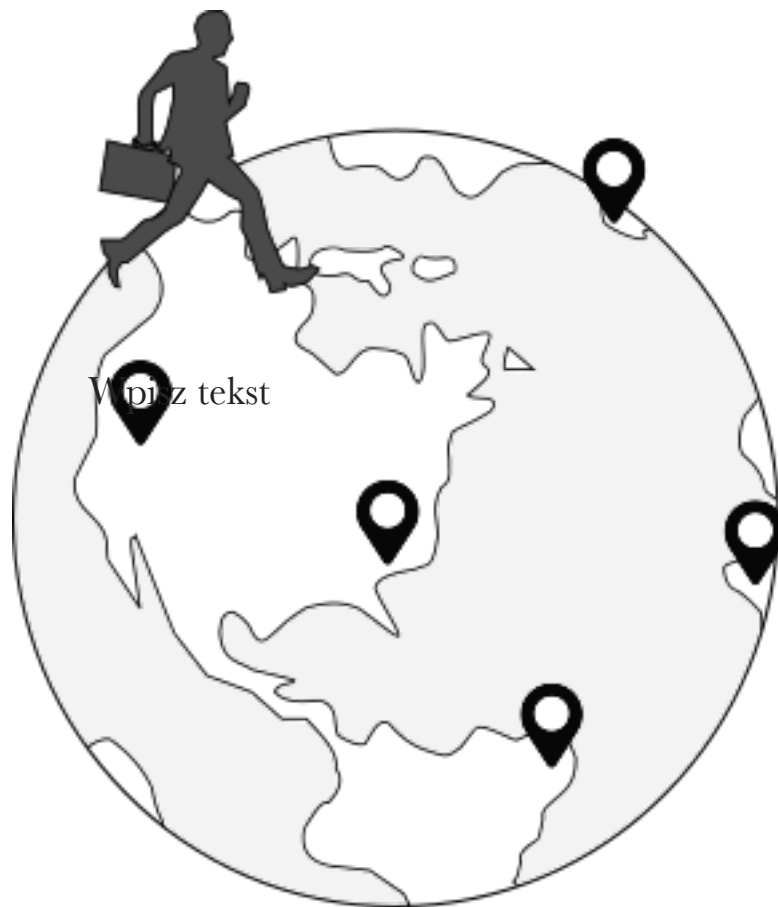


Problem Komiwożera

Projektowanie Efektywnych Algorytmów

Algorytmy genetyczne



Karol Kłusek

26 luty 2019

Wstęp

Celem projektu było wykonanie programu, wykorzystującego algorytm Genetyczny (ang. Tabu Search) dla problemu komiwojażera. Wskazaniem było, aby algorytm pozwalał na modyfikację swojego działania w zależności od wprowadzonych parametrów.

Specyfikacja Techniczna

- Struktury przechowujące dane alokowane są dynamicznie, zależnie od rozmiaru problemu.
- Program posiada możliwość wczytywania danych z pliku, w celu weryfikacji poprawności.
- Algorytm zostały zaimplementowany zgodnie z paradygmatami programowania obiektowego.
- Czas wykonania algorytmu mierzony był z dokładnością do nanosekund, przy wykorzystaniu bibliotek systemowych.

Założenia projektowe

Dla wykonania algorytmu genetycznego:

- Populacje generowane po każdym krzyżowaniu wyłaniają najlepsze osobniki,
- 2 metody krzyżowania i 2 metody mutacji,
- Mutacje realizowane przez definicję sąsiedztwa swap i insert.

Wstęp teoretyczny

Problem komiwojażera (ang. travelling salesman problem, w skrócie TSP) – problem obliczeniowy polegający na poszukiwaniu w grafie takiego cyklu, który zawiera wszystkie wierzchołki (każdy dokładnie raz) i ma jak najmniejszy koszt. Bardziej formalnie, problem komiwojażera polega na poszukiwaniu w grafie cyklu Hamiltona o najmniejszej wadze. Problem ma liczne zastosowania w życiu codziennym. Najlepszym przykładem jest praca kuriera, który musi wyjechać z magazynu, zawieźć przesyłki w różne miejsca i wrócić do magazynu.

