AGH, Kraków, 09.01.2023 r.



Badania Operacyjne 2

Zagadnienie optymalizacji wyboru sprzedawców w sklepie internetowym na podstawie zadanej listy zakupów z wykorzystaniem algorytmu ewolucyjnego

Prowadzący:  
dr inż. Wojciech Chmiel  
Autorzy:  
Łukasz Gakan  
Wiktor Kacprzak  
Mateusz Król

Spis Treści

[1. Model zagadnienia 3](#_Toc124193794)

[1.1. Wstęp teoretyczny 3](#_Toc124193795)

[1.2. Struktury danych 4](#_Toc124193796)

[1.3. Model matematyczny 4](#_Toc124193797)

[2. Algorytm 6](#_Toc124193798)

[2.1. Pseudokod 6](#_Toc124193799)

[2.2. Opracowane elementy algorytmu 7](#_Toc124193800)

[2.3. Parametry algorytmu 10](#_Toc124193801)

[3. Aplikacja 11](#_Toc124193802)

[3.1. Uruchamianie 11](#_Toc124193803)

[3.2. Format danych oraz interfejsy 11](#_Toc124193804)

[3.3. Opis funkcjonalności 11](#_Toc124193805)

[4. Testy 11](#_Toc124193806)

[4.1. Wykaz scenariuszy 11](#_Toc124193807)

[4.2. Wyniki testów i charakterystyka instancji 12](#_Toc124193808)

Podział pracy

|  |  |  |  |
| --- | --- | --- | --- |
| Etap prac | Łukasz Gakan | Wiktor Kacprzak | Mateusz Król |
| Model zagadnienia |  |  |  |
| Algorytm opracowanie |  |  |  |
| Implementacja aplikacji |  |  |  |
| Testy |  |  |  |
| Dokumentacja |  |  |  |

# Model zagadnienia

## Wstęp teoretyczny

Międzynarodowa strona internetowa Bricklink zajmuje się zrzeszaniem sprzedawców i kupców wycofanych z produkcji jak i wciąż produkowanych zestawów LEGO. Sprzedawcy tworzą indywidualne katalogi uzupełniając je o dostępne w ich asortymencie zestawy bądź pojedyncze elementy zestawów. Zarówno sprzedawcy jak i kupcy mogą zbywać/nabywać towary wyłącznie w obrębie bazy produktów dostępnej na stronie internetowej.

W związku z dość skomplikowanym charakterem wyboru sprzedawców dla interesujących klienta produktów postanowiono zoptymalizować ten proces. Celem optymalizacji doboru dostawców było wyznaczenie na podstawie zadanej "listy zakupów" najtańszego zestawu artykułów z inwentarza, biorąc pod uwagę zarówno cenę produktu, jak i cenę przesyłki. Optymalizacja jest zatem jednokryterialna - minimalizacja sumarycznej ceny produktów i przesyłek.

Dodatkowe uwarunkowania wpływające na znalezienie najlepszego rozwiązania to m. in. wprowadzanie przez indywidualnych sprzedawców tzw. zniżek ilościowych pozwalających na zmniejszenie ceny zamówionych u niego artykułów na podstawie ich sumarycznej ilości. Następnie pod uwagę należało wziąć zmienną cenę przesyłki, która silnie zależy od sumarycznej masy zamówionych produktów, przy czym zakłada się, że sprzedawcy nie łączą swoich zamówień w celu obniżenia ceny przesyłki dla klienta. Istotnym aspektem był również fakt, że zwiększenie ilości produktów kupionych u jednego sprzedawcy, zwiększa cenę przesyłki, jednak zmniejszenie tej ilości wpływa na zwiększenie ogólnej liczby sprzedawców, a co za tym idzie ilości indywidualnych przesyłek.

Podczas implementacji reprezentacji strony internetowej zastosowano następujące uproszczenia:

* Stosowane okresowe promocje u sprzedawców nie były brane pod uwagę, w cenie produktu w momencie dokonywania wyboru nie istnieje adnotacja, że jest to cena po obniżce;
* Utrzymywane przez niektórych sprzedawców minimalne wartości zamówień nie były przestrzegane, ponieważ nawet w rzeczywistych przypadkach najczęściej niemożliwym jest kupienie pewnych przedmiotów, jeśli takowe znajdują się w inwentarzu tylko jednego sprzedawcy, który taką minimalną wartość ma dość wysoką;
* Wskazane na liście zakupów produkty traktowane były jako wyłącznie kompletne/nieuszkodzone. Jest to o tyle istotne, że na rzeczywistej stronie można kupić nawet takie produkty, a wtedy ich cena jest nierealistycznie zaniżona;
* Wyeliminowano informację o lokalizacji sprzedawców, zamiast tego zastosowano ich regionalizację poprzez odpowiednią modyfikację ceny przesyłki, np. 5% sprzedawców mieszka w USA, zatem dla nich cena przesyłki dla klienta domyślnie zlokalizowanego w Polsce będzie większa niż dla 5% sprzedawców zamieszkałych w Niemczech.

## Struktury danych

Danymi wejściowymi w zastosowanym algorytmie były:

* Lista zakupów – przechowywana w postaci listy par, pewnego numeru ID produktu, wraz z zamówioną ilością;
* Bazy danych strony internetowej – początkowo planowano zastosować tylko dwa rodzaje baz: produkty i sprzedawcy, jednakowoż okazało się, że będą one niewystarczające, a ich względna nieczytelność doprowadziła do zwiększenia ilości baz do 3 ([???] = [big,medium,small]):
  + **[???]\_database.csv** – przechowująca informacje o ilości i cenie danych produktów w inwentarzach wszystkich sprzedawców;
  + **[???]\_unique\_items\_file.csv** – przechowująca informacje o wszystkich nazwach produktów oraz ich wadze;
  + **[???]\_unique\_sellers.csv** – przechowująca informacje o cenach przesyłek i funkcjach zniżek ilościowych dla wszystkich sprzedawców;

Co ważne, zostały one stworzone w trzech wariantach, co umożliwiło zmianę rozmiaru problemu na poziomie testów algorytmu. Różniły się one zarówno rozmiarem jak i różnorodnością, gdzie przez rozmiar rozumie się ilość produktów x ilość sprzedawców, a przez różnorodność rozumie się m. in. regionalizację, zachowanie różnorodności cen produktów w regionie. Stworzone warianty to:

1. Przestrzeń mała, z zachowaniem różnorodności (rozmiar 20x15);
2. Przestrzeń duża, zachowaniem różnorodności (rozmiar 50x30);
3. Przestrzeń średnia, bez zachowania różnorodności (rozmiar 30x20).

* Zadany maksymalny budżet – jego wartość posłużyła do rozpoznania rozwiązań niedopuszczalnych i zastosowania funkcji kary.

## Model matematyczny

**Postać rozwiązania:**

Macierz w rozmiarze  *n x m*, gdzie *n* – liczba produktów w inwentarzu strony internetowej, *m* – liczba sprzedawców w bazie sprzedawców. Macierz wypełniona będzie wartościami typu całkowitoliczbowego, określającymi ilość danego produktu zamówionego u danego sprzedawcy, np., oznacza, że zamówiono produkt o ID = 4 u sprzedawcy o ID = 5 w ilości 2 sztuki.

**Funkcja celu:**

– i-ta należność za produkt z listy u j-tego sprzedawcy [PLN]

– j-ta należność za przesyłkę (pobierana jest z informacji o przesyłkach przechowywanej przez obiekt klasy Seller) [PLN]

𝑛 – liczba produktów na liście [-]

𝑚 – liczba sprzedawców [-]

**Funkcja kary:**

Funkcja kary wynika bezpośrednio z warunku ograniczającego, mówiącego, że maksymalny określony budżet, który klient chce przeznaczyć na swoje zakupy, musi być mniejszy od wartości funkcji celu obliczonej dla najlepszego rozwiązania.

