Implementacja i analiza efektywności algorytmu genetycznego.

Zadanie projektowe nr 3

Prowadzący: dr inż. Jarosław Mierzwa

Grupa: Piątek 15.15-16.55

Autor: Izabela Pałubicka 252771

1. Wstęp teoretyczny.

1.1. Algorytm genetyczny

Algorytm ten opiera swe działanie na biologicznym procesie ewolucji. Osobniki pewnej populacji krzyżując ze sobą swoje chromosomy tworzą potomstwo, którym przekazują swoje cechy. O tym czy mają szanse przetrwać w danym środowisku informuje ich przystosowanie. Dla problemu komiwojażera osobnikami są ścieżki, chromosomami są miasta, a przystosowaniem jest koszt drogi. Algorytm genetyczny jest to algorytm heurystyczny co oznacza, że nie daje on gwarancji odnalezienia najlepszego rozwiązania dla danego problemu.

Aby skonstruować algorytm genetyczny dla problemu komiwojażera potrzebne są elementy takie jak:

- podstawowa reprezentacja potencjalnych rozwiązań zadania, których ilością jest początkowa liczebność populacji, a jej ścieżki generowane są w sposób losowy.
- funkcję oceniającą, która gra rolę środowiska i ocenia rozwiązania według ich dopasowania, czyli oblicza koszt ścieżek - im mniejszy koszt tym ścieżka jest lepiej przystosowana
- operatory genetyczne (metoda selekcji, operator krzyżowania i mutacji)
- parametry: rozmiar populacji, współczynnik krzyżowania i współczynnik mutacji
- warunków zakończenia działania (warunków stopu)

1.2. Selekcja

Selekcje polega na wyborze z bieżącej populacji osobników, potencjalnych rodziców, których materiał genetyczny zostanie poddany z pewnym prawdopodobieństwem operacji krzyżowania oraz mutacji, a następnie zostanie przekazany osobnikom potomnym. Zastosowana została turniejowa metoda selekcji, która pobiera losowo osobników, których ilość zależna jest od rozmiaru populacji. Powstaje grupa, z której pobierany jest zwycięzca, osobnik o najmniejszym koszcie drogi i dodawany jest do grupy potencjalnych rodziców.

1.3. Krzyżowanie

Krzyżowanie polega na wymianie materiału genetycznego pomiędzy dwoma losowo wybranymi osobnikami. W wyniku krzyżowania powstają nowe osobniki, które mogą wejść w skład nowej populacji.

Metoda OX – Order Crossover:

- Losowanie z grupy rodziców osobników do krzyżowania:

Rodzic1 = [1 2 3 4 5 6 7 8 9]
Rodzic2 = [4 5 2 1 8 7 6 9 3]

- Wylosowanie dwóch punktów tworzących przedział:

Rodzic1 = [1 2 3 4 5 6 7 8 9]
Rodzic2 = [4 5 2 1 8 7 6 9 3]

- Dodanie wyznaczonych przedziałów do potomków w odpowiednich miejscach:

Potomek1 = [x x x 4 5 6 7 x x]
Potomek2 = [x x x 1 8 7 6 x x]

- Uzupełnienie potomków o wyznaczone elementy zaczynając od drugiego cięcia u drugiego rodzica, w taki sposób, aby nie doszło do konfliktów (powtórzenia miast, czyli zamknięcia cyklu):

Potomek1 = [2 1 8 4 5 6 7 9 3]
Potomek2 = [3 4 5 1 8 7 6 9 2]

1.4. Mutacja

Mutacja polega na zamianie wartości losowo wybranego genu. Jej celem jest zapewnienie zmienności chromosomów. Zachodzi z prawdopodobieństwem $p_m \in (0.01,\ 0.1)$. Przykładowe metody mutacji:

- Transposition – wzajemna wymiana, zamienione zostają miejscami dwa losowo wybrane miasta:

[1 2 3 4 5 6 7 8 9] => [1 2 7 4 5 6 3 8 9]

- Insertion – wstawianie, przestawiane jest losowo wybrane miasto na losową pozycję, pozostałe miasta są rozsuwane:

[1 2 3 4 5 6 7 8 9] => [1 2 6 4 5 3 8 9]

2. Opis najważniejszych klas w projekcie

Klasa DataMatrix

Jest to klasa, która przechowuje dane wczytane z pliku, a także ustawione parametry algorytmu (kryterium stopu, rozmiar populacji początkowej, współczynnik mutacji, współczynnik krzyżowania i rodzaj mutacji). Posiada metodę, która wczytuje dane z pliku, a także wyświetla wczytane dane.