Nie jest znany efektywny (tj. działający w czasie co najwyżej wielomianowym) algorytm dający gwarancję znalezienia optymalnego rozwiązania problemu komiwojażera. Problem ten jest bowiem zaliczany do klasy problemów NP-trudnych. W wersji decyzyjnej (czy istnieje cykl o długości mniejszej od x) problem jest zaliczany do klasy problemów NP-zupełnych. W grafie pełnym mającym n wierzchołków liczba możliwych cykli Hamiltona wynosi aż $(n-1)!/2$. W praktyce sprawdzenie wszystkich możliwości jest zatem wykonalne tylko dla niewielkiej liczby wierzchołków.

Kolejne ścieżki wybierane z grafu mogą być swoimi sąsiadami. Definicja sąsiedztwa jest umową w jaki sposób tworzy się kolejne ścieżki na podstawie ścieżki podstawowej. Najpopularniejszym przykładem sąsiedztwa ścieżek jest sąsiedztwo typu **swap**. Polega ono na zamianie dwóch wierzchołków w grafie miejscami. Przykład poniżej

$$r := 0-5-4-6-2-3-1-0$$

$$r' := 0-1-4-6-2-3-5-0$$

Kolejnym typem jest sąsiedztwo **insert**, definiowane jako wstawienie dowolnego wierzchołka między dwa inne w ścieżce. Przykład:

$$r := 0-5-4-6-2-3-1-0$$

$$r' := 0-5-1-4-6-2-3-0$$

Ostatnim jest sąsiedztwo typu **inverse** zdefiniowane jako odwrócenie kolejności wierzchołków w ścieżce. przykład:

$$r := 0-5-4-6-2-3-1-0$$

$$r' := 0-5-3-2-6-4-1-0$$

Cechy dobrego sąsiedztwa

Aby sąsiedztwo zostało uznane za dobre, musi spełniać następujące cechy:

- zawiera co najmniej jedno rozwiązanie i nie obejmuje całej przestrzeni rozwiązań,
- rozwiązania różnią się niewiele oddanego rozwiązania,
- niezależnie od początkowego rozwiązania, powinno być osiągalne każde rozwiązanie z przestrzeni rozwiązań algorytmu.

Algorytm genetyczny

Problem definiuje środowisko, w którym istnieje pewna populacja osobników. Każdy z osobników ma przypisany pewien zbiór informacji stanowiących jego genotyp, a będących podstawą do utworzenia fenotypu. Fenotyp to zbiór cech podlegających ocenie funkcji przystosowania modelującej środowisko. Innymi słowy - genotyp opisuje proponowane rozwiązanie problemu, a funkcja przystosowania ocenia, jak dobre jest to rozwiązanie.

Genotyp składa się z chromosomów, gdzie zakodowany jest fenotyp i ewentualnie pewne informacje pomocnicze dla algorytmu genetycznego. Chromosom składa się z genów.

Wspólnymi cechami algorytmów ewolucyjnych, odróżniającymi je od innych, tradycyjnych metod optymalizacji, są:

- stosowanie operatorów genetycznych, które dostosowane są do postaci rozwiązań,
- przetwarzanie populacji rozwiązań, prowadzące do równoległego przeszukiwania przestrzeni rozwiązań z różnych punktów,
- w celu ukierunkowania procesu przeszukiwania wystarczającą informacją jest jakość aktualnych rozwiązań,
- celowe wprowadzenie elementów losowych.