**Funkcja kryterialna:**

**Cena produktu:**

– bazowa cena produktu u wybranego sprzedawcy [PLN]

ℎ(𝑛′) – funkcja określająca zniżkę na podstawie zamówionej ilości produktów

Przyjmujemy, że w przypadku braku zniżek ilościowych u danego sprzedawcy ℎ(𝑛′) jest funkcją stałą, równą 1.

Przykładowa postać funkcji ℎ(𝑛′):

𝑛′ - liczba przedmiotów zamówionych od tego samego sprzedawcy.

**Cena przesyłki:**

Należność za przesyłkę obliczana jest na podstawie funkcji wykazującej relację między wartością przesyłki, a łączną wagą zamówienia.

Przykładowe funkcje cenowe dla przesyłek:

1° - Funkcja nieciągła

2° - Funkcja ciągła

– łączna waga produktów [g]

# Algorytm

## Pseudokod

Opracowany w ramach projektu algorytm ewolucyjny pozwala przy zastosowaniu narzędzi tj. selekcja, krzyżowanie i mutacja odszukać przybliżoną lokalizację minimum funkcji kryterialnej. Pseudokod przedstawiony został poniżej:

## Opracowane elementy algorytmu

Wstępna wersja algorytmu zakładała możliwość wykorzystania wspomnianych narzędzi w taki sposób, aby rozwiązania uzyskiwane przez krzyżowania i mutacje mogły być niedopuszczalne. Ich niedopuszczalność wynikała z zamówienia niewystarczającej lub zbyt dużej ilości danych produktów, co natomiast wiązało się z koniecznością ciągłej walidacji rozwiązań, znacznie zwiększając przy tym złożoność obliczeniową całego algorytmu. Biorąc jednak pod uwagę to, że otrzymywanie w populacjach rozwiązań niedopuszczalnych, jest atutem w kontekście lepszego przeszukiwania przestrzeni rozwiązań, a co za tym idzie zwiększonego prawdopodobieństwa odnalezienia punktu optymalnego, postanowiono zachować tę cechę, jednakże w znacznym stopniu ją ograniczając. Opracowane sposoby krzyżowania i mutacji umożliwiają otrzymanie rozwiązań niedopuszczalnych z rozwiązań dopuszczalnych, jak i odwrotnie, jednakże zapewniają stałą ilość zamówionych produktów, a ich niedopuszczalność określana jest wyłącznie przez niezerową wartość funkcji kary.

1. Metody selekcji

Metoda elitarna – selection\_elite(population, amount, budget)

Przyjmuje 3 parametry: populacja w formie listy macierzy rozwiązań, maksymalny budżet, który klient może przeznaczyć na zakup klocków oraz ilość rodziców   
w formie liczby procentowej, którzy zostaną wybrani z tej populacji.

Algorytm: Dla każdej macierzy obliczamy wartości funkcji celu z uwzględnieniem funkcji kary i zapisujemy je do listy przechowującej krotki (wartość funkcji celu, macierz rozwiązania). Następnie sortujemy tę listę rosnąco po wartościach funkcji celu i zwracamy tyle pierwszych elementów na ile pozwala parametr ilość rodziców

Metoda turniejowa – selection\_tournament(population, amount, budget)

Tak samo jw. przyjmuje 3 parametry,

Algorytm: Tak samo jw. tworzymy listę krotek. Następnie w pętli while porównujemy elementy stojące obok siebie i gorszego z nich, czyli tego o większej wartości funkcji celu, usuwamy z tej listy. Wykonujemy to dopóki w liście zostanie nam liczba elementów równa parametrowi określającemu liczbę rodziców.

Metoda rankingowa – selection\_ranking(population, amount, budget)

Tak samo jw. przyjmuje 3 parametry,

Algorytm: tak samo jak wyżej tworzymy listę krotek. Następnie na podstawie wartości funkcji celu tworzymy nową listę krotek, w której lepsze elementy występują więcej razy od tych gorszych (najlepszy występuje tyle razy, ile wynosi liczebność populacji, następny o 1 mniej razy, itd.). Zatem prawdopodobieństwo wylosowania lepszego elementu z tej listy jest większe niż gorszego zgodnie   
z definicją tej metody. Na koniec losujemy określoną liczbę rodziców, uwzględniając to, że po wylosowaniu jednego usuwamy wszystkie jego wystąpienia w starej liście.

1. Metody krzyżowania

Krzyżowanie „crossover\_every\_2nd(matrix\_1, matrix\_2)” – zamiana naprzemiennych rzędów

matrxi\_1 matrix\_2

Krzyżowanie “ crossover\_basic(matrix\_1, matrix\_2, order\_length=5, choice=’random’, ‘mode=’rows\_idx\_and\_number’)”

matrxi\_1 matrix\_2

Ta metoda krzyżowania przyjmuje 2 parametry macierzowe oraz 3 opisane szczegółowo poniżej:

* **order\_length**

liczba unikalnych produktów na liście zakupów

* **choice = [‘random’, ‘choice’]**

Parametr określający sposób wyboru rzędów do krzyżowania:

1. **random**

W zależności od parametru **mode** wybierane są rzędy z elementem losowości

1. **choice**

Użytkownik sam wybiera rzędy

* **mode = [‘rows\_idx’, ‘rows\_number’, ‘rows\_idx\_and\_number’]**

1. **rows\_idx**

Wybór **order\_length//2** rzędów w sposób losowy

np. mając order\_length=5, losowane są 2 rzędy

1. **rows\_number**

Wybieranie co drugiego rzędu, aż ich ilość będzie równa losowej liczbie całkowitej z zakresu **(1, order\_length)**

np. mając order\_length=5 i wylosowując liczbę 2 wybierane są wiersze 1 i 3

1. **rows\_idx\_and\_number**

Wylosowanie zarówno rzędów jak ich ilości (z zakresu **(1, order\_length)**)

np. . mając order\_length=5 i wylosowując liczbę 2 losujemy jakie rzędy zostaną wzięte pod uwagę (np. 2 i 3)

1. Metody mutacji

Mutacja jednostkowa

**mutate\_singular\_product(sol\_matrix: np. array) -> np.array**

Parametry:

sol\_matrix – macierz rozwiązania (osobnik) w populacji [struktura typu numpy.array]

Wyjście: sol\_matrix - osobnik po mutacji [struktura typu numpy.array]

Sposób mutowania osobnika oparty na losowym wyborze rzędu macierzy osobnika (czyli losowego produktu) oraz na losowym przeniesieniu między kolumnami (czyli między konkretnymi sprzedawcami, u których produkt jest dostępny) pojedynczej sztuki produktu.

Mutacja z eliminacją

**mutate\_with\_seller\_elimination(sol\_matrix: np.array) -> np.array**

Parametry:

sol\_matrix – macierz rozwiązania (osobnik) w populacji [struktura typu numpy.array]

Wyjście: sol\_matrix - osobnik po mutacji [struktura typu numpy.array]

Sposób mutowania osobnika oparty na losowym wyborze rzędu macierzy osobnika (czyli losowego produktu) oraz na losowym wyborze kolumny (konkretnego sprzedawcy, u którego dostępny jest produkt), której wartość zostanie zmniejszona na jeden z dwóch sposobów:

* wyzerowanie – jeśli pozostali sprzedawcy są w stanie zaspokoić zapotrzebowanie
* maksymalne zmniejszenie wartości – jeśli pozostali sprzedawcy nie są w stanie zaspokoić zapotrzebowania w całości

1. Generacja populacji początkowej

Dla zadanego rozmiaru populacji w zmiennej population\_size tworzone są rozwiązania w ramach klasy Individual, zawierającej zarówno wartość funkcji celu jak i macierz osobnika. Każda macierz osobnika tworzona jest w sposób losowy, a przy tym deterministyczny. Dla każdego produktu wskazanego w liście zakupów spośród sprzedawców oferujących ten produkt losuje się zarówno sprzedawców, z których usług będzie się korzystać, jak i konkretne ilości przedmiotów u nich zamawianych.

## Parametry algorytmu

Pełen zestaw parametrów stworzonego algorytmu zawarty został w Tabeli 2-1.

