

# Metaheurystyki i ich zastosowania

## Zadanie 6: Rzeczywisty problem (VRPTW)

### SPIS TREŚCI

<b>Zadanie 6: Rzeczywisty problem (VRPTW) .....</b>	<b>- 1 -</b>
<b>SPIS TREŚCI.....</b>	<b>- 1 -</b>
<b>1 WSTĘP I OPIS PROBLEMU .....</b>	<b>- 2 -</b>
1.1 Definicja problemu .....	2 -
1.2 Ograniczenie i funkcja celu .....	3 -
1.3 Charakterystyka instancji testowych (Benchmark Solomona) .....	3 -
<b>2 OPIS ZASTOSOWANEJ METAHEURYSTYKI .....</b>	<b>- 3 -</b>
2.1 Zasada działania algorytmu .....	3 -
2.2 Uzasadnienie wyboru metody .....	4 -
2.3 Pseudokod algorytmu .....	4 -
2.4 Architektura hybrydy i integracja lokalnego przeszukiwania .....	5 -
<b>3 ADAPTACJA ALGORYTMU DO SPECYFIKI PROBLEMU .....</b>	<b>- 5 -</b>
3.1 Reprezentacja rozwiązania.....	5 -
3.2 Funkcja celu i ocena tras.....	5 -
3.3 Uwzględnianie okien czasowych i opóźnień.....	6 -
3.4 Dodatkowe ograniczenia i obsługa przeładowania .....	6 -
<b>4 IMPLEMENTACJA I INSTRUKCJA OBSŁUGI .....</b>	<b>- 7 -</b>
4.1 Kluczowe miejsca implementacji .....	7 -
4.2 Instrukcja użytkownika .....	7 -
<b>5 PROJEKT EKSPERYMENTÓW BADAWCZYCH .....</b>	<b>- 8 -</b>
<b>6 PREZENTACJA I ANALIZA WYNIKÓW .....</b>	<b>- 9 -</b>
6.1 Badania dla pliku r101.txt.....	9 -
6.2 Badanie dla pliku c101.txt .....	29 -
6.3 Badanie dla pliku rc101.txt .....	49 -

<b>7</b>	<b>Analiza Wyników Eksperymentalnych .....</b>	<b>- 69 -</b>
<b>7.1</b>	<b>Wpływ poszczególnych parametrów na jakość rozwiązania.....</b>	<b>- 69 -</b>
7.1.1	Wielkość populacji (N) .....	69 -
7.1.2	Współczynnik mutacji (Pm) .....	70 -
7.1.3	Elitaryzm .....	70 -
<b>7.2</b>	<b>Stabilność działania algorytmu .....</b>	<b>- 71 -</b>
<b>7.3</b>	<b>Trudność grup danych (R vs C vs RC).....</b>	<b>- 71 -</b>
	Grupa C (Clustered) – Najłatwiejsza optymalizacja dystansu.....	71 -
	Grupa R (Random) – Wyzwanie dla eksploracji .....	71 -
	Grupa RC (Mixed) – Najwyższy stopień trudności .....	71 -
<b>8</b>	<b>PORÓWNANIE Z ROZWIĄZANIAMI LITERATUROWYMI (BKS) .....</b>	<b>- 72 -</b>
<b>9</b>	<b>WNIOSKI I PODSUMOWANIE.....</b>	<b>- 72 -</b>
<b>9.1</b>	<b>Skuteczność algorytmu.....</b>	<b>- 72 -</b>
<b>9.2</b>	<b>Najlepsze ustawienia parametrów.....</b>	<b>- 72 -</b>
<b>9.3</b>	<b>Ograniczenia rozwiązania.....</b>	<b>- 72 -</b>
<b>9.4</b>	<b>Kierunki dalszej poprawy .....</b>	<b>- 73 -</b>

# 1 WSTĘP I OPIS PROBLEMU

## 1.1 Definicja problemu

Problem układania tras dla pojazdów z oknami czasowymi (ang. Vehicle Routing Problem with Time Windows - VRPTW) jest rozszerzeniem klasycznego problem VRP. Polega on na wyznaczeniu optymalnego zestawu tras dla floty pojazdów stacjonujących w centralnym depozycie, które muszą obsłużyć rozproszony zestaw klientów.

W przeciwieństwie do podstawowej wersji problemu, VRPTW nakłada na każdą wizytę u klienta sztywne ograniczenia czasowe. Każdy odbiorca  $i$  definiuje przedział  $[e_i, l_i]$  gdzie  $e_i$  to najwcześniejsza możliwa godzina rozpoczęcia obsługi, a  $l_i$  to czas graniczny (najpóźniejszy), którego nie można przekroczyć. Jeśli pojazd przyjedzie przed czasem  $e_i$ , musi zaczekać do momentu otwarcia okna. Przyjazd po czasie  $l_i$  jest uznawany za naruszenie ograniczeń problemu (rozwiążanie niewykonalne).

## 1.2 Ograniczenie i funkcja celu

Główym celem optymalizacji jest minimalizacja całkowitego kosztu eksploatacyjnego, który w niniejszym opracowaniu jest definiowany przede wszystkim przez sumaryczny dystans pokonany przez wszystkie pojazdy.

Kluczowe ograniczenia zaimplementowane w modelu:

- Ograniczenie pojemności (Capacity): Suma zapotrzebowania (demand) wszystkich klientów na danej trasie nie może przekroczyć maksymalnej ładowności pojazdu.
- Ograniczenia czasowe (Time Windows): Obsługa musi rozpocząć się wewnątrz okna  $[e_i, l_i]$ . Uwzględniany jest również czas obsługi (service time) oraz czasu przejazdu między punktami.
- Ograniczenie floty: Liczba użytych pojazdów nie może przekroczyć zasobów dostępnych w depozycie.
- Spójność tras: Każda trasa musi rozpoczynać się i kończyć w depozycie (punkt, którego ID = 0), a każdy klient musi zostać obsłużony dokładnie raz.

### 1.3 Charakterystyka instancji testowych (Benchmark Solomona)

Do przeprowadzenia eksperymentów wykorzystano benchmark M. Solomona (100 klientów). Zbiory danych w tym benchmarku dzielą się na trzy specyficzne kategorie, które pozwalają ocenić uniwersalność algorytmu:

1. Grupa C (Clustered): Klienci są rozmieszczeni w grupach (klastrach).
2. Grupa R (Random): Klienci są rozmieszczeni losowo (jednostajnie) na mapie.
3. Grupa RC (Random-Clustered): Mieszanka obu powyższych podejść, najlepiej oddającą rzeczywiste warunki logistyczne.

## 2 OPIS ZASTOSOWANEJ METAHEURYSTYKI

### 2.1 Zasada działania algorytmu

Zastosowana metaheurystyka do rozwiązania omawianego problemu to Algorytm Genetyczny (ang. Genetic Algorithm), dodatkowo wspomagany lokalnym przeszukiwaniem. Metoda ta łączy populacyjne podejście do przeszukiwania przestrzeni rozwiązań z lokalnymi heurystykami optymalizacyjnymi. Algorytm operuje na populacji osobników, gdzie każdy osobnik reprezentuje kompletne rozwiązanie problemu VRPTW.

W odróżnieniu od klasycznego algorytmu genetycznego, niniejsza implementacja kładzie silny nacisk na fazę mutacji, która pełni rolę operatora lokalnego przeszukiwania. Proces ewolucyjny opiera się na mechanizmie elitarystmu, gdzie tylko najlepsze jednostki z danej generacji mają szansę stać się rodzicami dla kolejnej pokolenia.

### 2.2 Uzasadnienie wyboru metody

Algorytm Genetyczny wsparty lokalnym przeszukiwaniem jest jedną z najskuteczniejszych metod rozwiązywania problemów klasy VRP ze względu na:

- Odporność na optima lokalne: Dzięki operowaniu na całej populacji, algorytm może jednocześnie przeszukiwać różne obszary przestrzeni rozwiązań.
- Możliwość hybrydyzacji: Struktura tras VRPTW pozwala na łatwe włączenie dedykowanych operatorów, takich jak 2-opt czy inteligenta relokacja klientów, co znacząco przyspiesza zbieżność algorytmu.
- Efektywną obsługę ograniczeń: Wykorzystanie „inteligentnego wstawiania” w trakcie mutacji pozwala na aktywne poszukiwanie miejsc w trasie, które minimalizują kary za naruszenie okien czasowych.

## 2.3 Pseudokod algorytmu

Poniżej przedstawiono opis przebiegu algorytmu zaimplementowanego w metodzie `vrptw.genetic_algorithm.run()`, z uwzględnieniem wykonalności rozwiązań (minimalizacja kar przed minimalizacją dystansu).

```
# PROCEDURA GŁÓWNA (GENETICALGORITHM.RUN)
1. INICJALIZUJ POPULACJE:
   - DODAJ ROZWIĄZANIE BAZOWE.
   - WYPEŁNIJ RESZĘ POPULACJI (POP_SIZE) ZMUTOWANYMI WERSJAMI BAZY.
2. DLA KAŻDEJ GENERACJI (0 DO G):
   A. OCEŃ KAŻDEGO OSOBNIKA (SCORE: KARA, POJAZDY, DYSTANS).
   B. POSORTUJ POPULACJĘ WEDŁUG KLUCZA:
      (CZY_POSIADA_KARE, LICZBA_PODJAZDÓW, WARTOŚĆ_KARY, DYSTANS)
   C. WYBIERZ ELITY:
      - ZACHOWAJ 20% (ELITISM_RATE) NAJLEPSZYCH OSOBNIKÓW.
   D. BUDUJ NOWĄ GENERACJĘ:
      - DOPÓKI NOWA POPULACJA < POP_SIZE:
         I. WYLOSUJ RODZICA Z PULI ELIT.
         II. STWÓRZ KOPIĘ I ZASTOSUJ MUTACJE (PRAWDOPODOBIENSTWO MUTATION_RATE).
         III. USUŃ PUSTE TRASY I DODAJ DZIECKO DO NOWEJ GENERACJI.
   E. ZAPISZ STATYSTYKI (NAJLEPSZY DYSTANS, KARA, ŚREDNIA) W HISTORII.
3. ZWRÓĆ NAJLEPSZE ZNALEZIONE DOTYCZĄCE ROZWIĄZANIE (BEST_OVERALL).
```

## 2.4 Architektura hybrydy i integracja lokalnego przeszukiwania

Rozwiązanie ma charakter hybrydowy, ponieważ w proces ewolucyjny wbudowano dwa operatory optymalizacji lokalnej, które działają jako mutacje:

1. 2-opt: Metoda `_2opt_mutation()` wybiera losową trasę i wykonuje odwrócenie kolejności segmentu klientów. Pozwala to na optymalizację dystansu wewnętrz pojedynczej trasy bez wpływu na okna czasowe innych pojazdów.
2. Relocate z optymalnym wstawieniem: Metoda `_relocate_mutation_get_best_insertion_pos()`, która przeszukuje wszystkie możliwe pozycje w docelowej trasie i wybiera, tę która generuje najmniejszy przyrost kary czasowej i dystansu.

Dzięki takiemu podejściu, każda mutacja nie jest jedynie losową zmianą, ale ukierunkowaną próbą poprawy lokalnej struktury tras.

### 3 ADAPTACJA ALGORYTMU DO SPECYFIKI PROBLEMU

W tej sekcji opisano sposób w jaki teoretyczne założenia problemu VRPTW zostały odwzorowane w strukturach danych oraz logice oceny rozwiązań w zaimplementowanym systemie.

#### 3.1 Reprezentacja rozwiązania

W programie zastosowano strukturę obiektową opartą na klasach *Solution* i *Route*. Każdy osobnik w populacji algorytmu genetycznego składa się z listy obiektów *Route*, które przechowują sekwencję identyfikatorów klientów (bez uwzględniania depozytu wewnętrz listy). Depozyt jest traktowany jako stały punkt startowy i końcowy.

```
@dataclass 7 usages  ± irek bartoszek
class Route:
    stops: List[int] # lista ID klientów (bez Depot)

@dataclass 14 usages  ± irek bartoszek
class Solution:
    routes: List[Route]
```

#### 3.2 Funkcja celu i ocena tras

Kluczowym elementem adaptacji jest klasa *Score*, która przechowuje trzy parametry oceny: liczbę pojazdów, wartość kary (penalty) oraz dystans.

Definicja funkcji celu: Zastosowano hierarchiczną funkcję celu. Porównywanie dwóch rozwiązań (metoda *key()* w klasie *Score*) odbywa się według priorytetów:

1. Liczba pojazdów (im mniej, tym lepiej).
2. Suma kar (Penalty) – za naruszenie okien czasowych i przeładowanie (priorytet uzyskania rozwiązania dopuszczalnego).
3. Całkowity dystans – optymalizowany dopiero dla rozwiązań o tej samej liczbie pojazdów i karze.

```
@dataclass(frozen=True) 2 usages  ± irek bartoszek
class Score:
    vehicles: int
    penalty: float
    distance: float

    def key(self) -> tuple[int, float, float]: 2 usages  ± irek bartoszek
        return (self.vehicles, self.penalty, self.distance)
```

### 3.3 Uwzględnianie okien czasowych i opóźnień

Mechanizm obsługi czasu został zaimplementowany w funkcji `simulate_route()`. Algorytm iteruje po kolejnych klientach w trasie, obliczając czas przyjazdu ( $t$ ):

- Oczekiwanie: Jeśli pojazd dotrze do klienta przed `ready_time`, następuje oczekiwanie:  
$$t = \max(t + travel\_time, ready\_time)$$
- Opóźnienie: Jeśli pojazd dotrze po `due_date`, różnica ( $t - due\_date$ ) jest dodawana do sumarycznego opóźnienia trasy (`late_time`).

Wartość opóźnienia jest następnie mnożona przez współczynnik kary (w kodzie przyjęto wagę 100.0 dla spóźnień), co drastycznie obniża ocenę osobnika i wymusza na procesie ewolucyjnym poszukiwanie tras spełniających rygory czasowe.

### 3.4 Dodatkowe ograniczenia i obsługa przeładowania

Poza oknami czasowymi, algorytm uwzględnia:

- Limit pojazdów: Każde rozwiązanie, które przekroczy `max_vehicles` zdefiniowane w pliku Solomona, jest uznawane za niepoprawne (walidacje odbywa się w `validate_solution()`).
- Ograniczenie pojemności (Capacity): Podczas symulacji trasy sumowane jest zapotrzebowanie (demand) klientów. Jeśli `load > capacity`, naliczana jest bardzo wysoka kara (współczynnik 1000.0), co sprawia, że takie rozwiązanie jest natychmiast eliminowane przez mechanizm selekcji naturalnej.
- Czas powrotu do depozytu: Algorytm sprawdza również okno czasowe depozytu – pojazd musi wrócić do bazy przed jej zamknięciem, w przeciwnym razie naliczana jest dodatkowa kara za spóźnienie.

## 4 IMPLEMENTACJA I INSTRUKCJA OBSŁUGI

### 4.1 Kluczowe miejsca implementacji

`vrptw/models.py`

Zawiera definicje struktur danych (Customer, Route, Solution).

`vrptw/evaluation.py`

Główny element systemu oceny. Funkcja `simulate_route()` odpowiada za symulację fizycznego przejazdu pojazdu, uwzględniając narastający czas, oczekiwanie na otwarcie okna oraz naruszenia `due_date`.

### `vrptw/genetic_algorithm.py`

Zawiera klasę `GeneticAlgorithm`, implementującą cykl ewolucyjny, mechanizm elitaryzmu oraz hybrydowe operatory mutacji.

### `vrptw/construction.py`

Implementuje algorytm zachłanny `build_initial_solution_ready_time()`, który sortuje klientów według czasu gotowości i buduje pierwsze poprawne rozwiązanie, stanowiące punkt startowy dla GA.

## 4.2 Instrukcja użytkownika

Program został przygotowany jako skrypt konsolowy w języku Python. Aby przeprowadzić eksperymenty, należy wykonać poniższe kroki:

### Wymagania wstępne

Należy upewnić się, że zainstalowane są niezbędne biblioteki.

### Konfiguracja eksperymentu

Głównym punktem wejścia do aplikacji jest plik `src/main.py`. Przed uruchomieniem można dostosować parametry w słowniku `config`:

- `instance_file`: nazwa pliku z instancją (np. "r101.txt").
- `num_runs`: liczba powtórzeń algorytmu dla statystyki (zalecane min. 5).
- `ga_params`: parametry algorytmu (wielkość populacji, liczba generacji).

### Uruchomienie programu

Aby uruchomić obliczenia, należy przejść do folderu `/src` i wykonać komendę:

```
python main.py
```

### Odczyt wyników

Po zakończeniu działania algorytmu:

1. Konsola: Wyświetli statystyki z każdego cyklu (Best, Worst, Mean, Std Dev) oraz tabelę zbieżności GA.
2. Wizualizacja: Program automatycznie otworzy okna z wykresem najlepszej znalezionej trasy.
3. Pliki: W folderze `plots/` (generowanym automatycznie) zostaną zapisane wykresy poprawy funkcji kary oraz dystansu w formacie `.png`.

## 5 PROJEKT EKSPERYMENTÓW BADAWCZYCH

Celem eksperymentów jest analiza wpływu parametrów sterujących algorytmu genetycznego na jakość uzyskiwanych tras (minimalizacja dystansu i liczby pojazdów) oraz ocenienie stabilności algorytmu w obliczu twardych ograniczeń okien czasowych.

W ramach badań statystycznych monitorowane i konfigurowane będą następujące parametry:

- Liczba generacji (G): Maksymalna liczba iteracji ewolucyjnych. Określa czas, jaki algorytm ma na eksplorację przestrzeni rozwiązań i dopracowanie szczegółów tras.
- Wielkość populacji (N): Liczba równolegle rozważanych wariantów rozwiązań. Wpływ na różnorodność genetyczną i zdolność algorytmu do unikania minimów lokalnych.
- Współczynnik mutacji (Pm): Prawdopodobieństwo, że dany osobnik zostanie poddany modyfikacji strukturalnej.
- Wskaźnik elitaryzmu (E): Odsetek najlepszych osobników z bieżącej generacji, które są kopiowane do nowej populacji bez żadnych zmian.

Eksperymenty zostaną przeprowadzone na trzech reprezentatywnych instancjach Solomona: R101 (rozmieszczenie losowe), C101 (klastry klientów) oraz RC101 (mieszane) w sposób parametryczny dla każdych różnych wartości wielkości populacji, współczynnika mutacji i wskaźnika elitaryzmu.

Wartości bazowe zostały przyjęte jako:

- Liczba generacji (G): 1000,
- Wielkość populacji (N): 100,
- Współczynnik mutacji (Pm): 0.1,
- Wskaźnik elitaryzmu (E): 0.2

W trakcie prowadzenia eksperymentu współczynnik mutacji został zmieniony jako bazowa wartość na wartość 0.4.

## 6 PREZENTACJA I ANALIZA WYNIKÓW

### 6.1 Badania dla pliku r101.txt

#### 1. Badanie wpływu wielkości populacji

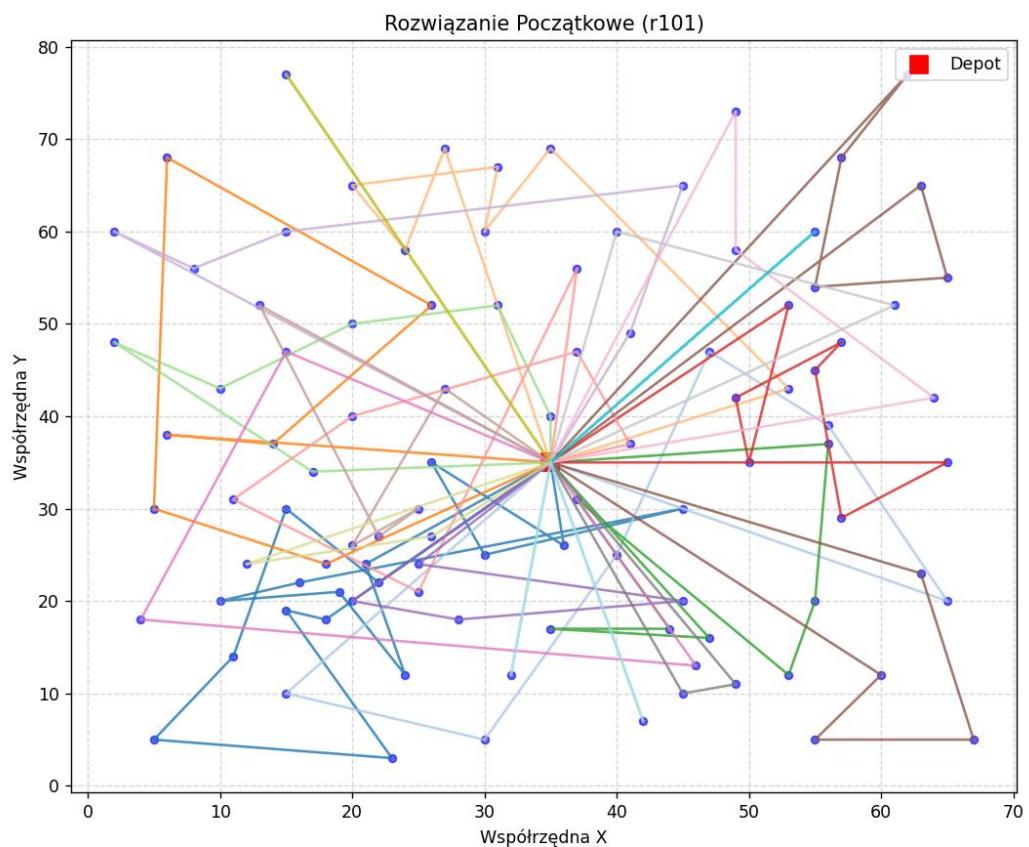
Badania wpływu wielkości populacji (N). Ten parametr decyduje o balansie między "szerokością" poszukiwań (eksploracją) a kosztem obliczeniowym. Wartości zmieniane będą wynosiły: N = [50,100,200].

