Porównanie algorytmów Branch&Bound, TabuSearch, symulowanego wyżarzania i genetycznego dla problemu komiwojażera w kontekście optymalizacji czasu robienia zakupów w sklepie.

Grupa 3

Prezentacja nr 2 - 27.01.2022

Seminarium - Systemy Wspomagania Decyzji i Symulacja Komputerowa

Konspekt

1. Wstęp
2. Dane wejściowe
3. Algorytmy
4. Symulator
5. Badania
6. Wyniki
7. Diagram Gantta
8. Podsumowanie

Wstęp

- Niniejszy projekt zakłada porównanie algorytmów Branch&Bound, TabuSearch, symulowanego wyżarzania i algorytmu genetycznego dla rzeczywistego przykładu problemu komiwojażera zaobserwowanego w życiu codziennym.
- Problemem tym jest optymalizacja czasu robienia zakupów w sklepie.
- Trasa komiwojażera jest cyklem przechodzącym przez każdy wierzchołek grafu dokładnie jeden raz - jest to zatem cykl Hamiltona.
- Celem projektu jest porównanie algorytmów znajdowania optymalnej ścieżki robienia typowych, codziennych zakupów w supermarkecie.

Dane wejściowe

- pliki .txt
- przykłady bazujące na rzeczywistych danych, stworzone na podstawie rzeczywistych list zakupów oraz planów sklepów
- Format pliku:

Pierwsza linia to liczba X (całkowita, dodatnia, różna od zera) oznaczająca ilość miast.

Kolejne linie to macierz X na X, liczby w linijce są oddzielone spacja,
zawierają odległości pomiędzy miastami (całkowite, dodatnie, różne od zera).
Przekątna macierzy zawiera liczby -1.

Zawartość przykładowego pliku:
3
-1 10 15
20 -1 5
25 30 -1

Symulator

```
--- TRAVELLING SALESMAN PROBLEM ---
1. Wczytaj macierz miast z pliku
2. Generuj losowa macierz miast
3. Wyświetl macierz miast
4. Uruchom algorytm Brute-force
5. Uruchom algorytm Branch and bound
6. Zmień ustawienia algorytmu Tabu Search
7. Uruchom algorytm Tabu Search
8. Zmień ustawienia algorytmu Genetic
9. Uruchom algorytm Genetic
S. Uruchom algorytm Simulated Annealing
F. Testy
0. Wyjście
Wybór:
```

Algorytmy

- Branch&Bound,
- TabuSearch,
- Algorytm genetyczny,
- Symulowane wyżarzanie

Branch and bound

- Implementacja na bazie rekurencyjnego algorytmu brute force
- Dolne ograniczenie wyliczane jako suma minimalnego kosztu odwiedzenia pozostałych wierzchołków, wyliczonego ze wzoru:

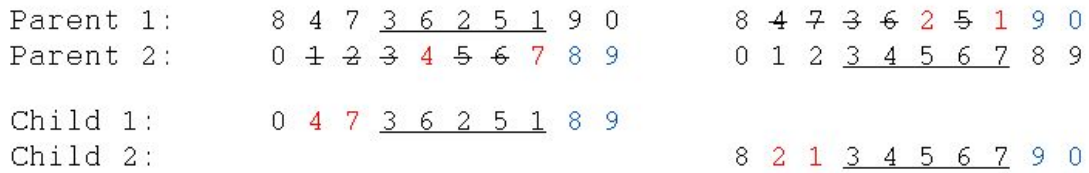
$$(min(wagaKrawedziWchodzacej) + min(wagaKrawedziWychodzacej)) / 2$$

Tabu Search

- otoczenie typu swap (złożoność obliczeniowa jednej iteracji - $O(n^2)$)
- początkowe rozwiązanie znajdywane algorytmem najbliższego sąsiada
- lista tabu to pary miast
- zaimplementowane kryterium aspiracji i dywersyfikacji
- koniec pracy algorytmu to upłynięcie zadanego czasu

Algorytm genetyczny

- Genotypem osobnika jest trasa komiwojażera
- Fenotypem jest długość tej trasy
- Populacja startowa jest generowana całkowicie losowo
- Zastosowaliśmy algorytm krzyżowania OX (ordered crossover)



