261, 260, 258, 3, 256, 208, 225, 236, 229, 213, 218, 220, 222, 217, 197, 191, 189, 193, 190, 184, 180, 194, 195, 192, 198, 142, 143, 199, 135, 134, 133, 269, 16, 17, 36, 37, 34, 39, 48, 50, 46, 61, 115, 114, 42, 66, 44, 58, 33, 26, 124, 119, 121, 41, 138, 148, 145, 139, 137, 201, 224, 235, 233, 232, 238, 231, 230, 245, 250, 248, 257, 247, 242, 241, 237, 240, 234, 223, 196, 202, 214, 209, 251, 264, 265, 268, 132, 126, 125, 123, 38, 29, 153, 12, 13, 14, 272, 9, 5, 278, 2, 277, 279, 1, 0, 6, 10, 7, 11, 15, 21, 19, 128, 130, 131, 25, 24, 175, 176, 156, 173, 106, 89, 97, 90, 108, 79, 74, 86, 88, 163, 186, 183, 161, 92, 95, 87, 118, 150, 149, 155, 122, 27, 30, 35, 49, 57, 71, 117, 116, 112, 65, 69, 56, 51, 53, 55, 63, 64, 80, 102, 160, 170, 167, 166, 164, 171, 105, 104, 93, 107, 162, 188, 185, 144, 140, 253, 255, 4, 254, 252, 210, 219, 206, 205, 263, 136, 146, 141, 147, 267, 22, 18, 151, 54, 68, 67, 52, 46]

evaluation=8503

Genotype2=[45, 43, 42, 31, 25, 19, 22, 17, 15, 259, 0, 279, 2, 242, 243, 245, 228, 255, 254, 209, 248, 251, 219, 220, 223, 232, 234, 230, 249, 258, 257, 260, 263, 136, 16, 23, 128, 126, 20, 18, 269, 139, 262, 275, 4, 10, 12, 14, 273, 8, 7, 5, 247, 244, 235, 237, 236, 246, 253, 266, 267, 11, 270, 135, 152, 153, 59, 161, 179, 174, 113, 89, 169, 90, 79, 91, 78, 97, 73, 71, 85, 82, 111, 105, 167, 166, 74, 95, 102, 168, 100, 94, 96, 98, 170, 165, 164, 187, 163, 160, 172, 106, 110, 57, 66, 70, 52, 35, 30, 125, 204, 144, 183, 149, 180, 155, 118, 120, 60, 117, 65, 67, 40, 39, 38, 124, 123, 36, 50, 48, 63, 62, 115, 114, 83, 86, 75, 108, 103, 104, 107, 87, 64, 76, 93, 81, 109, 92, 99, 72, 69, 68, 51, 49, 56, 55, 44, 37, 27, 132, 268, 256, 252, 213, 217, 203, 197, 175, 156, 121, 21, 134, 131, 24, 26, 32, 34, 53, 33, 28, 127, 129, 29, 119, 150, 151, 176, 159, 185, 162, 173, 178, 196, 198, 146, 147, 265, 130, 154, 122, 158, 186, 189, 190, 137, 138, 142, 191, 195, 218, 225, 205, 215, 210, 212, 200, 201, 194, 193, 140, 148, 264, 133, 261, 271, 272, 13, 274, 3, 276, 1, 6, 9, 177, 199, 143, 141, 145, 216, 221, 231, 238, 240, 239, 278, 277, 250, 241, 229, 206, 227, 224, 233, 226, 222, 214, 208, 207, 211, 202, 192, 188, 184, 181, 157, 182, 101, 171, 112, 84, 88, 80, 77, 116, 61, 58, 41, 54, 47, 46]

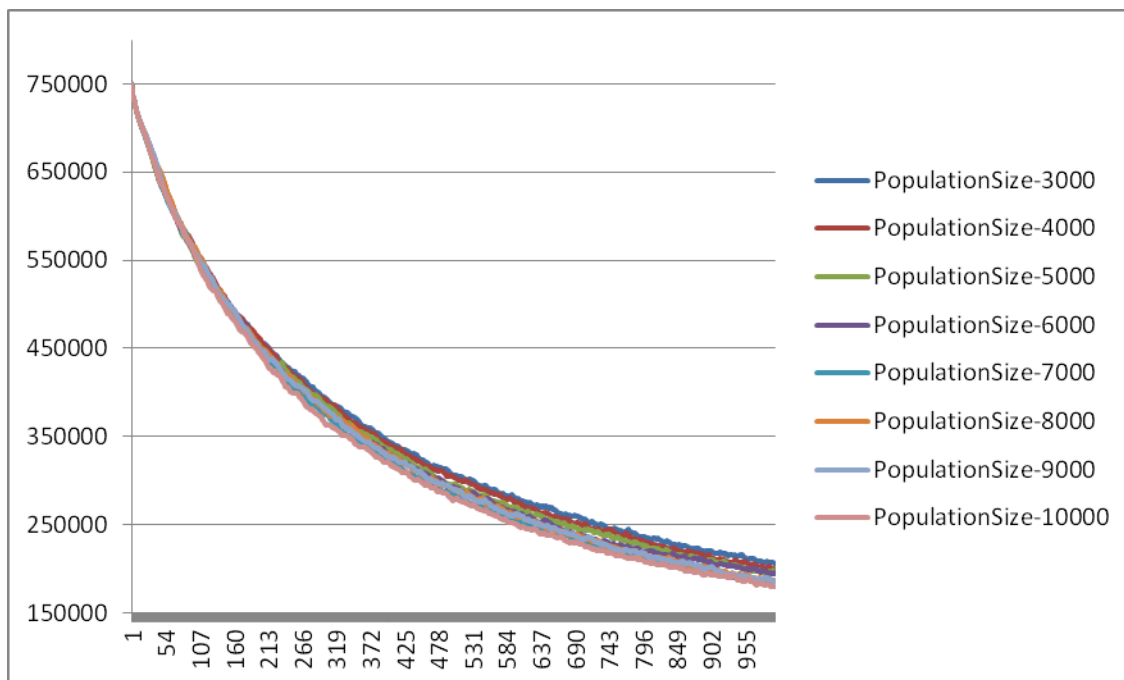
evaluation=8535

Po uzyskaniu najlepszego doboru parametrów czyli:

15 prawd. Mutacji

50 prawd. Krzyżowania

Wykonano jeszcze jedno doświadczenie zwiększając wielkość populacji w celu osiągnięcia jak najlepszego wyniku. Poniżej wyniki ostatniego doświadczenia.



#	Populatio	Populatio	Populatio	Populatio	Populatio	Populatio	Populatio	Populatio
meanTime	1881	2534	3198	3853	4606	5300	5908	6475
meansOfE	18346	17755	17276	17255	16246	16393	16481	15964
meansOfV	72932	73067	73342	73116	73230	73288	73441	73249

Oraz najlepszy wniosek uzyskany tą metodą:

Genotype1=[13, 23, 270, 16, 130, 21, 10, 259, 214, 222, 221, 216, 218, 223, 238, 236, 232, 2, 3, 7, 12, 126, 20, 26, 124, 119, 152, 125, 51, 49, 48, 50, 52, 34, 59, 40, 41, 45, 53, 47, 56, 57, 114, 112, 89, 76, 75, 79, 84, 83, 86, 92, 98, 168, 101, 166, 170, 164, 161, 163, 173, 113, 87, 111, 108, 169, 167, 187, 190, 189, 184, 162, 174, 156, 157, 144, 195, 186, 176, 180, 181, 175, 149, 137, 205, 207, 229, 230, 277, 278, 256, 0, 228, 235, 227, 206, 199, 192, 191, 145, 147, 141, 263, 264, 140, 136, 134, 269, 268, 19, 129, 127, 32, 121, 122, 123, 30, 36, 35, 33, 28, 38, 37, 46, 117, 62, 104, 99, 97, 96, 95, 74, 94, 77, 80, 78, 90, 107, 105, 106, 103, 172, 63, 67, 69, 72, 85, 116, 61, 43, 44, 54, 55, 42, 118, 153, 159, 182, 158, 4, 5, 275, 276, 6, 1, 247, 246, 241, 244, 239, 243, 249, 250, 240, 242, 279, 274, 273, 11, 8, 260, 257, 255, 237, 210, 251, 208, 209, 211, 219, 226, 233, 234, 231, 254, 253, 261, 271, 15, 258, 262, 135, 133, 128, 120, 150, 177, 160, 171, 100, 91, 93, 81, 65, 109, 115, 58, 39, 31, 131, 22, 14, 132, 154, 29, 27, 25, 24, 18, 17, 139, 155, 110, 60, 64, 70, 68, 66, 71, 73, 88, 102, 165, 188, 185, 200, 196, 138, 148, 179, 183, 178, 151, 265, 266, 203, 194, 220, 224, 213, 217, 225, 245, 248, 252, 267, 142, 204, 212, 215, 143, 146, 197, 193, 198, 201, 202, 272, 9, 10]

evaluation=7898

Genotype2=[13, 270, 12, 14, 180, 110, 60, 117, 61, 58, 64, 85, 86, 84, 41, 39, 46, 53, 52, 38, 36, 42, 57, 62, 70, 83, 114, 80, 88, 103, 95, 92, 169, 111, 66, 69, 72, 109, 79, 74, 76, 77, 78, 91, 81, 174, 162, 172, 106, 107, 113, 112, 116, 67, 56, 63, 55, 35, 51, 48, 33, 28, 126, 29, 27, 22, 23, 9, 6, 274, 5, 4, 275, 7, 8, 271, 272, 258, 260, 261, 268, 265, 204, 206, 252, 251, 207, 211, 214, 208, 255, 247, 239, 237, 250, 234, 223, 218, 198, 142, 18, 155, 159, 182, 186, 185, 167, 99, 101, 166, 170, 171, 89, 102, 90, 82, 108, 160, 158, 144, 148, 137, 149, 150, 151, 176, 179, 178, 177, 118, 152, 120, 121, 125, 128, 26, 32, 37, 49, 50, 47, 44, 45, 68, 65, 115, 87, 71, 73, 75, 96, 93, 94, 97, 98, 165, 163, 168, 104, 105, 100, 173, 175, 181, 184, 188,

192, 196, 197, 200, 210, 209, 254, 256, 2, 277, 278, 1, 241, 246, 243, 244, 236, 226, 203, 202, 143, 190, 187, 164, 161,
157, 156, 119, 59, 43, 54, 31, 21, 24, 19, 131, 132, 154, 183, 189, 145, 140, 147, 136, 138, 146, 141, 262, 264, 139, 135,
127, 129, 123, 34, 40, 30, 124, 25, 153, 122, 20, 17, 130, 267, 266, 134, 133, 16, 273, 248, 245, 229, 249, 242, 279, 0, 3,
276, 240, 231, 232, 233, 224, 225, 235, 230, 238, 228, 227, 212, 213, 215, 221, 220, 219, 222, 217, 199, 201, 195, 194,
193, 191, 216, 205, 253, 257, 259, 263, 269, 15, 11, 10]

evaluation=7143