Tabela ‑ Zestawienie parametrów algorytmu

|  |  |  |
| --- | --- | --- |
| L.p. | Symbol/nazwa | Opis słowny |
| 1 | population\_size | Rozmiar populacji, wartość stała w trakcie działania algorytmu, określa liczbę osobników w populacji |
| 2 | selection\_method | Odpowiada za wybór typu selekcji, która zostanie użyta w algorytmie |
| 3 | parent\_percentage | Liczba z zakresu (0;1], określająca, jaki procent populacji będzie miał w danej iteracji się mutować i/lub krzyżować |
| 4 | chance\_to\_crossover | Liczba z zakresu [0;1], określająca, jakie jest prawdopodobieństwo wykonania krzyżowania osobników w populacji przejściowej (rodziców) |
| 5 | crossover\_type | Odpowiada za wybór typu krzyżowania, które zostanie użyte w algorytmie |
| 6 | mutation\_type | Odpowiada za wybór typu mutacji, która zostanie użyta w algorytmie |
| 7 | max\_iters | Maksymalna ilość iteracji algorytmu |
| 8 | iters\_without\_change | Maksymalna ilość iteracji algorytmu, jaką pojedyncze najlepsze rozwiązanie musi pozostawać bez zmian, aby spełniony został drugi warunek stopu |
| 9 | budget | Maksymalny budżet klienta – bezpośrednio wpływający na funkcję kary |

Podczas implementacji algorytmu zauważono, że rozmiar i struktura listy zakupów (ilość unikalnych produktów i ilość sztuk tych produktów) bardzo silnie wpływa na kształtowanie się przestrzeni rozwiązań. Co ważne, postać listy zakupów jest w przypadku rozpatrywanego zagadnienia częścią projektowanej instancji, a zatem wpływa na te dwie sprawy w sposób nierozłączny, dlatego też postanowiono nie kwalifikować jej do parametrów algorytmu. Podobną decyzję podjęto także w przypadku wspomnianych w poprzednim rozdziale 3 rodzajów baz danych, które nie tyle wpływają na sposób działania algorytmu, co, podobnie jak lista zakupów, kreują przestrzeń rozwiązań, komplikując, bądź upraszczając algorytmowi jego zadanie.

# Aplikacja

## Uruchamianie

## Format danych oraz interfejsy

## Opis funkcjonalności

# Testy

## Wykaz scenariuszy

W ramach projektu zaprojektowano i zrealizowano 14 testów, umożliwiających zarówno badanie wpływu parametrów algorytmu na sposób jego wykonywania oraz jakość rozwiązań, jak i sprawdzenie wpływu czynników kreujących przestrzeń rozwiązań tj. postać listy zakupów na sprawność jego wykonywania. Wykaz testów przedstawiony został poniżej:

1. Test\_01 – badanie zależności jakości rozwiązania od rozmiaru populacji
2. Test\_02 – badanie zależności jakości rozwiązania od rozmiaru populacji przejściowej
3. Test\_03 – badanie zależności jakości rozwiązania od szansy na wykonanie krzyżowania
4. Test\_04 – badanie wpływu wyboru metody selekcji na jakość rozwiązania
5. Test\_05 – badanie wpływu wyboru metody mutacji na jakość rozwiązania
6. Test\_06 – badanie wpływu wyboru metody krzyżowania na jakość rozwiązania
7. Test\_07 – znalezienie najlepszej i najgorszej kombinacji operatorów selekcji, krzyżowania i mutacji dla zadanej instancji testowej
8. Test\_08 – znalezienie najlepszej i najgorsze kombinacji operatorów (j. w.) oraz parametrów algorytmu dla zadanej instancji testowej
9. Test\_09 – badanie zależności jakości rozwiązania od rozmiaru problemu, mała liczba unikalnych produktów z podziałem na metody selekcji
10. Test\_10 – badanie zależności jakości rozwiązania od rozmiaru problemu, duża liczba unikalnych produktów
11. Test\_11 – badanie zależności jakości rozwiązania dla różnej ilości unikalnych produktów w liście zakupów, przy zachowaniu stałej sumarycznej ilości produktów
12. Test\_12 – badanie zależności jakości rozwiązania dla różnej ilości unikalnych produktów od wybranej metody mutacji
13. Test\_13 – badanie złożoności czasowej w zależności od postaci listy zakupów
14. Test\_14 – badanie działania algorytmu dla przypadku złośliwego, ze znanym rozwiązaniem optymalnym

W wykonywanych testach brano pod uwagę wartość funkcji kryterialnej najlepszego znalezionego rozwiązania oraz ilość iteracji, po której to rozwiązanie zostało znalezione przez algorytm.

## Wyniki testów i charakterystyka instancji

Podczas wykonywania testów skorzystano z różnych wariantów populacji początkowej oraz pewnego stałego wektora parametrów algorytmu. Warianty populacji oraz zestawienie parametrów przedstawione zostały w Tabelach 4-1.-4-6. Należy nadmienić, że przedstawione wartości parametrów są wartościami domyślnymi, tzn. jeśli dane badanie nie uwzględnia zmienności danego parametru w celu wyznaczenia jakiejś tendencji, to przez wszystkie iteracje przyjmowana jest wartość domyślna.

Tabela ‑ Zestawienie parametrów algorytmu zastosowanych w testach

|  |  |  |
| --- | --- | --- |
| L.p. | Symbol/nazwa | Wartość |
| 1 | population\_size | 10 |
| 2 | selection\_method | ranking |
| 3 | parent\_percentage | 0.3 |
| 4 | chance\_to\_crossover | 0.2 |
| 5 | mutation\_type | singular |
| 6 | max\_iters | 300 |
| 7 | iters\_without\_change | 30 |
| 8 | budget | 1000 |

Tabela ‑ Parametry populacji początkowej dla testów: 1.1, 2, 7, 8, 10, 12.1, 13

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Funkcje celu dla populacji początkowej | | | | | | | | | | | | |
| 2936.06 | 3171.68 | 2760.50 | | 3351.97 | 2585.49 | | 3121.30 | 3385.85 | 2273.09 | | 1928.89 | 3177.07 |
| Średnia | | | Wartość MAX | | | Wartość MIN | | | | Odchyl. Stand. | | |
| 2869.19 | | | 3385.85 | | | 1928.89 | | | | 385.758 | | |

Tabela ‑ Parametry populacji dla testu 1.2 przy rozmiarze populacji równym 20

|  |  |  |  |
| --- | --- | --- | --- |
| Średnia | Wartość MAX | Wartość MIN | Odchyl. Stand. |
| 3124.16 | 3512.66 | 1673.20 | 418.78 |

Tabela ‑ Parametry populacji dla testu 1.3 przy rozmiarze populacji równym 40

|  |  |  |  |
| --- | --- | --- | --- |
| Średnia | Wartość MAX | Wartość MIN | Odchyl. Stand. |
| 2611.86 | 3722.41 | 1533.12 | 350.23 |

Tabela ‑ Parametry populacji początkowej dla testów: 3, 4, 5, 6, 11, 12.2, 14

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Funkcje celu dla populacji początkowej | | | | | | | | | | | | |
| 1261.49 | 3337.96 | 2646.83 | | 1148.48 | 3165.69 | | 3568.29 | 2473.03 | 1510.64 | | 2927.44 | 1400.75 |
| Średnia | | | Wartość MAX | | | Wartość MIN | | | | Odchyl. Stand. | | |
| 2344.06 | | | 3568.29 | | | 1148.48 | | | | 810.98 | | |

Tabela ‑ Parametry populacji początkowej dla testu: 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Funkcje celu dla populacji początkowej | | | | | | | | | | | | |
| 3382.83 | 3305.11 | 6464.31 | | 4803.97 | 4030.23 | | 3340.64 | 3334.48 | 3384.86 | | 4607.39 | 2397.72 |
| Średnia | | | Wartość MAX | | | Wartość MIN | | | | Odchyl. Stand. | | |
| 3905.15 | | | 6464.31 | | | 2397.72 | | | | 857.06 | | |

### Test\_01 - badanie zależności jakości rozwiązania od rozmiaru populacji

**Autor:** Mateusz Król  
**Cel badania:** Wykrycie i scharakteryzowanie tendencji między rosnącą liczebnością populacji, a jakością rozwiązania.