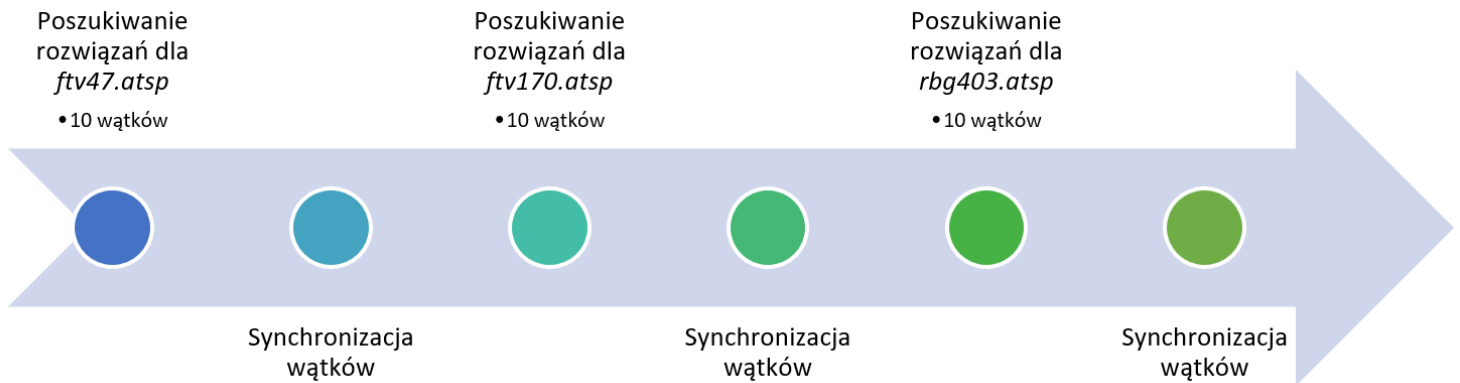
Plan eksperymentu

Pomiary czasu wykonania zadania przez program są obarczone pewnymi niepewnościami. Pierwszą z nich jest prawdopodobieństwo wylosowania szczególnego przypadku danych, dla którego algorytm może być skrajnie nieefektywny lub odwrotnie. Druga to nieprzewidziane zachowanie urządzenia testowego np. zmiana taktowania w trakcie wykonywania eksperymentu. Aby zapobiec tym przekłamaniom będę wykonywał testy dla każdej instancji problemu o zadanym rozmiarze 10-krotnie i wyciągał średnią oraz szukał najlepszego wyniku.

Czas będzie mierzony poprzez funkcję z biblioteki systemowej w Rust'cie nazywaną *PreciseTime::now()*;. Zapewnia ona dokładność do jednej nanosekundy.

Wszystkie wykorzystane struktury w programie są alokowane dynamicznie co pozwala na modyfikację ich rozmiaru w trakcie działania programu. Do reprezentacji grafu wykorzystałem macierz sąsiedztwa.