Klasa GeneticAlgorithm

Jest to najważniejsza klasa. To ona posiada metodę runAlgorithm(), która wykonuje algorytm genetyczny. Klasa ta posiada strukturę Specimen, która przechowuje wektor miast, a także koszt drogi. Klasa posiada również metody, które są wywoływane przez funkcję runAlgorithm(). Są to metody

- generateInitialPopulation() - generującą populację początkową.

- adaptationAssessment() – liczącą koszt drogi dla podanego wektora.
- selection() – wybierającą za pomocą metody turniejowej potencjalnych rodziców.
- crossing() - zawierającą metodę OX.
- mutating() – zawierającą dwie metody mutacji Transposition i Insertion.
- generateProbability() – generującą prawdopodobieństwo z jakim dojdzie do krzyżowani i mutacji.

Klasa Timer

Jest odpowiedzialna za pomiar czasu wewnątrz działania algorytmu genetycznego dla sprawdzenia kryterium stopu, dzięki czemu algorytm może zostać przerwany. Korzysta z biblioteki chrono, a mierzony czas podawany jest w sekundach. Do pomiaru upływu czasu wykorzystana została metoda startTimer(), stopTimer(), a także timeCounter(), która zwraca różnicę czasu między początkiem a końcem odmierzania czasu.

3. Wyniki

3.1. Pomiary błędu względnego w funkcji czasu

Błąd względny został policzony ze wzoru:

$$\delta = \frac{|f_{zn} - f_{opt}|}{f_{opt}}$$

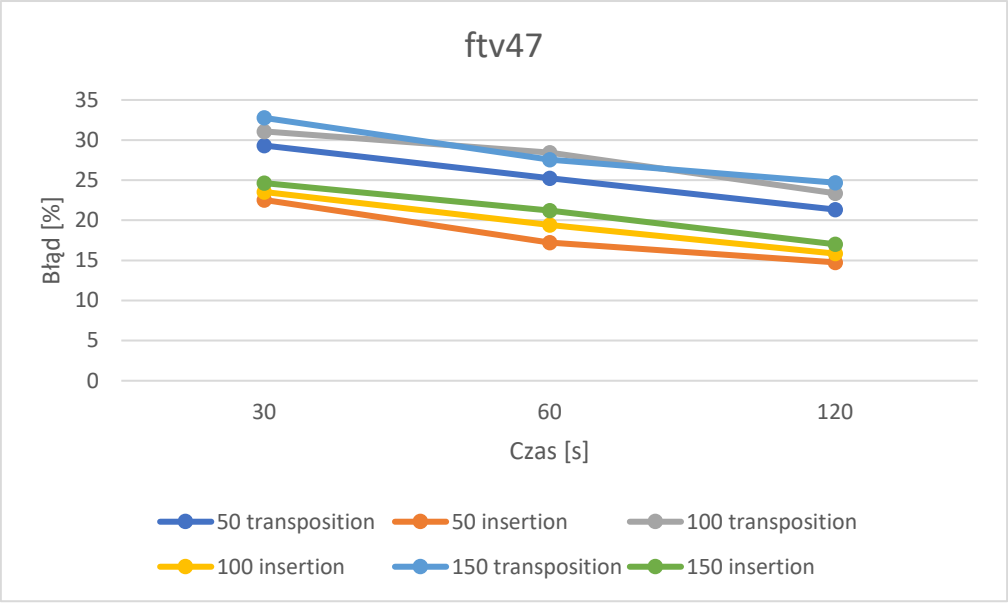
gdzie f_{zn} – wartość policzona przez algorytm, f_{opt} – najlepsze znane rozwiązanie

Badanie zostały 3 wielkości populacji: 50, 100 i 150, dla 3 warunków stopu: 30s, 60s i 120s dla dwóch rodzajów mutacji transposition i insertion.

