Mateusz Krupa 256280

Projektowanie efektywnych algorytmów
Asymetryczny Problem Komiwojażera
Algorytm genetyczny

Spis treści

1 Wstęp teoretyczny

1.1 Cel projektu

Celem zadania projektowego była implementacja oraz analiza efektywności algorytmy genetycznego. Należało zbadać jak dobre rozwiązania algorytm ten otrzymuje w określonym czasie i przedstawić je na wykresie jako zależność błędu względnego (podanego w procentach) w funkcji czasu wykonywania algorytmu. Błąd względny wyrażono jako:

$$\frac{|f - f_{opt}|}{f_{opt}}$$

gdzie:
f - wartość obliczona przez testowany algorytm
f_{opt} - wartośc optymalna - najlepsze znane rozwiązanie

Problem komiwojażera (eng. TSP) to zagadnienie polegające na znalezieniu najkrótszej drogi w grafie nieskierowanym, w którym znamy wszystkie wierzchołki oraz krawędzie. Dzięki tym informacjom jesteśmy w stanie wyznaczyć różne drogi w zależności o interesujące nas kryteria. W projekcie skupimy się na znalezieniu najkrótszej drogi od miasta początkowego (czyli 0) poprzez wszystkie pozostałe miasta i powrót do miasta początkowego.

1.2 Algorytm genetyczny

Algorytm genetyczny(eng. genetic algorithm) - jest to algorytm heurystyczny polegający na przeszukiwaniu przestrzeni rozwiązań, który przypomina zjawisko ewolucji.
Populacja tworzona jest poprzez tworzenie wielu rozwiązań o losowej kolejności a następnie selekcja poprzez wybranie osobników o najkrótszym wyniku. Po wstępnej selekcji pętla algorytmu to: mutacja, krzyżowanie oraz tworzenie nowej populacji. Mutacja i krzyżowanie występuje z pewnym prawdopodobieństwem a tworzenie nowej populacji zawsze gdy poddamy rozmnażaniu wszystkich osobników. Mutacja polega na zamianie kolejności dwóch losowych genów danego osobnika np:

$$0 - 1 - 2 - 3 - 4 - 0$$
$$0 - 4 - 2 - 3 - 1 - 0$$

Krzyżowanie odbywa się za pomocą dwóch osobników(rodziców). Losowany jest indeks który oznacza miejsce podziału. Geny do tego miejsca pozostają bez zmian reszta jest usuwana. Brakujące geny uzupełniamy w takiej kolejności w jakiej występują u drugiego osobnika. Przykład dla miejsca podziału równego 2:

$0 - 1 - 2 - 3 - 4 - 0$ - pierwszy osobnik
$0 - 4 - 1 - 2 - 3 - 0$ - drugi osobnik
$0 - 1 - 2 - 4 - 3 - 0$ - utworzony osobnik

2 Opis najważniejszych klas w projekcie

Aplikacja posiada trzy klasy. Klasa Result pomogła mi łatwiej przechowywać i działać na danych które były osobnikami i posiadała ścieżkę dla danego osobnika. Klasa Program służy do wyświetlania menu i komunikacji z użytkownikiem oraz przechowuje domyślne wartości dla algorytmu. Klasa GeneticAlgorithm posiaga główne pętle algorytmu, która zawiera wywołanie funkcji takich jak tworzenie populacji, krzyżowanie, mutacja.

3 Wyniki eksperymentu i wnioski

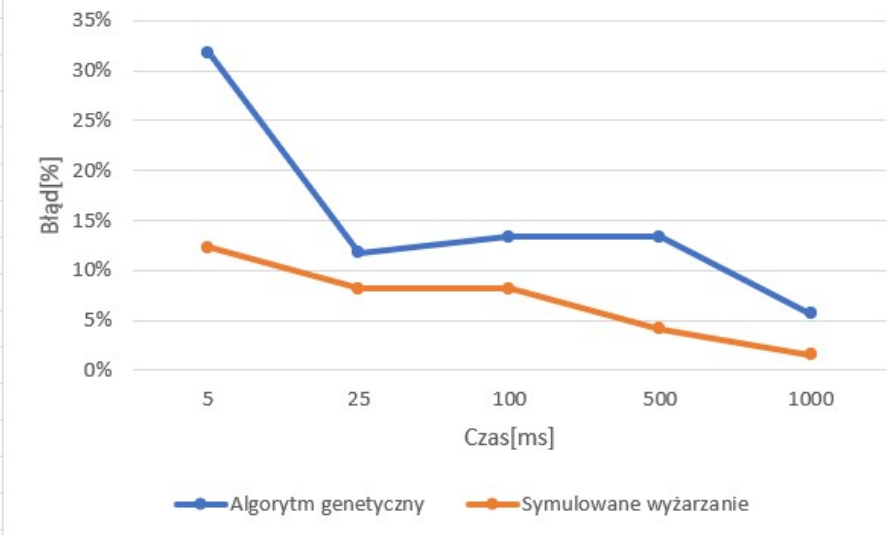
3.1 Testowanie

Dla wszystkich testów współczynnik krzyżowania wynosi 0.8 oraz współczynnik mutacji 0,01.