Symulowane wyżarzanie

```
najlepszyKosztGlobalny = INT_MAX
najlepszaKolejnosc
x = losowe rozwiązanie (np. porządek naturalny)
while (temperatura > temperaturaKońcowa
       while (numerPowtórzenia < liczbaPowtórzeńDlaJednejTemperatury)
              y = stwórzNoweRozwiązanieNaPodstawieObecnego(x)
              yKoszt = obliczKoszt(y)
              if (yKoszt < najlepszyKosztGlobalny)
                     najlepszyKosztGlobalny = yKoszt
                     najlepszaKolejnosc = y.kolejnosc
              if (yKoszt < obliczKoszt(x) )
                     x = y
              else if (exp ((obliczKoszt (x) - yKoszt) / temperatura)) >losowaLiczbaOd0Do1())
                     x = y
              numerPowtórzenia++
       temperatura = temperatura * wspolczynnikStygnieciaAlpha
return najlepszyKosztGlobalny, najlepszaKolejnosc
```

Badania

- Nie wszystkie parametry algorytmów wzięto pod uwagę podczas tworzenia scenariuszy testowych. Niektóre z nich zawsze dają lepsze wyniki, a niektóre z nich mają praktycznie zerowy wpływ na wynik.
- Po wybraniu parametrów badawczych zbadano wpływ zmiany jednego z nich na długość drogi uzyskaną przez algorytm.

Branch&Bound	TabuSearch	Algorytm Genetyczny	Symulowane Wyżarzanie
brak	**czas obecności na liście:** 0.25n, 0.5n, n, 2n **czas pracy:** 5s, 10s, 30s	**wielkość populacji:** 25, 50, 100 **współczynnik krzyżowania:** 0.33, 0.80, 0.99 **czas pracy:** 10s, 30s, 60s,	**Temperatura początkowa:** 1000, 10000, 5000 **Temperatura końcowa:** 0.1, 0.01, 0.001 **Liczba powtórzeń dla jednej T:** 25, 50, 100

Wyniki

1. Dla małych zbiorów danych zmiana parametrów nie miała wpływu na osiągane wyniki.
2. Różnice pojawiały się dla średnich i dużych zestawów.

Wyniki - Branch&Bound

1. Dla średnich zbiorów danych algorytm uzyskiwał gorsze wyniki niż reszta algorytmów. Prawdopodobnie spowodowane jest to dosyć niską złożonością sposobu funkcjonowania algorytmu.

Tabela nr 1 - wyniki dla Branch&Bound

	Wyniki dla poszczególnych plików z zestawami danych			
	12-1.txt	12-2.txt	17.txt	24.txt
Uzyskany wynik	54	47	62	80

Wyniki - TabuSearch

1. Ustawienie kadencji na 0.5*iloscMiast prowadziło do osiągnięcia o średnio 3% lepszych wyników niż dla kadencji 0.25*iloscMiast, 1.00*iloscMiast oraz 2.00*iloscMiast. Im większy zbiór danych tym mniejsza różnica.
2. Im dłuższy czas pracy algorytmu tym lepszy wynik udało się osiągnąć.
3. Włączenie aspiracji i dywersyfikacji wpływa pozytywnie na wyniki osiągane przez algorytm.

Tabela nr 2 - wyniki dla TabuSearch

Parametry algorytmu	Wyniki dla poszczególnych plików z zestawami danych					
	12-1.txt	12-2.txt	17.txt	24.txt	31.txt	50.txt
Kadencja: 0.25*iloscMiast Czas: 10s Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61,1	81	98,14	154
Kadencja: 0.50*iloscMiast Czas: 10s Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61	78,88	94	154
Kadencja: 1.00*iloscMiast Czas: 10s Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61	79,9	98,7	157
Kadencja: 2.00*iloscMiast Czas: 10s Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61	81	98	157
Kadencja: 0.50*iloscMiast **Czas: 5s** Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61	79,6	94	154
Kadencja: 0.50*iloscMiast **Czas: 10s** Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61	79,4	94	154
Kadencja: 0.50*iloscMiast **Czas: 30s** Aspiracja: włączona Dywersyfikacja: włączona Iteracje bez poprawy do dywersyfikacji: 10000	54	47	61	79	94	153,9

Wyniki - Alg. Genetyczny

1. Im większa populacja tym lepszy wynik.
2. Im wyższy współczynnik krzyżowania tym lepszy wynik.
3. Dłuższa praca algorytmu prowadziła do nieznacznego pogorszenia wyniku.