- współczynnik błędu względnego dla pliku ftv48 dla dwóch badanych mutacji:

		Czas [s]		
		30	60	120
Populacja	Mutacja	Błąd[%]		
50	transposition	29,32	25,24	21,34
50	insertion	22,54	17,23	14,76
100	transposition	31,08	28,42	23,37
100	insertion	23,53	19,42	15,85
150	transposition	32,77	27,56	24,69
150	insertion	24,64	21,23	17,01

Tabela 1: Współczynnik błędu względnego w funkcji czasu działania algorytmu dla ftv47.atsp

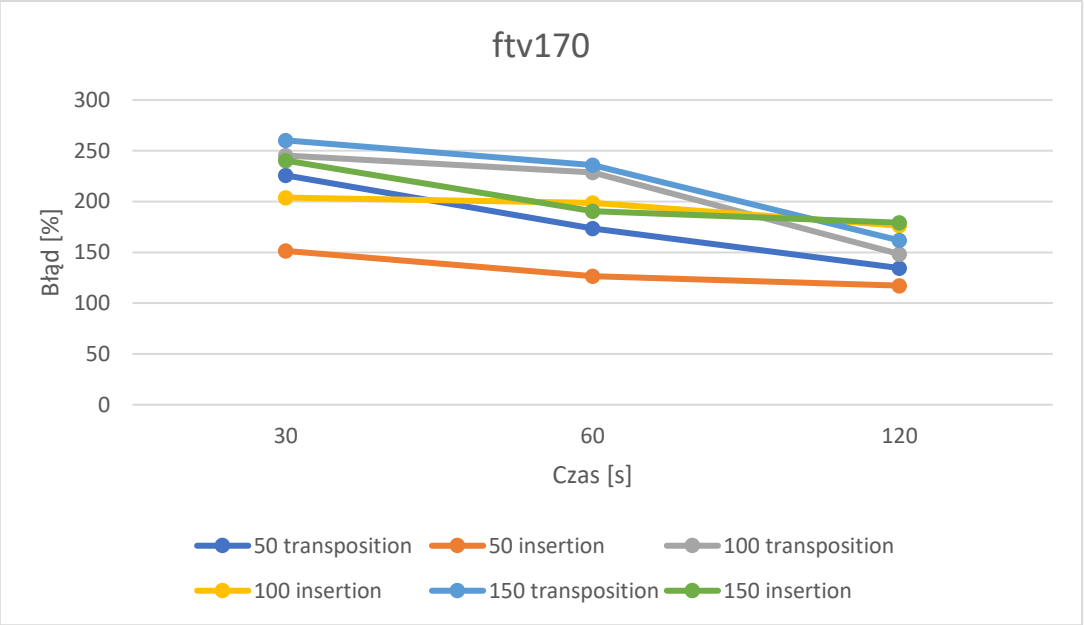


Wykres 1: Współczynnik błędu w funkcji czasu dla uśrednionych wyników dla pliku ftv47.atsp i dla wszystkich 6 konfiguracji.

- współczynnik błędu względnego dla pliku ftv170 dla dwóch badanych mutacji:

		Czas [s]		
		30	60	120
Populacja	Mutacja	Błąd[%]		
50	transposition	225,75	173,53	134,54
50	insertion	151,38	126,46	117,24
100	transposition	245,32	228,54	148,32
100	insertion	203,88	198,66	176,23
150	transposition	260,21	235,75	161,72
150	insertion	240,33	190,49	179,20

Tabela 2: Współczynnik błędu względnego w funkcji czasu działania algorytmu dla ftv170.atsp