3.1.1 br17

populacja początkowa = 10

Algorytm genetyczny		**Symulowane wyżarzanie**	
Czas[ms]	**Błąd[%]**	**Czas[ms]**	**Błąd[%]**
5	32%	5	12%
25	12%	25	8%
100	13%	100	8%
500	13%	500	4%
1000	6%	1000	2%

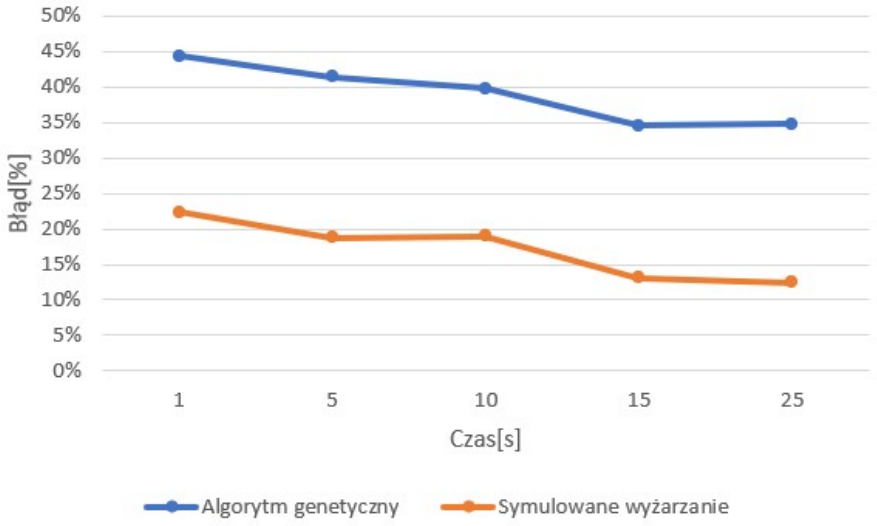


Rysunek 1: Błąd względny od funkcji czasu

3.1.2 ftv45

populacja początkowa = 100

Algorytm genetyczny		**Symulowane wyżarzanie**	
Czas[s]	**Błąd[%]**	**Czas[s]**	**Błąd[%]**
1	44%	1	22%
5	41%	5	19%
10	40%	10	19%
15	35%	15	13%
25	35%	25	12%

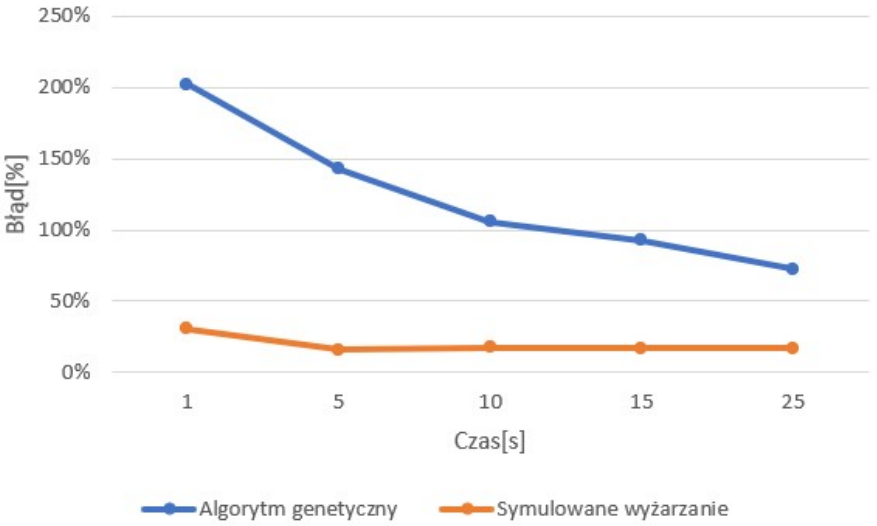


Rysunek 2: Błąd względny od funkcji czasu

3.1.3 rbg323

populacja początkowa = 150

Algorytm genetyczny		**Symulowane wyżarzanie**	
Czas[s]	**Błąd[%]**	**Czas[s]**	**Błąd[%]**
1	202%	1	30%
5	143%	5	16%
10	106%	10	18%
15	93%	15	17%
25	72%	25	17%



Rysunek 3: Błąd względny od funkcji czasu

3.2 Wnioski

Podczas testowania algorytmu zauważyłem, że większa wartość początkowej populacji wpływa pozytywnie dla większych instancji problemu, jednak rozwiązania jakie proponował algorytm w te-

stowanym czasie nie były tak dobre jak w przypadku symulowanego wyżarzania. Jednoznacznie z wykresów widać, że algorytm genetyczny potrzebuje więcej czasu by osiągnąć wyniki bliskie optymalnemu podczas gdy symulowane wyżarzanie działa zadowalająco nawet w krótkim czasie. Z tego wynika, że algorytm genetyczny działa na określonej populacji a symulowane wyżarzanie tylko na jednej instancji. Z moich obserwacji obydwu algorytmów wynika, że nie da się znaleźć parametrów, które by dobrze sprawdzały się dla różnych instancji, najlepiej jest określać osobno parametry dla każdej instancji oraz kryteriów działania algorytmu.

4 Literatura

[1] $https://www.obitko.com/tutorials/genetic-algorithms/operators.php$
[2] $https://www.rubicite.com/Tutorials/GeneticAlgorithms.aspx$
[3] $https://pl.wikipedia.org/wiki/Algorytm_genetyczny$