Tabela nr 3 - wyniki dla Algorytmu Genetycznego

Parametry algorytmu	Wyniki dla poszczególnych plików z zestawami danych					
	12-1.txt	12-2.txt	17.txt	24.txt	31.txt	50.txt
Czas: 30s **Populacja: 25** Współczynnik krzyżowania: 0.8 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	63,2	88,8	113	184,7
Czas: 30s **Populacja: 50** Współczynnik krzyżowania: 0.8 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	62,8	87,8	107,3	176
Czas: 30s **Populacja: 100** Współczynnik krzyżowania: 0.8 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	61,8	83,9	104	169,1
Czas: 30s Populacja: 50 **Współczynnik krzyżowania: 0.33** Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	62,7	87,5	109,4	179,8
Czas: 30s Populacja: 50 **Współczynnik krzyżowania: 0.80** Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	62,6	87	108,1	177,4
Czas: 30s Populacja: 50 **Współczynnik krzyżowania: 0.99** Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	62,4	83,8	107,8	174,1
Czas: 10s Populacja: 50 Współczynnik krzyżowania: 0.80 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	63	85,8	107,3	177,7
Czas: 30s Populacja: 50 Współczynnik krzyżowania: 0.80 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	63,7	85,9	109,5	176,8
Czas: 60s Populacja: 50 Współczynnik krzyżowania: 0.80 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	54	47	63	86,1	110	179,6

Wyniki - Sym. Wyżarzanie

1. Wzrost temperatury początkowej zwykle powodował nieznaczne pogorszenie wyników osiąganych przez algorytm.
2. Spadek temperatury końcowej zwykle powodował nieznaczne polepszenie wyników osiąganych przez algorytm.
3. Zwiększanie ilości powtórzeń pozytywnie wpływało na wyniki osiągane przez algorytm.

Tabela nr 4 - wyniki dla Symulowanego Wyżarzania

Parametry algorytmu	Wyniki dla poszczególnych plików z zestawami danych					
	12-1.txt	12-2.txt	17.txt	24.txt	31.txt	50.txt
Temperatura początkowa: 1000 Temperatura końcowa: 0.01 Ilość powtórzeń: 25 Współczynnik alpha: 0.999	54	47	61,1	78,7	94,2	144,1
Temperatura początkowa: 5000 Temperatura końcowa: 0.01 Ilość powtórzeń: 25 Współczynnik alpha: 0.999	54	47	61,1	78,7	93,6	144,3
Temperatura początkowa: 10000 Temperatura końcowa: 0.01 Ilość powtórzeń: 25 Współczynnik alpha: 0.999	54	47	61,2	78,8	93,7	143,2
Temperatura początkowa: 1000 **Temperatura końcowa: 0.1** Ilość powtórzeń: 25 Współczynnik alpha: 0.999	54	47	61	78,7	94,5	144,6
Temperatura początkowa: 1000 **Temperatura końcowa: 0.01** Ilość powtórzeń: 25 Współczynnik alpha: 0.999	54	47	61	78,6	94	143,9
Temperatura początkowa: 1000 **Temperatura końcowa: 0.001** Ilość powtórzeń: 25 Współczynnik alpha: 0.999	54	47	61	79	94	143,8
Temperatura początkowa: 1000 Temperatura końcowa: 0.01 **Ilość powtórzeń: 25** Współczynnik alpha: 0.999	54	47	61,2	79	93,9	144,5
Temperatura początkowa: 1000 Temperatura końcowa: 0.01 **Ilość powtórzeń: 50** Współczynnik alpha: 0.999	54	47	61	78,4	93,8	142,2
Temperatura początkowa: 1000 Temperatura końcowa: 0.01 **Ilość powtórzeń: 100** Współczynnik alpha: 0.999	54	47	61	78,4	93,2	141,7

Wyniki - zbiorcze

Obok zaprezentowano zbiorczą dla wszystkich algorytmów tabelę (tabela nr 5) zawierającą zwycięskie (cechujące się najlepszymi wynikami) konfiguracje każdego algorytmu dla poszczególnych zestawów danych.

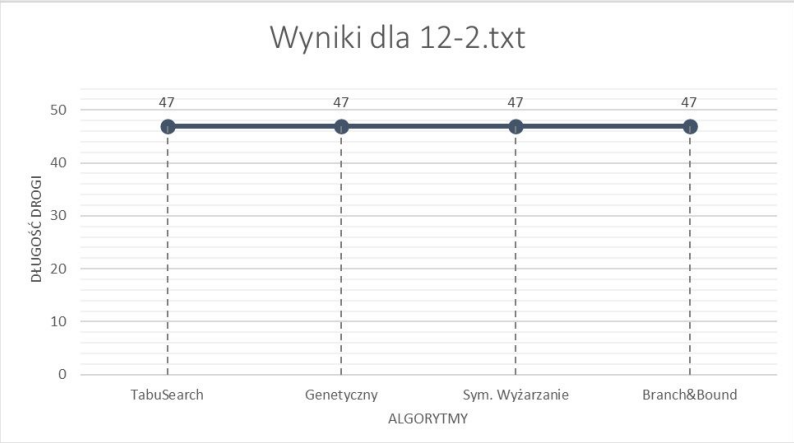
Na podstawie tabeli zostały sporządzone wykresy obrazujące różnice pomiędzy wynikami osiągniętymi przez algorytmy - zostaną zaprezentowane na kolejnych slajdach.