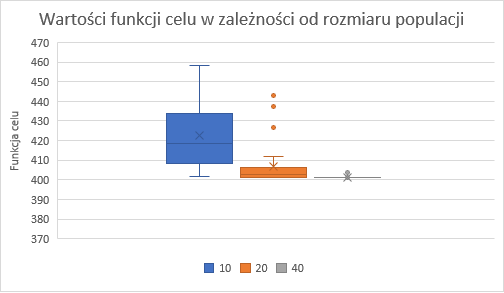


Fig. ‑ Wykres wartości funkcji celu w zależności od rozmiaru populacji

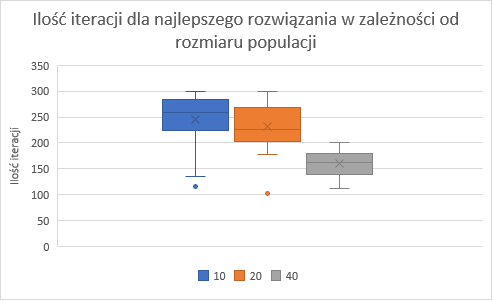


Fig. ‑ Wykres ilości iteracji w zależności od rozmiaru populacji

Tabela ‑ Wartości statystyczne dla rozmiaru populacji=10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Populacja = 10 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 422.65 | 458.47 | 401.97 | 13.97 | 5.14% |
| Ilość iteracji | 246.27 | 300 | 116 | 37.48 | - |

Tabela ‑ Wartości statystyczne dla rozmiaru populacji=20

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Populacja = 20 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 406.85 | 443.16 | 401.06 | 6.60 | 1.44% |
| Ilość iteracji | 232.23 | 299 | 103 | 35.52 | - |

Tab. 5. Wartości statystyczne dla rozmiaru populacji=40

Tabela ‑ Wartości statystyczne dla rozmiaru populacji=40

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Populacja = 40 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 401.32 | 404.01 | 401.06 | 0.45 | 0.07% |
| Ilość iteracji | 161.03 | 201 | 111 | 20.17 | - |

**Interpretacja:**

Najniższą wartość funkcji kryterialnej osiągnięto dla najbardziej liczebnej populacji. Dodatkowo wyniki dla rozmiaru równego 40 osobników miały najmniejszą rozbieżność – wartość odchylenia standardowego poniżej 1. Dodatkowo błąd względny okazał się być niewielki. Dzięki zastosowaniu większej ilości osobników zmniejszeniu uległa średnia liczba iteracji. Na podstawie wyników badania można wysunąć wniosek, że zwiększenie liczebności populacji wpływa korzystnie na efektywność algorytmu.

### Test\_02 - badanie zależności jakości rozwiązania od rozmiaru populacji przejściowej

**Autor:** Mateusz Król  
**Cel badania:** Wykrycie i scharakteryzowanie tendencji między rosnącą liczebnością populacji przejściowej, a jakością rozwiązania.

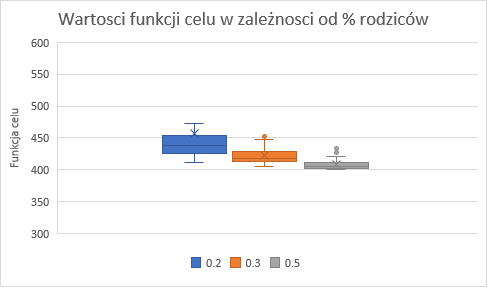


Fig. ‑ Wykres wartości funkcji celu w zależności od rozmiaru populacji przejściowej

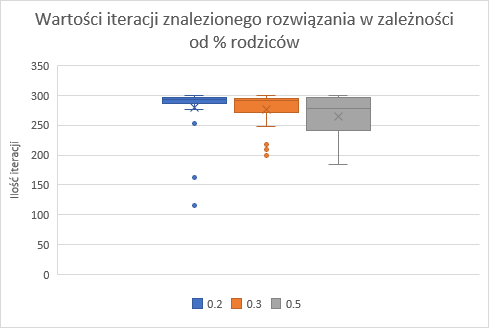


Fig. ‑ Wykres ilości potrzebnych iteracji w zależności od rozmiaru populacji przejściowej

Tabela ‑ Wartości statystyczne dla procentu rodziców = 20%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| % rodziców = 20% | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 457.36 | 821.2 | 411.87 | 37.80 | 11.04% |
| Ilość iteracji | 280 | 300 | 116 | 22.60 | - |

Tabela ‑ Wartości statystyczne dla procentu rodziców = 30%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| % rodziców = 30% | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 422.20 | 451.43 | 404.37 | 9.81 | 4.41% |
| Ilość iteracji | 276.23 | 300 | 199 | 22.39 | - |

Tabela ‑ Wartości statystyczne dla procentu rodziców = 50%

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| % rodziców = 50% | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 408.45 | 433.55 | 401.11 | 6.61 | 1.83% |
| Ilość iteracji | 265.33 | 300 | 184 | 29.64 | - |

**Interpretacja:**

Najniższe wartości funkcji kryterialnej osiągnięto dla największej procentowej liczebności populacji przejściowej, czyli dla 50% liczebności populacji startowej. Wraz ze zwiększeniem liczebności spada wartość odchylenia standardowego, wskazując na zmniejszenie rozpiętości wyników algorytmu. Co ciekawe, odchylenie standardowe ilości iteracji rosło w miarę zwiększania liczebności populacji przejściowej. Na podstawie wyników można stwierdzić, że lepsze rezultaty otrzymałoby się dla większej maksymalnej liczby iteracji w algorytmie, na co wskazuje spłaszczenie wykresów blisko wartości granicznej równej 300.

### Test\_03 – badanie zależności jakości rozwiązania od szansy na wykonanie krzyżowania

**Autor:** Łukasz Gakan  
**Cel badania:** Wykrycie i scharakteryzowanie tendencji między rosnącym prawdopodobieństwem wykonania krzyżowania osobników, a jakością rozwiązania.

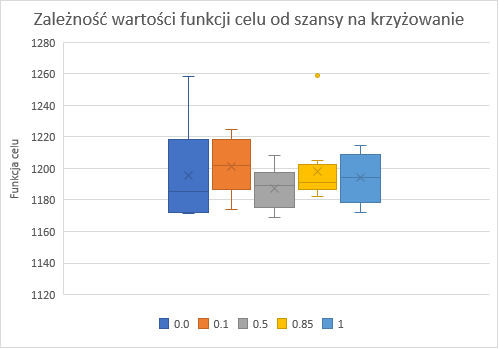


Fig. ‑ Wykres zależności funkcji celu od szansy na krzyżowanie.

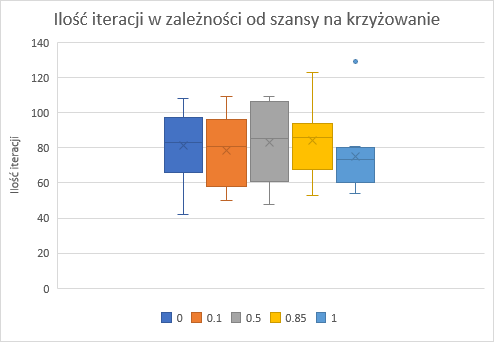


Fig. ‑ Wykres ilości iteracji algorytmu w zależności od szansy na krzyżowanie.