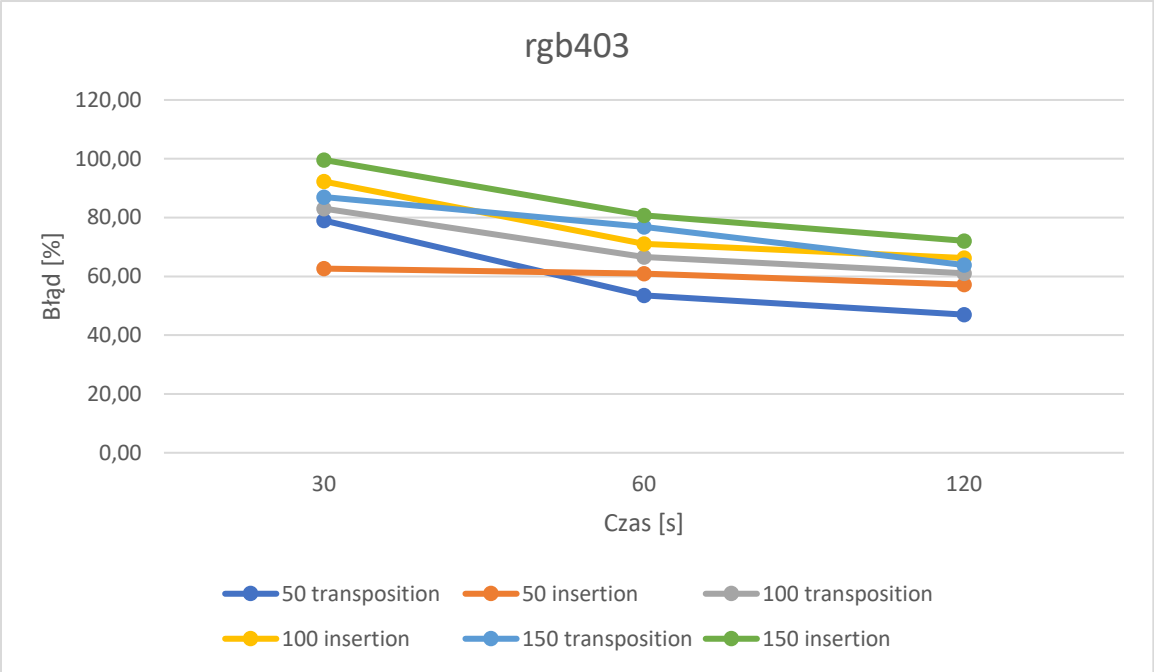


Wykres 2: Wykres błędu w funkcji czasu dla uśrednionych wyników dla pliku ftv170.atsp i dla wszystkich sześciu konfiguracji.

- współczynnik błędu względnego dla pliku rgb403 dla dwóch badanych mutacji:

		Czas [s]		
		30	60	120
Populacja	Mutacja	Błąd[%]		
50	transposition	78,99	53,49	46,99
50	insertion	62,64	60,93	57,20
100	transposition	83,04	66,57	61,05
100	insertion	92,25	71,08	66,25
150	transposition	86,99	76,88	63,85
150	insertion	99,60	80,77	72,09

Tabela 3: Współczynnik błędu względnego w funkcji czasu działania algorytmu dla rgb403.atsp



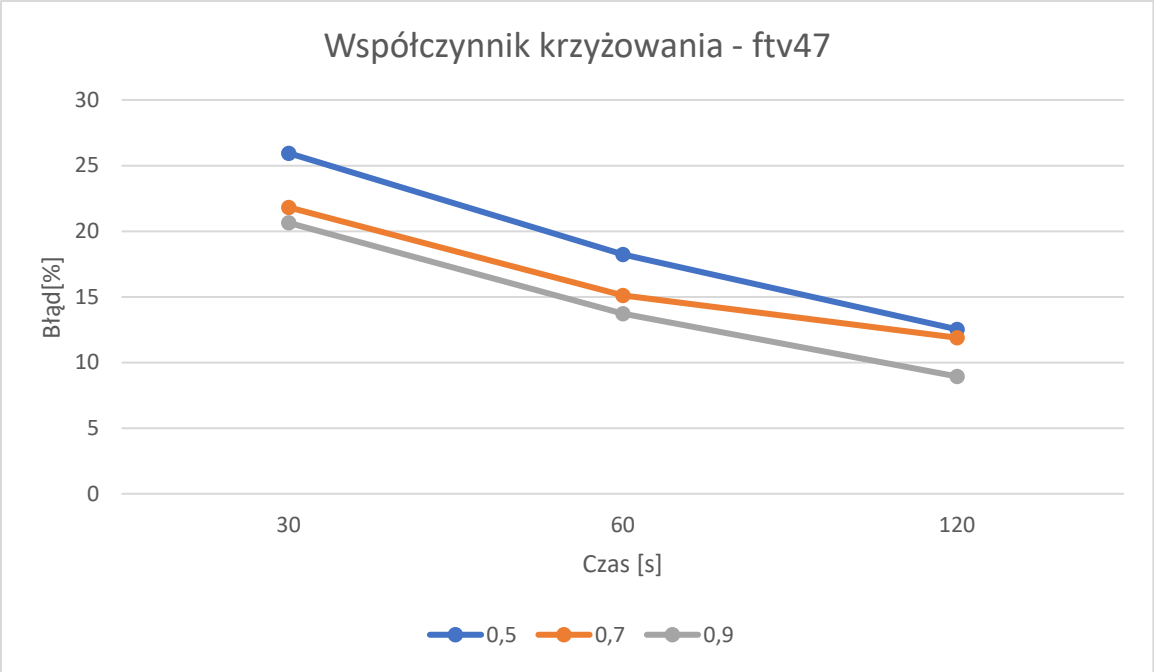
Wykres 3: Wykres błędu w funkcji czasu dla uśrednionych wyników dla pliku rgb403.atsp i dla wszystkich sześciu konfiguracji.

