

WSI – Raport z zadania 2

Michał Pędziwiatr

Numer indeksu: 331421

Wstępne założenia

Problem komiwojażera, znany także jako problem TSP (od ang. „travelling salesman problem”), to zagadnienie optymalizacyjne, polegające na odnalezieniu minimalnego cyklu Hamiltona w danym pełnym grafie ważonym. W kontekście treści naszego zadania jednak ów cykl przyjmuje postać najkrótszej trasy, która zaczynając od określonego miasta A, przechodzi przez każde miasto dokładnie raz, by zakończyć się w wyznaczonym mieście B.

Jednym z najefektywniejszych sposobów na rozwiązanie owego zagadnienia, są algorytmy ewolucyjne, które dzięki symulacji znanej nam z natury ewolucji, potrafią sprawnie selekcjonować potencjalne rozwiązania i ostatecznie wybrać najlepsze (najkrótsze).

Hiperparametry Algorytmu

Algorytmy ewolucyjne w swoim działaniu korzystają z zestawu określonych hiperparametrów, które dyktują ich działanie. Zgodnie z poleceniem, po wykonaniu wielu testów znalazłem ich ustawienie, zwracające przyzwoite wyniki:

- Ilość generacji = 1000,
- Rozmiar populacji (ilość rozwiązań na generację) = 250,
- Szansa na krzyżowanie = 75%,
- Szansa na mutację = 50%,
- Wzór na współczynnik przystosowania (ang. fitness) = $\frac{1}{s^9}$, gdzie s – długość rozwiązania,

Dodatkowo, wybranym punktem startowym zostały Skierniewice, a miastem końcowym – Warszawa.

Wpływ prawdopodobieństwa krzyżowania na proces optymalizacji

Hiperparametr szansy na krzyżowanie ma istotny wpływ na optymalizację i sposób działania naszego algorytmu ewolucyjnego. Niezależnie od wybranego sposobu w jaki je zaimplementujemy, to właśnie krzyżowanie umożliwia skuteczne wymienianie się danymi pomiędzy rozwiązaniami, co doprowadza do większej ilości oraz różnorodności ocenionych rozwiązań.

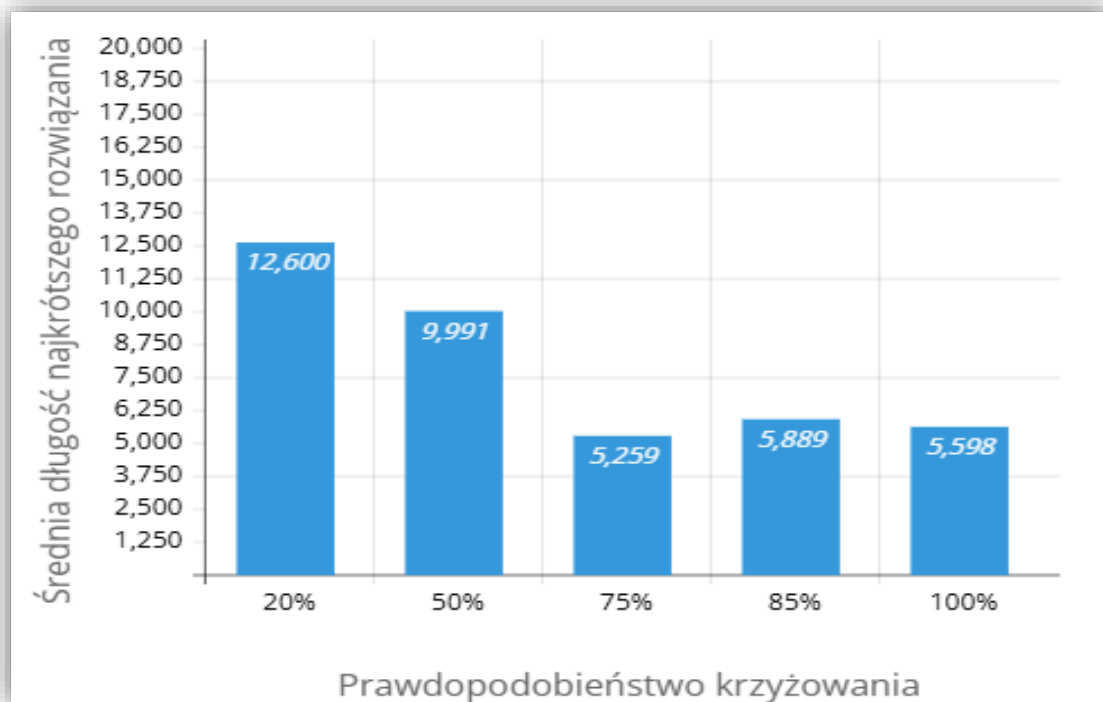
W celu dokładniejszej oceny wpływu ów parametru na proces optymalizacji mojej implementacji algorytmu, wykonałem pomiary przedstawione w tabeli widocznej na następnej stronie.