Wielowątkowość



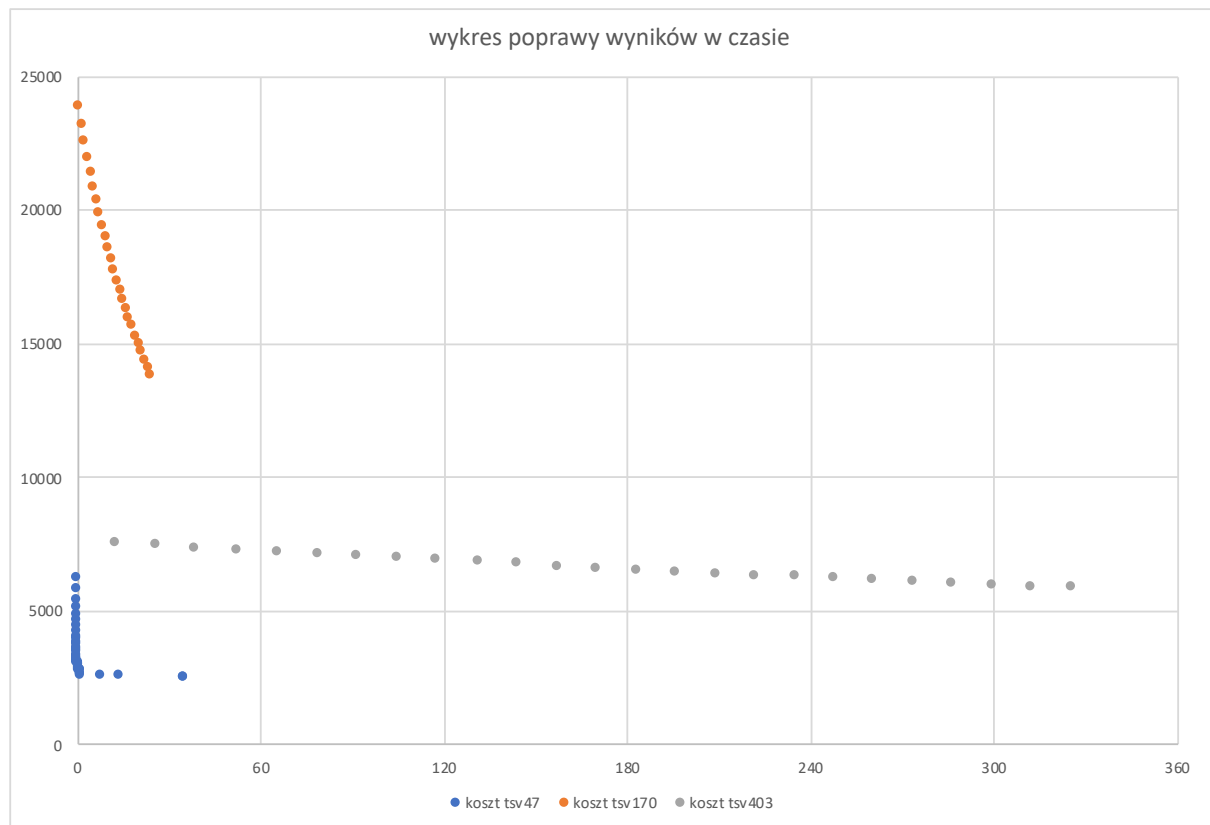
Wyniki eksperymentu

Tabu Search

ftv47.atsp – parametry uruchomienia: limit czasu – 120 s, limit iteracji – 10000, kadencja blokowanej ścieżki – 20 iteracji, ilość błędów krytycznych – 25.

ftv170.atsp – parametry uruchomienia: limit czasu – 240s, limit iteracji – 10000, kadencja blokowanej ścieżki – 20 iteracji, ilość błędów krytycznych – 25.

rbg403.atsp – parametry uruchomienia: limit czasu – 360s, limit iteracji – 10000, kadencja blokowanej ścieżki – 20 iteracji, ilość błędów krytycznych – 25.