**Tabela 1.** Wyniki 5 eksperymentów dla badanego współczynnika.

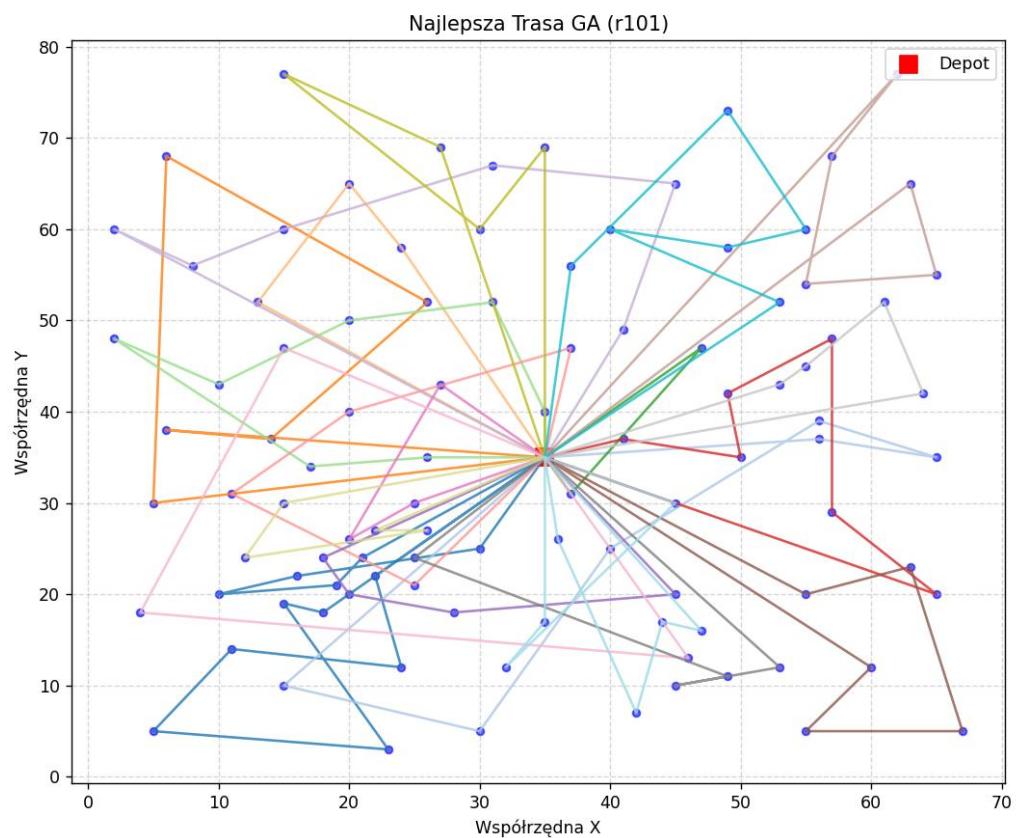
Lp.	Populacja (N)	Próba	Liczba aut (Veh)	Dystans (Dist)
1	50	1	23	1994.56
2		2	22	2006.12
3		3	23	2083.93
4		4	23	2095.24
5		5	23	2052.50
6	100	1	22	1986.55
7		2	23	1877.08
8		3	22	1872.89
9		4	22	1985.86
10		5	21	1906.70
11	200	1	21	1855.82
12		2	22	1851.43
13		3	22	1935.72
14		4	22	1805.67
15		5	22	1850.27

**Tabela 2.** Wyniki podsumowywające.

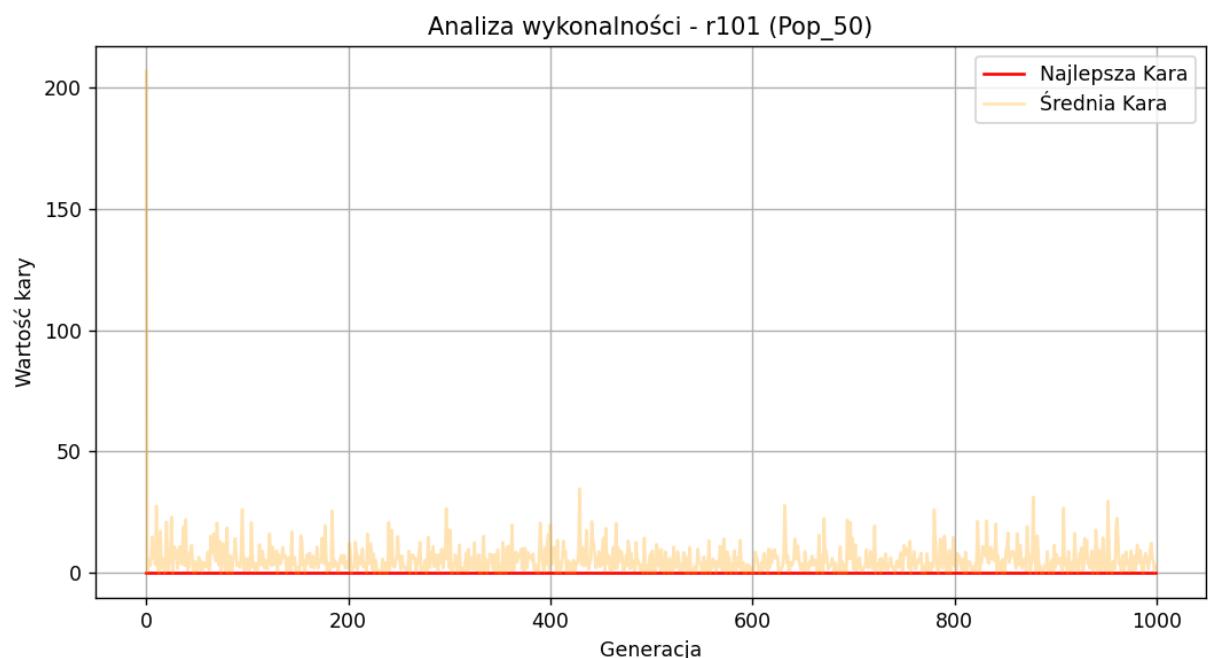
Populacja (N)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)
50	1994.56	2046.47	40.35
100	1872.89	1925.82	50.67
200	1805.67	1859.78	42.12



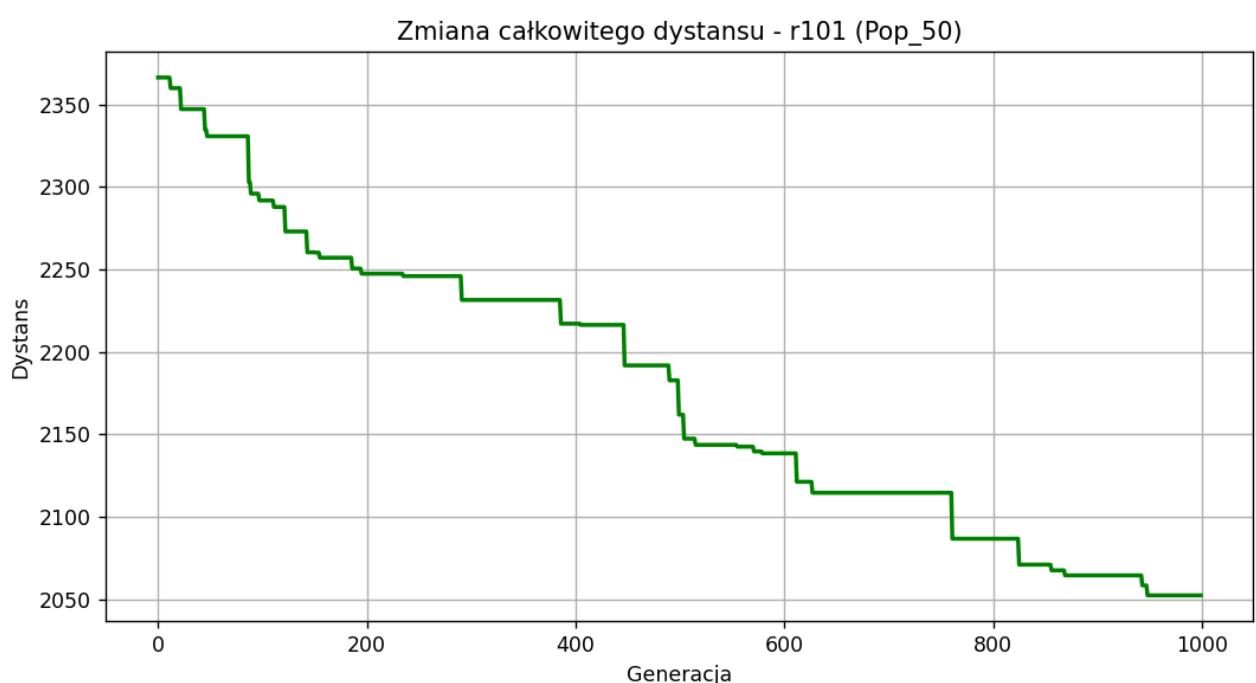
**Wykres 1.** Wizualizacja rozwiązania początkowego.



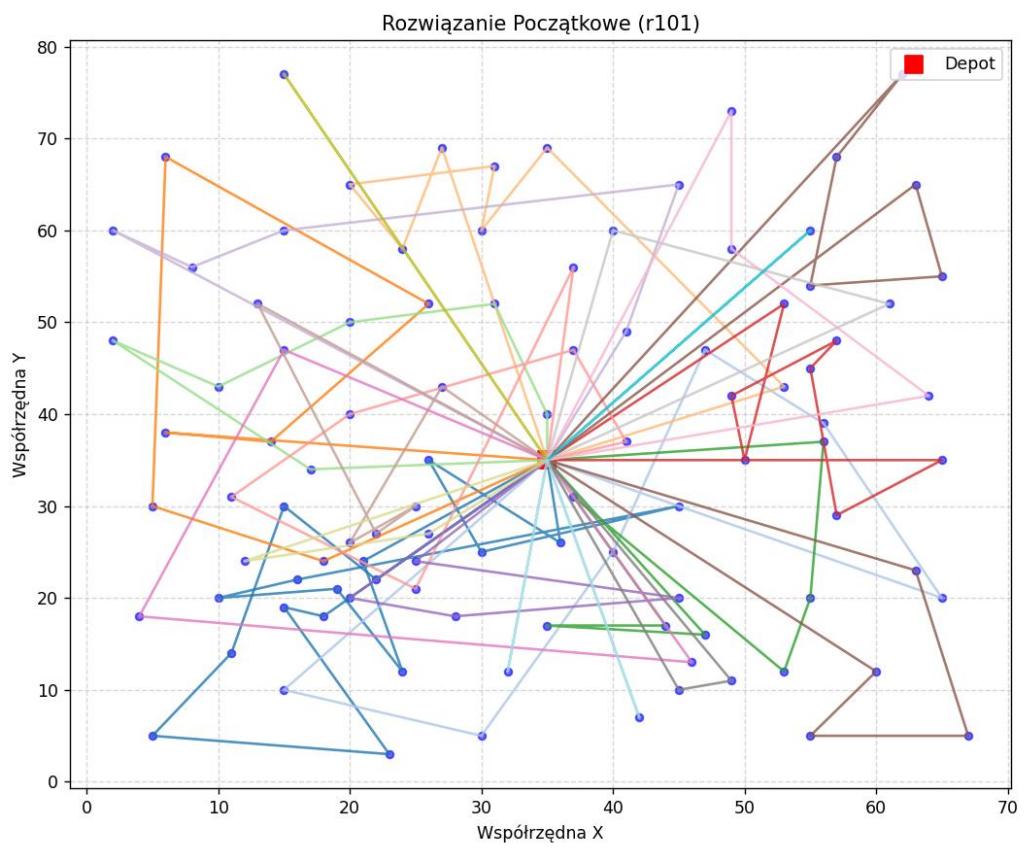
**Wykres 2.** wizualizacja najlepszego rozwiązania.



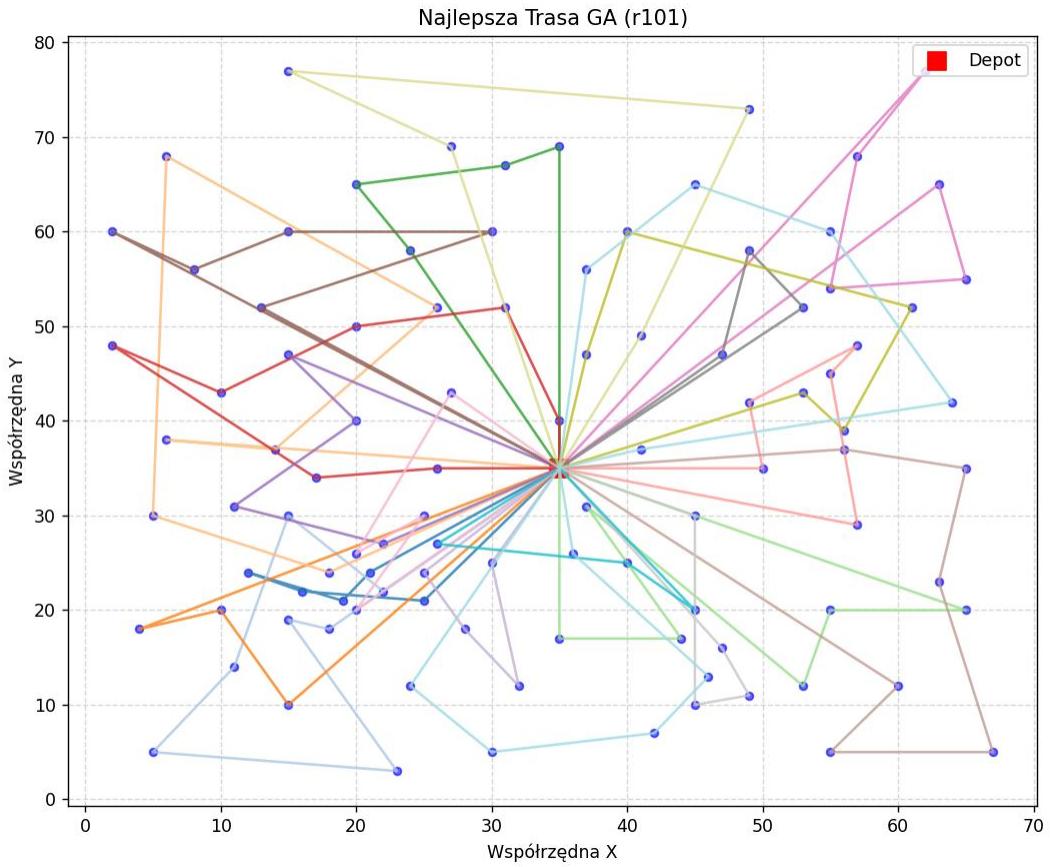
**Wykres 3.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 4.** Wizualizacja zmiany dystansu dla najlepszej trasy.

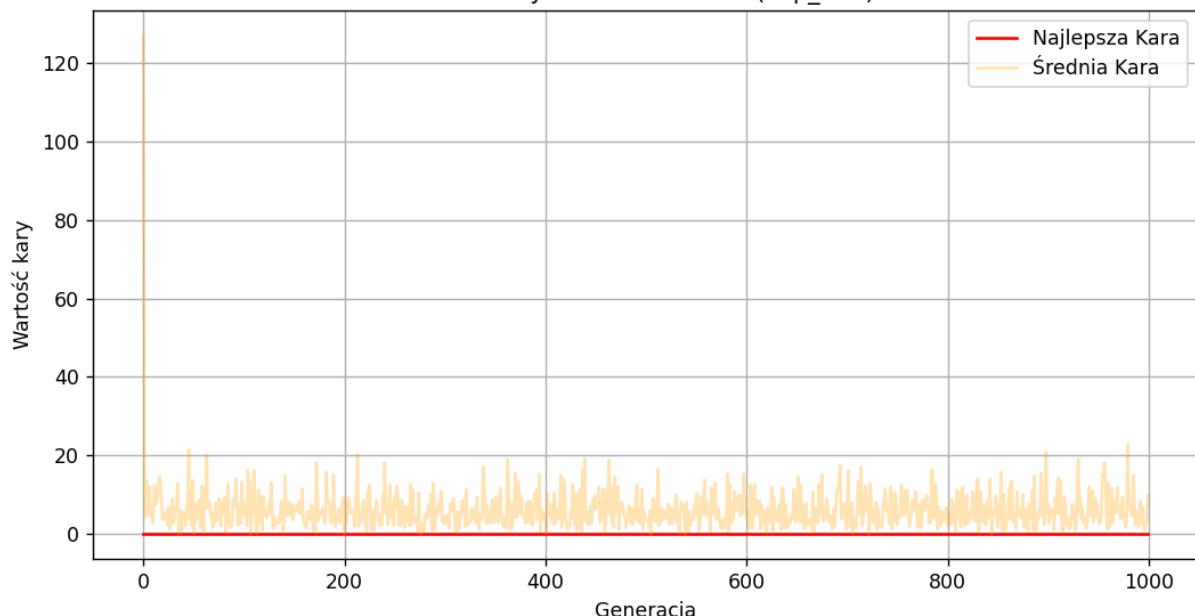


**Wykres 5.** Wizualizacja rozwiązania początkowego.



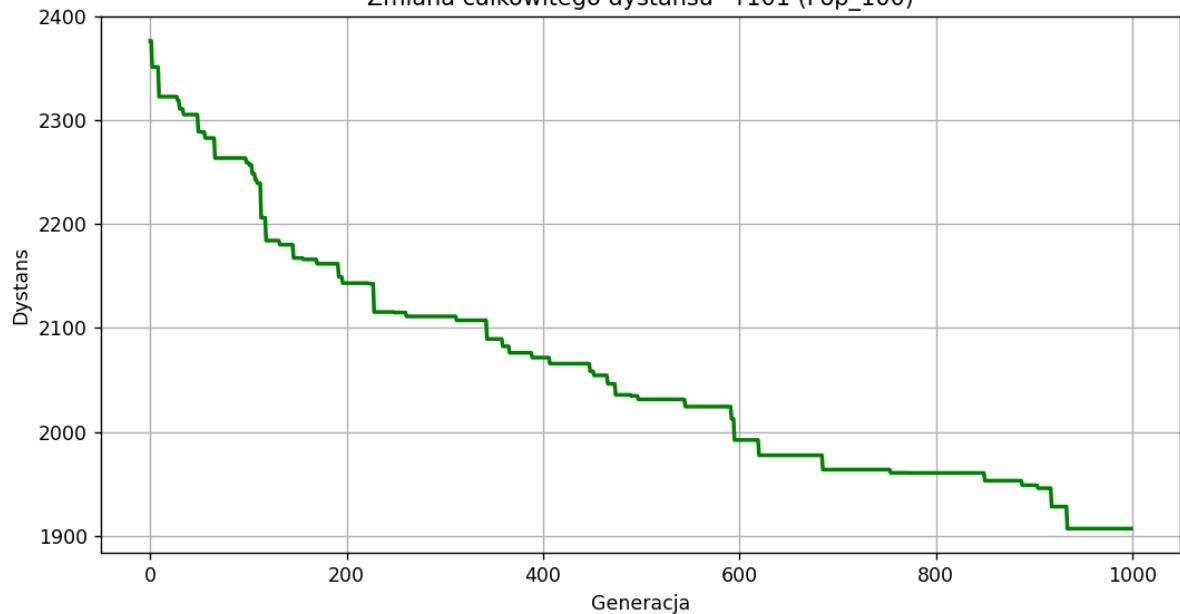
**Wykres 6.** Wizualizacja najlepszego rozwiązania.

Analiza wykonalności - r101 (Pop\_100)

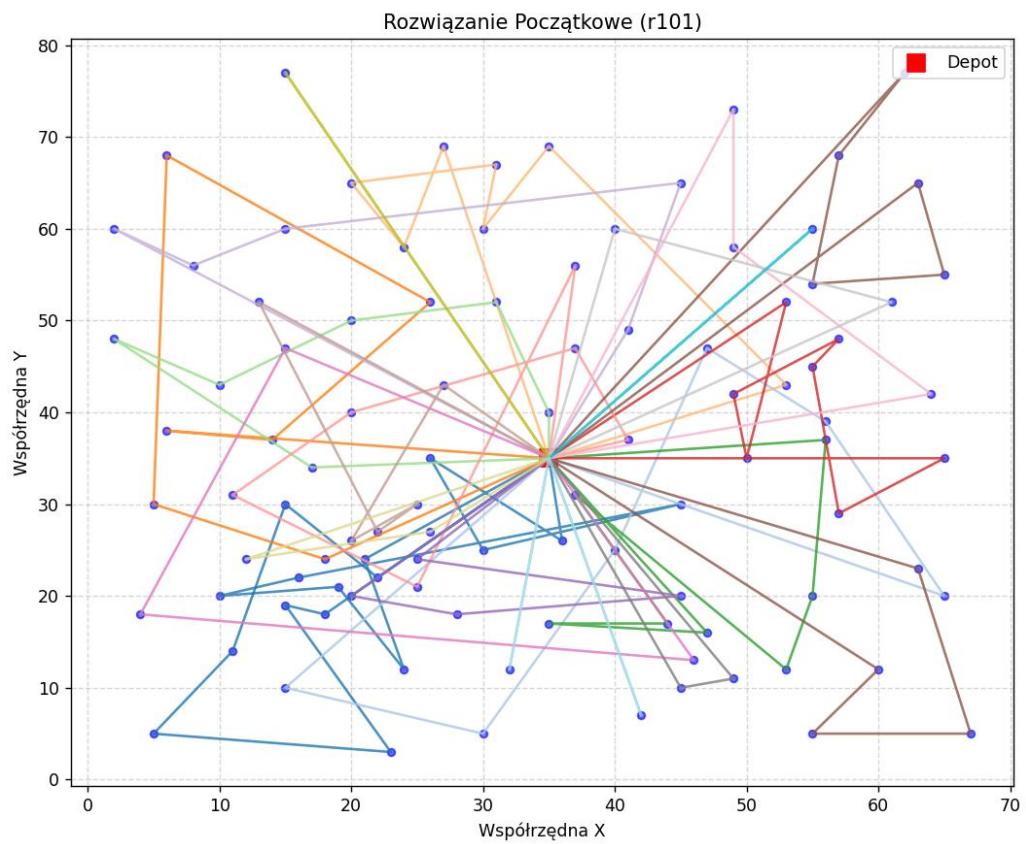


**Wykres 7.** Wizualizacja średniej kary populacji dla najlepszej trasy.

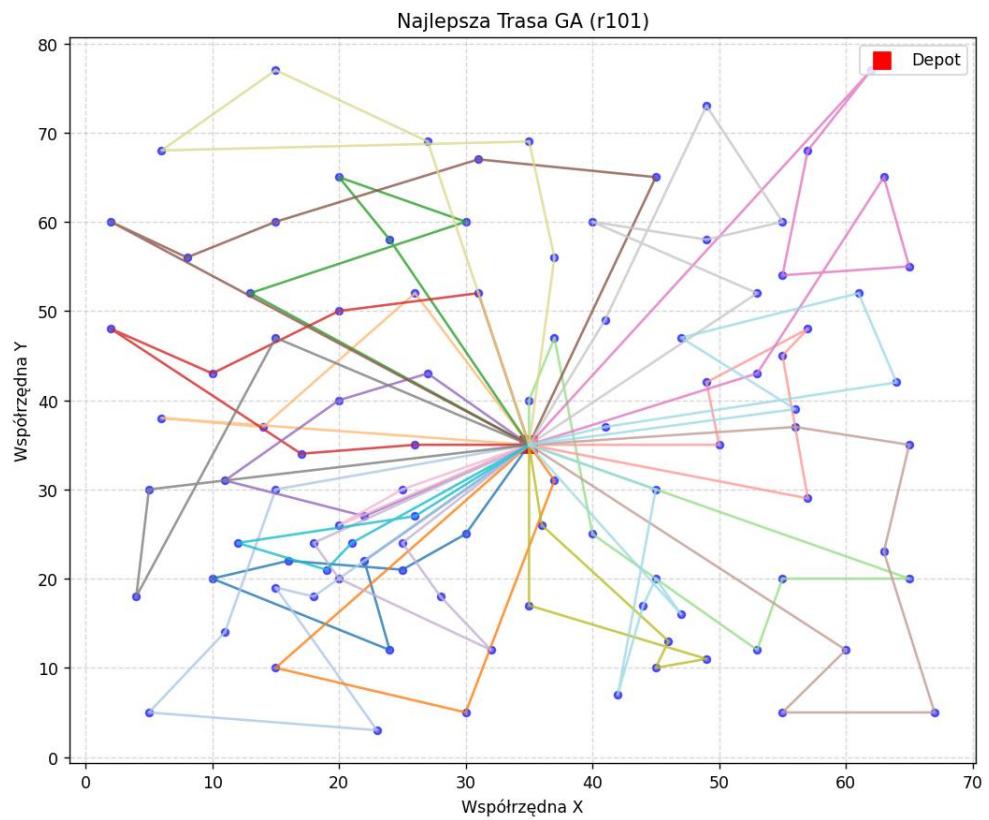
Zmiana całkowitego dystansu - r101 (Pop\_100)



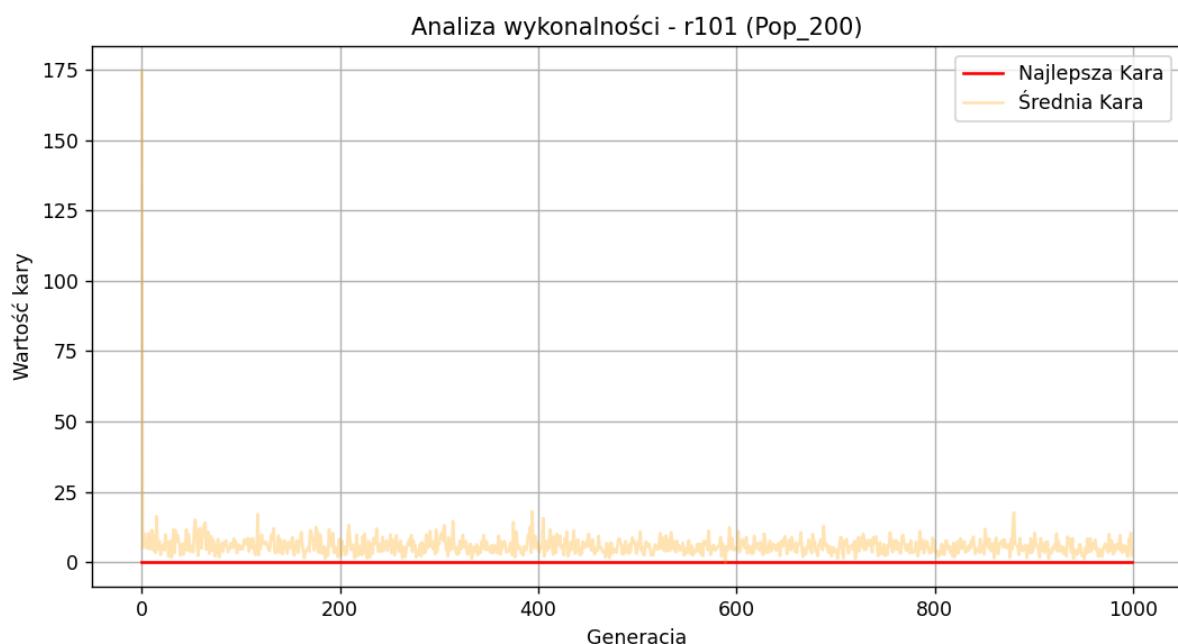
**Wykres 8.** Wizualizacja zmiany dystansu dla najlepszej trasy.



