

Laboratorium 1

Implementacja i badanie Algorytmów Ewolucyjnych

1. Cel ćwiczenia

Celem ćwiczenia było zapoznanie się z metaheurystyką algorytmu ewolucyjnego w praktyczny sposób poprzez ich samodzielną implementację.

2. Wstęp

Problem komiwojażera – jest to problem polegający na znalezieniu najkrótszej ścieżki łączącej wszystkie punkty w taki sposób, że ostatni punkt łączy się z pierwszym. Problem ten może być rozwiązany na wiele sposobów, jednak przez swoją dużą złożoność obliczeniową często do jego rozwiązania wykorzystuje się metody niedeterministyczne.

Algorytmy ewolucyjne – są grupą algorytmów wykorzystujących czynnik losowy i zaliczane są do narzędzi sztucznej inteligencji. Podstawą algorytmów ewolucyjnych jest fakt, że największe prawdopodobieństwo przeżycia i stworzenia potomstwa mają te osobniki, które są najlepiej przystosowane.

3. Opis projektu i analiza wyników.

W ramach projektu zrealizowano następujące zadania:

- zbudowano model EA,
- zbadano działania EA na 5 plikach (1 łatwy, 3 średnie, 1 trudny),
- zbadano wpływ rozmiaru populacji pop_size i liczby pokoleń na wyniki działania EA
- zbadano wpływu 2 metod selekcji na skuteczność EA – turniej i ruletka
- zbadano jeden operator krzyżowania i dwa operatory mutacji

Do implementacji wykorzystano język C++.

Przy badaniu wyników dla danej metody brano pod uwagę 10 uruchomień algorytmu z tymi samymi parametrami początkowymi.

Tabela 1. Zestawienie działania algorytmu losowego i ewolucyjnego dla różnych zbiorów danych.

Nazwa zbioru	Wynik optymalny	Algorytm losowy 10k				Algorytm genetyczny [100 osobnikow, 100 epok]			
		best	worst	avg	std	best	worst	avg	std
berlin52	7542	22793	23892	23182	339	9719	11730	11025	688
kroA100	21282	136849	142292	139465	1733	55925	63294	60677	2634
kroA150	26524	213598	221997	217865	2655	107764	115081	107818	3926
kroA200	29368	285405	297735	293262	4911	150694	160750	155889	3987
fl417	11861	438003	445478	442634	2196	248712	272906	259695	6823

Działanie algorytmu ewolucyjnego jest znacznie skuteczniejsze niż algorytmu losowego dla podobnej ilości iteracji (patrz Tabela 1). Złożoność obliczeniowa algorytmu losowego jest mniejsza i

pojedyncze iteracje wykonują się o wiele szybciej, jednak po pewnym czasie wyniki przestają się polepszać. Dla algorytmów ewolucyjnych zwiększenie ilości epok powoduje znaczące polepszenie zwracanych wyników. W tabeli 2 zestawiono wyniki działania algorytmu ewolucyjnego dla różnych zbiorów danych przy jednoczesnym zwiększeniu liczby iteracji. Dla małych zbiorów wejściowych wyniki zaczęły zbliżać się do wartości optymalnych. Zwiększenie liczby iteracji zmniejszyło odchylenie standardowe.

Tabela 2. Zestawienie działania algorytmu ewolucyjnego dla różnych zbiorów danych przy wydłużeniu czasu obliczeń.

Nazwa zbioru	Wynik optymalny	Algorytm genetyczny [200 osobników, 1000 epok]			
		best	worst	avg	std
berlin52	7542	7991	9047	8303	323
kroA100	21282	23083	25359	24500	811
kroA150	26524	32282	35154	33868	995
kroA200	29368	43168	47704	45500	1949
fl417	11861	37964	45044	42510	2253

W celu sprawdzenia wpływu różnych parametrów na działanie algorytmów ewolucyjnych wykonano szereg testów. W tabeli 3 dokonano porównania wpływu prawdopodobieństw krzyżowania na skuteczność działania algorytmu.

Tabela 3. Wpływ prawdopodobieństwa krzyżowania na działanie algorytmu genetycznego dla zbioru kroA150 dla 1000 epok.