Wnioski

- Szansa na krzyżowanie = 20% (n = 5)
 - Średnia długość rozwiązania: 12600,743km
 - Średni czas działania algorytmu: 152,8s
 - Odchylenia standardowe: 308,476km, 3,9s
- Szansa na krzyżowanie = 50% (n = 5)
 - Średnia długość rozwiązania: 9991,061km
 - Średni czas działania algorytmu: 180,9s
 - Odchylenia standardowe: 345,218km, 2,6s
- Szansa na krzyżowanie = 75% (n = 10)
 - Średnia długość rozwiązania: 5259,722km
 - Średni czas działania algorytmu: 189,1s
 - Odchylenia standardowe: 348,193km, 6,9s
- Szansa na krzyżowanie = 85% (n = 10)
 - Średnia długość rozwiązania: 5889,650km
 - Średni czas działania algorytmu: 192,6s
 - Odchylenia standardowe: 466,272km, 6,6s
- Szansa na krzyżowanie = 100% (n = 10)
 - Średnia długość rozwiązania: 5598,060km
 - Średni czas działania algorytmu: 179,2s
 - Odchylenia standardowe: 388,058km, 8s

Prawdopodobieństwo krzyżowania	Długość najkrótszego rozwiązania [km]		Czas kompilacji algorytmu [s]
	20%	12058,497	155
	20%	12671,649	158
	20%	12713,061	153
	20%	12827,727	149,5
	20%	12732,783	148,5
	50%	9919,195	183,4
	50%	9631,067	180,2
	50%	10498,401	176,8
	50%	10155,435	182,4
	50%	9751,208	181,6
	75%	4962,375	187,8
	75%	5005,206	177,1
	75%	5519,208	190,4
	75%	5214,133	185,8
	75%	5766,471	201
	75%	4895,349	190,5
	75%	5611,422	193,7
	75%	5209,298	190,3
	75%	5449,345	195,4
	75%	4667,983	178,8
	75%	5556,157	189,2
	85%	6374,732	201,4
	85%	5634,37	191,2
	85%	5968,065	188,7
	85%	6724,81	203,05
	85%	5244,304	185,3
	85%	5426,835	185,2
	85%	5760,126	188,2
	85%	6293,557	199,4
	85%	5517,987	189,8
	85%	5951,714	193,9
	100%	5749,229	173,5
	100%	5226,276	186,3
	100%	5186,763	184,5
	100%	5784,307	170,9
	100%	6218,941	187,6
	100%	5388,871	171,8
	100%	5554,090	170,5
	100%	6215,440	191,5
	100%	5200,804	172,9
	100%	5455,880	182,3

Poza oczywistymi zmianami w wartościach prawdopodobieństwa krzyżowania, każdy pomiar wykonany został z omówionym we wcześniejszym punkcie ustawieniem hiperparametrów.



Wnioski cd.

Zarówno teoria algorytmów ewolucyjnych, jak i pomiary wskazują nam więc jak bardzo istotną rolę ma hiperparametr prawdopodobieństwa krzyżowania na proces optymalizacji. Znacznie mniej zadowalające wyniki jakie dały nam pomiary przy jego wartościach równych 20% oraz 50%, wyraźnie wskazują na to, iż wcześniej omówiona różnorodność rozwiązań jaką daje nam mechanizm krzyżowania jest kluczowa do optymalnego działania algorytmu.