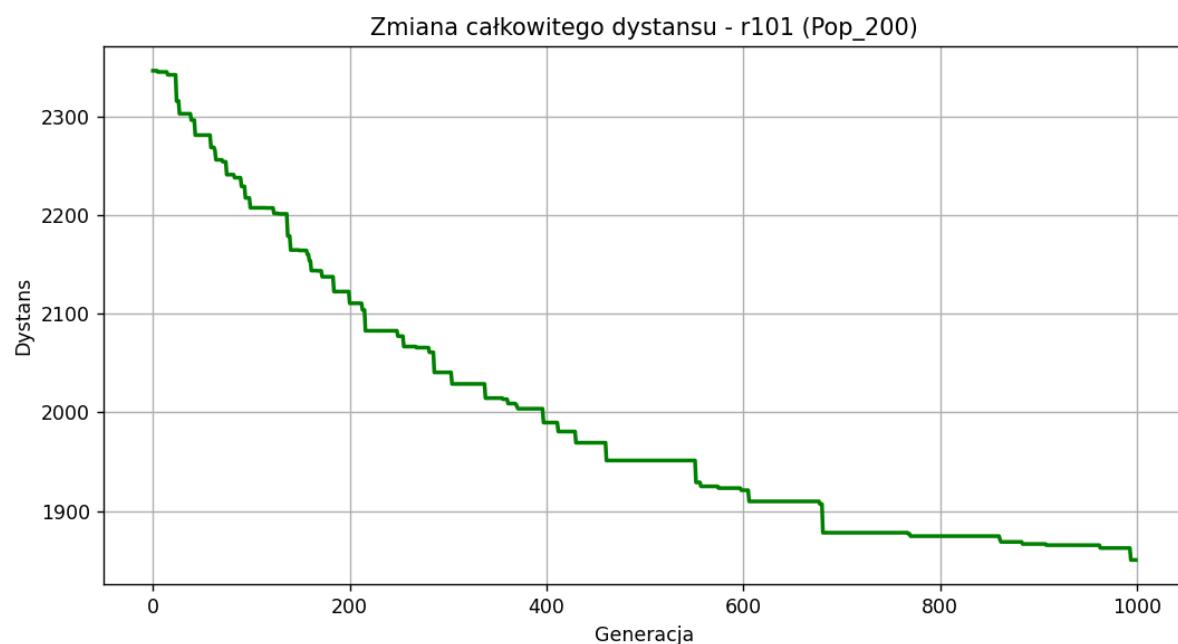
**Wykres 9.** Wizualizacja rozwiązania początkowego.



**Wykres 10.** Wizualizacja najlepszego rozwiązania.



**Wykres 11.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 12.** Wizualizacja zmiany dystansu dla najlepszej trasy.

## 2. Badanie wpływu współczynnika mutacji

Współczynnik mutacji jest kluczowym parametrem kontrolującym balans między eksploracją (przeszukiwaniem nowych obszarów przestrzeni rozwiązań) a eksploatacją (poprawianiem aktualnie najlepszych tras). Wartości zmieniane będą wynosiły:  $Pm = [0.05, 0.1, 0.4]$ .

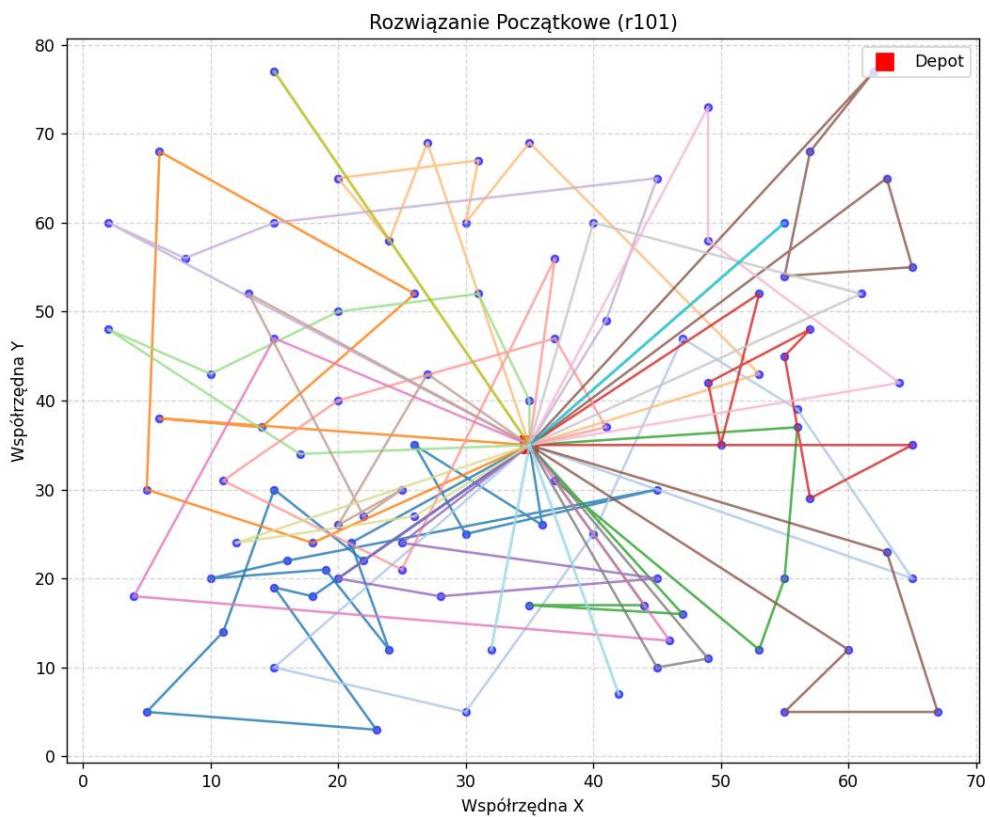
Eksperyment zostanie wykonany po 5 razy dla każdej badanej wartości dla 3 plików.

**Tabela 3.** Wyniki 5 eksperymentów dla badanego współczynnika.

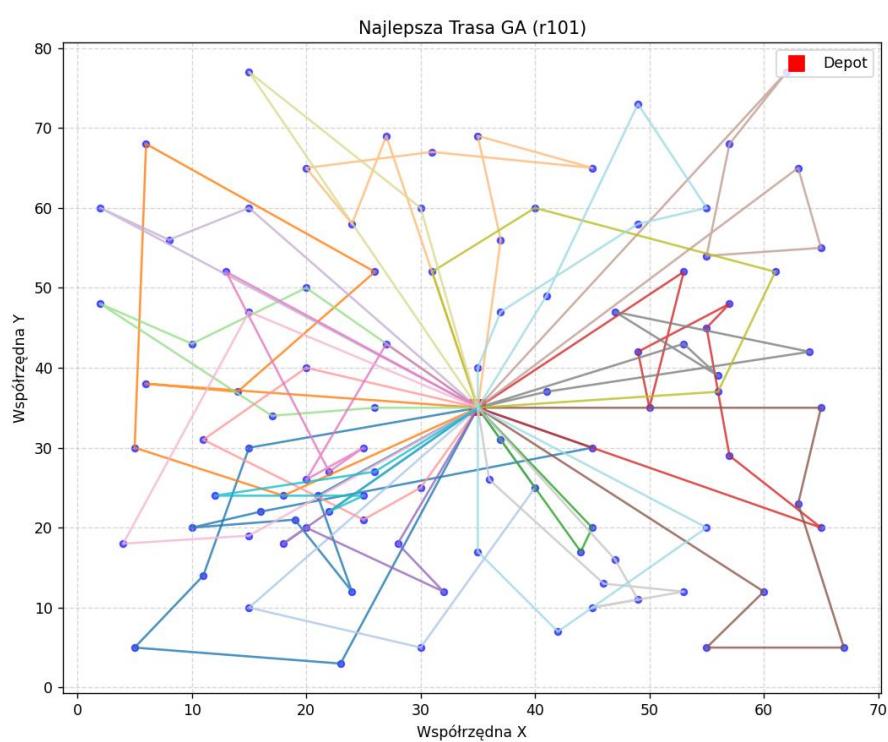
Lp.	Mutacja (Pm)	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	0.05	1	23	2017.51	0.00
2		2	23	2078.32	0.00
3		3	23	2065.47	0.00
4		4	22	2025.96	0.00
5		5	23	2016.18	0.00
6	0.10	1	23	1953.96	0.00
7		2	23	1942.37	0.00
8		3	22	1936.50	0.00
9		4	21	1868.30	0.00
10		5	22	1928.53	0.00
11	0.40	1	22	1769.25	0.00
12		2	22	1854.37	0.00
13		3	21	1810.56	0.00
14		4	21	1829.63	0.00
15		5	22	1824.34	0.00

**Tabela 4.** Wyniki podsumowujące.

Mutacja (Pm)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
0.05	2025.96	2040.69	26.02	22
0.10	1868.30	1925.93	29.99	21
0.40	1769.25	1817.63	28.03	22

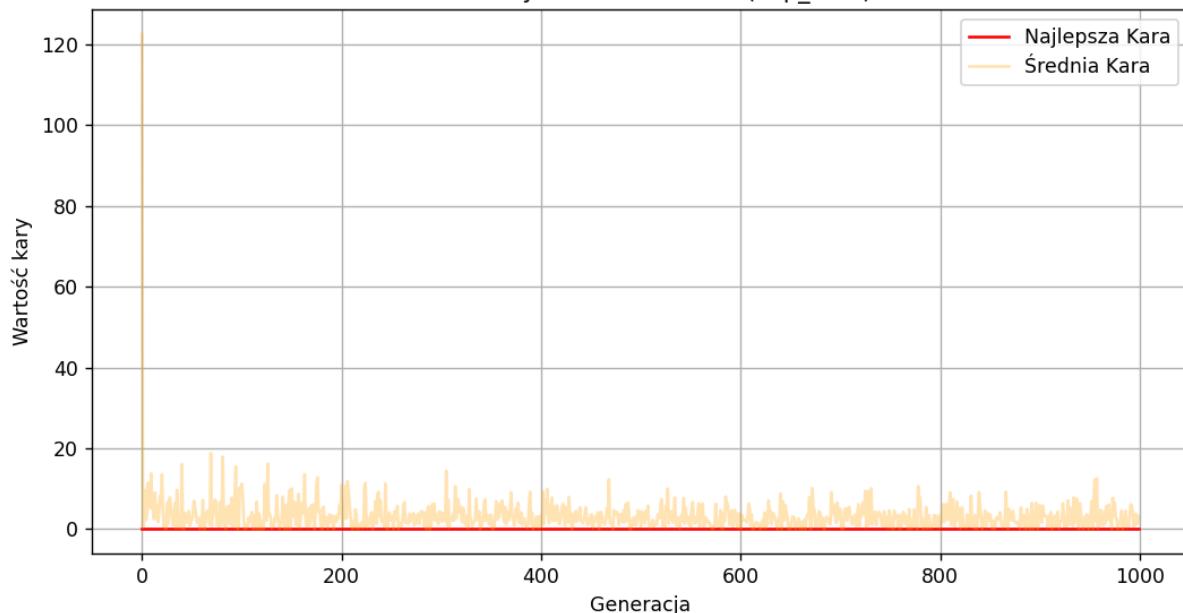


**Wykres 13.** Wizualizacja rozwiązania początkowego.



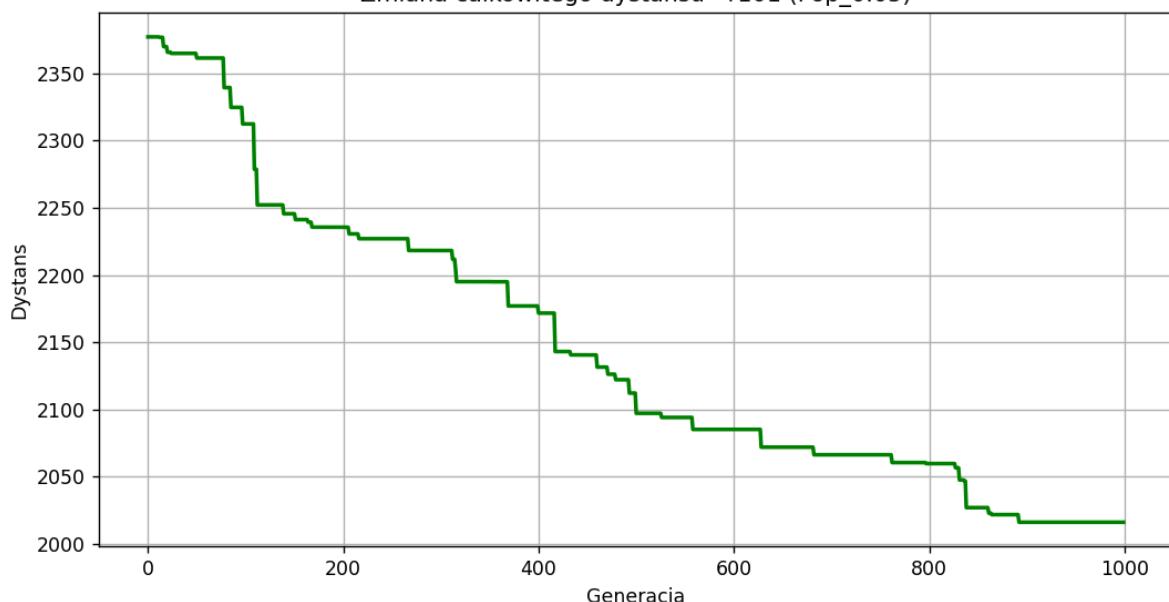
**Wykres 14.** Wizualizacja najlepszego rozwiązania.

Analiza wykonalności - r101 (Pop\_0.05)

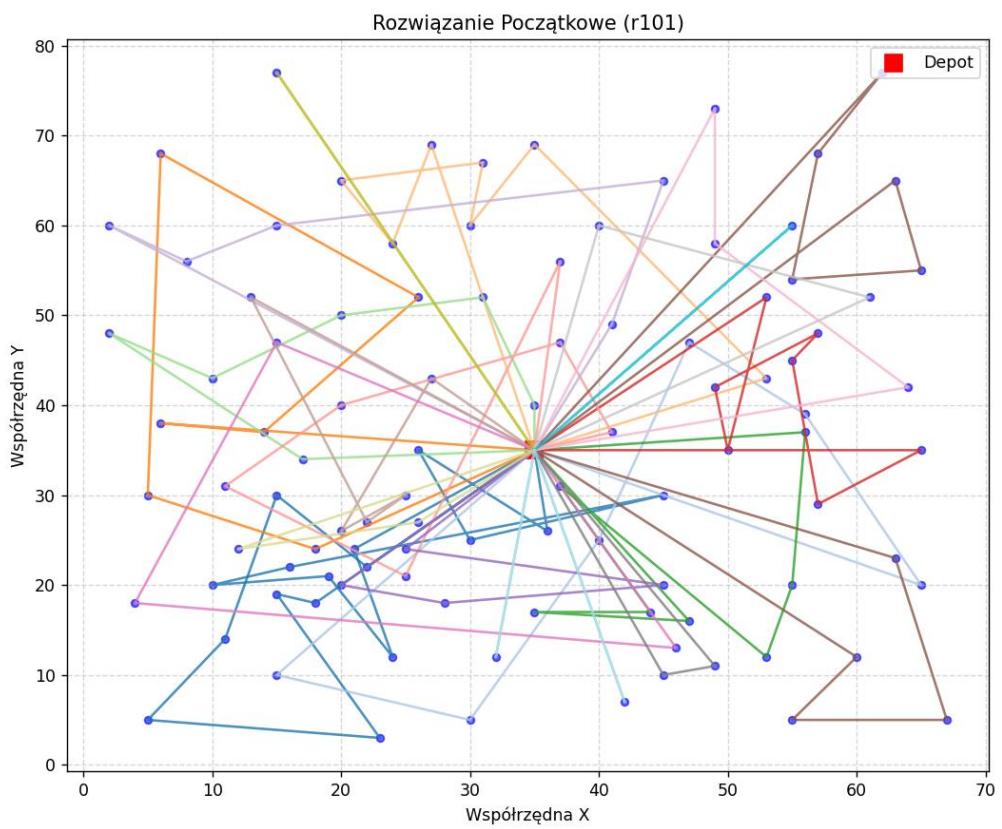


Wykres 15. Wizualizacja średniej kary populacji dla najlepszej trasy.

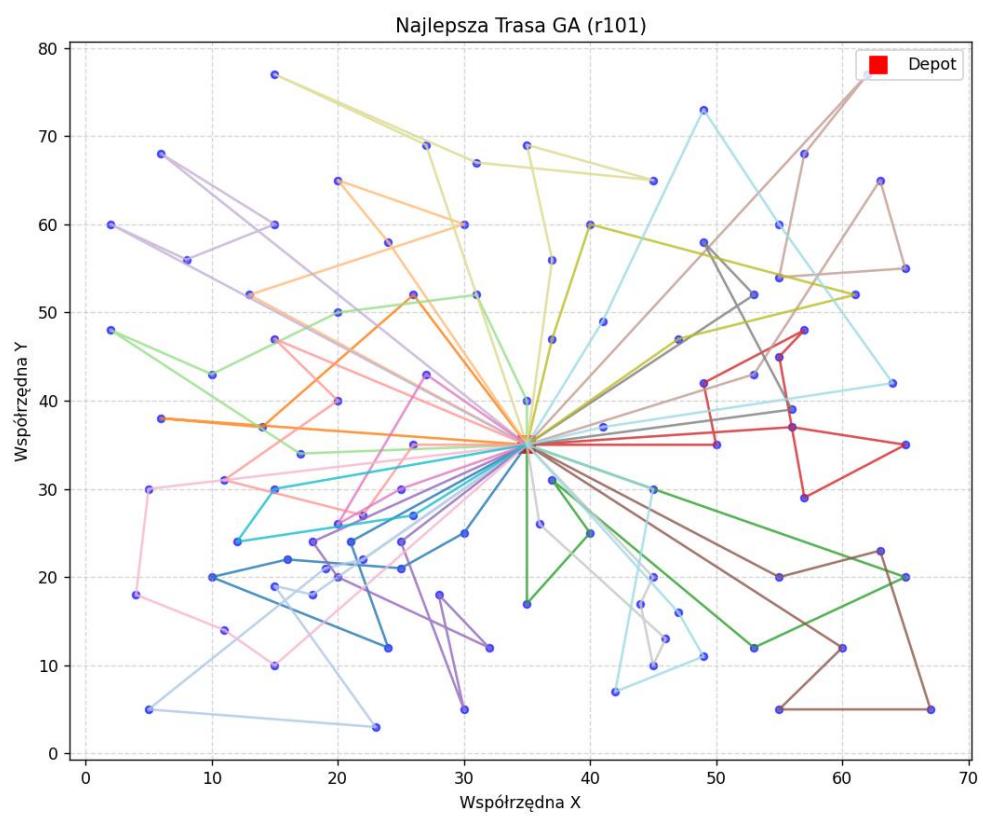
Zmiana całkowitego dystansu - r101 (Pop\_0.05)



Wykres 16. Wizualizacja zmiany dystansu dla najlepszej trasy.

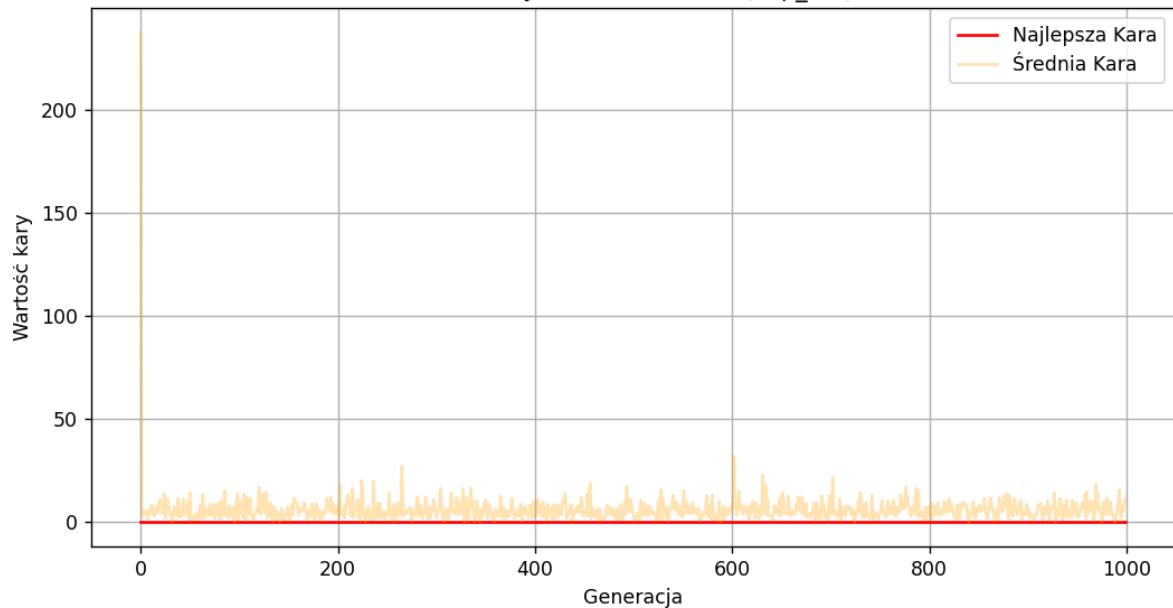


**Wykres 17.** Wizualizacja rozwiązania początkowego.



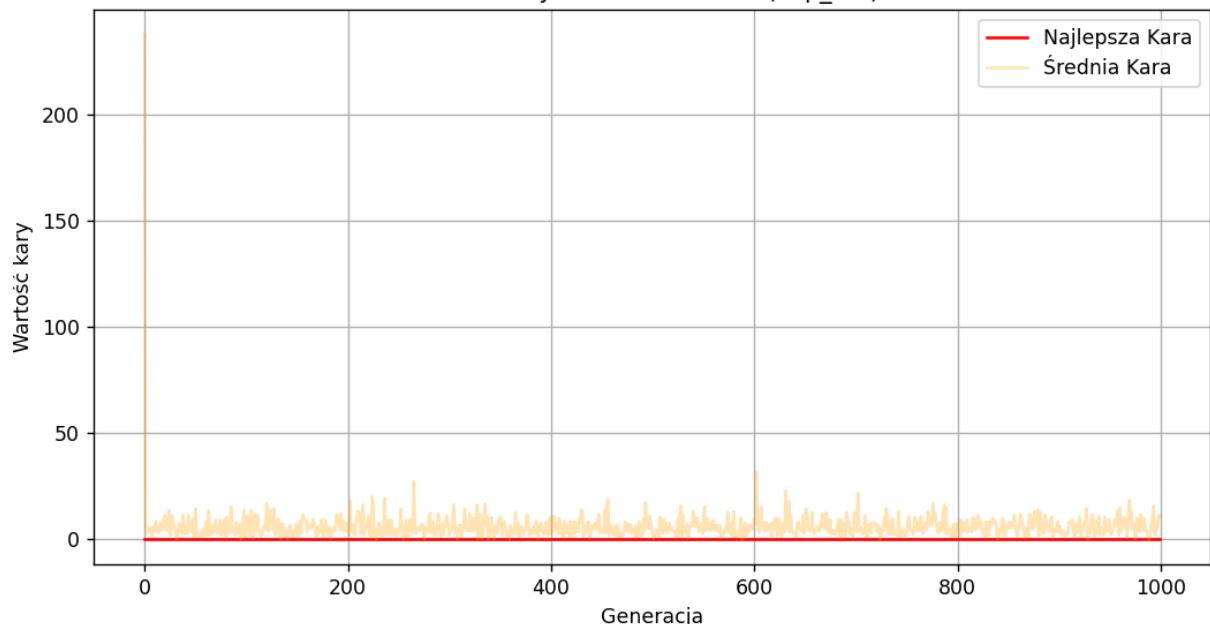
**Wykres 18.** Wizualizacja najlepszego rozwiązania.

Analiza wykonalności - r101 (Pop\_0.1)



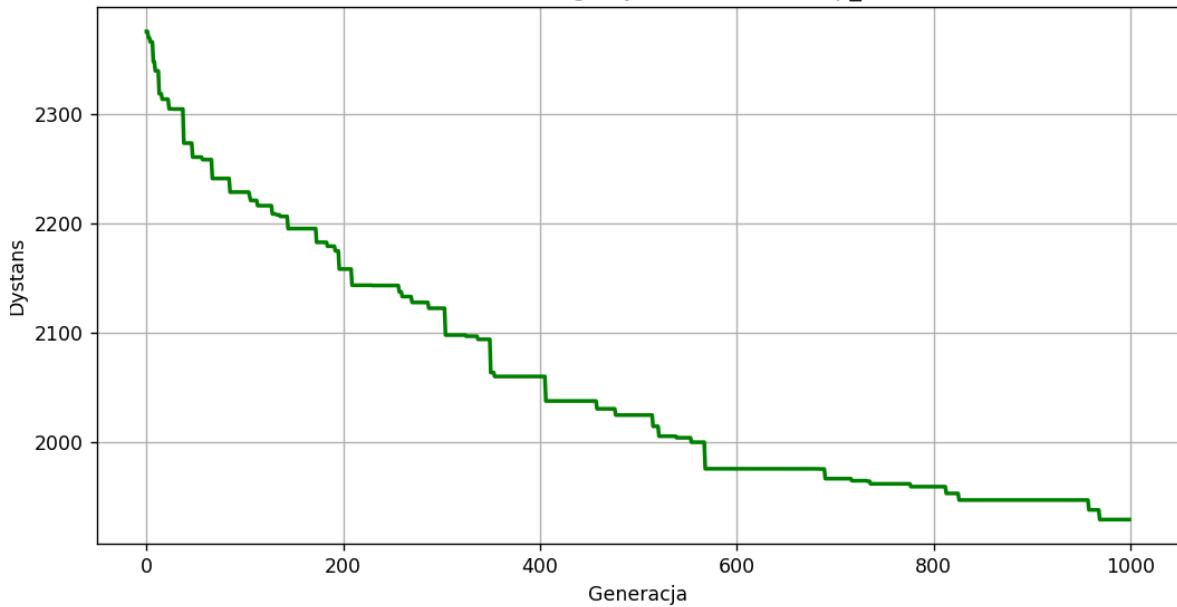
**Wykres 19.** Wizualizacja średniej kary populacji dla najlepszej trasy.