Tabela nr 5 - konfiguracje algorytmów, które osiągnęły najlepsze wyniki

Plik (zestaw danych)	Wynik oraz konfiguracja algorytmu			
	TabuSearch	Algorytm Genetyczny	Symulowane Wyżarzanie	Branch&Bound
12-1.txt	Wynik: **54** Konfiguracje osiągnęły ten sam wynik	Wynik: **54** Konfiguracje osiągnęły ten sam wynik	Wynik: **54** Konfiguracje osiągnęły ten sam wynik	Wynik: **54** Brak konfiguracji
12.2.txt	Wynik: **47** Konfiguracje osiągnęły ten sam wynik	Wynik: **47** Konfiguracje osiągnęły ten sam wynik	Wynik: **47** Konfiguracje osiągnęły ten sam wynik	Wynik: **47** Brak konfiguracji
17.txt	Wynik: **61** Kadencja: 0.50*iloscMiast Czas: 10s Aspiracja: tak Dywersyfikacja: tak Iteracje bez poprawy do dywersyfikacji: 10000	Wynik: **61,8** Czas: 30s Populacja: 100 Współczynnik krzyżowania: 0.8 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	Wynik: **61** Temperatura początkowa: 1000 Temperatura końcowa: 0.1 Ilość powtórzeń: 100 Współczynnik alpha: 0.999	Wynik: **62** Brak konfiguracji
24.txt	Wynik: **78,88** Kadencja: 0.50*iloscMiast Czas: 10s Aspiracja: tak Dywersyfikacja: tak Iteracje bez poprawy do dywersyfikacji: 10000	Wynik: **83,8** Czas: 30s Populacja: 50 Współczynnik krzyżowania: 0.99 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	Wynik: **78,4** Temperatura początkowa: 1000 Temperatura końcowa: 0.01 Ilość powtórzeń: 100 Współczynnik alpha: 0.999	Wynik: **80** Brak konfiguracji
31.txt	Wynik: **94** Kadencja: 0.50*iloscMiast Czas: 10s Aspiracja: tak Dywersyfikacja: tak Iteracje bez poprawy do dywersyfikacji: 10000	Wynik: **104** Czas: 30s Populacja: 100 Współczynnik krzyżowania: 0.8 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	Wynik: **93,2** Temperatura początkowa: 1000 Temperatura końcowa: 0.01 Ilość powtórzeń: 100 Współczynnik alpha: 0.999	
50.txt	Wynik: **153,9** Kadencja: 0.50*iloscMiast Czas: 30s Aspiracja: tak Dywersyfikacja: tak Iteracje bez poprawy do dywersyfikacji: 10000	Wynik: **169,1** Czas: 30s Populacja: 100 Współczynnik krzyżowania: 0.8 Współczynnik mutacji 0.01 Mutacja: wierzchołkowa	Wynik: **141,7** Temperatura początkowa: 1000 Temperatura końcowa: 0.01 Ilość powtórzeń: 100 Współczynnik alpha: 0.999	