Uważam, że jest to szczególnie ważne przy wybranym przeze mnie, ze względu na osiąganę wyniki, bardzo „surowym” sposobie oceniania współczynnika fitness. Krzyżowanie w tej sytuacji na ucieczkę z potencjalnych minimów lokalnych efektywności, w które współczynnik ten w połączeniu z selekcją ruletkową mógłby nas zaprowadzić. Owe minima, czyli pewnego rodzaju pułapki na nasz algorytm, wymagają bowiem często gwałtownej zmiany genów rozwiązania, czego zaimplementowany przeze mnie prosty algorytm mutacji może sam w sobie nie dostarczyć.

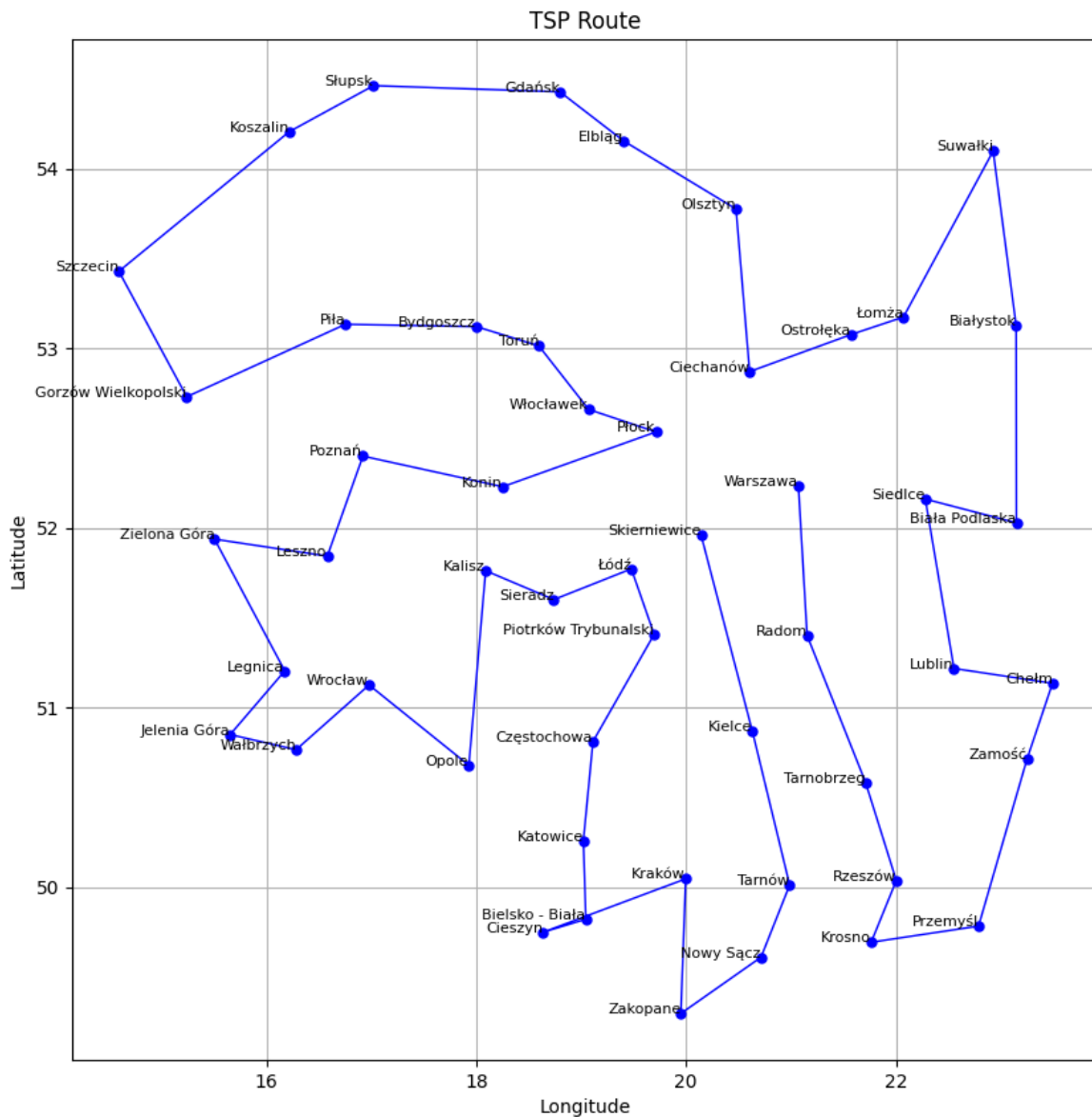
Analizując otrzymane wyniki warto zauważyć także bardzo gwałtowny przeskok w jakości rozwiązań pomiędzy wartością 50% i 75%, prowadzący do prawie dwukrotnego zmniejszenia się ich średniej długości z jednoczesną, proporcjonalnie nieporównywalnie mniejszą, zmianą czasu kompilacji. Pozostałe zmiany, w szczególności dla wartości $\geq 75\%$, mimo swej wyrazistości, są dużo łagodniejsze.