Analiza wykonalności - r101 (Pop\_0.1)

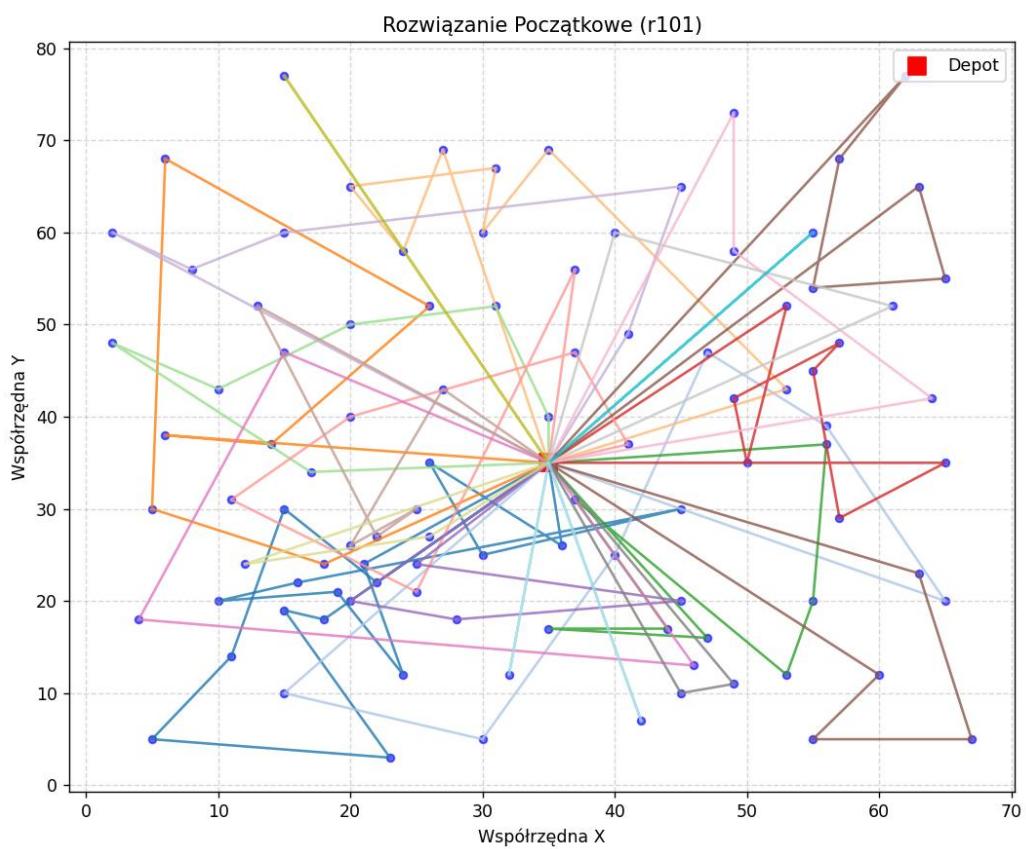


**Wykres 20.** Wizualizacja zmiany dystansu dla najlepszej trasy.

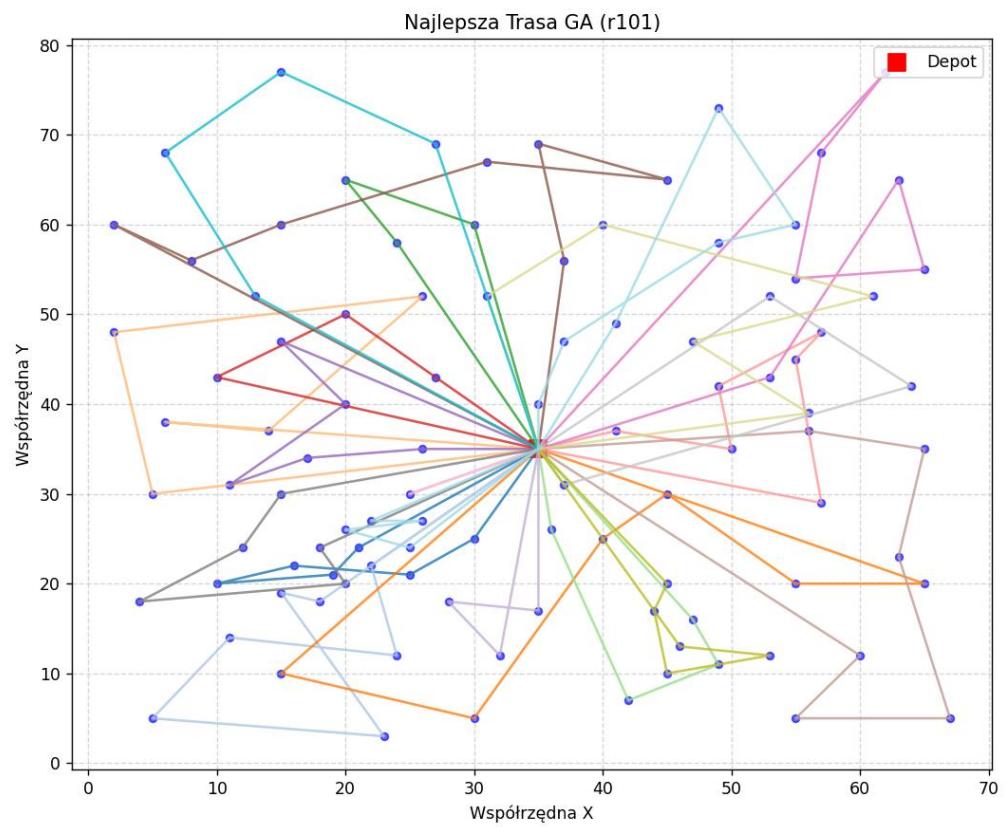
Zmiana całkowitego dystansu - r101 (Pop\_0.1)



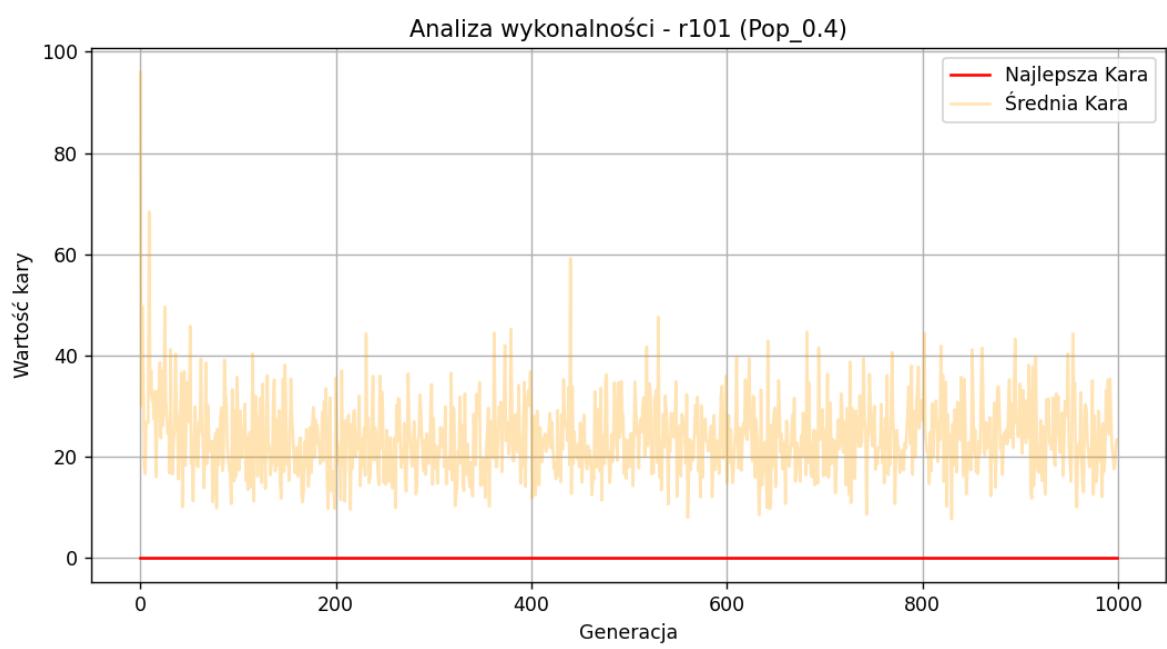
Wykres 21. Wizualizacja zmiany dystansu dla najlepszej trasy.



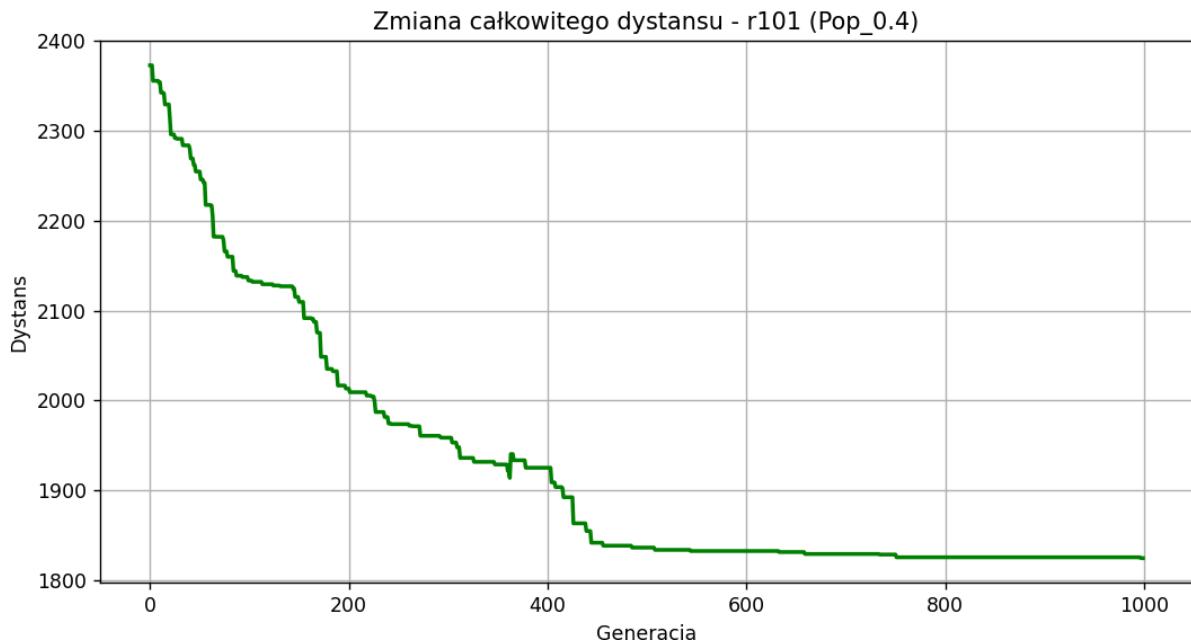
Wykres 22. Wizualizacja rozwiązania początkowego.



**Wykres 23.** Wizualizacja najlepszego rozwiązania.



**Wykres 24.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 24.** Wizualizacja zmiany dystansu dla najlepszej trasy.

### 3. Badanie wpływu współczynnika elitaryzmu.

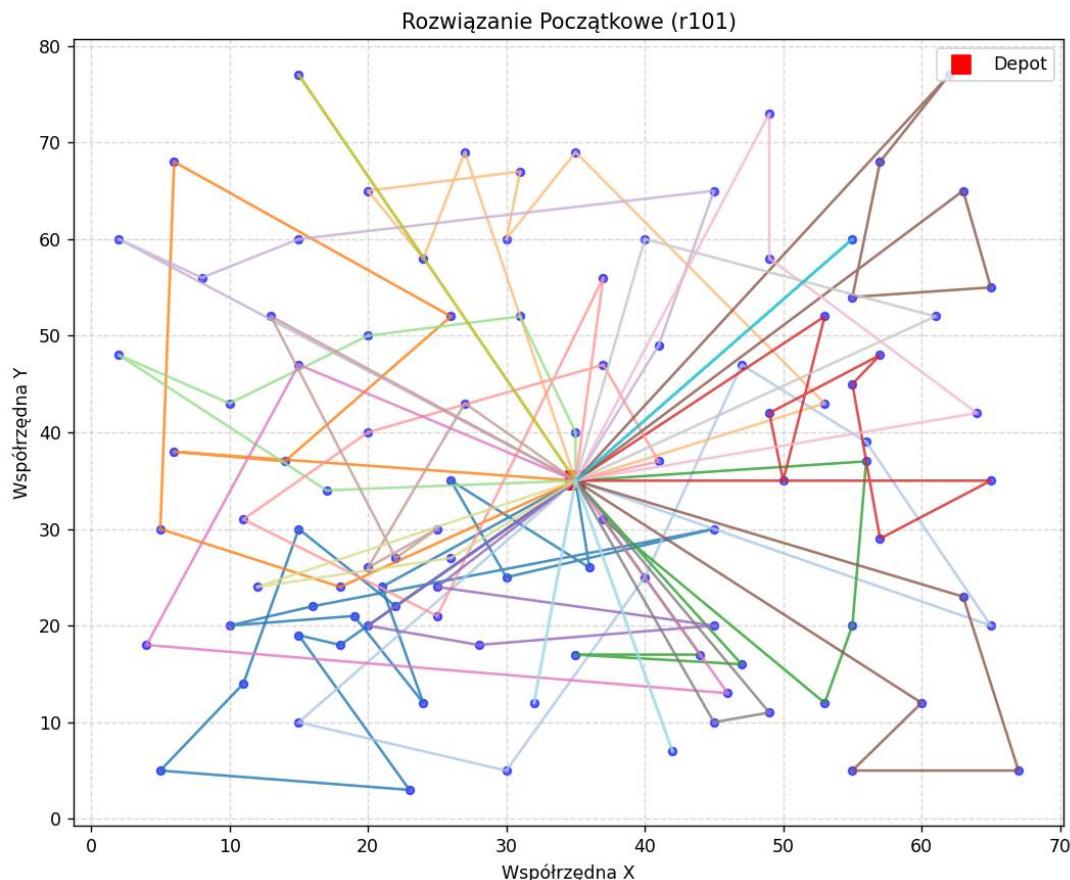
Elitaryzm chroni najlepsze osobniki przed negatywnymi skutkami losowości. W problemie VRPTW, gdzie znalezienie trasy spełniającej wszystkie okna czasowe (Penalty = 0) jest trudne, elitaryzm pełni rolę "bezpiecznika". Niski elitaryzm: Pozwala na dużą rotację w populacji, ale ryzykuje utratą rekordowego wyniku ("regresja"). Wysoki elitaryzm: Gwarantuje zachowanie najlepszych cech, ale może prowadzić do zbyt szybkiej dominacji jednego osobnika i utraty różnorodności (stagnacja w minimum lokalnym). Przeprowadzane będzie badanie wartości E = [0.05, 0.2, 0.4].

**Tabela 5.** Wyniki 5 eksperymentów dla badanego współczynnika.

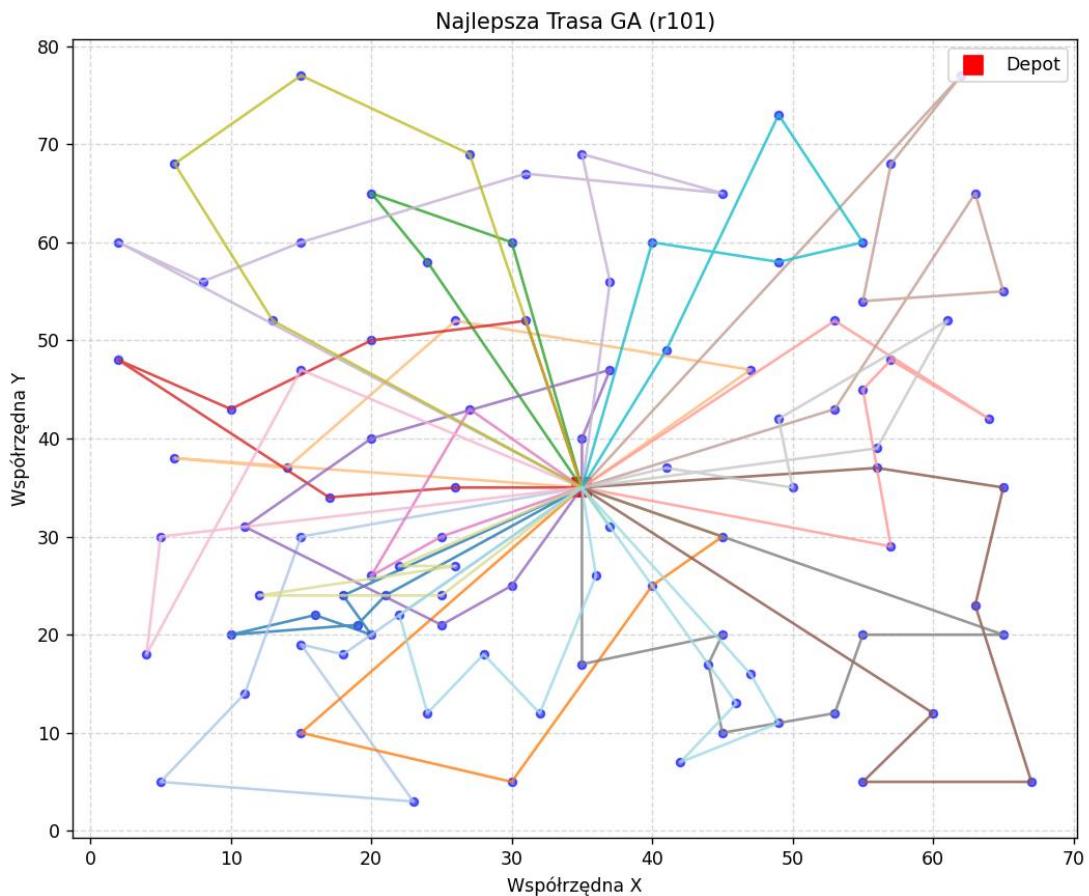
Lp.	Elitaryzm (E)	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	0.05	1	22	1831.08	0.00
2		2	21	1882.66	0.00
3		3	21	1862.63	0.00
4		4	21	1800.75	0.00
5		5	22	1818.29	0.00
6	0.20	1	21	1783.43	0.00
7		2	22	1784.20	0.00
8		3	22	1846.29	0.00
9		4	21	1841.95	0.00
10		5	22	1826.87	0.00
11	0.40	1	21	1793.75	0.00
12		2	22	1846.93	0.00
13		3	22	1769.41	0.00
14		4	21	1825.10	0.00
15		5	22	1862.48	0.00

**Tabela 6.** Wyniki podsumowujące.

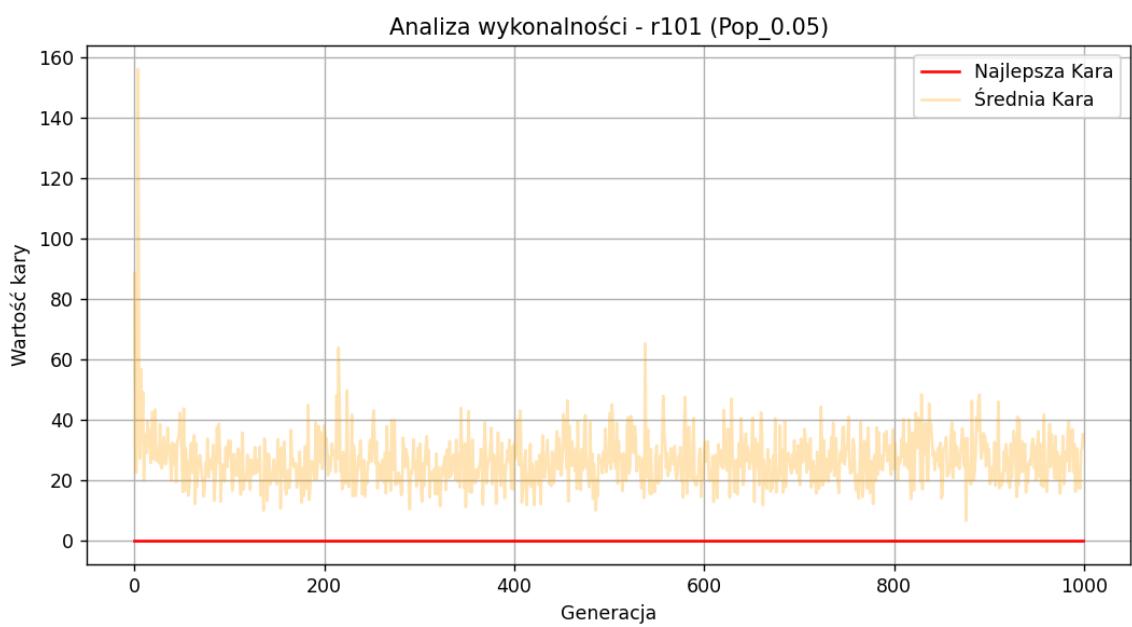
Elitaryzm (E)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
0.05	1800.75	1839.08	29.73	21
0.20	1783.43	1816.55	27.49	21
0.40	1793.75	1819.54	34.07	21



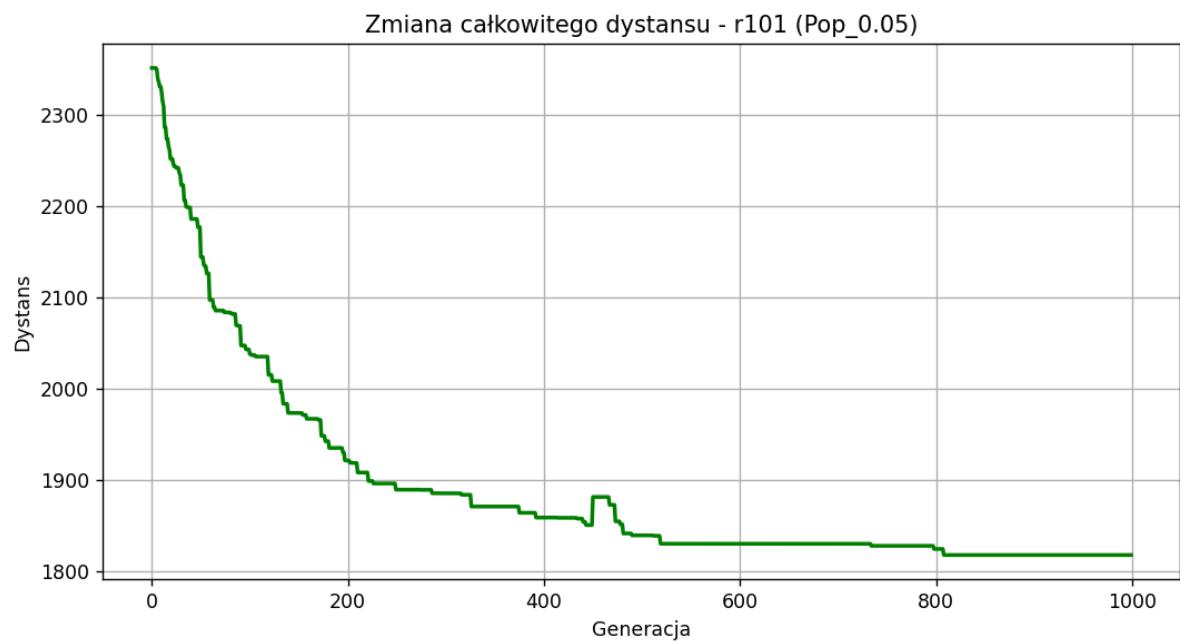
**Wykres 25.** Wizualizacja rozwiązania początkowego.