Tabela ‑ Wartości statystyczne dla szansy krzyżowania = 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Szansa krzyżowania = 0 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1195.57 | 1258.24 | 1171.70 | 22.02 | 2.04% |
| Ilość iteracji | 81.10 | 108 | 116 | 15.90 | - |

Tabela ‑ Wartości statystyczne dla szansy krzyżowania = 0.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Szansa krzyżowania = 0.1 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1201.57 | 1224.85 | 1173.65 | 13.69 | 2.38% |
| Ilość iteracji | 78.5 | 109 | 50 | 17.30 | - |

Tabela ‑ Wartości statystyczne dla szansy krzyżowania = 0.5

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Szansa krzyżowania = 0.5 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1187.38 | 1208.45 | 1168.94 | 10.89 | 1.58% |
| Ilość iteracji | 83 | 109 | 48 | 21.00 | - |

Tabela ‑ Wartości statystyczne dla szansy krzyżowania = 0.85

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Szansa krzyżowania = 0.85 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1198.36 | 1258.70 | 1182.33 | 14.11 | 1.36% |
| Ilość iteracji | 84 | 123 | 53 | 15.60 | - |

Tabela ‑ Wartości statystyczne dla szansy krzyżowania = 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Szansa krzyżowania = 1 | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1194.41 | 1214.74 | 1172.30 | 14.45 | 1.89% |
| Ilość iteracji | 75.1 | 129 | 54 | 13.70 | - |

**Interpretacja:**

Na podstawie badania nie udało się wykazać bezpośredniego liniowego wpływu wartości prawdopodobieństwa na wykonanie krzyżowania na jakość rozwiązania. Dysponując otrzymanymi wynikami niemożliwym jest wytyczenie tendencji ani dla wartości funkcji kryterialnej, ani dla ilości iteracji, zatem przyjęto, że rola krzyżowania w przypadku rozpatrywanego algorytmu ma charakter wspomagający i musi być wartość tego prawdopodobieństwa musi być dopasowana do pozostałych parametrów algorytmu.

### Test\_04 – badanie wpływu wyboru metody selekcji na jakość rozwiązania

**Autor:** Mateusz Król  
**Cel badania:** Scharakteryzowanie rezultatów uzyskiwanych przez algorytm dla różnych typów selekcji

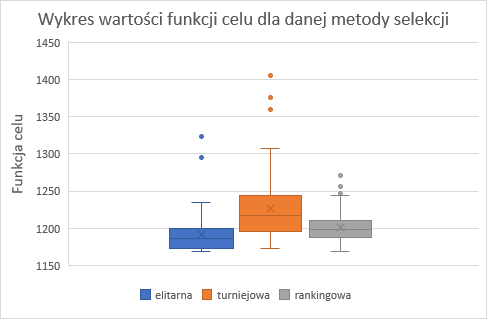


Fig. ‑ Wykres wartości funkcji celu w zależności od wybranej metody selekcji.

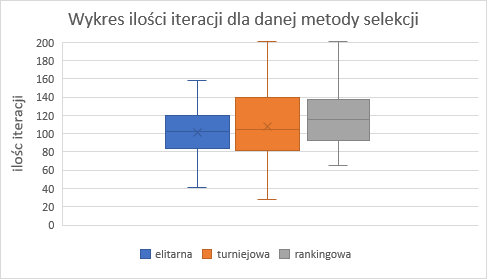


Fig. ‑ Wykres ilości iteracji w zależności od metody selekcji

Tabela ‑ Wartości statystyczne dla metody elitarnej

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metoda elitarna | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1191.35 | 1323.25 | 1168.94 | 17.57 | 1.92% |
| Ilość iteracji | 101.15 | 158 | 41 | 22.15 | - |

Tabela ‑ Wartości statystyczne dla metody turniejowej

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metoda turniejowa | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1226.86 | 1404.94 | 1172.30 | 33.10 | 4.65% |
| Ilość iteracji | 107.37 | 201 | 28 | 31.59 | - |

Tabela ‑ Wartości statystyczne dla metody rankingowej

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metoda rankingowa | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1201.42 | 1270.67 | 1168.94 | 15.77 | 2.78% |
| Ilość iteracji | 119.27 | 201 | 65 | 26.21 | - |

**Interpretacja:**

Najlepsze wyniki zarówno pod względem wartości funkcji kryterialnej, jak i ilości iteracji osiągnięto dla selekcji elitarnej, jednakże wyniki dla pozostałych metod nie są dużo gorsze, zatem ciężko definitywnie wnioskować, że wskazana metoda jest najlepsza, bo możliwe, że akurat w przypadku tej przestrzeni rozwiązań poradziła sobie ona najlepiej. Dodatkowo na uwagę zasługuje wartość odchylenia standardowego dla wartości funkcji kryterialnej dla selekcji rankingowej, której wartość jest najniższa, zatem otrzymanie oczekiwanego wyniku najpewniejsze. Najgorsze wyniki otrzymano przy użyciu selekcji turniejowej, jednak średnia wyników funkcji kryterialnej różni się nie więcej niż 30 PLN od pozostałych dwóch.

### Test\_05 - badanie wpływu wyboru metody mutacji na jakość rozwiązania

**Autor:** Wiktor Kacprzak  
**Cel badania:** Scharakteryzowanie rezultatów uzyskiwanych przez algorytm dla różnych typów mutacji

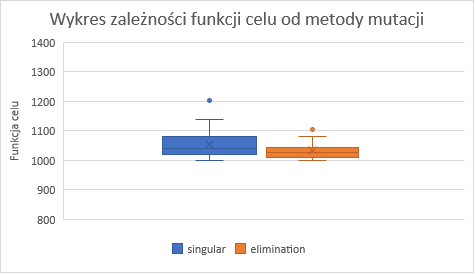


Fig. ‑ Wykres zależności funkcji celu od metody mutacji

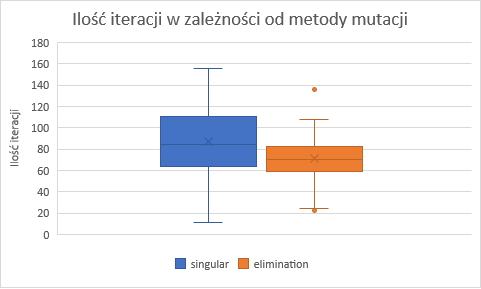


Fig. ‑ Wykres zależności ilości iteracji od metody mutacji

Tabela ‑ Wartości statystyczne dla mutacji pojedynczej

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mutacja pojedyncza | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1055.00 | 1203.32 | 1000.10 | 34.88 | 5.49% |
| Ilość iteracji | 87.43 | 156 | 11 | 29.73 | - |

Tabela ‑ Wartości statystyczne dla mutacji z eliminacją

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mutacja z eliminacją | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1033.14 | 1114.36 | 1000.35 | 20.64 | 3.28% |
| Ilość iteracji | 71.67 | 136 | 23 | 17.64 | - |

**Interpretacja:**

Lepsze wyniki zostały osiągnięte dla mutacji z eliminacją. Wskazuje na to zarówno niższa średnia wartość funkcji celu, jaki i jej mniejsze rozproszenie. Podobny efekt odnotowano na wykresie ilości iteracji potrzebnej do znalezienia końcowego rozwiązania. Wartości dla mutacji   
z eliminacją są niższe i dużo mniej rozproszone.

### Test\_06 - badanie wpływu wyboru metody krzyżowania na jakość rozwiązania

**Autor:** Łukasz Gakan  
**Cel badania:** Scharakteryzowanie rezultatów uzyskiwanych przez algorytm dla różnych typów krzyżowania

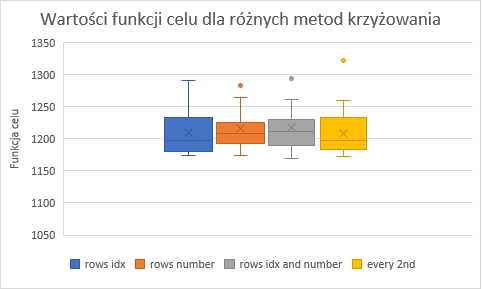


Fig. ‑ Wykres zależności funkcji celu od metody krzyżowania.