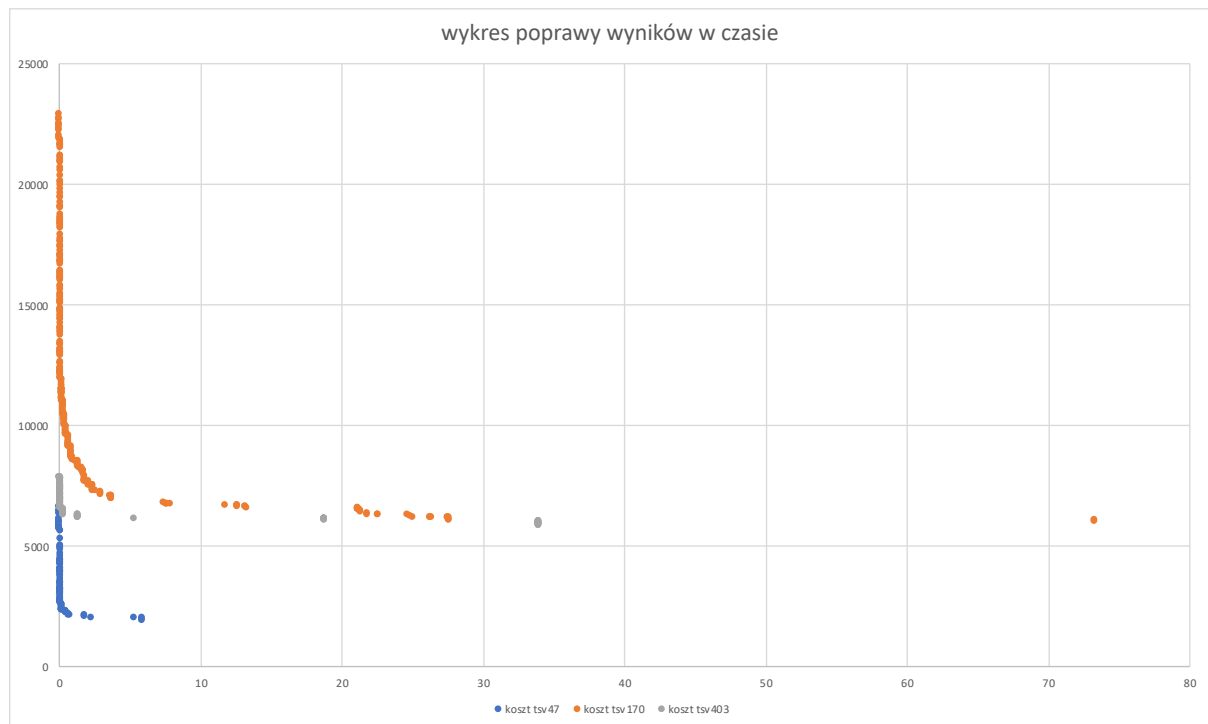


Simulated Annealing

ftv47.atsp – parametry uruchomienia: limit czasu – 120 s,
temperatura początkowa – 20, szybkość wyżarzania – 0.999999.

ftv170.atsp – parametry uruchomienia: limit czasu – 240 s,
temperatura początkowa – 20, szybkość wyżarzania – 0.999999.

rbg403.atsp – parametry uruchomienia: limit czasu – 360 s,
temperatura początkowa – 20, szybkość wyżarzania – 0.999999.



Uzyskane Wyniki

Tabu Search

maksimum iteracji	maksimum błędów	czas zakazu	rozmiar miasta			błąd względny		
			48	168	403	48	168	403
5000	10	10	3309	18061	6554	86%	556%	166%
		20	3298	17827	6636	86%	547%	169%
		40	3255	18638	6624	83%	577%	169%
	25	10	2468	13783	6500	39%	400%	164%
		20	2503	13638	6484	41%	395%	163%
		40	2342	13917	6486	32%	405%	163%
	75	10	2306	7963	6639	30%	189%	169%
		20	2294	8063	6414	29%	193%	160%
		40	2233	8120	6606	26%	195%	168%
10000	10	10	3256	18306	6574	83%	564%	167%
		20	3305	18220	6725	86%	561%	173%
		40	3285	18451	6663	85%	570%	170%
	25	10	2461	13809	6390	39%	401%	159%
		20	2414	13742	6548	36%	399%	166%
		40	2409	13507	6511	36%	390%	164%
	75	10	2371	8122	6614	34%	195%	168%
		20	2311	8074	6543	30%	193%	165%
		40	2386	8084	6474	34%	193%	163%
20000	10	10	3251	17864	6591	83%	548%	167%
		20	3204	18444	6757	80%	569%	174%
		40	3207	18499	6677	81%	571%	171%
	25	10	2423	13374	6576	36%	385%	167%
		20	2400	13392	6485	35%	386%	163%
		40	2489	13749	6553	40%	399%	166%
	75	10	2191	7929	6562	23%	188%	166%
		20	2371	8293	6381	34%	201%	159%
		40	2304	8134	6468	30%	195%	162%

Najlepsze uzyskane rozwiązania:

- dla pliku *ftv47.atsp*: 2191 (błąd względny 23%) – ścieżka
25-37-39-19-44-15-34-35-45-11-8-32-7-23-13-46-36-14-16-21-40-38-18-17-12-6-3-
24-4-29-30-31-5-9-33-27-28-2-41-43-42-22-20-0-47-26-1-10,
- dla pliku *ftv170.atsp*: 7929 (błąd względny 188%) – ścieżka
164-4-6-58-66-29-70-26-72-32-105-126-77-51-28-114-110-10-46-73-151-147-87-1
65-96-13-148-47-16-2-127-125-92-158-18-137-104-22-7-160-150-33-119-107-106-
131-118-109-19-130-93-163-17-139-76-115-100-142-68-53-64-21-145-35-108-132-
123-102-27-113-121-101-30-117-153-88-1-78-15-146-44-25-111-124-116-140-81-0
-3-167-85-157-43-94-155-84-24-129-65-41-69-48-11-42-31-99-141-63-38-112-136-
74-133-79-8-54-52-49-23-89-57-60-59-45-80-9-83-162-134-86-5-95-166-14-154-82
-156-62-40-50-20-143-90-55-61-56-67-37-71-39-98-149-36-135-75-152-97-34-103-
128-122-138-120-144-91-161-159-12,
- dla pliku *rbg403.atsp*: 6381 (błąd względny 159%) – ścieżka
193-98-135-254-332-360-148-238-343-186-366-382-208-81-400-245-197-288-132-
150-373-87-36-9-359-286-20-229-44-72-1-122-140-357-297-166-67-114-385-28-26
5-230-167-75-54-126-251-250-277-103-84-293-354-185-49-5-115-42-352-175-70-2
69-236-258-241-306-157-174-321-188-237-58-315-50-65-183-260-160-394-16-121
-386-40-104-4-387-52-324-59-284-170-240-347-10-89-317-221-389-199-266-344-
125-323-361-105-51-295-220-233-252-369-117-308-47-154-163-144-63-336-64-91
-268-355-341-177-113-96-82-156-32-145-196-85-29-264-328-309-106-311-397-24
7-222-165-348-30-318-172-182-218-398-201-365-25-246-118-299-303-136-61-2-2
67-362-329-259-206-334-57-232-74-194-283-307-402-45-24-203-92-130-273-358-
180-71-158-6-159-326-243-179-26-35-27-271-255-33-17-119-169-95-192-340-128-
8-377-316-171-375-231-152-181-338-99-78-60-205-34-213-298-263-80-38-120-29
0-333-69-371-107-292-401-73-97-204-319-224-368-116-108-191-287-257-211-217
-22-15-187-133-207-153-0-335-79-53-190-249-13-102-346-296-21-178-100-226-2
12-124-314-18-285-56-55-134-253-173-242-200-383-131-209-12-195-395-129-363
-19-68-198-41-384-101-62-235-210-77-327-239-294-289-94-66-155-275-272-93-1
51-393-302-278-378-396-23-337-123-313-310-356-379-111-3-112-219-274-46-262
-320-261-349-146-305-110-301-304-374-256-388-364-391-31-149-312-300-380-22
5-141-39-248-282-164-350-325-161-223-139-381-143-392-351-281-331-279-90-22
7-399-76-228-280-14-270-322-370-83-162-189-43-339-48-138-244-137-109-215-3
67-330-202-127-372-147-184-214-390-345-276-86-11-216-7-291-37-342-376-88-2
34-168-142-353-176.

Simulated Annealing

Temperatura początkowa	Szybkość wyżarzania	Najlepsze rozwiązanie w zależn. od rozmiaru problemu		
		48 (rozw. najl. 1776)	168 (rozw. najl. 2755)	403 (rozw. najl. 2465)
10	0,999990	<u>1951</u>	<u>6341</u>	<u>5180</u>
		<u>10%</u>	<u>130%</u>	<u>110%</u>
	0,999994	1968	<u>6043</u>	<u>5186</u>
		<u>11%</u>	<u>119%</u>	<u>110%</u>
	0,999999	<u>1860</u>	<u>5540</u>	<u>5167</u>
		<u>5%</u>	<u>101%</u>	<u>110%</u>
20	0,999990	1988	6369	6025
		<u>12%</u>	<u>131%</u>	<u>144%</u>
	0,999994	<u>1885</u>	6262	5980
		<u>6%</u>	<u>127%</u>	<u>143%</u>
	0,999999	1899	5803	5972
		<u>7%</u>	111%	142%
40	0,999990	2537	10135	6448
		<u>43%</u>	<u>268%</u>	<u>162%</u>
	0,999994	2516	10041	6474
		42%	<u>264%</u>	<u>163%</u>
	0,999999	2530	9657	6408
		<u>42%</u>	251%	160%

- dla pliku *ftv47.atsp*: 1860 (błąd względny 5%) – ścieżka
23-12-32-7-31-30-5-6-24-4-29-3-8-11-10-0-25-47-26-1-9-33-27-28-2-41-43-19-44-1
5-16-45-39-21-40-42-22-20-38-37-18-17-34-13-46-36-35-14,
- dla pliku *ftv170.atsp*: 5540 (błąd względny 101%) – ścieżka
116-113-110-144-55-68-34-54-6-166-16-151-149-45-21-135-84-12-153-130-121-14
1-61-49-75-20-143-57-60-158-9-10-162-122-106-107-4-120-102-150-43-85-1-70-1
1-156-66-140-63-38-41-92-3-124-95-155-32-62-50-90-161-146-48-26-119-105-19-
18-138-69-31-104-127-89-160-134-53-74-23-128-71-25-118-109-125-93-154-86-5-
163-131-129-126-99-37-97-139-94-52-59-44-79-13-147-78-14-145-42-100-164-88-
167-7-159-17-24-137-72-29-112-133-81-0-77-22-142-67-40-36-83-28-108-123-101
-30-111-136-47-80-8-157-56-65-39-98-152-96-148-76-15-114-103-132-87-2-33-58-
46-82-165-91-64-35-51-73-27-117-115,