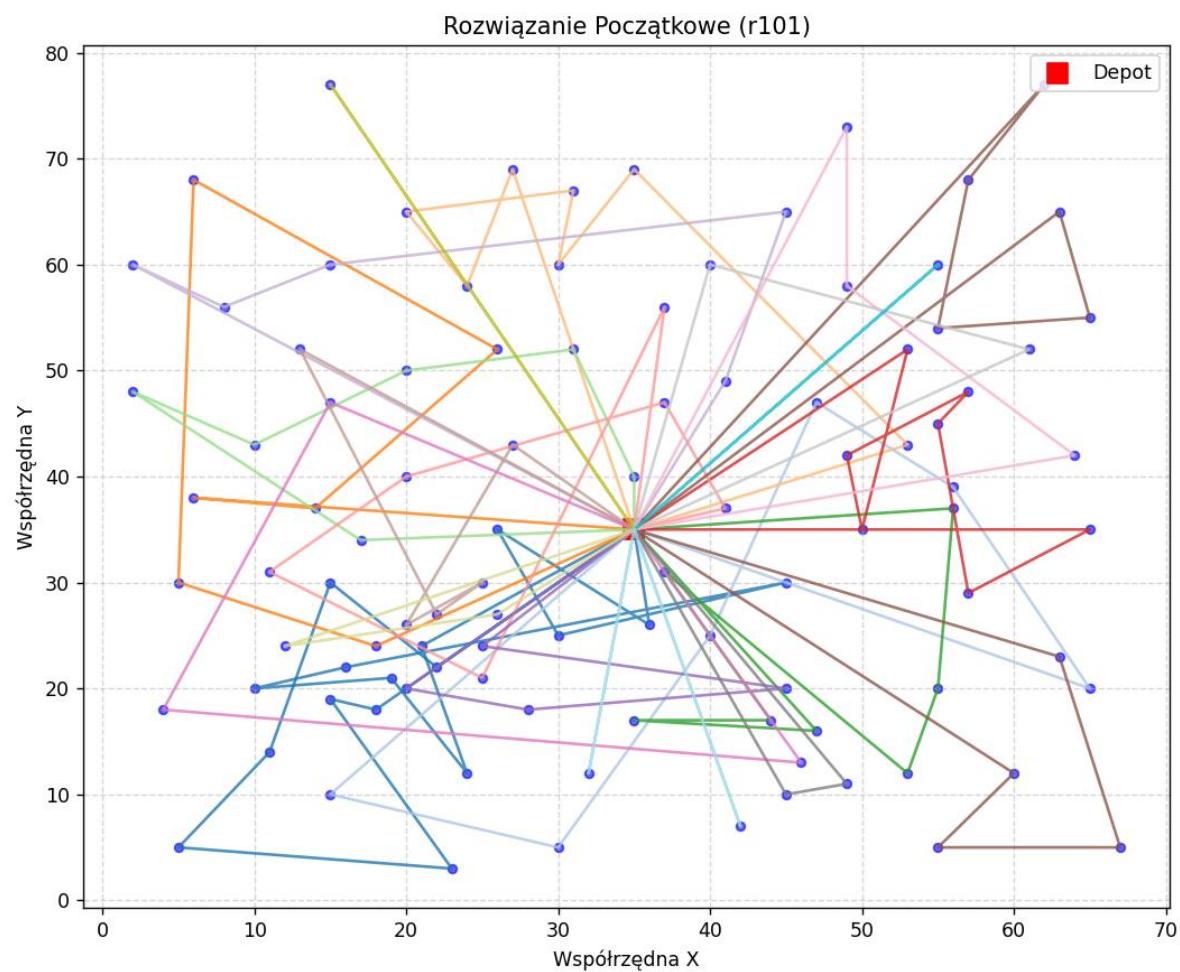
**Wykres 26.** Wizualizacja najlepszego rozwiązania.



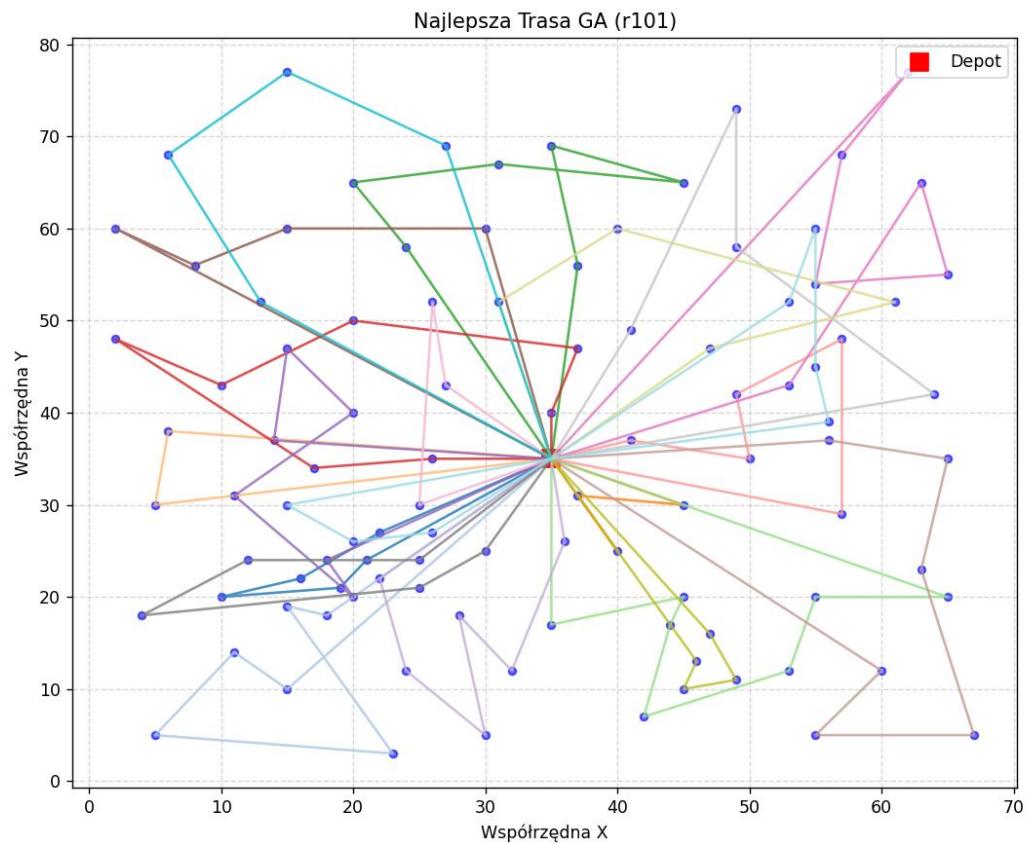
**Wykres 27.** Wizualizacja średniej kary populacji dla najlepszej trasy.



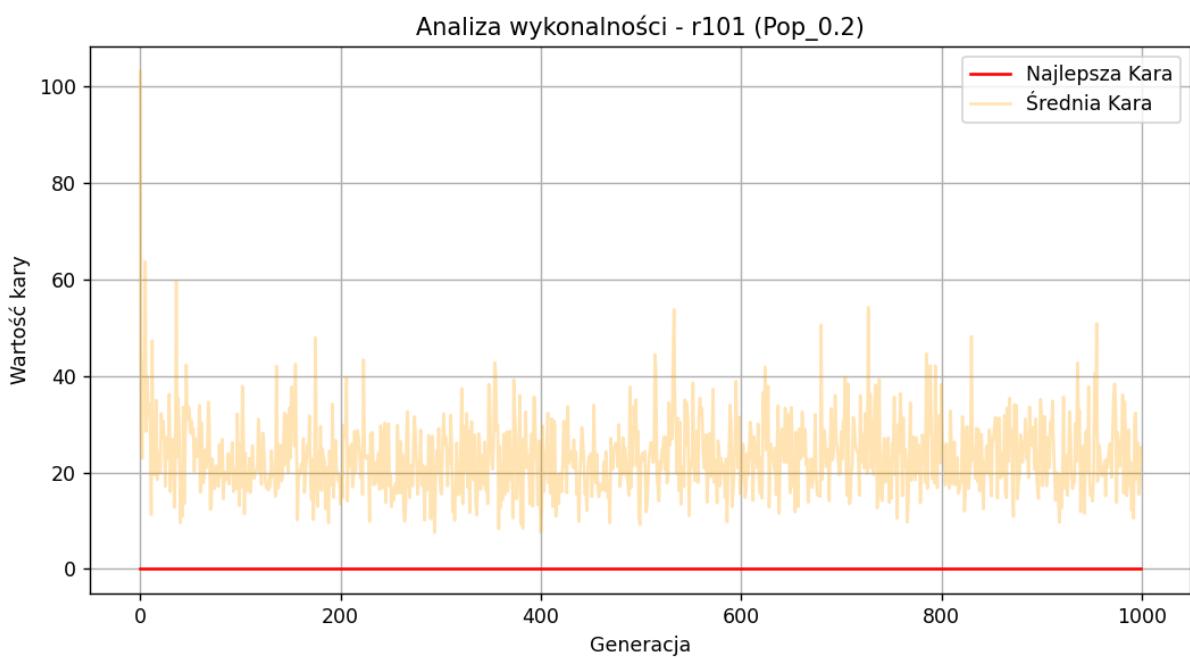
**Wykres 28.** Wizualizacja zmiany dystansu dla najlepszej trasy.



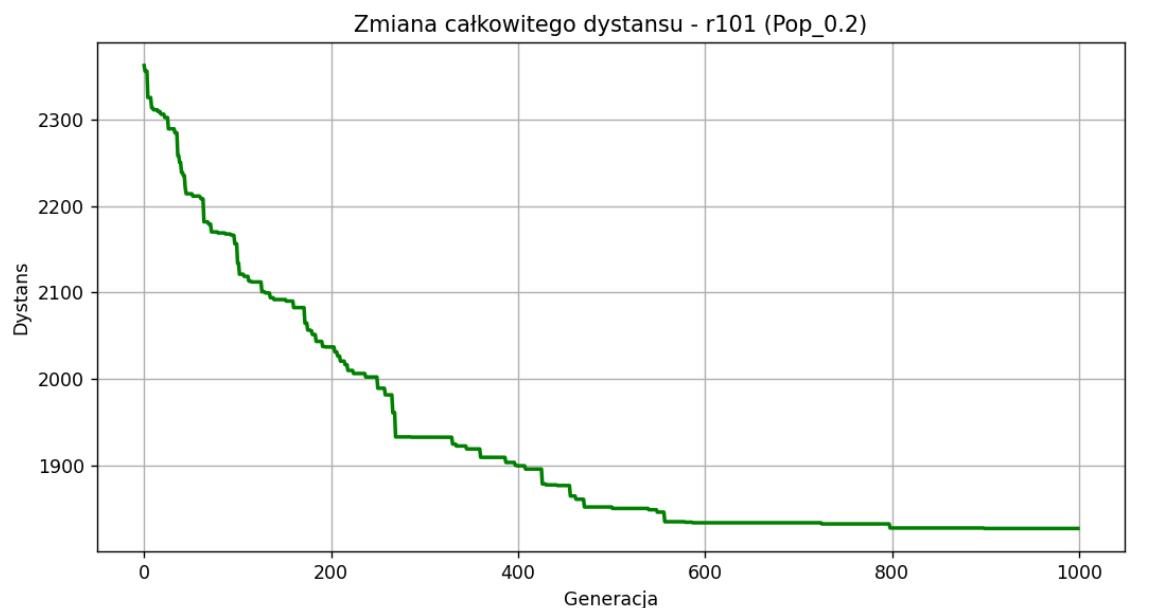
**Wykres 29.** Wizualizacja rozwiązania początkowego.



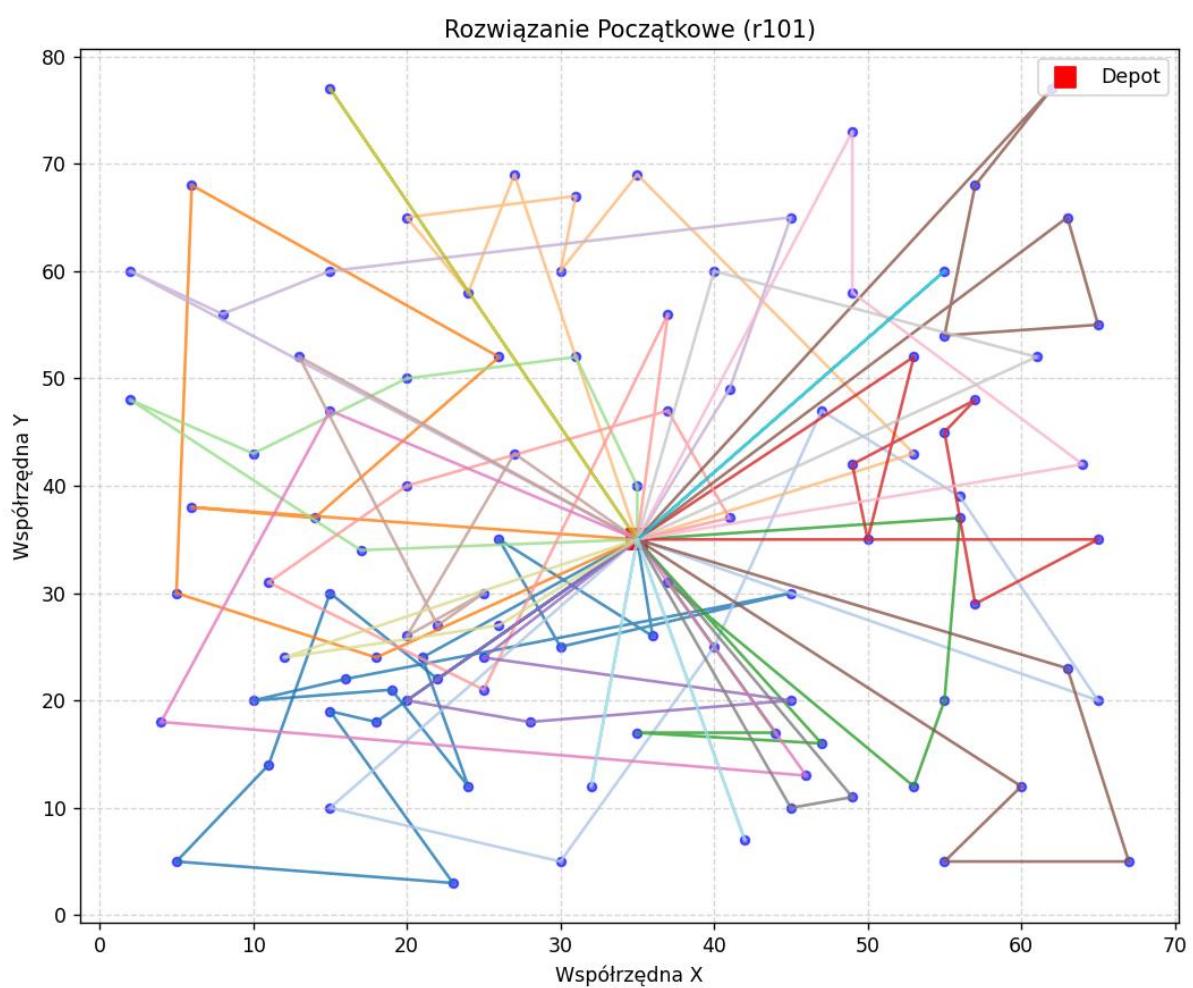
**Wykres 30.** Wizualizacja najlepszego rozwiązania



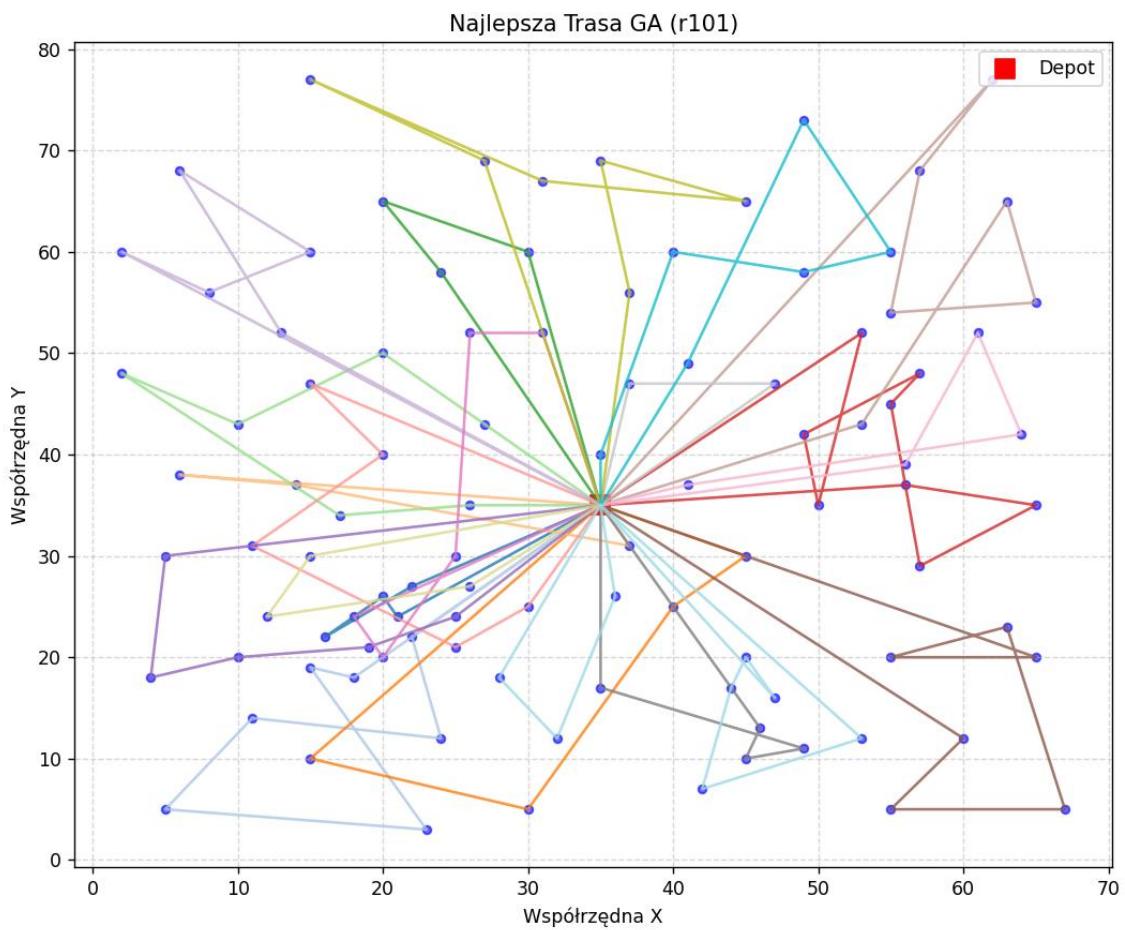
**Wykres 31.** Wizualizacja średniej kary populacji dla najlepszej trasy.



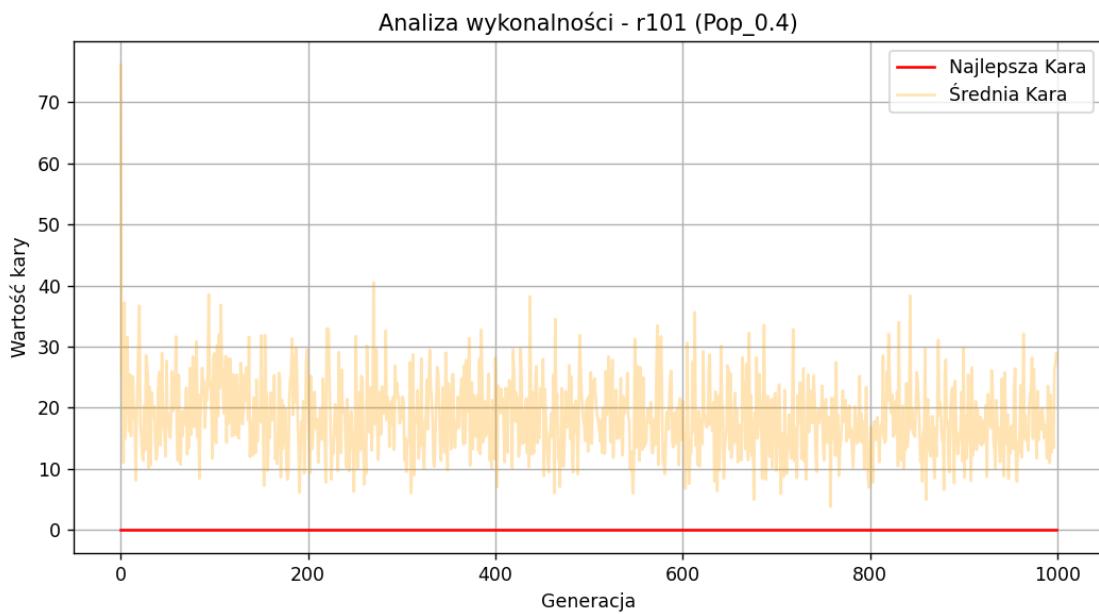
**Wykres 32.** Wizualizacja zmiany dystansu dla najlepszej trasy.



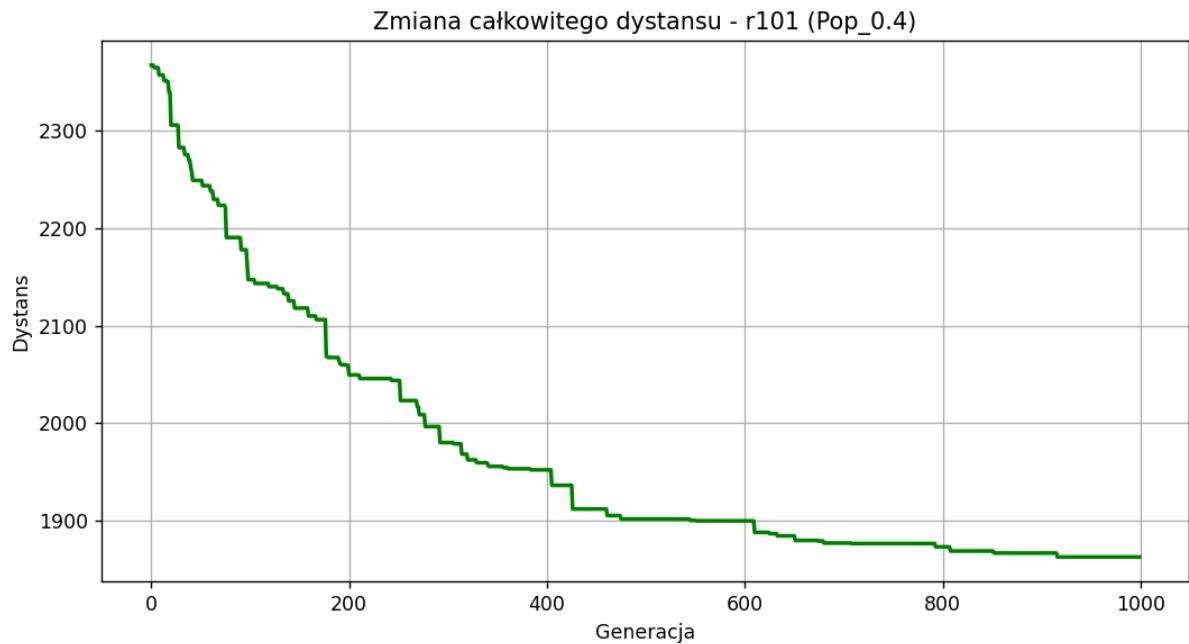
**Wykres 33.** Wizualizacja rozwiązania początkowego.



**Wykres 34.** Wizualizacja najlepszego rozwiązania.



**Wykres 35.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 36.** Wizualizacja zmiany dystansu dla najlepszej trasy.

## 6.2 Badanie dla pliku c101.txt

### 1. Badanie wpływu wielkości populacji

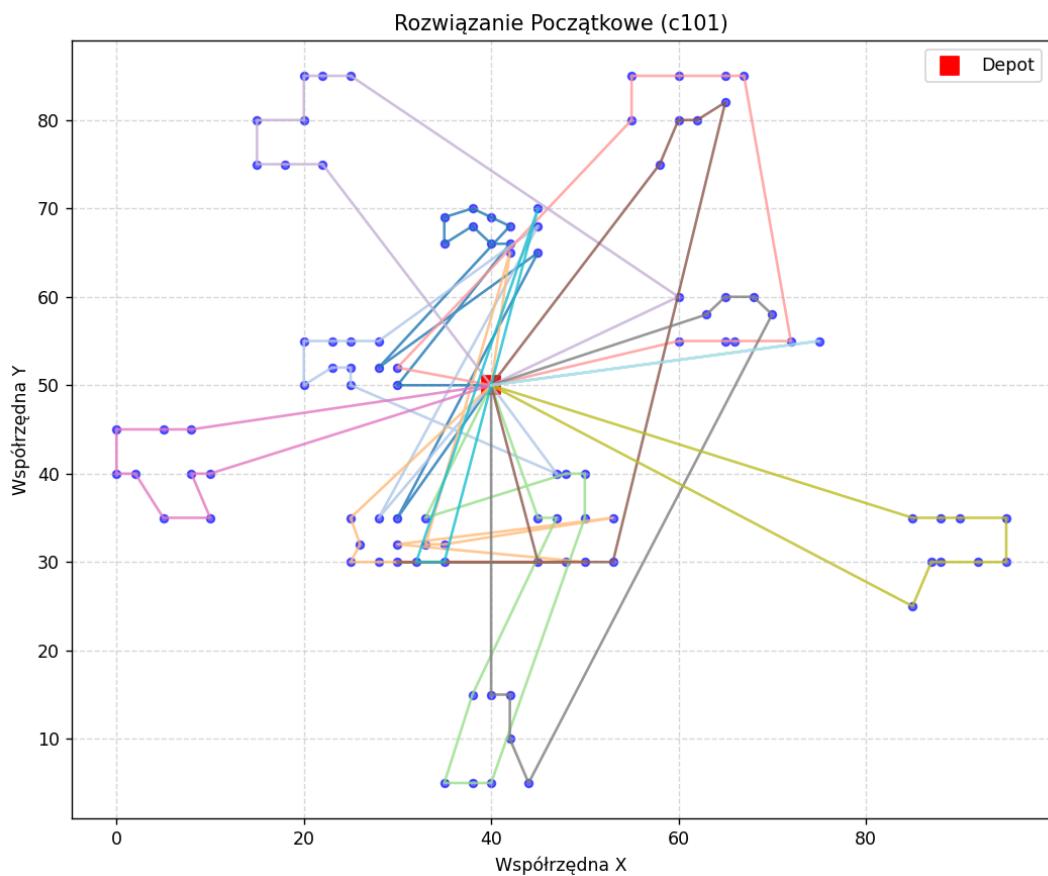
Badania wpływu wielkości populacji (N). Ten parametr decyduje o balansie między "szerokością" poszukiwań (eksploracją) a kosztem obliczeniowym. Wartości zmieniane będą wynosiły: N = [50,100,200].