3.2. Wyniki dla współczynników krzyżowania dla najlepszych populacji i mutacji dla danego problemu

- Plik ftv47: insertion 50

	Czas [s]		
	30	60	120
Krzyżowanie	Błąd[%]		
0,5	25,95	18,24	12,53
0,7	21,82	15,12	11,89
0,9	20,64	13,73	8,94

Tabela 4: Współczynnik błędu względnego w funkcji czasu działania algorytmu dla ftv47.atsp dla najlepszej konfiguracji: populacja 50, mutacja insertion.

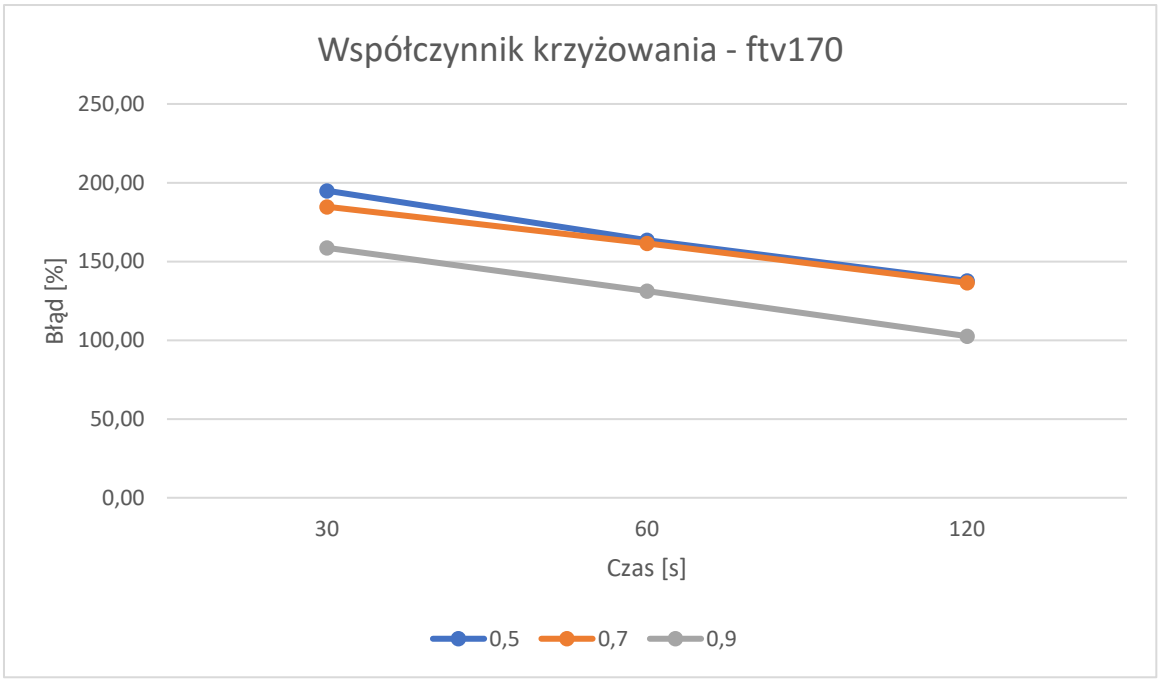


Wykres 4: Wykres błędu w funkcji czasu dla uśrednionych wyników dla pliku ftv47.atsp i dla trzech współczynników mutacji 0.5, 0.7 i 0.9.