- dla pliku *rbg403.atsp*: 5167 (błąd względny 110%) – ścieżka
159-173-3-306-225-178-390-286-154-220-342-176-219-47-54-231-257-161-136-21
0-137-138-40-48-203-275-68-188-246-326-96-88-39-236-227-14-66-42-106-29-12
9-322-293-179-70-8-233-380-269-214-174-185-384-261-119-146-247-375-294-289
-313-113-80-377-361-296-45-349-305-334-277-385-22-153-356-2-170-50-364-216-
195-31-297-124-123-386-324-63-372-283-368-376-55-388-204-167-24-36-142-84-
272-209-118-186-242-144-213-64-4-339-315-171-330-389-16-44-234-304-383-162
-314-11-207-378-78-52-341-110-340-166-59-359-351-325-95-76-152-366-128-260-
26-87-56-5-51-43-255-382-320-254-311-240-288-148-75-105-391-281-86-279-92-3
98-41-321-130-363-116-252-30-93-120-125-60-61-98-90-271-379-151-143-135-18
7-241-327-302-74-194-121-53-309-169-17-218-7-266-399-18-285-0-333-177-183-
316-355-149-329-205-344-81-278-34-182-72-62-310-164-282-395-239-208-73-155
-112-280-189-163-15-192-251-357-317-190-348-307-228-373-267-352-335-10-292
-362-184-202-198-253-374-71-103-82-259-264-274-350-97-381-345-215-319-224-
160-238-299-328-35-145-273-83-217-212-248-13-100-300-117-353-65-37-19-244-
28-158-397-9-114-108-46-85-226-197-132-243-323-206-115-230-338-126-301-393
-91-141-371-290-402-38-131-262-181-12-6-235-287-147-127-196-400-318-172-39
4-157-57-222-200-122-32-67-79-396-270-156-298-369-337-104-139-284-134-232-
49-221-229-354-1-332-20-107-370-358-193-250-223-111-308-343-140-360-263-21
1-365-133-245-77-367-295-312-346-401-150-392-256-249-258-33-268-94-99-21-2
3-69-265-199-336-27-165-168-175-303-201-89-58-276-25-387-180-191-109-102-2
91-347-237-331-101.

Wnioski

- Algorytmy genetyczne są niedokładne, to znaczy, że nie zawsze znajdują najlepsze rozwiązanie. W zamian znajdują rozwiązania bliskie optymalnemu w krótkim czasie (w porównaniu do algorytmów dokładnych)
- Implementacja algorytmu to jedynie połowa sukcesu, bardzo ważnym aspektem jest dobór parametrów najlepszych dla problemu. W zależności o rozmiarze grafu jak i odchylenia wartości średniej przejść można wnioskować jakie parametry będą dawały najlepsze wyniki
- Strategia wykorzystania tych algorytmów opiera się na ich szybkości. Są na tyle szybkie, że najlepiej jest uruchamiać je wiele razy dla tego samego problemu i wyciągać najbardziej optymalne rozwiązanie.

Źródła

https://www.youtube.com/watch?v=1FEP_sNb62k

<https://www.youtube.com/watch?v=XaXsJJh-Q5Y>

<http://cs.pwr.edu.pl/zielinski/lectures/om/mow10.pdf>

<https://doc.rust-lang.org/book/2018-edition/index.html>

<https://docs.rs/permutohedron/0.2.4/permutohedron/>

<https://docs.rs/time/0.1.40/time/>

https://docs.rs/num-traits/0.2.6/num_traits/

<https://docs.rs/rand/0.6.0/rand/>