**Tabela 7.** Wyniki 5 eksperymentów dla badanego współczynnika.

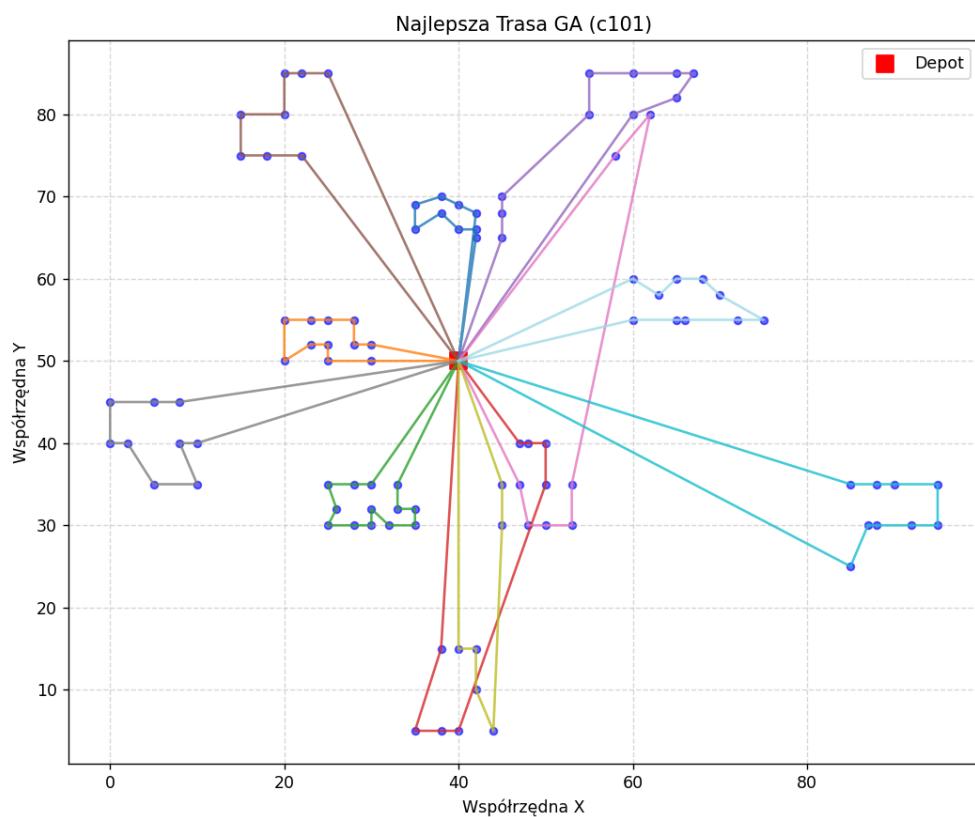
Lp.	Populacja (N)	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	50	1	12	1142.51	0.00
2		2	11	972.70	0.00
3		3	12	1011.32	0.00
4		4	11	997.97	0.00
5		5	12	983.09	0.00
6	100	1	12	947.62	0.00
7		2	11	1014.98	0.00
8		3	12	1048.84	0.00
9		4	12	1090.57	0.00
10		5	12	984.80	0.00
11	200	1	12	1049.94	0.00
12		2	12	1025.29	0.00
13		3	11	929.49	0.00
14		4	11	972.51	0.00
15		5	12	981.76	0.00

**Tabela 8.** Wyniki podsumowujące.

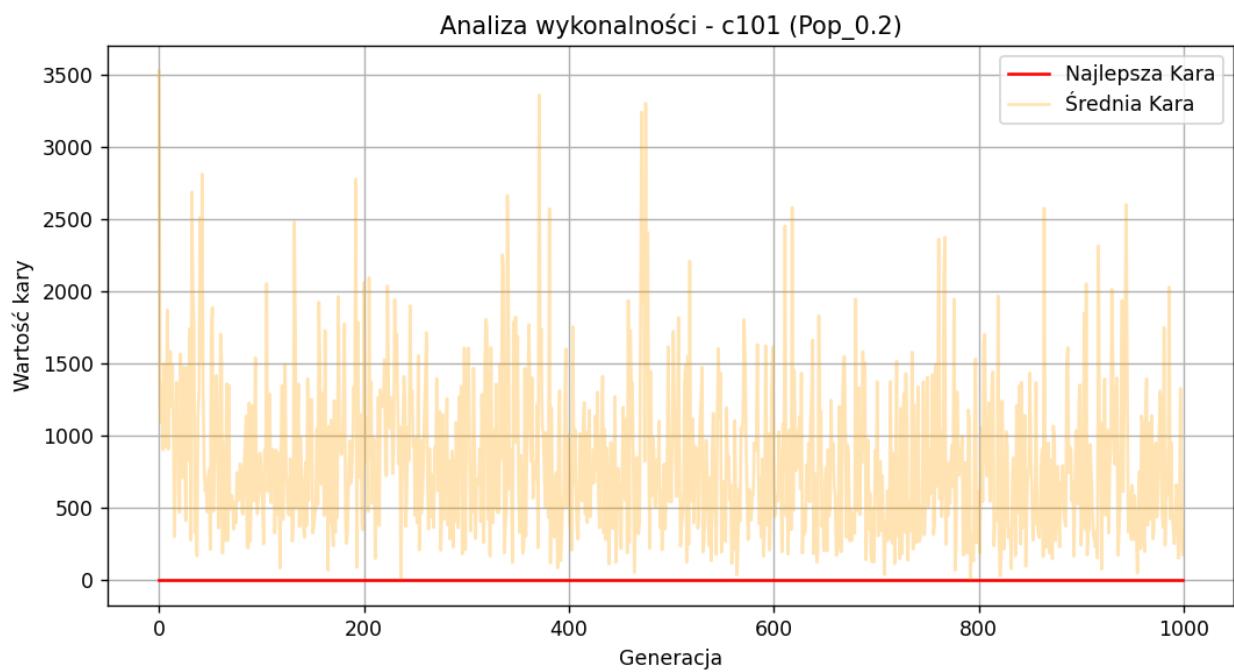
Populacja (N)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
50	972.70	1021.52	61.90	11
100	947.62	1017.36	49.56	12
200	929.49	991.80	42.09	11



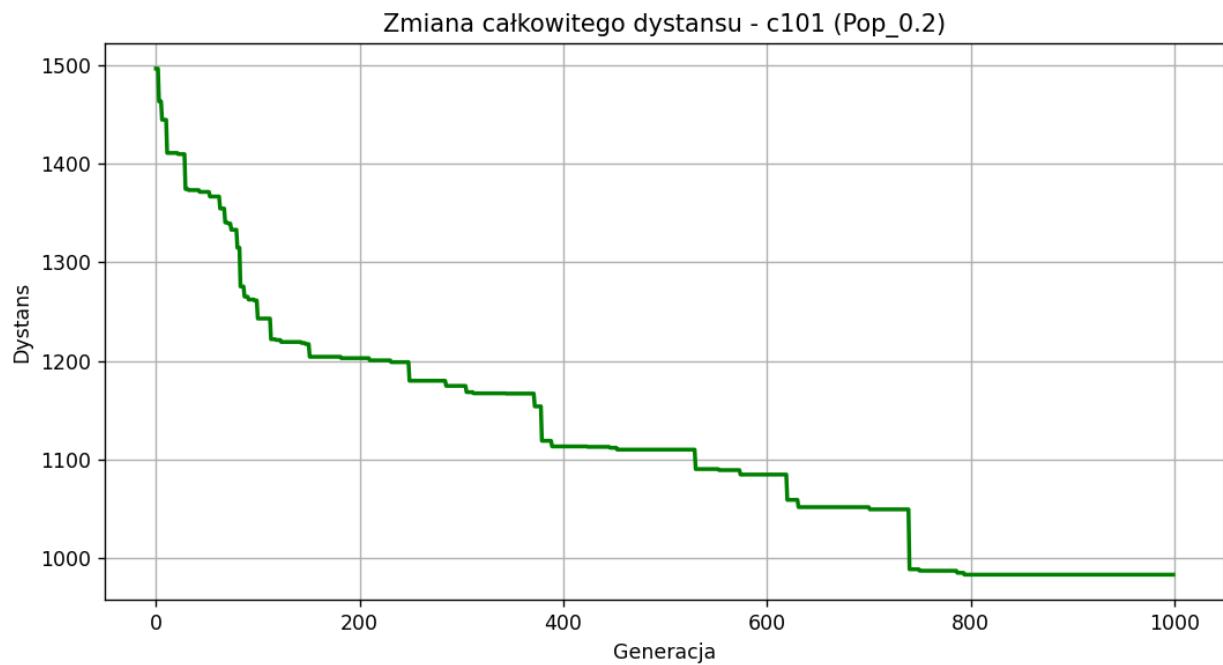
**Wykres 37.** Wizualizacja rozwiązania początkowego.



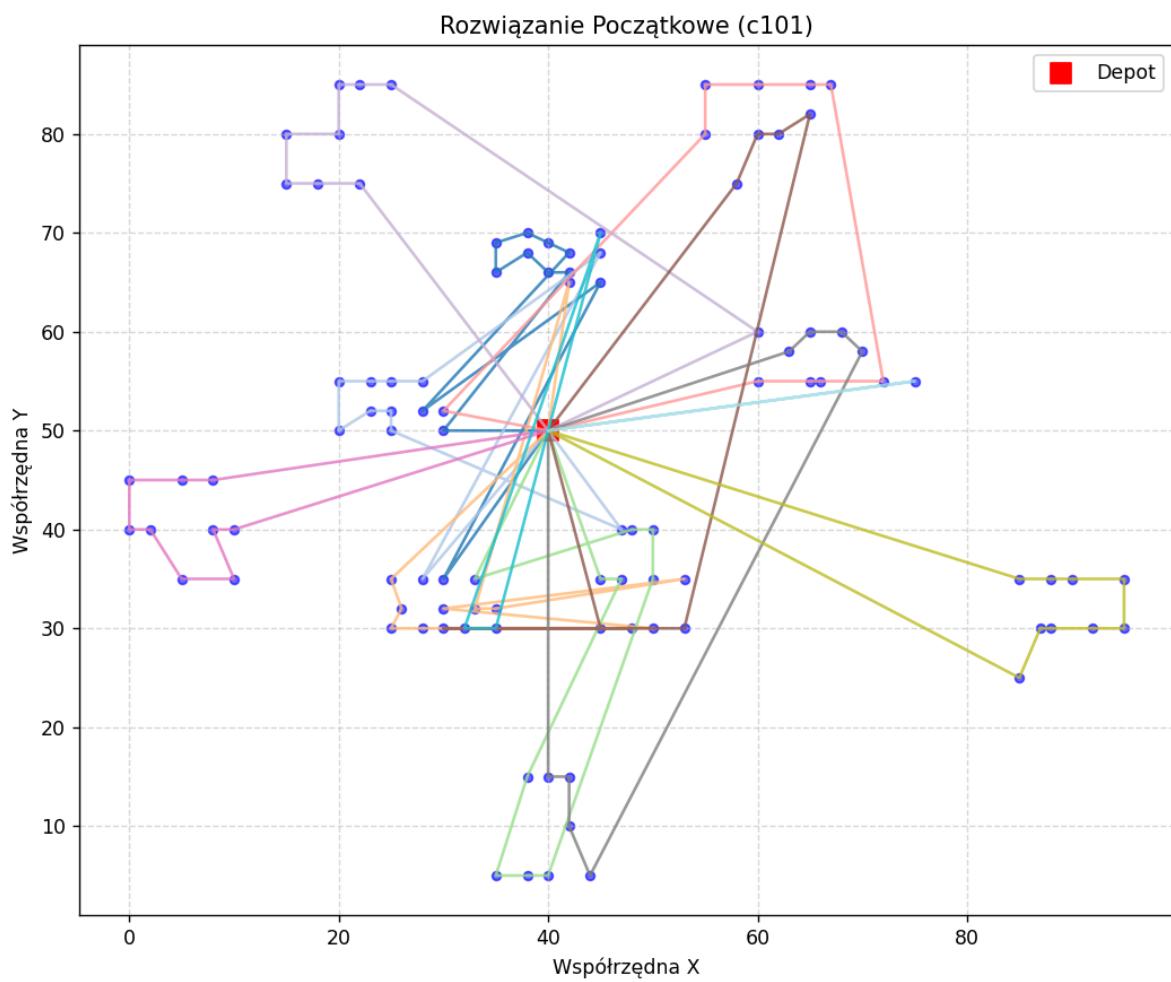
**Wykres 38.** Wizualizacja najlepszego rozwiązania.



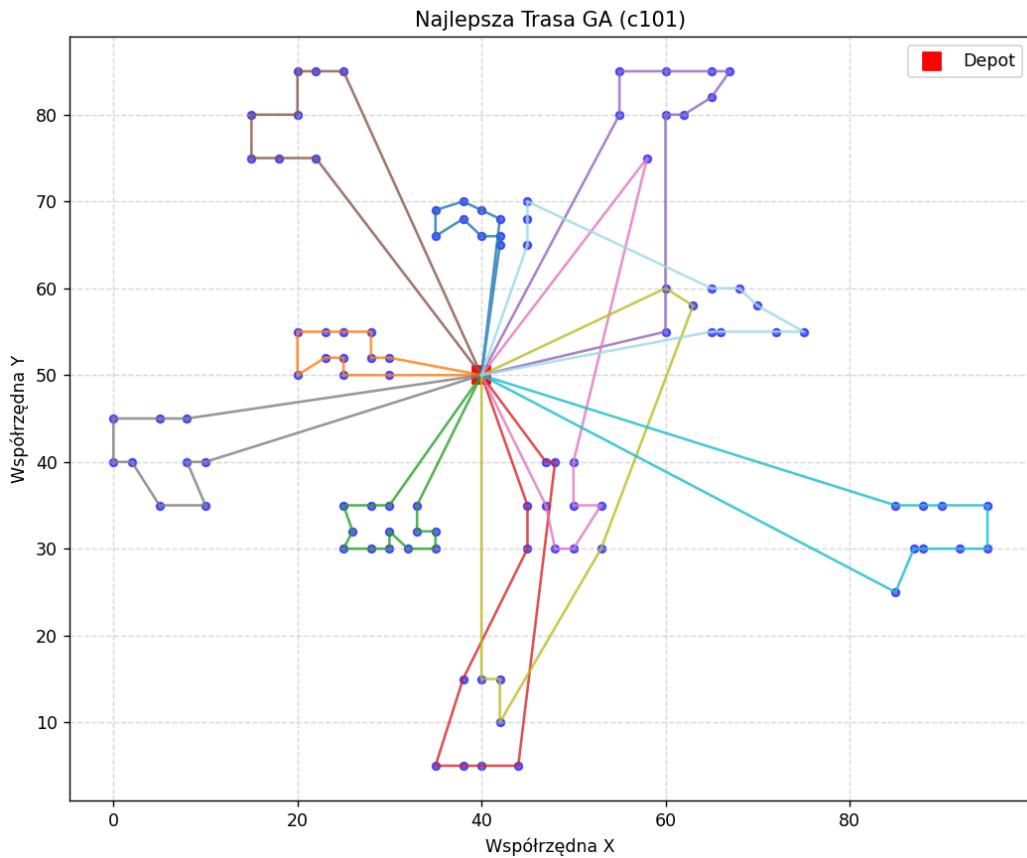
**Wykres 39.** Wizualizacja średniej kary populacji dla najlepszej trasy.



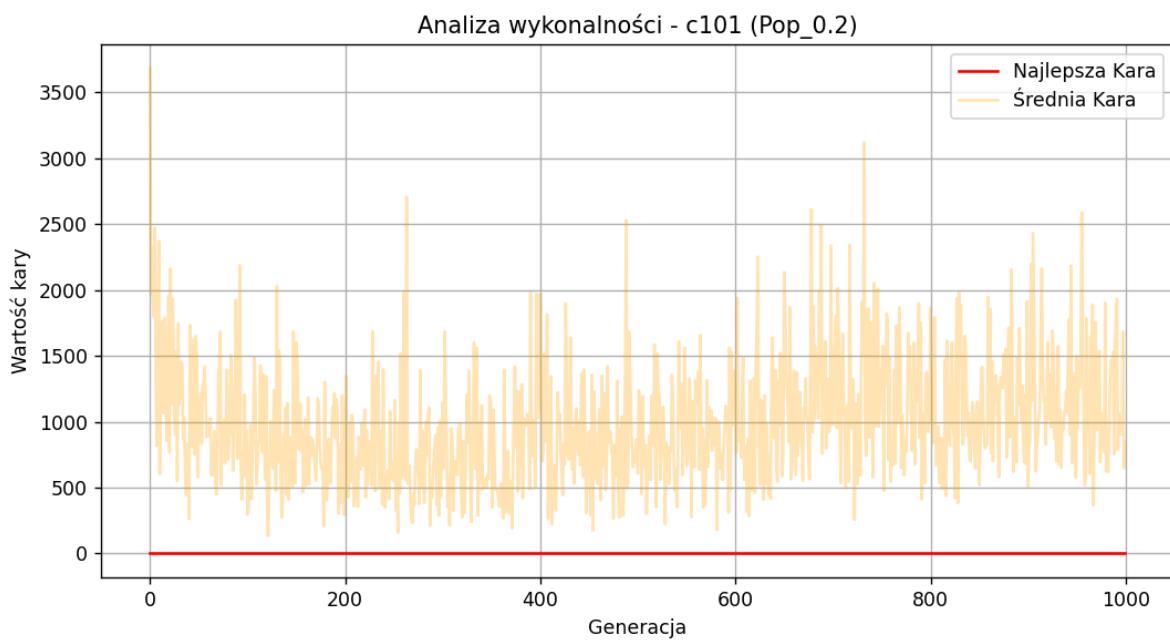
**Wykres 40.** Wizualizacja zmiany dystansu dla najlepszej trasy.



**Wykres 41.** Wizualizacja rozwiązania początkowego.

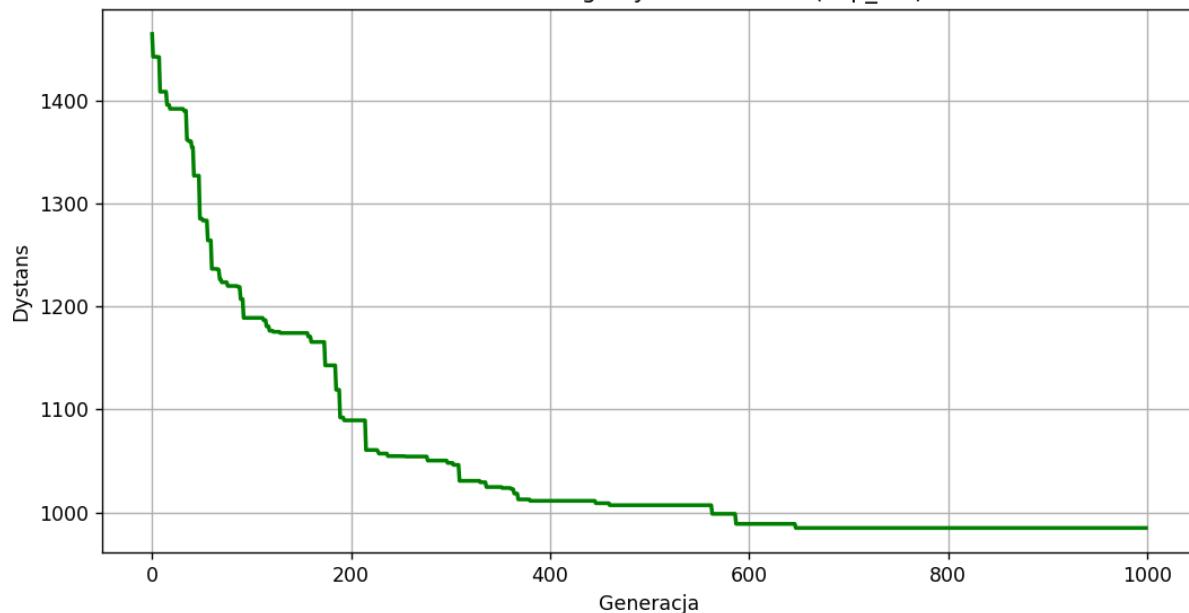


**Wykres 42.** Wizualizacja najlepszego rozwiązania.



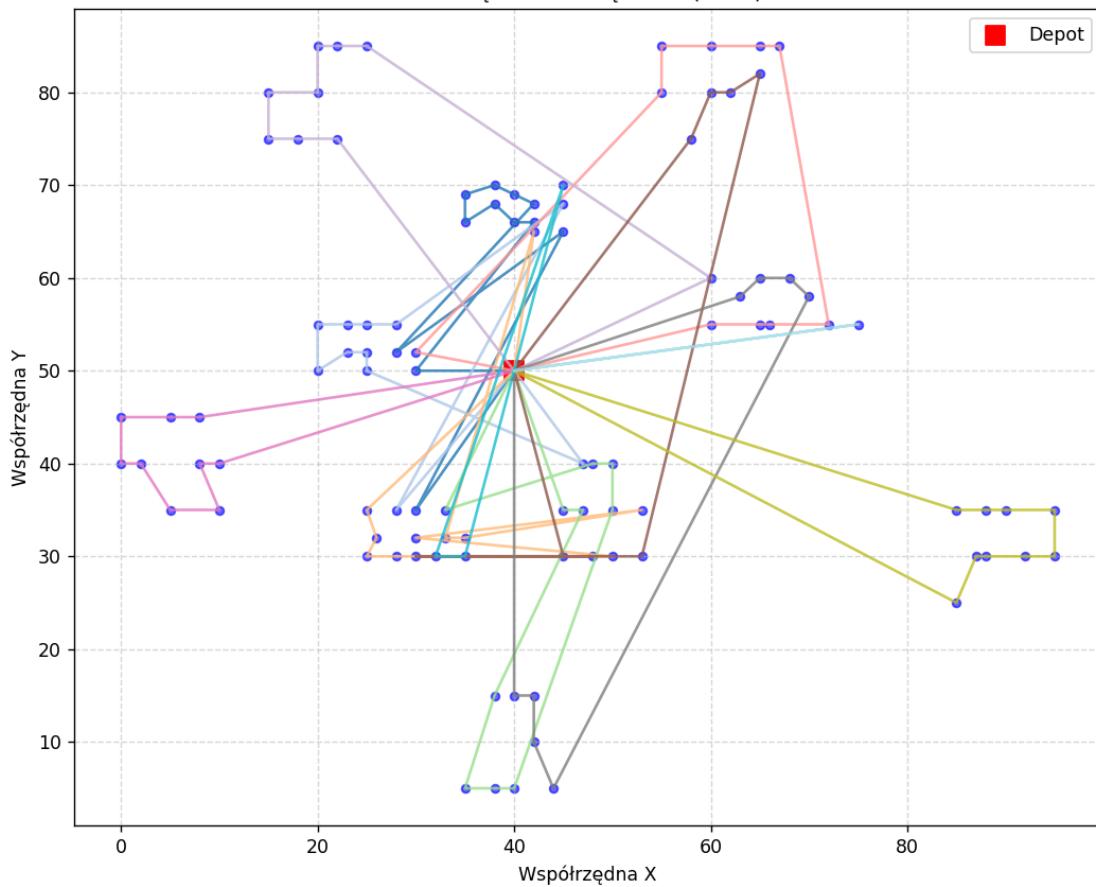
**Wykres 43.** Wizualizacja średniej kary populacji dla najlepszej trasy.