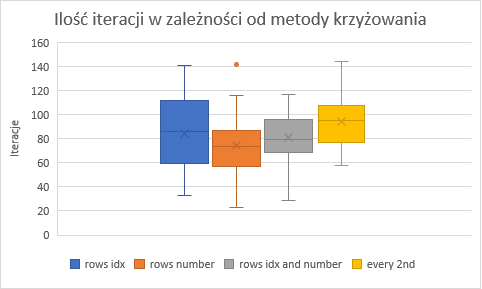


Fig. ‑ Wykres zależności funkcji celu od metody krzyżowania.

Tabela ‑ Wartości statystyczne dla krzyżowania rows\_idx

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Krzyżowanie rows\_idx | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1209.03 | 1290.28 | 1173.22 | 27.02 | 3.05% |
| Ilość iteracji | 84.15 | 141 | 33 | 24.44 | - |

Tabela ‑ Wartości statystyczne dla krzyżowania rows\_number

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Krzyżowanie rows\_number | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1215.22 | 1285.48 | 1173.22 | 23.82 | 3.58% |
| Ilość iteracji | 74.70 | 142 | 23 | 19.40 | - |

Tabela ‑ Wartości statystyczne dla krzyżowania rows\_idx\_number

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Krzyżowanie rows\_idx\_number | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1217.19 | 1297.94 | 1168.85 | 25.38 | 4.13% |
| Ilość iteracji | 81.10 | 117 | 29 | 17.90 | - |

Tabela ‑ Wartości statystyczne dla krzyżowania every\_2nd

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Krzyżowanie every\_2nd | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1207.94 | 1322.12 | 1171.7 | 27.35 | 3.09% |
| Ilość iteracji | 94.30 | 144 | 58 | 16.50 | - |

**Interpretacja:**

Powyższy wykres funkcji kryterialnej pokazuje, że wszystkie metody krzyżowania działają bardzo podobnie i uzyskiwane przez nie wartości wiele się od siebie nie różnią. Najniższe wartości może osiągać metoda wybierania losowych rzędów rows idx, jednak przez duże rozproszenie wyników nie można uznać jej za najlepszą. W przypadku ilości iteracji, na szczególną uwagę zasługuje metoda rows number, która zarówno średnio, jak i sumarycznie uzyskała najlepsze wyniki.

### Test\_07 - znalezienie najlepszej i najgorszej kombinacji operatorów selekcji, krzyżowania i mutacji dla zadanej instancji testowej

**Autor:** Łukasz Gakan  
**Cel badania:** Wykreowanie na podstawie poprzednich testów optymalnego wektora operacji selekcji, krzyżowania i mutacji oraz wektora potencjalnie najmniej korzystnego ze względu na jakość rozwiązania

Tabela ‑ Opis instancji testowych dla Test\_07

|  |  |  |
| --- | --- | --- |
| Przyjęto: | Najlepszy dobór operacji: | Najgorszy dobór operacji: |
| Procent rodziców = 30% | Selekcja elitarna | Selekcja turniejowa |
| Szansa krzyżowania = 0.5 | Mutacja z eliminacją | Mutacja pojedyncza |
| Rozmiar populacji = 40 | Krzyżowanie – losowe rzędy | Krzyżowanie – przez wybór |

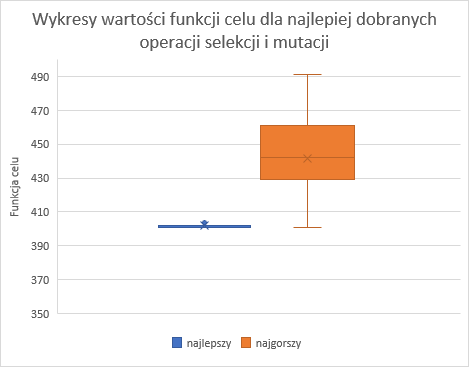


Fig. ‑ Wykres zależności funkcji celu dla najlepszej i najgorszej kombinacji wybranych operacji.

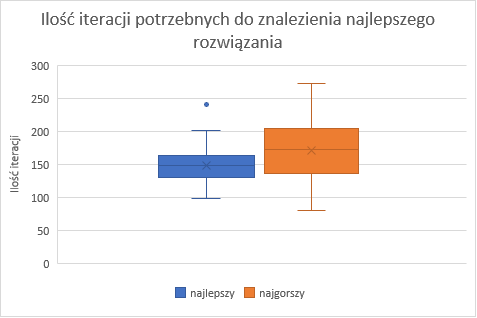


Fig. ‑ Wykres zależności ilości iteracji dla najlepszej i najgorszej kombinacji wybranych operacji.

Tabela ‑ Wartości statystyczne dla najlepszych operacji

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Najlepsze operacje | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 401.76 | 404.01 | 401.06 | 1.05 | 0.17% |
| Ilość iteracji | 148.33 | 241 | 98 | 21.87 | - |

Tabela ‑ Wartości statystyczne dla krzyżowania najgorszych operacji

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Najgorsze operacje | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 441.28 | 490.89 | 401.06 | 18.49 | 10.02% |
| Ilość iteracji | 170.94 | 273 | 81 | 33.88 | - |

**Interpretacja:**

Zgodnie z przewidywaniami, wyniki są skrajnie różne, szczególnie w przypadku wartości funkcji celu. Dla lepszego doboru operacji wartości mieszczą się w bardzo małym zakresie, natomiast w drugim przypadku zwracane są wyniki z bardzo dużą amplitudą, a przy tym w ogóle nie osiągają minimalnej wartości. Dodatkowo dla ilości iteracji również widoczna jest różnica, ponieważ przedział iteracji w lepszym przypadku jest mniejszy.

### Test\_08 - znalezienie najlepszej i najgorsze kombinacji operatorów oraz parametrów algorytmu dla zadanej instancji testowej

**Autor:** Łukasz Gakan  
**Cel badania:** Wykreowanie na podstawie poprzednich testów optymalnego wektora operacji selekcji, krzyżowania i mutacji i parametrów algorytmu oraz wektora potencjalnie najmniej korzystnego ze względu na jakość rozwiązania

Tabela ‑ Opis instancji testowych dla Test\_08

|  |  |
| --- | --- |
| Najlepszy dobór parametrów | Najgorszy dobór parametrów |
| Selekcja elitarna | Selekcja turniejowa |
| Mutacja z eliminacją | Mutacja z eliminacją |
| Krzyżowanie – losowe rzędy | Krzyżowanie – przez wybór |
| Procent rodziców = 50% | Procent rodziców = 20% |
| Szansa krzyżowania = 0.5 | Szansa krzyżowania = 0.1 |
| Rozmiar populacji = 40 | Rozmiar populacji = 10 |

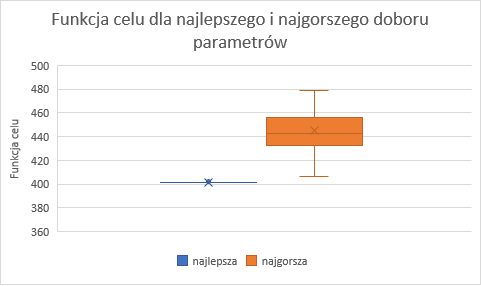


Fig. ‑ Wykres zależności funkcji celu dla najlepszej i najgorszej kombinacji wszystkich parametrów.

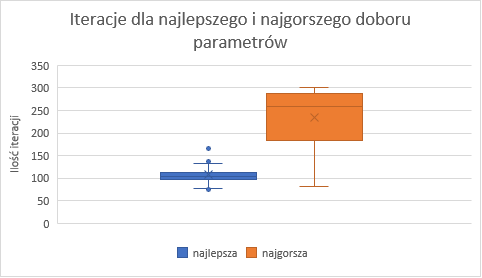


Fig. ‑ Wykres ilości iteracji dla najlepszej i najgorszej kombinacji wszystkich parametrów.