- Plik ftv170: insertion 50

	Czas [s]		
	30	60	120
Krzyżowanie	Błąd[%]		
0,5	194,92	163,59	137,71
0,7	184,75	161,45	136,44
0,9	158,66	131,32	102,61

Tabela 5: Współczynnik błędu względnego w funkcji czasu działania algorytmu dla ftv47.atsp dla najlepszej konfiguracji: populacja 50, mutacja insertion.

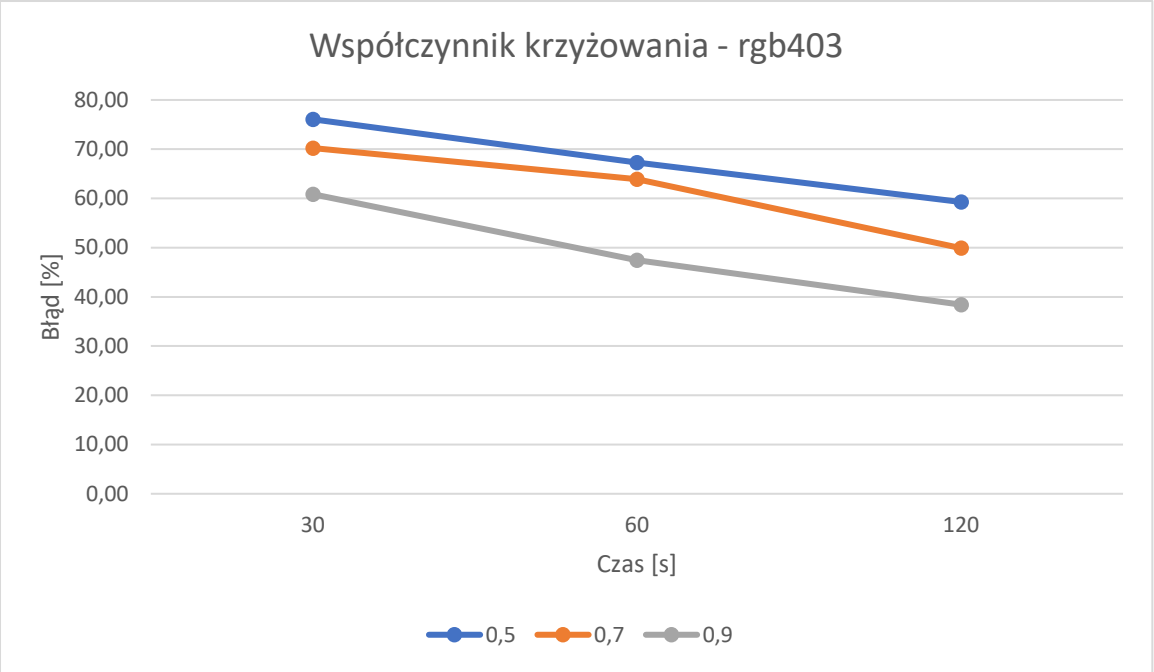


Wykres 5: Wykres błędu w funkcji czasu dla uśrednionych wyników dla pliku ftv170.atsp i dla trzech współczynników mutacji 0.5, 0.7 i 0.9.

- Plik rgb403: transposition 50

	Czas [s]		
	30	60	120
Krzyżowanie	Błąd[%]		
0,5	76,06	67,30	59,27
0,7	70,22	63,87	49,90
0,9	60,85	47,42	38,02

Tabela 6: Współczynnik błędu względnego w funkcji czasu działania algorytmu dla rgb403.atsp dla najlepszej konfiguracji: populacja 50, mutacja transposition.



Wykres 4: Wykres błędu w funkcji czasu dla uśrednionych wyników dla pliku rgb403.atsp i dla trzech współczynników mutacji 0.5, 0.7 i 0.9.