Zmiana całkowitego dystansu - c101 (Pop\_0.2)

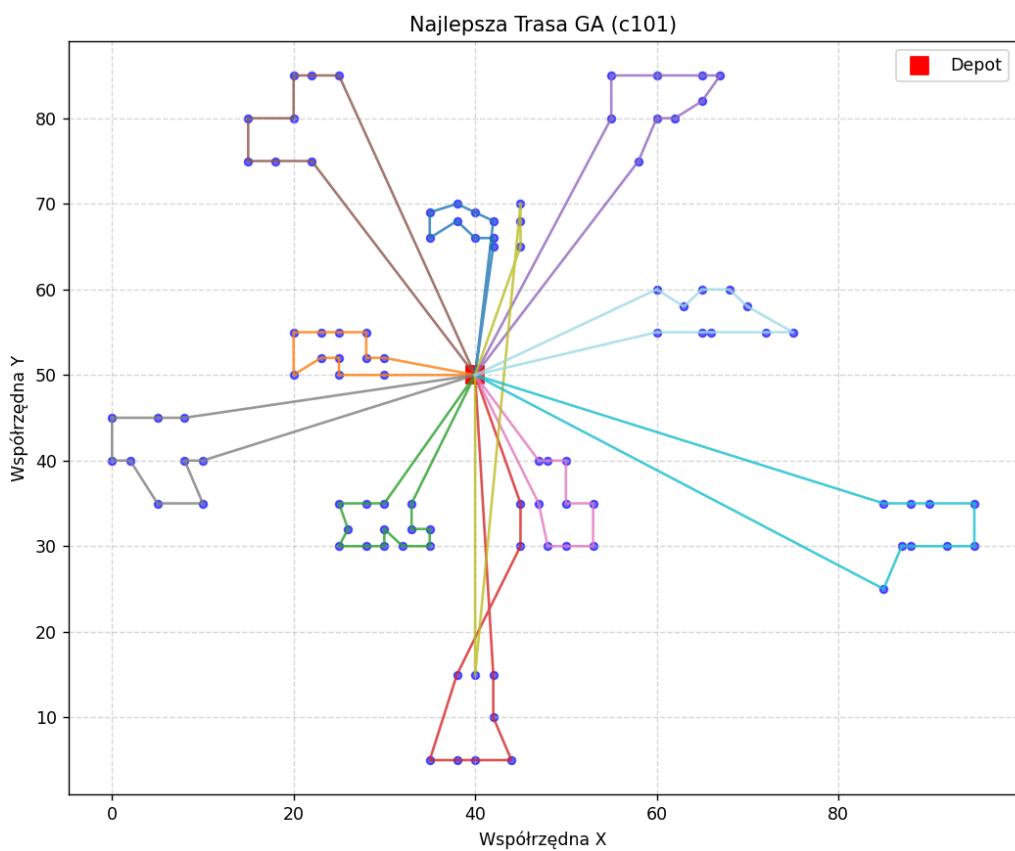


Wykres 44. Wizualizacja zmiany dystansu dla najlepszej trasy.

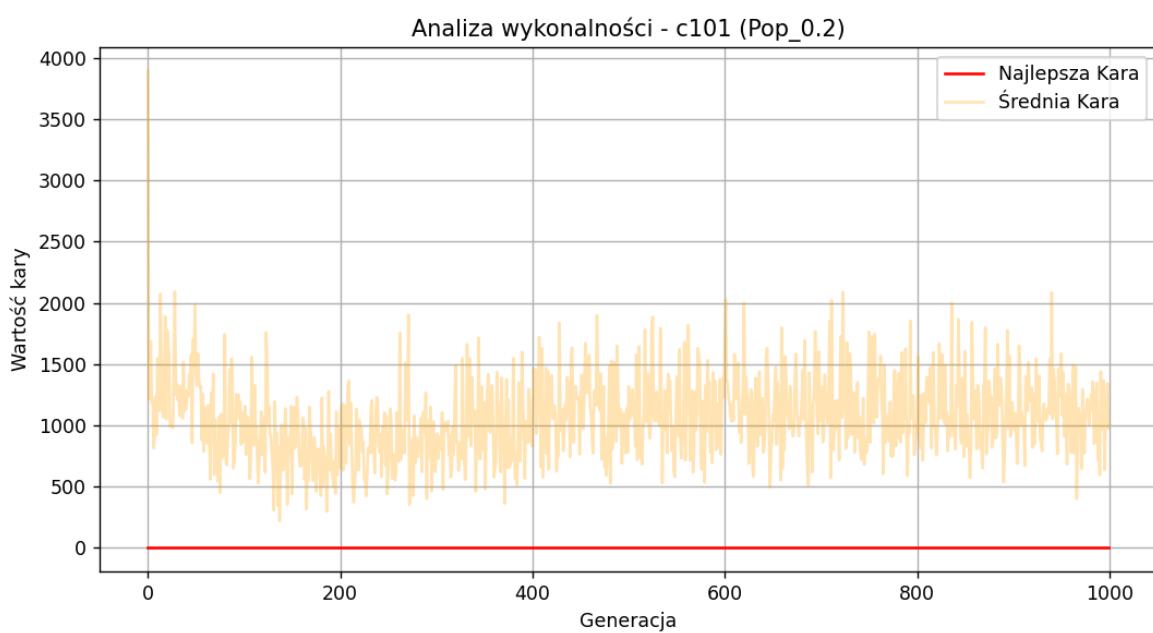
Rozwiążanie Początkowe (c101)



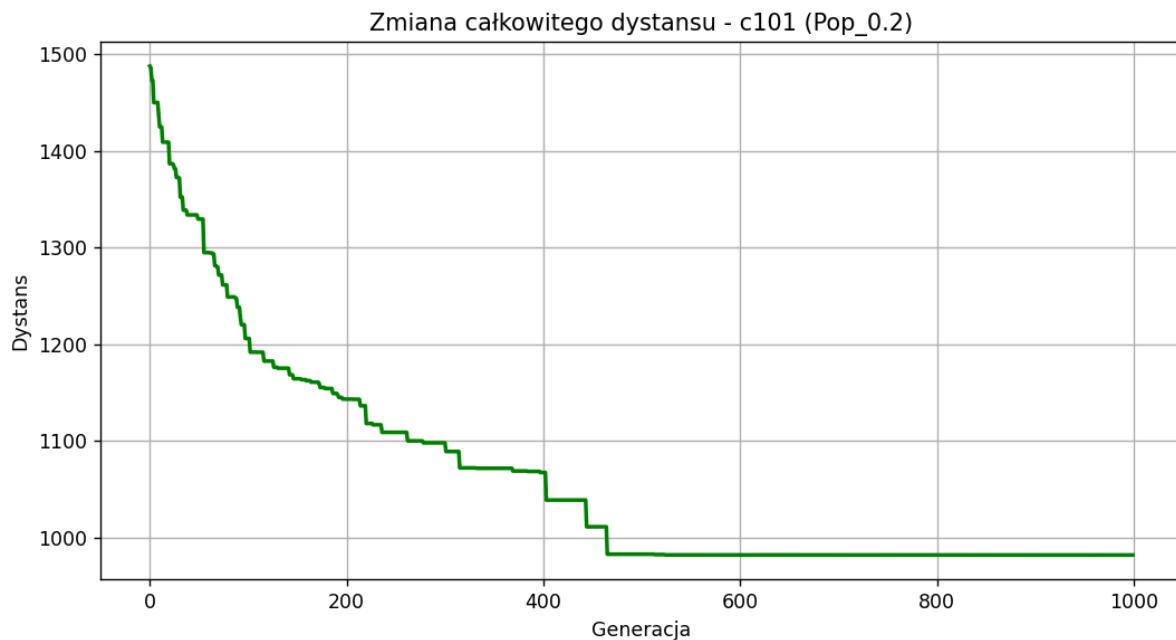
Wykres 45. Wizualizacja rozwiązania początkowego.



**Wykres 46.** Wizualizacja najlepszego rozwiązania.



**Wykres 47.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 48.** Wizualizacja zmiany dystansu dla najlepszej trasy.

## 2. Badanie wpływu współczynnika mutacji

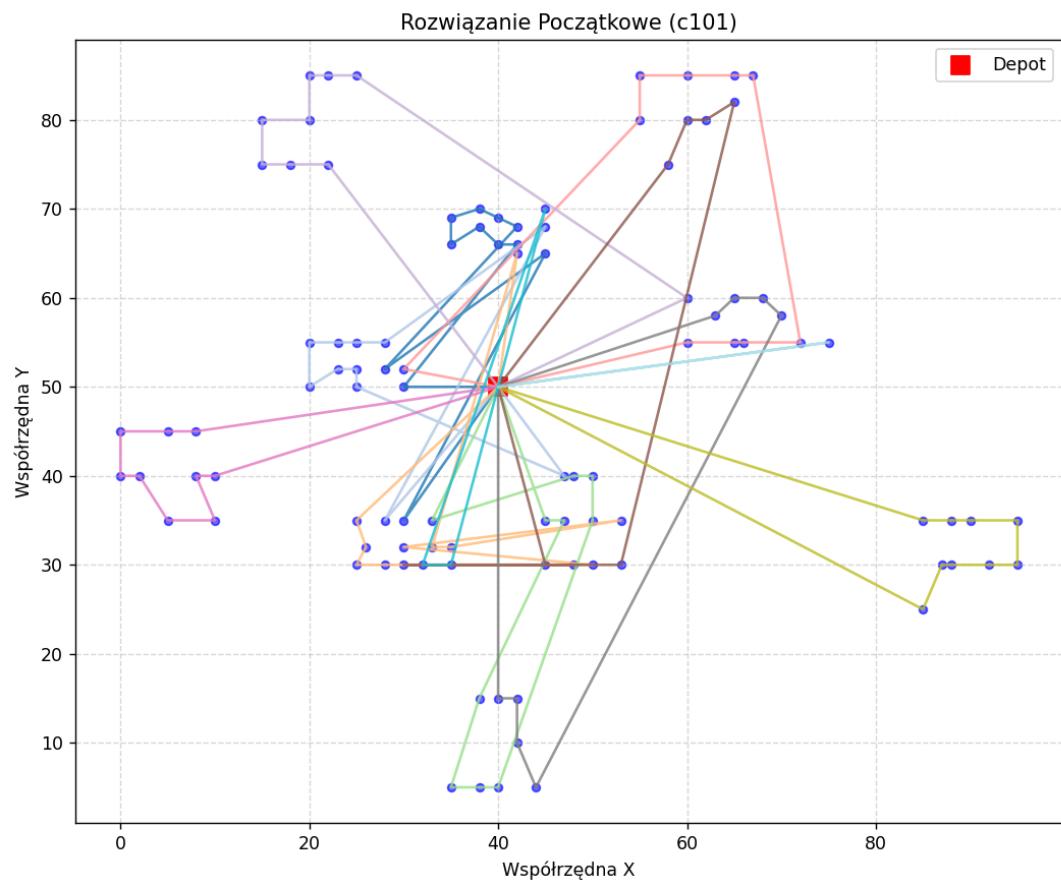
Współczynnik mutacji jest kluczowym parametrem kontrolującym balans między eksploracją (przeszukiwaniem nowych obszarów przestrzeni rozwiązań) a eksploatacją (poprawianiem aktualnie najlepszych tras). Wartości zmieniane będą wynosiły:  $P_m = [0.05, 0.1, 0.4]$ . Eksperyment zostanie wykonany po 5 razy dla każdej badanej wartości dla 3 plików.

**Tabela 9.** Wyniki 5 eksperymentów dla badanego współczynnika.

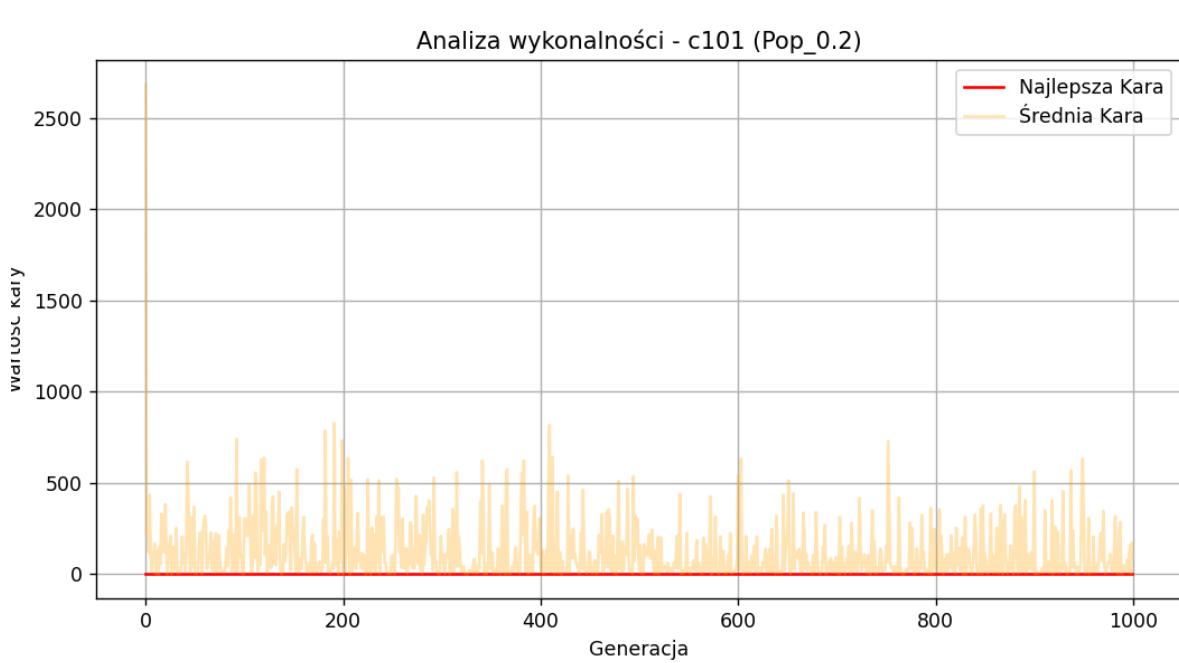
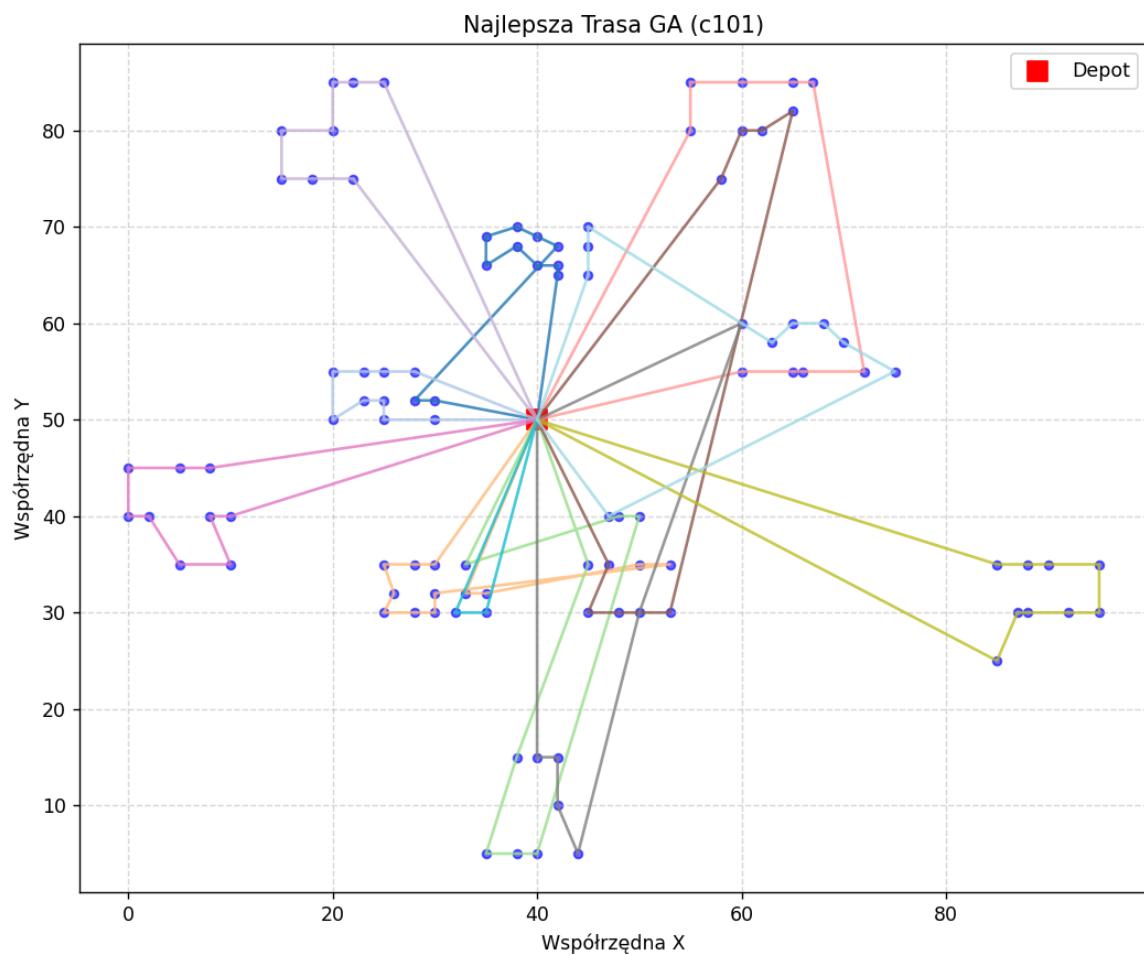
Lp.	Mutacja ( $P_m$ )	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	0.05	1	12	1170.92	0.00
2		2	12	1194.80	0.00
3		3	12	1171.71	0.00
4		4	12	1225.32	0.00
5		5	12	1186.81	0.00
6	0.10	1	12	1089.09	0.00
7		2	12	1050.53	0.00
8		3	12	1079.99	0.00
9		4	12	1088.66	0.00
10		5	12	1065.28	0.00
11	0.40	1	11	861.70	0.00
12		2	11	861.70	0.00
13		3	11	948.43	0.00
14		4	12	1033.76	0.00
15		5	12	969.92	0.00

**Tabela 10.** Wyniki podsumowujące.

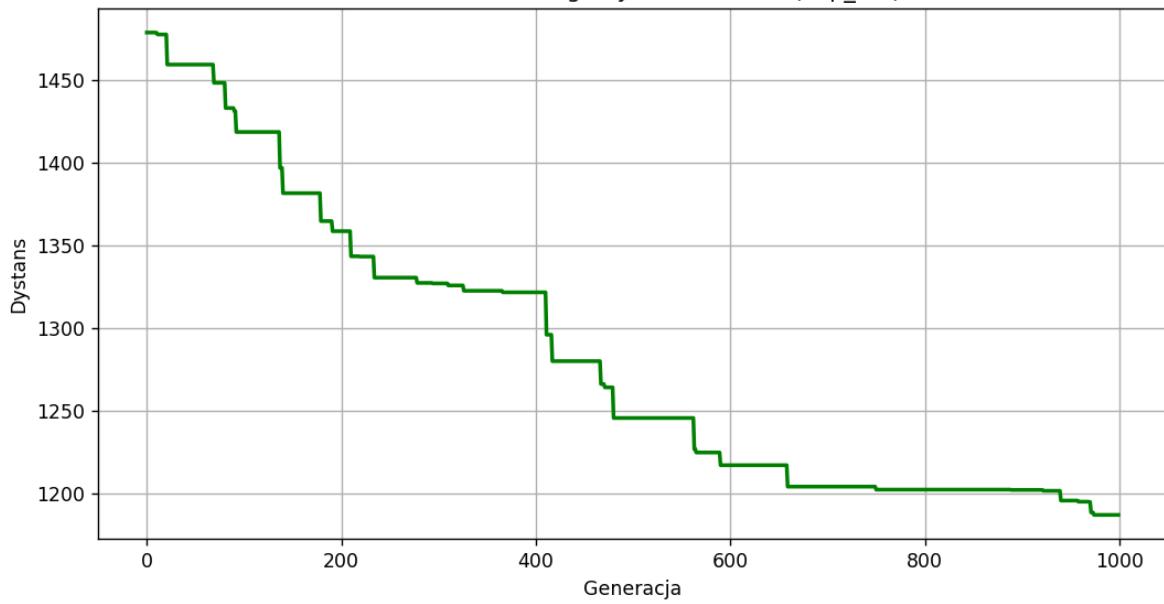
Mutacja (Pm )	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
0.05	1170.92	1189.91	19.89	12
0.10	1050.53	1074.71	14.85	12
0.40	861.70	935.10	66.18	11



**Wykres 49.** Wizualizacja rozwiązania początkowego.

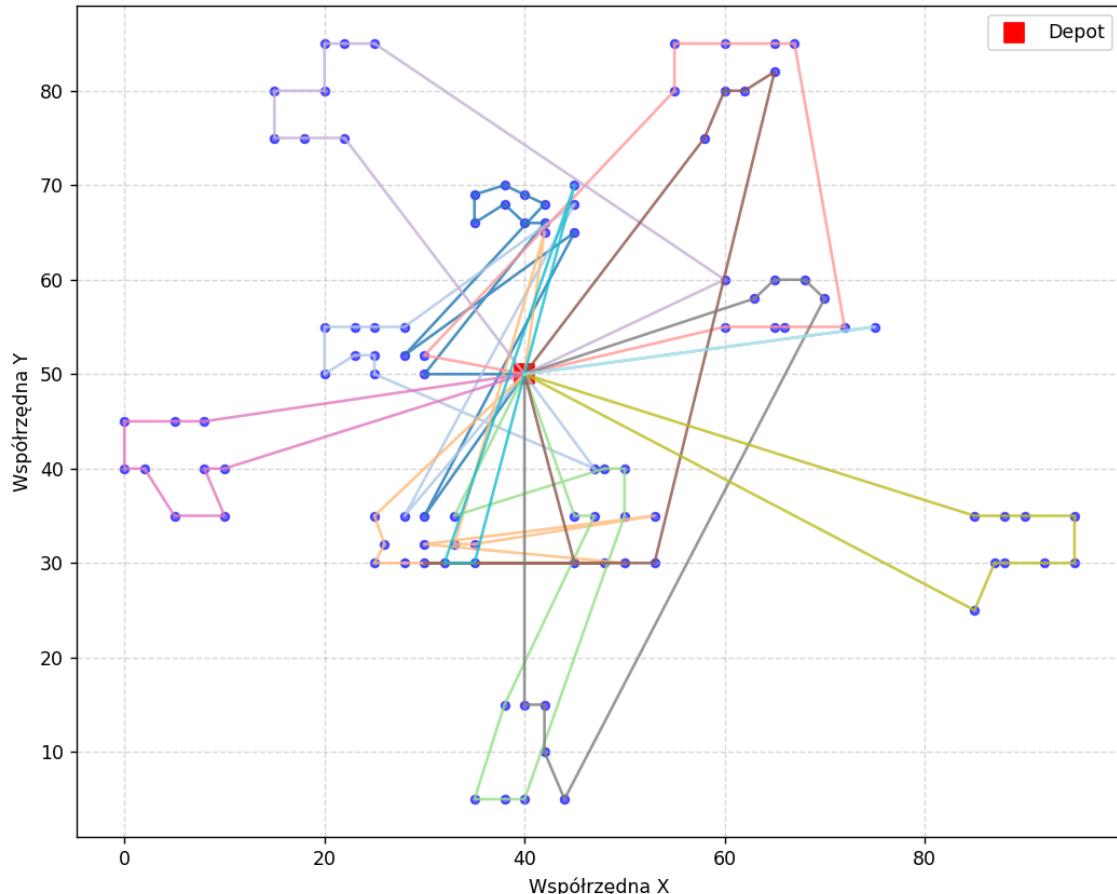


Zmiana całkowitego dystansu - c101 (Pop\_0.2)