Rodzaj selekcji	Prawdopodobieństwo krzyżowania	Typ mutacji	Prawdopodobieństwo mutacji	Rozmiar populacji	Najlepsze	Średnia	Odchylenie standardowe
turniej	1.0	inwersja	0.1	100.0	37174.7	40252.8	1619.9
turniej	0.7	inwersja	0.1	100.0	38745.8	40663.5	1358.3
turniej	0.3	inwersja	0.1	100.0	40788.3	42496.3	1526.0

Najlepsze wyniki uzyskano dla prawdopodobieństwa krzyżowania równego 1. Działanie takie może spowodować szybsze przystosowanie populacji, jednak wiąże się też z ryzykiem. Jeśli nie zastosujemy zasady elitaryzmu, do kolejnego pokolenia nie przejdzie żaden osobnik ze starego zbioru. Osobniki w nowej populacji nie koniecznie muszą być lepszej. Duża wartość odchylenia standardowego dla najwyższego prawdopodobieństwa krzyżowania wskazuje na większą różnorodność populacji.

Tabela 4. Wpływ rodzaju selekcji na działanie algorytmu genetycznego dla zbioru kroA150 dla liczby epok 1000.

Rodzaj selekcji	Prawdopodobieństwo krzyżowania	Typ mutacji	Prawdopodobieństwo mutacji	Rozmiar populacji	Najlepsze	Średnia	Odchylenie standardowe
turniej	0.7	inwersja	0.1	100.0	38745.8	40663.5	1358.3
ruletka	0.7	inwersja	0.1	100.0	39921.3	41224.3	1252.3

Dla testowanego zbioru danych selekcja poprzez turniej okazała się nieznacznie skuteczniejsza. Algorytm ruletki jest bardziej skomplikowany i złożony obliczeniowo niż turniej, jednak ma on zasadniczą przewagę. W przypadku ruletki, osobniki słabo przystosowane mogą utworzyć potomstwo. Jest to mało prawdopodobne, jednak możliwe. Dla algorytmu ruletki osobniki o słabym przystosowaniu są pomijane i nie mogą mieć potomstwa.

Tabela 5. Wpływ rozmiaru populacji działanie algorytmu genetycznego dla zbioru kroA150 dla 1000 epok.

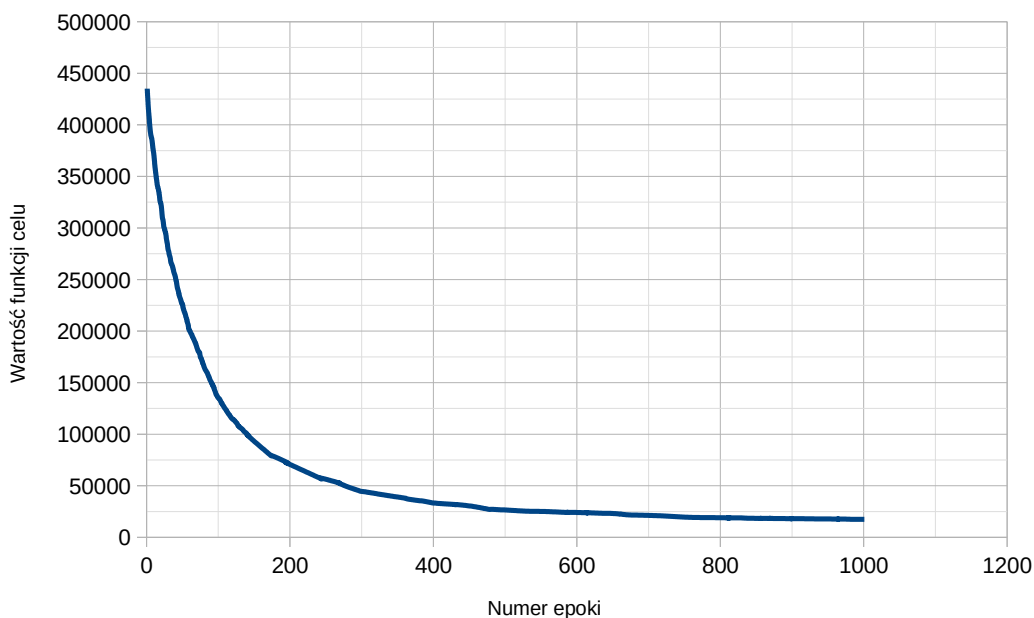
Rodzaj selekcji	Prawdopodobieństwo krzyżowania	Typ mutacji	Prawdopodobieństwo mutacji	Rozmiar populacji	Najlepsze	Średnia	Odchylenie standardowe
turniej	0.7	inwersja	0.1	1000.0	28968.9	29513.2	428.2
turniej	0.7	inwersja	0.1	100.0	39801.1	42471.4	1440.2
turniej	0.7	inwersja	0.1	10.0	96185.4	103302.5	4586.9