Kolejnym ciekawym, a zarazem niespodziewanym zjawiskiem, jest także swojego rodzaju minimum lokalne średniej długości rozwiązań w okolicach wartości prawdopodobieństwa krzyżowania równych 100%. Mimo tego, iż tak ustawiony hiperparametr zwraca w porównywalnym czasie mniej dokładne wyniki od wartości 75%, różnica pomiędzy nimi jest subtelniejsza niż mogłoby się wydawać. Może to wskazywać na to, iż dla naszej sytuacji, lub przynajmniej dla mojej implementacji algorytmu, eksploracja kompletnie nowych rozwiązań (wspomagana krzyżowaniem) jest ważniejsza od eksploatacji już poznanych rozwiązań. Osobiście uważam, że również w tym przypadku, znaczącą rolę mógł odegrać dość „ekstremalny” wzór na współczynnik fitness, który stawia silny nacisk na eksploatację najlepszych rozwiązań. W tej sytuacji więc, wysokie prawdopodobieństwo krzyżowania w pewnym sensie balansuje to podejście, dodając do puli wcześniej niespotkane krzyżówki tras.

Mimo tego jednak, iż wartość prawdopodobieństwa krzyżowania, rzeczywiście może znacząco wpływać na różnorodność oraz jakość rozwiązań, należy pamiętać, że nie jest to jedyny parametr naszego algorytmu. Zmiana wartości innych hiperparametrów takich jak rozmiar populacji czy prawdopodobieństwo mutacji, a także mechanizmów algorytmu, takich jak przykładowo wcześniej wspomniana ocena parametru fitness, może także odegrać ważną rolę w procesie optymalizacji. Jednocześnie, parametry te mogą mieć specyficzne ustawienia i kombinacje, dla których przebiega on bardziej lub mniej efektywnie.

Nie zmienia to faktu jednak, że hiperparametr prawdopodobieństwa krzyżowania ma olbrzymie znaczenie dla wyników naszego algorytmu (oraz mniejsze na czas jego działania) i nie należy lekceważyć procesu dobierania dla niego odpowiednich wartości.

Najlepsza znaleziona przez algorytm trasa



```
Best found solution: ['Skierniewice', 'Kielce', 'Tarnów', 'Nowy Sącz', 'Zakopane', 'Kraków', 'Cieszyn', 'Bielsko - Biała', 'Katowice', 'Częstochowa', 'Piotrków Trybunalski', 'Łódź', 'Sieradz', 'Kalisz', 'Opole', 'Wrocław', 'Wałbrzych', 'Jelenia Góra', 'Legnica', 'Zielona Góra', 'Leszno', 'Poznań', 'Konin', 'Płock', 'Włocławek', 'Toruń', 'Bydgoszcz', 'Piła', 'Gorzów Wielkopolski', 'Szczecin', 'Koszalin', 'Słupsk', 'Gdańsk', 'Elbląg', 'Olsztyn', 'Ciechanów', 'Ostrołęka', 'Łomża', 'Suwałki', 'Białystok', 'Biała Podlaska', 'Siedlce', 'Lublin', 'Chełm', 'Zamość', 'Przemyśl', 'Krosno', 'Rzeszów', 'Tarnobrzeg', 'Radom', 'Warszawa']
```

Distance: 4667.983km

Execution time: 178.78207850456238s

Generation count: 1000

Population size: 250

Crossover probability: 75.0%

Mutation probability: 50.0%