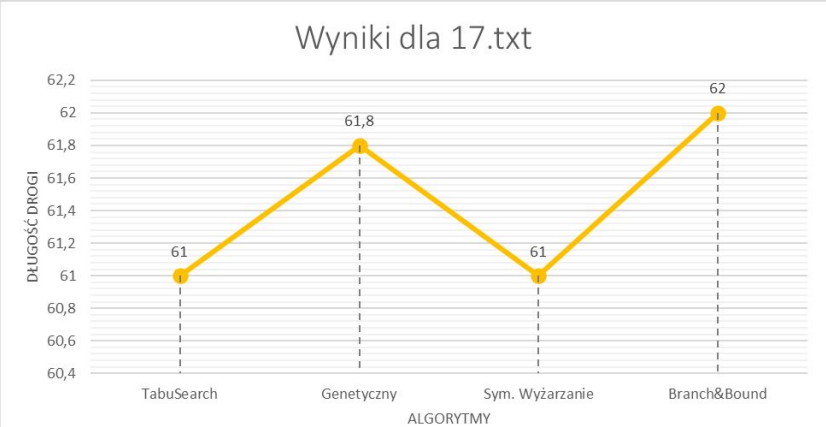
Wyniki - zbiorcze

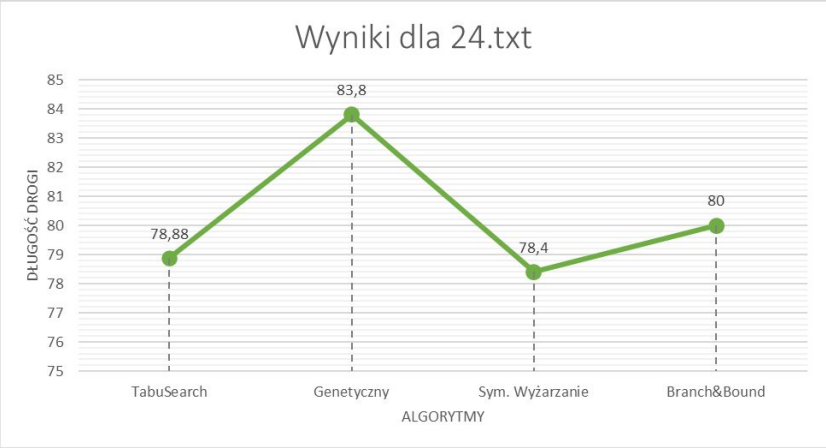
Jak widać na wykresach obok, dla mniejszych zestawów danych (z plików 12-1.txt oraz 12-2.txt) algorytmy osiągnęły te same wyniki. Warto jednak podkreślić, że nasza implementacja Symulowanego Wyżarzania osiągnęła wynik znacznie szybciej od pozostałych algorytmów.



Wyniki - zbiorcze

Jak widać na wykresach obok dla średnich zestawów danych (z plików 17.txt i 24.txt) najgorszy okazał się algorytm genetyczny. Pozostałe algorytmy osiągnęły podobne wyniki. Różnica pomiędzy TabuSearch oraz Symulowanym Wyżarzaniem była szczególnie mała. Najlepszy okazało się jednak Symulowane Wyżarzanie. Dodatkowo, to właśnie ten algorytm znalazł optymalny wynik najszybciej.





Wyniki - zbiorcze

Dla dużych zbiorów danych zrezygnowano z testów algorytmu Branch&Bound, ponieważ czas działania algorytmu był zbyt długi - co znacząco obniża użyteczność oraz sens wykorzystywania algorytmu. W tym przypadku ponownie najlepszy okazał się algorytm Symulowanego Wyżarzania z małą stratą dla TabuSearch.

Dodatkowo zaobserwowano, że im większy zbiór danych tym różnica pomiędzy wynikami TabuSearch i Symulowanego Wyżarzania jest większa na korzyść drugiego z algorytmów.

Diagram Gantta

kliknij, by otworzyć

Podsumowanie

- Na podstawie zaprojektowanych badań przeprowadzonych z wykorzystaniem autorskiego symulatora udało się stwierdzić, która z autorskich implementacji algorytmów rozwiązujących problem komiwojażera osiągał najlepsze wyniki. Najlepsze - zarówno pod względem osiągniętego wyniku jak i szybkości działania - okazało się Symulowane Wyżarzanie.
- W przyszłości w celu dalszego rozwoju projektu można by udoskonalić implementację algorytmu Branch&Bound. Dzięki temu mógłby on pełnić rolę benchmarka w eksperymencie. Być może dobrym pomysłem byłoby także rozszerzenie testów o dodatkowe algorytmy. Warto byłoby również pomyśleć o jeszcze większym zróżnicowaniu zbiorów danych testowych. Istnieje wtedy możliwość na zbadanie wszystkich algorytmów w szerszym zakresie danych i odkrycie ich cech ujawniających się dopiero na specyficznych zbiorach danych.

Literatura - źródła

1. R. Ahuja, O. Ergun, J. Orlin, A. Punnen, A survey of very large-scale neighborhood search techniques, Discrete Applied Mathematics 123 (2002) 75–102.
2. S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by Simulated Annealing, Science 220 (1983) 671–680.
3. F. Glover, T. Laguna, Tabu search, Kluwer Academic Publishers, 1997.
4. R. Wieczorkowski, Algorytmy stochastyczne w optymalizacji dyskretnej przy zaburzonych wartościach funkcji, Matematyka Stosowana 38 (1995) 119–153.
5. Balas, Egon, and Paolo Toth. "Branch and bound methods for the traveling salesman problem." (1983).
6. Zaawansowane programowanie, Wykład 5: Algorytmy Dokładne, prof. dr hab. inż. Marta Kasprzak, Instytut Informatyki, Politechnika Poznańska