Bardzo duży wpływ na skuteczność poszukiwania optymalnego rozwiązania ma rozmiar populacji (tabela 5). Wraz ze wzrostem rozmiaru populacji znacznie wzrasta czas obliczeń, jednak uzyskiwane wyniki mogą być warte fady. Dla populacji liczącej 1000 osobników uzyskano ponad trzykrotnie lepszy wynik.

Tabela 6. Wpływ typu mutacji działanie algorytmu genetycznego dla zbioru kroA150 dla 1000 epok.

Rodzaj selekcji	Prawdopodobieństwo krzyżowania	Typ mutacji	Prawdopodobieństwo mutacji	Rozmiar populacji	Najlepsze	Średnia	Odchylenie standardowe
turniej	0.7	inwersja	0.1	100.0	39468.3	40794.6	1284.2
turniej	0.7	zamiana	0.1	100.0	70014.3	75920.3	4590.0

Największym zaskoczeniem podczas wszystkich testów okazał się mechanizm mutacji poprzez inwersje. Dzięki losowym zmianom kolejności genów u osobników, bardzo wzrosła wydajność całego algorytmu. W problemie komiwojażera zastosowanie mutacji przez inwersje jest dużo skuteczniejsze niż zamiana.



Wykres 1. Wykres zależności funkcji celu od numeru epoki dla EA w rozwiązywaniu problemu TSP dla zbioru fl417. (pop_size = 1000, Px = 0.7, Pm = 0.1, mutacja przez inwersje)

Na wykresie 1 przedstawiono proces znajdowania najlepszej ścieżki dla problemu komiwożacza. Początkowo losowo utworzone osobniki ulegają szybkiej ewolucji i przystosowywaniu. Po około 400 iteracjach różnice między najlepszymi osobnikami z nowego pokolenia i starego nie były już tak duże, jednak algorytm dalej zbliżał się do rozwiązania optymalnego. Działanie algorytmu zostało przerwane po 1000 iteracji, jednak algorytm nie osiągnął jeszcze wartości granicznej. Dalsze jego działanie pozwoliłoby na uzyskanie lepszych rezultatów.

4. Podsumowanie i wnioski

- Algorytm genetyczny jest bardziej kosztowny obliczeniowo niż algorytm losowy, jednak zapewnia o wiele lepsze rezultaty. Dla niewielkiej liczby iteracji wyniki są podobne, jednak gdy ta ulega zwiększeniu, algorytm losowy nie zapewnia znacznej poprawy.
- Dobór odpowiednich parametrów dla algorytmów genetycznych jest bardzo ważny.
- Większe prawdopodobieństwo krzyżowania zapewnia lepsze wyniki, jednak nie zawsze warto ustawiać duże wartości tego parametru. W celu bardziej stabilnego działania lepiej jest utrzymywać wartość tego parametru w okolicach 0.7.
- Rozmiar populacji ma duży wpływ na skuteczność poszukiwania optymalnego rozwiązania. Im większa jest populacja, tym rezultaty są lepsze – wiąże się to jednak ze znacznym wydłużeniem obliczeń.
- Mechanizm mutacji wpływa bardzo pozytywnie na zachowanie populacji. Spośród zaimplementowanych metod, najlepszym mechanizmem okazała się inwersja. Jej zastosowanie na zbiorze kroA150 z prawdopodobieństwem 0.1 sprawiło, że skuteczność algorytmu uległa podwojeniu.