Tabela ‑ Wartości statystyczne dla najlepszych parametrów

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Najlepsze parametry | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 401.52 | 404.01 | 401.06 | 0.73 | 0.12% |
| Ilość iteracji | 108.30 | 166 | 76 | 14.00 | - |

Tabela ‑ Wartości statystyczne dla najgorszych parametrów

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Najgorsze parametry | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 444.81 | 479.23 | 406.79 | 14.93 | 10.25% |
| Ilość iteracji | 233.97 | 300 | 82 | 54.64 | - |

**Interpretacja:**

Również w tym przypadku wyniki dla różnych instancji testowych bardzo się różnią. Dodatnie dodatkowego rozróżnienia w postaci innych wartości parametrów algorytmu doprowadziło do znacznego pogłębienia różnicy między wynikami zarówno dla funkcji kryterialnej, jak i ilości iteracji. Dla najlepszego doboru parametrów optymalne rozwiązanie znajdowane jest znacznej większości iteracji oraz błąd względny jest bliski zeru.

### Test\_09 - badanie zależności jakości rozwiązania od rozmiaru problemu, mała liczba unikalnych produktów z podziałem na metody selekcji

**Autor:** Wiktor Kacprzak  
**Cel badania:** Zbadanie zachowania algorytmu dla listy zakupów z małą ilością unikalnych produktów, ale dużymi ilościami w obrębie jednego produktu

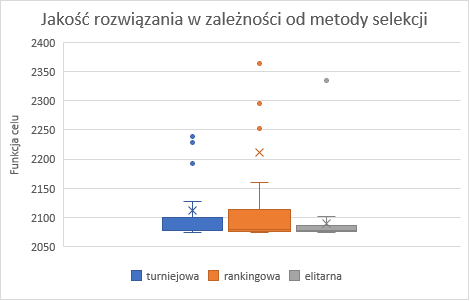


Fig. ‑ Wykres zależności funkcji celu dla różnych metod selekcji przy małej liście unikalnych produktów.

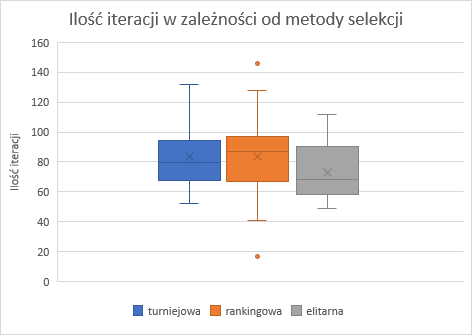


Fig. ‑ Wykres ilości iteracji dla różnych metod selekcji przy małej liście unikalnych produktów.

Tabela ‑ Wartości statystyczne dla turnieju i małej liczby unikalnych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Turniej + mało unikalnych | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 2112.66 | 2424.94 | 2074.19 | 48.77 | 1.85% |
| Ilość iteracji | 83.50 | 132 | 52 | 15.47 | - |

Tabela ‑ Wartości statystyczne dla rankingu i małej liczby unikalnych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ranking + mało unikalnych | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 2211.86 | 3855.05 | 2074.19 | 204.71 | 6.63% |
| Ilość iteracji | 83.70 | 146 | 17 | 19.04 | - |

Tabela ‑ Wartości statystyczne dla elity i małej liczby unikalnych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Elita + mało unikalnych | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 2089.79 | 2334.52 | 2074.19 | 18.91 | 0.75% |
| Ilość iteracji | 73.17 | 112 | 49 | 14.92 | - |

**Interpretacja:**

Z wykresu wartości funkcji celu można wywnioskować, że najlepsze wyniki uzyskano dzięki selekcji elitarnej, lecz pozostałe dały podobnie dobre wyniki. Mogło to wynikać z faktu, że mając mało przedmiotów unikalnych i biorąc dużą ilość ich sztuk do listy klienta, automatycznie spadła liczba możliwych kombinacji rozmieszenia artykułów pośród sprzedawców, zatem szansa na szybkie znalezienie optymalnego rozwiązania znacząco wzrosła. Jest to widoczne również na wykresie iteracji, którego wartości dla wszystkich trzech metod są bardzo zbliżone.

### Test\_10 - badanie zależności jakości rozwiązania od rozmiaru problemu, duża liczba unikalnych produktów

**Autor:** Wiktor Kacprzak  
**Cel badania:** Zbadanie zachowania algorytmu dla listy zakupów z dużą ilością unikalnych produktów, ale małymi ilościami w obrębie jednego produktu

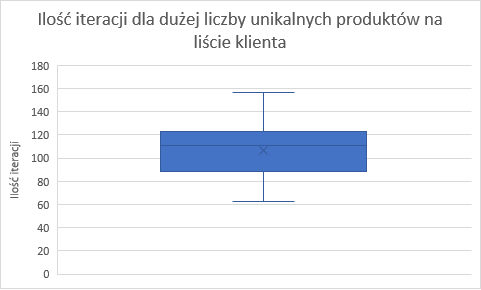


Fig. ‑ Wykres zależności ilości iteracji dla dużej liczby unikalnych produktów na liście klienta

Tabela ‑ Wartości statystyczne dla dużej liczby unikalnych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dużo unikalnych | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Ilość iteracji | 106.93 | 157 | 63 | 18.94 | - |

**Interpretacja:**

Ten test można porównać ze wcześniejszym tylko na podstawie ilości iteracji potrzebnej do znalezienia końcowego wyniku, ponieważ doszło do zmiany listy zakupów klienta, co jest równoznaczne ze zmianą cen i końcowym kosztem zakupu. Z tego wykresu można wywnioskować, że im większa jest ilość unikalnych produktów, tym średnio więcej iteracji potrzebuje algorytm, aby znaleźć optymalne rozwiązanie.

### Test\_11 - badanie zależności jakości rozwiązania dla różnej ilości unikalnych produktów w liście zakupów, przy zachowaniu stałej sumarycznej ilości produktów

**Autor:** Wiktor Kacprzak  
**Cel badania:** Zbadanie zachowania algorytmu dla stałej sumarycznej ilości produktów w liście zakupów, ale przy zmieniającej się liczbie produktów unikalnych.

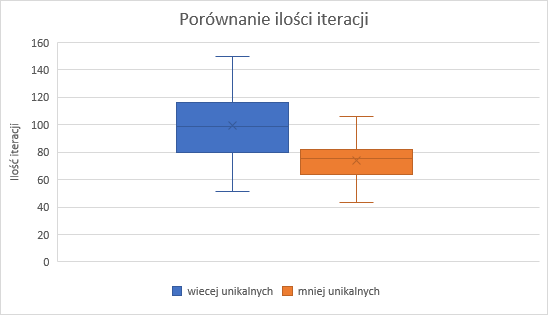


Fig. ‑ Wykres porównujący ilość iteracji dla przypadku listy z małą i dużą liczbą unikalnych produktów

Tabela ‑ Wartości statystyczne dla większej liczby unikalnych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Więcej unikalnych | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Ilość iteracji | 84.37 | 135 | 36 | 19.63 | - |

Tabela ‑ Wartości statystyczne dla mniejszej liczby unikalnych

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Mniej unikalnych | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Ilość iteracji | 58.66 | 91 | 28 | 10.85 | - |

**Interpretacja:**

Im więcej unikalnych produktów znajduje się na liście klienta, tym więcej iteracji potrzebuje algorytm, aby znaleźć końcowe optymalne rozwiązanie. Rozproszenie tej ilości iteracji również jest dużo większe.

### Test\_12 - badanie zależności jakości rozwiązania dla różnej ilości unikalnych produktów od wybranej metody mutacji

**Autor:** Mateusz Król  
**Cel badania:** Znalezienie tendencji między rodzajem użytej mutacji, a postacią listy zakupów – odpowiedzenie na pytanie, czy mutacja z eliminacją lepiej poradzi sobie dla dużej liczby produktów

Badanie podzielone zostało na 12.1 i 12.2, w jednym z których zmieniona została liczba unikalnych produktów na liście klienta (stąd rozbieżności między wartościami funkcji kryterialnej).