3.3. Najlepsze znalezione wyniki dla danych problemów

- **Plik ftv47:**

Koszt: 1972

Droga:

0->25->1->11->8->32->7->23->34->13->46->36->14->35->15->16->45->39->21->40->20->38->18->17->12->9->33->27->2->41->43->22->47->26->42->28->3->24->4->29->30->31->5->6->10->37->19->44->0

- **Plik ftv170:**

Koszt: 5392

Droga:

0->127->126->125->124->132->133->169->81->79->77->2->1->80->82->78->72->73->168->71->60->49->39->41->155->156->34->157->33->40->54->58->59->50->51->52->53->43->55->42->47->35->36->37->38->45->44->46->48->170->3->4->8->14->151->142->141->164->138->135->129->137->146->145->149->148->144->143->147->136->128->139->140->13->21->20->10->76->74->75->11->12->18->19->24->16->158->32->31->30->25->160->152->15->159->17->22->26->27->28->29->23->150->161->131->115->104->116->117->102->114->108->83->84->5->110->107->106->105->97->113->130->134->6->7->9->111->112->98->96->90->94->99->162->123->101->100->103->109->166->92->89->88->70->69->67->65->57->62->61->66->63->56->64->68->167->85->86->93->91->154->153->87->118->119->120->122->121->163->95->165->0

- **Plik rgb403:**

Koszt: 3365

Droga:

0->214->177->50->91->69->285->282->395->133->312->65->49->37->82->320->268->28->385->80->24->140->114->178->325->124->161->148->390->153->7->116->388->348->237->130->186->170->339->244->296->318->367->323->151->75->171->33->376->88->366->220->117->221->175->283->126->55->125->60->44->90->71->336->292->308->26->110->162->222->81->200->57->99->233->380->42->266->301->208->142->199->150->235->85->394->201->89->340->251->121->167->23->155->284->361->159->76->101->297->304->260->248->279->311->10->141->38->19->253->185->139->4->206->92->378->198->58->184->179->334->245->241->324->144->240->262->342->18->217->384->265->106->369->78->236->227->379->207->191->105->249->36->183->17->12->286->215->29->160->25->363->362->97->387->203->166->372->271->313->143->383->389->187->373->355->218->300->14->62->54->209->41->165->333->216->331->368->232->254->127->8->344->77->275->239->104->84->46->109->326->343->180->190->138->345->346->335->27->353->39->32->288->13->291->315->234->103->35->102->276->152->321->347->263->264->293->176->100->290->252->399->172->298->74->52->238->146->147->136->317->246->70->9->21->188->118->59->230->113->83->93->197->15->181->20->11->79->96->189->132->212->134->306->354->22->278->122->242->67->169->119->6->163->365->374->225->111->359->314->72->398->53->400->164->45->310->73->86->48->2->61->370->274->375->371->327->316->256->289->397->5->56->229->194->270->382->381->195->258->307->192->338->157->287->156->210->231->219->98->158->1->135->226->31->305->223->196->30->68->193->173->309->211->112->47->280->34->295->87->16->328->350->108->247->63->224->64->396->386->322->107->330->128->228->377->364->303->337->360->393->402->401->319->182->243->351->257->202->115->94->302->261->66->273->281->213->356->391->154->123->277->341->174->250->145->40->352->129->120->392->255->149->51->3->358->205->204->95->269->267->357->137->329->294->272->332->299->259->43->349->168->131->0;

3.4. Porównanie wyników z wynikami algorytmu Symulowanego Wyżarzania:

	ftv47	ftv170	rgb403
Najlepsze znane rozwiązanie	1776	2755	2465
Algorytm genetyczny	1972	5392	3365
Symulowane Wyżarzanie	1920	3972	2699

4. Wnioski

Wynik dla problemu ftv47 dla algorytmu generycznego był niewiele gorszy od wyniku uzyskanego przez algorytm SW. Najgorzej algorytm poradził sobie z problemem ftv170. Świadczy to o tym, że wyniki dla tego algorytm nie są całkowicie zależne od wielkości problemu.

Na wykresach z punktu 3.1. widać, że najlepszą populacją okazała się populacja o wielkości 50 i dla co raz większych populacji wyniki były gorsze. Można również zauważyć, że im dłużej wykonuje się algorytm tym osiąga lepsze wyniki. Różne metody mutacji także inaczej radzą sobie z problemami. Dla problemów ftv47 i ftv170 lepiej poradziła sobie metoda mutacji insertion, a dla problemu rgb403 metoda transposition.

Analiza wyników dla współczynnika krzyżowania z punktu 3.2. wykazała, że wyniki są zdecydowanie lepsze dla współczynnika krzyżowania o wartości 0.9. Algorytm ma wtedy większą szanse na dokonanie modyfikacji rozwiązań, dzięki czemu może znaleźć jeszcze lepsze rozwiązanie.