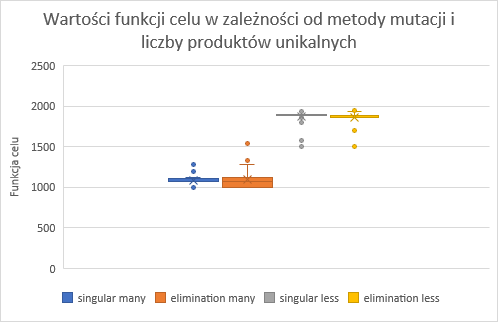


Fig. ‑ Wykres zależności funkcji celu dla różnych metod mutacji przy różnej wielkości listy klienta

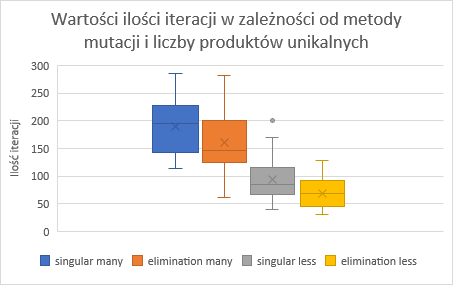


Fig. ‑ Wykres ilości iteracji dla różnych metod mutacji przy różnej wielkości listy klienta

Tabela ‑ Wartości statystyczne dla przypadku singular many

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Singular many | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1085.41 | 1277.20 | 994.10 | 36.92 | 9.18% |
| Ilość iteracji | 190.53 | 286 | 113 | 43.69 | - |

Tabela ‑ Wartości statystyczne dla przypadku elimination many

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Elimination many | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1089.10 | 1536.36 | 994.10 | 87.38 | 9.56% |
| Ilość iteracji | 161.40 | 282 | 62 | 50.01 | - |

Tabela ‑ Wartości statystyczne dla przypadku singular less

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Singular less | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1885.06 | 1979.77 | 1501.60 | 50.43 | 24.58% |
| Ilość iteracji | 93.23 | 201 | 40 | 27.66 | - |

Tabela ‑ Wartości statystyczne dla przypadku elimination less

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Elimination less | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 1857.58 | 1996.38 | 1501.60 | 57.34 | 23.71% |
| Ilość iteracji | 69.37 | 129 | 31 | 23.70 | - |

**Interpretacja:**

Analizując wyniki można tylko porównywać ze sobą wartości dla takiej samej liczby produktów unikalnych, bo wiadomo, że jeżeli zastosowano różne listy zakupów to całkowity koszt będzie różny. Z wykresu wartości funkcji kryterialnej odczytano, że mutacja z eliminacją może dać trochę lepsze wyniki od tej pojedynczej dla dużej liczby przedmiotów unikalnych. W przypadku mniejszej liczby tych produktów obie metody dają podobne wyniki. Biorąc pod uwagę ilość iteracji można wykazać, że w obu przypadkach lepsze rozwiązanie dostarczono dzięki metodzie mutacji z eliminacją.

### Test\_13 - badanie złożoności czasowej w zależności od postaci listy zakupów

**Autor:** Wiktor Kacprzak  
**Cel badania:** Znalezienie zależności między ilością iteracji, a postacią listy zakupów

W tym przypadku przyjęto 4 różne listy klienta, różne w zależności od ilości produktów unikalnych i ilości sztuk produktów.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Fig. ‑ Cztery użyte w testach listy zakupów

Tabela ‑ Wartości statystyczne dla testu złożoności czasowej

|  |  |  |
| --- | --- | --- |
|  | Średnia ilość iteracji | Średni czas wykonania algorytmu [s] |
| Client 0 | 58.9 | 0.6477 |
| Client 1 | 86.6 | 0.8829 |
| Client 2 | 86.3 | 0.9528 |
| Client 3 | 61.8 | 0.6947 |

**Interpretacja:**

Zgodnie z przewidywaniami listy zakupów z większą ilością unikalnych produktów wpłynęły na wydłużenie działania algorytmu. Ilość sztuk danego ma zdecydowanie drugorzędny wpływ na ilość iteracji.

### Test\_14 - badanie działania algorytmu dla przypadku złośliwego, ze znanym rozwiązaniem optymalnym

**Autor:** Mateusz Król  
**Cel badania:** Sprawdzenie jak sprawnie algorytm rozwiąże problem ze znanym rozwiązaniem w postaci zamówienia u pojedynczego sprzedawcy.

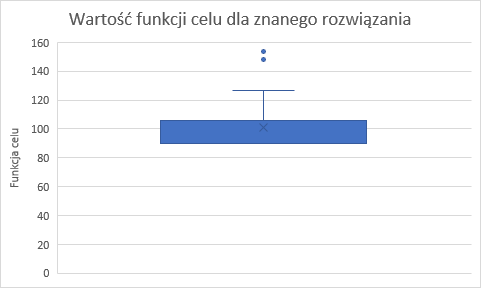


Fig. ‑ Wykres wartości funkcji celu dla przypadku ze znanym rozwiązaniem.

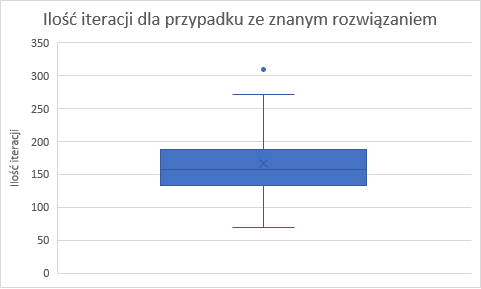


Fig. ‑ Wykres ilości iteracji dla przypadku ze znanym rozwiązaniem.

Tab. 38. Wartości statystyczne dla testu ze znanym rozwiązaniem

Tabela ‑ Wartości statystyczne dla testu ze znanym rozwiązaniem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Znane rozwiązanie | Średnia | Max | Min | Odchylenie standardowe | Błąd względny |
| Funkcja celu | 100.68 | 154 | 90 | 13.53 | 11.87% |
| Ilość iteracji | 166.73 | 300 | 70 | 39.83 | - |

**Interpretacja:**

Test został wykonany dla zmienionej bazy danych tak, aby jeden sprzedawca sprzedawał swoje produkty po zaniżonej cenie, co powinno poskutkować jego znalezieniem w każdym przypadku wykonania algorytmu. W nielicznych przypadkach algorytm nie znajdował znanego z góry optymalnego rozwiązania. Jest to potwierdzenie istnienia i prawidłowego działania losowości jaka jest w nim zawarta. Być może po zwiększeniu liczby iteracji algorytm byłby w stanie za każdym razem znaleźć szukane minimum. W porównaniu do innych testów błąd względny jest dość wysoki, jednak po przeanalizowaniu macierzy wynikowych okazało się, że różnice między rozwiązaniem pożądanym, a uzyskanym są jednostkowe tj. pojedynczy produkt przeniesiony do innego sprzedawcy.

# Podsumowanie

## Wnioski

## Stwierdzone problemy

Podczas implementacji największą trudnością okazało się wymyślenie schematu krzyżowań i mutacji w taki sposób, żeby pozostawać jak najbliżej rozwiązań dopuszczalnych. Z racji odstąpienia od pomysłu operowania na rozwiązaniach niedopuszczalnych w sposób niespełniający wymogów listy zakupów możliwe, że narażono się na zbieganie algorytmu do ekstremów lokalnych, jednakże przeprowadzone testy pokazują, że struktury danych dobrze współgrają ze stworzonymi operatorami.

Jednym z większych problemów wykorzystania generalnie heurystyk jest brak możliwości sprawdzenia, czy odnalezione rozwiązanie jest optymalne czy nie. Nawet jeśli przeprowadzone testy wykazują względną szybką zbieżność algorytmu, to nie można stwierdzić, że uzyskane rozwiązanie jest tym poszukiwanym.

Szczególnie problematyczne jest zagadnienie edycji baz danych, określających pośrednio przestrzeń rozwiązań w algorytmie. Duża ilość danych, skomplikowany format przechowywania informacji o cenie przesyłek i funkcji zniżek, powoduje, że edycja z poziomu aplikacji jest

## Perspektywy dalszego rozwoju