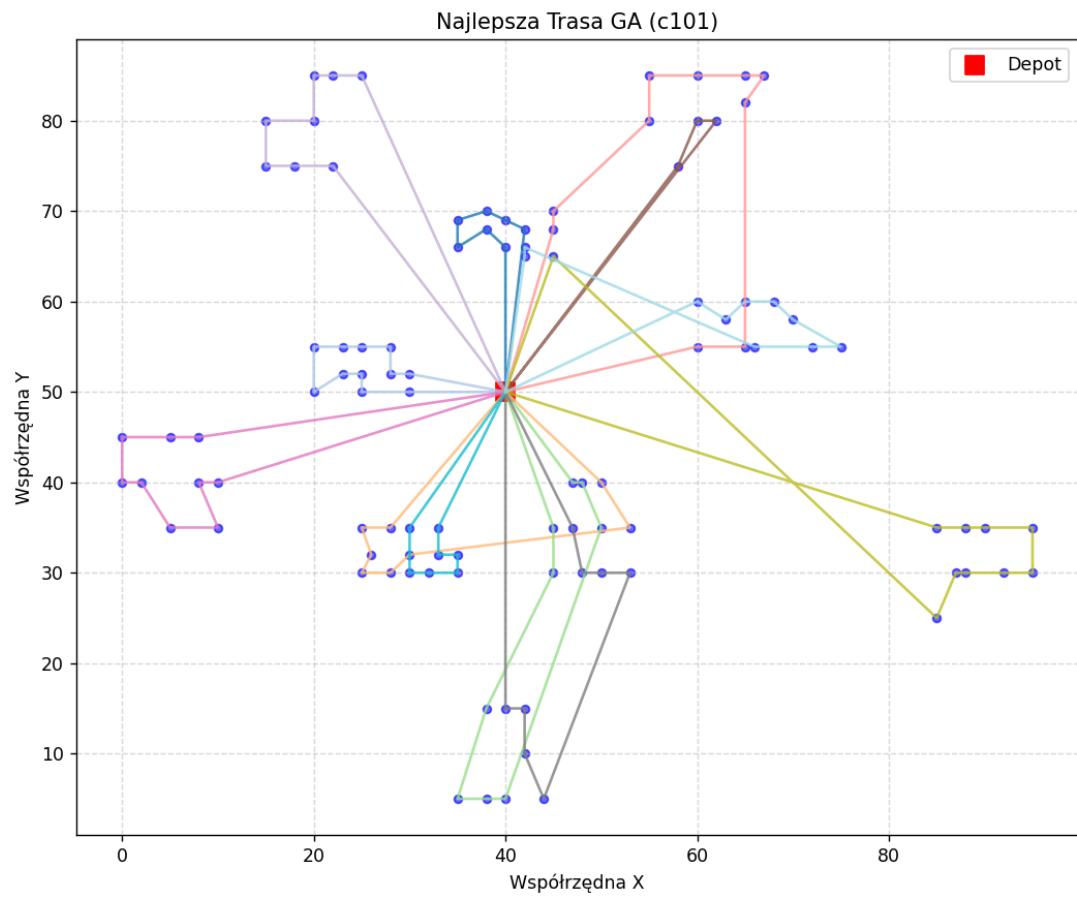


**Wykres 52.** Wizualizacja zmiany dystansu dla najlepszej trasy.

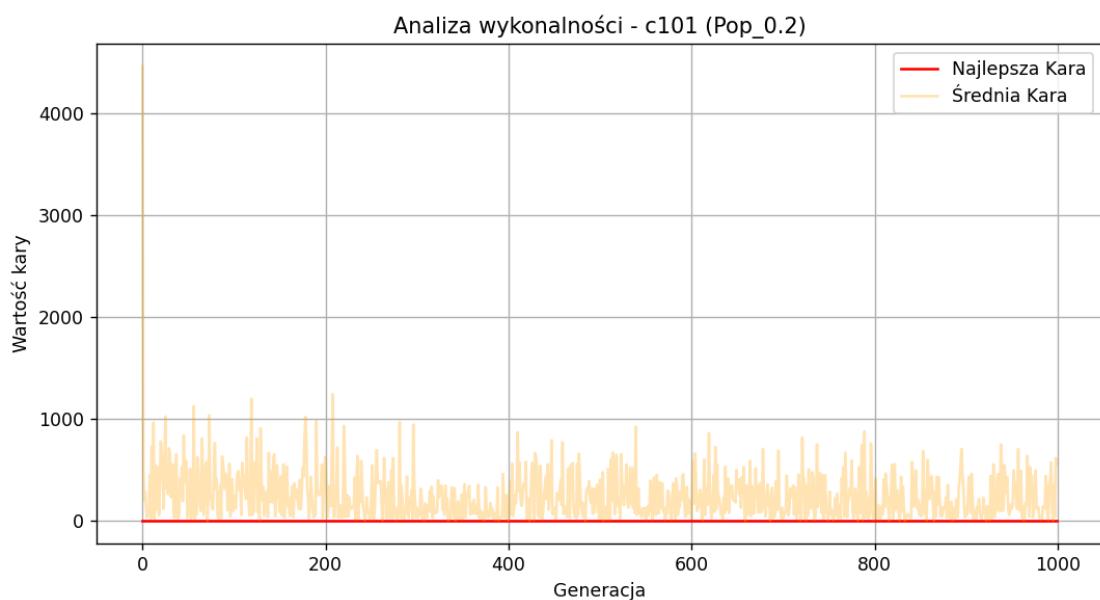
Rozwiązywanie Początkowe (c101)



**Wykres 53.** Wizualizacja rozwiązania początkowego.

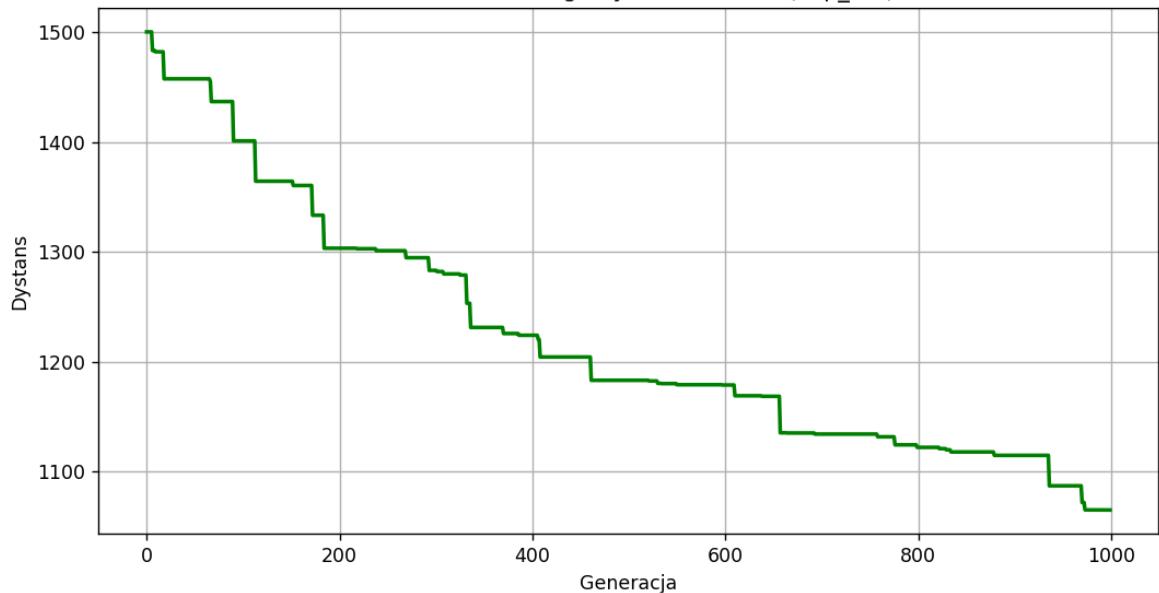


**Wykres 54.** Wizualizacja najlepszego rozwiązania.



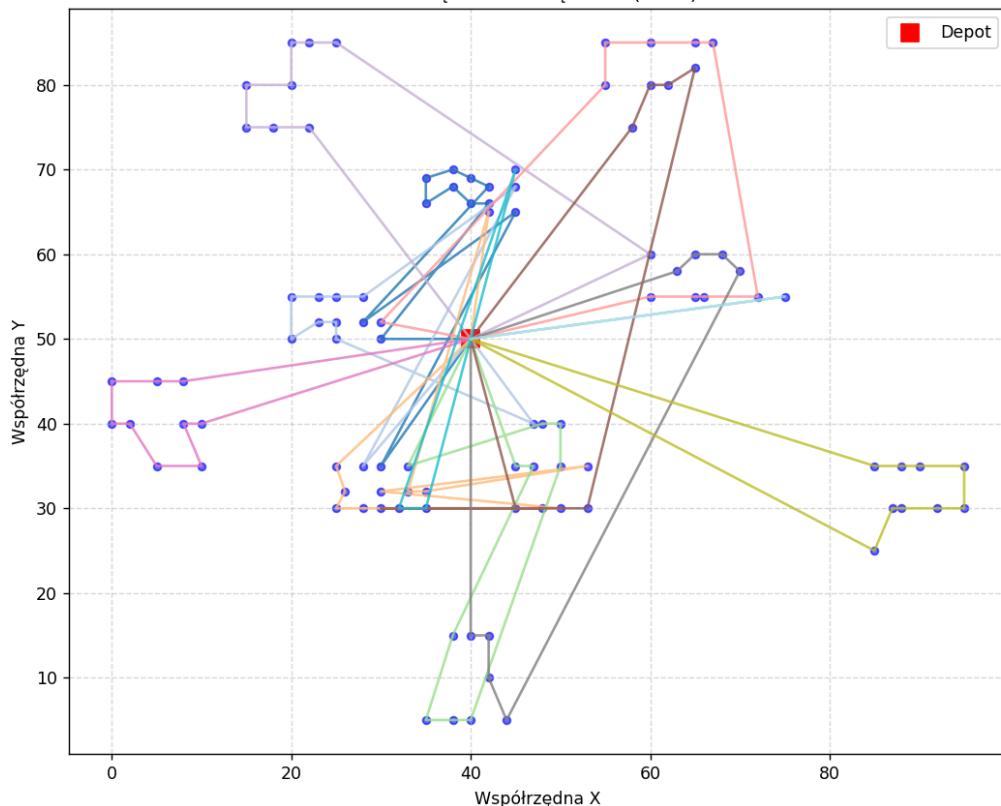
**Wykres 55.** Wizualizacja średniej kary populacji dla najlepszej trasy.

Zmiana całkowitego dystansu - c101 (Pop\_0.2)

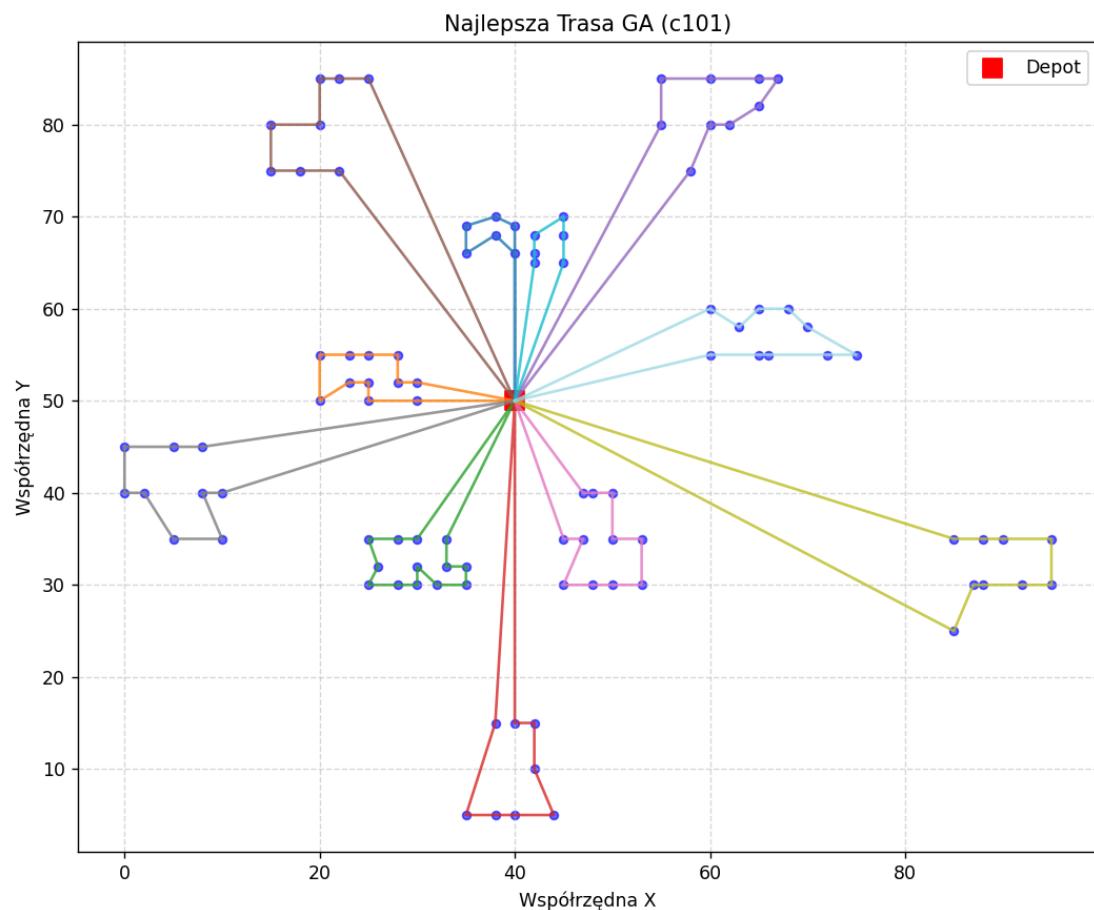


Wykres 56. Wizualizacja zmiany dystansu dla najlepszej trasy.

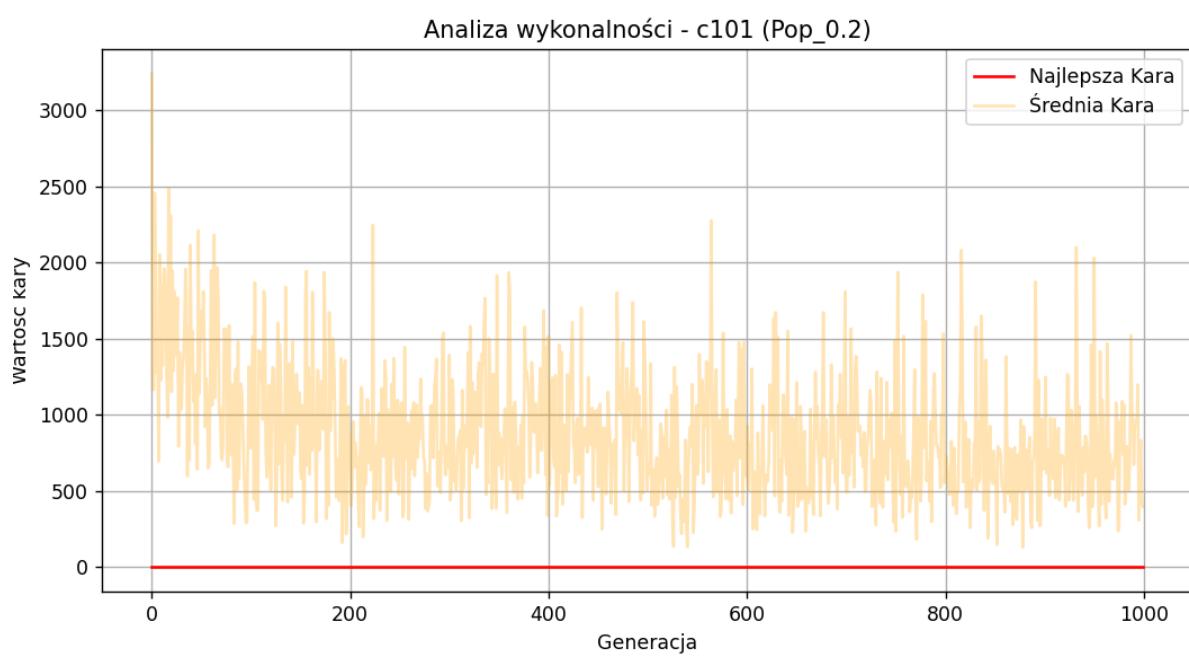
Rozwiążanie Początkowe (c101)



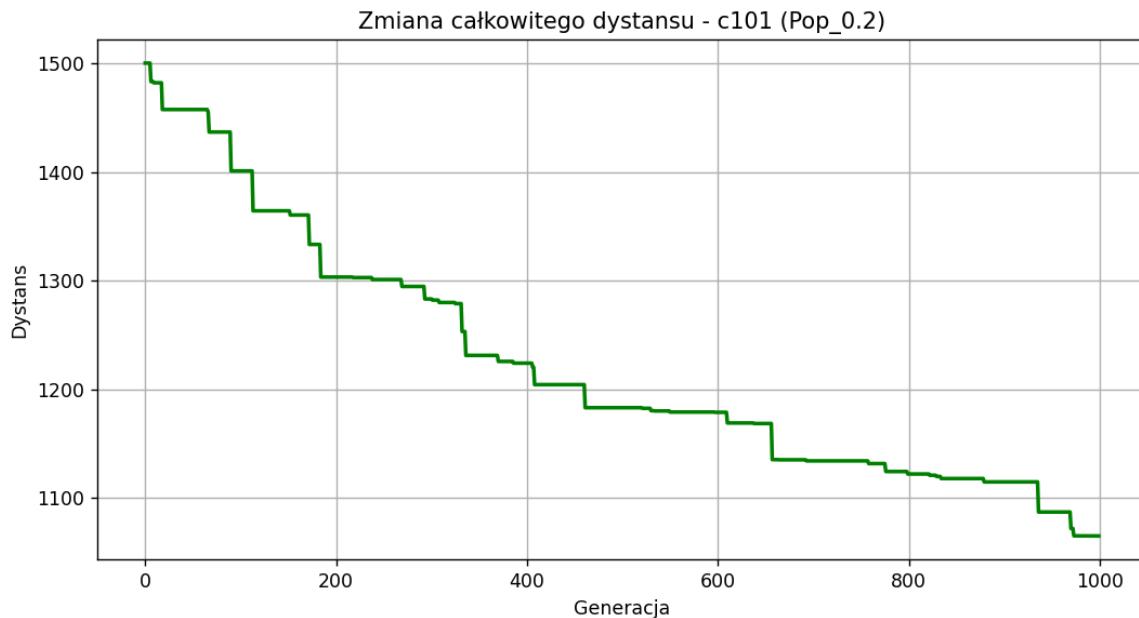
Wykres 57. Wizualizacja rozwiązania początkowego.



**Wykres 58.** Wizualizacja najlepszego rozwiązania.



**Wykres 59.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 60.** Wizualizacja zmiany dystansu dla najlepszej trasy.

### 3. Badanie wpływu współczynnika elitaryzmu.

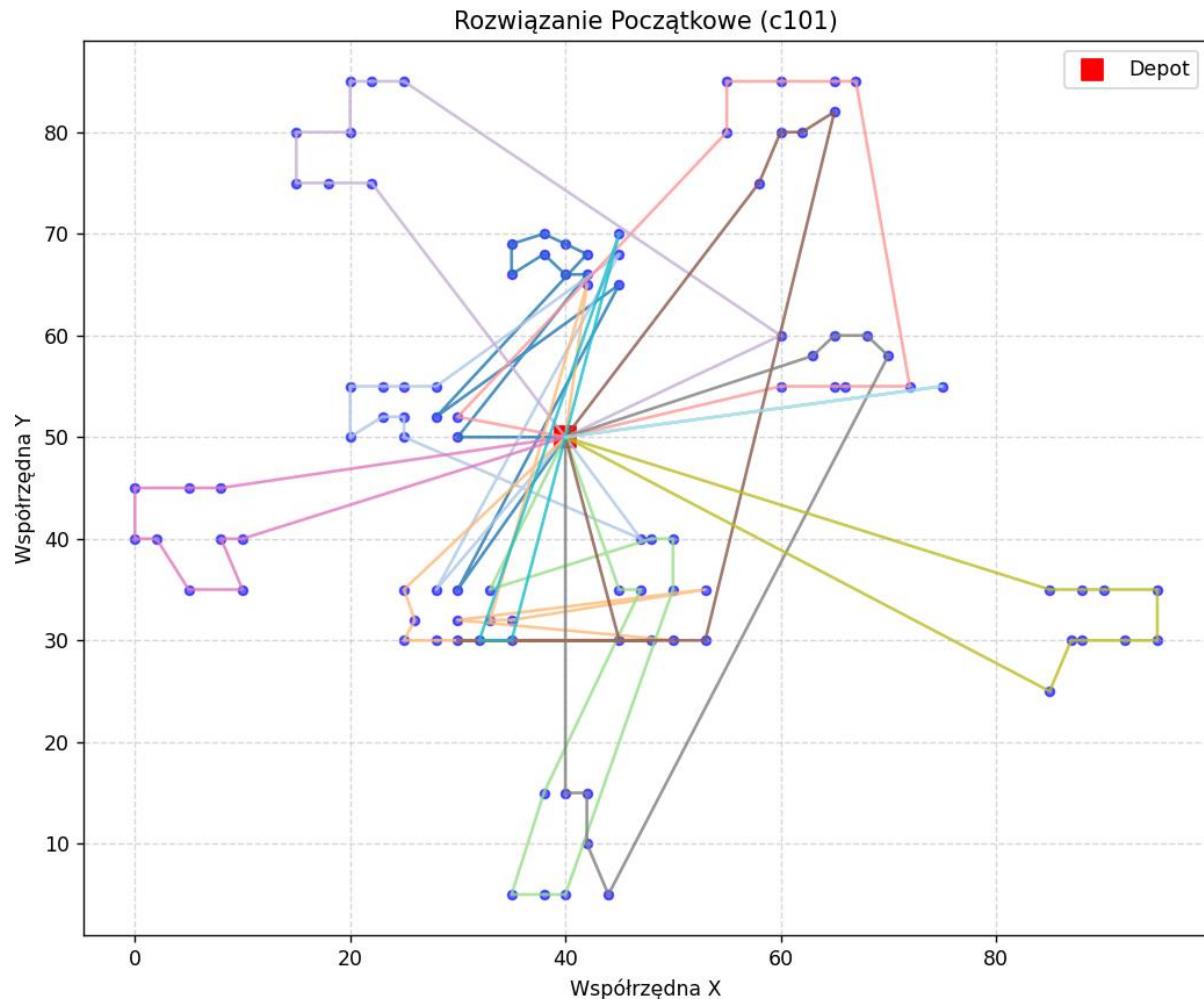
Elitaryzm chroni najlepsze osobniki przed negatywnymi skutkami losowości. W problemie VRPTW, gdzie znalezienie trasy spełniającej wszystkie okna czasowe (Penalty = 0) jest trudne, elitaryzm pełni rolę "bezpiecznika". Niski elitaryzm: Pozwala na dużą rotację w populacji, ale ryzykuje utratą rekordowego wyniku ("regresja"). Wysoki elitaryzm: Gwarantuje zachowanie najlepszych cech, ale może prowadzić do zbyt szybkiej dominacji jednego osobnika i utraty różnorodności (stagnacja w minimum lokalnym). Przeprowadzane będzie badanie wartości E = [0.05,0.2,0.4].

**Tabela 11.** Wyniki 5 eksperymentów dla badanego współczynnika.

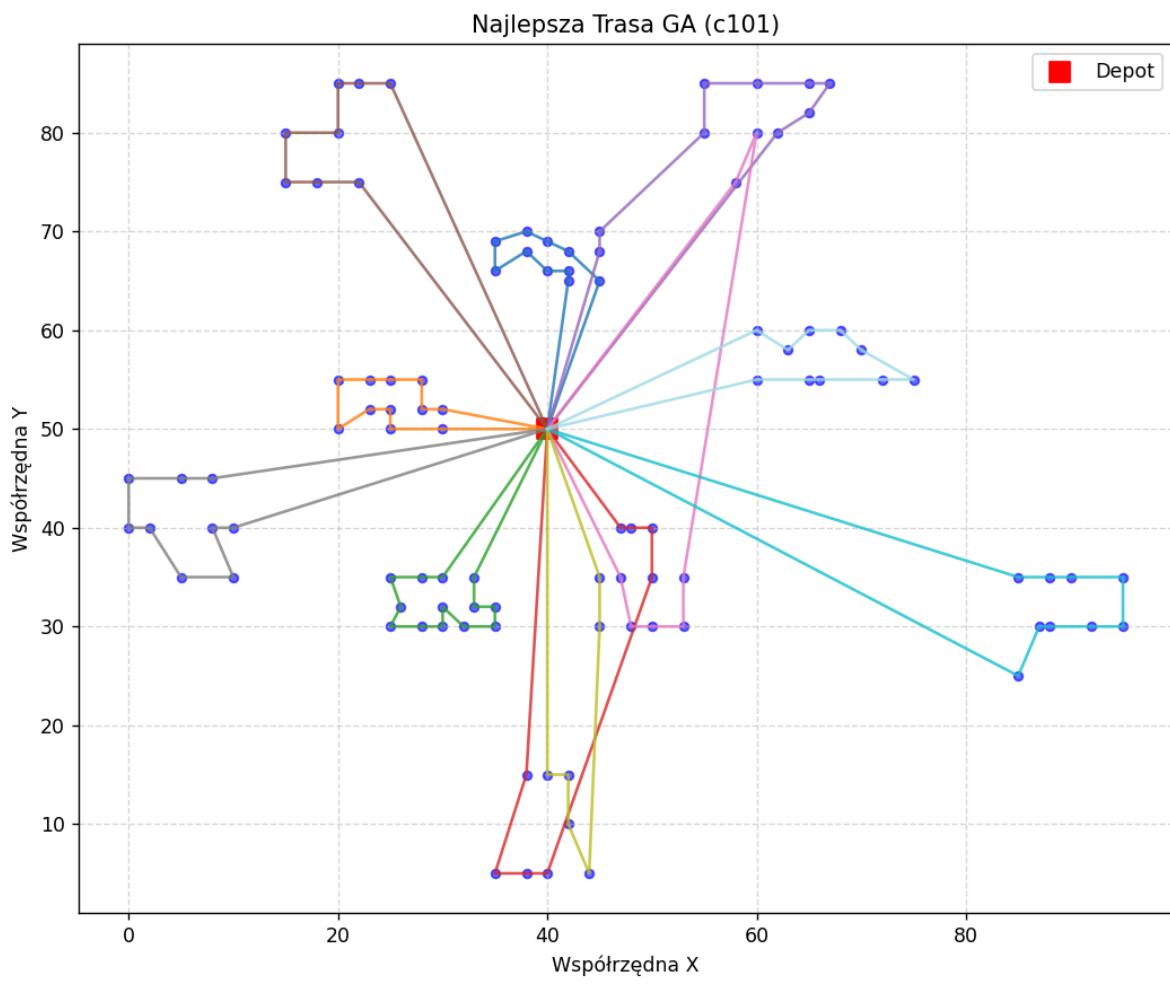
Lp.	Elitaryzm (E)	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	0.05	1	11	1000.45	0.00
2		2	11	1049.85	0.00
3		3	12	1045.62	0.00
4		4	11	999.39	0.00
5		5	11	972.51	0.00
6	0.20	1	11	981.10	0.00
7		2	11	1007.01	0.00
8		3	11	1019.72	0.00
9		4	11	986.25	0.00
10		5	11	1023.16	0.00
11	0.40	1	12	1041.46	0.00
12		2	12	997.67	0.00
13		3	11	1024.75	0.00
14		4	12	949.73	0.00
15		5	12	1032.51	0.00

**Tabela 12.** Wyniki podsumowujące.

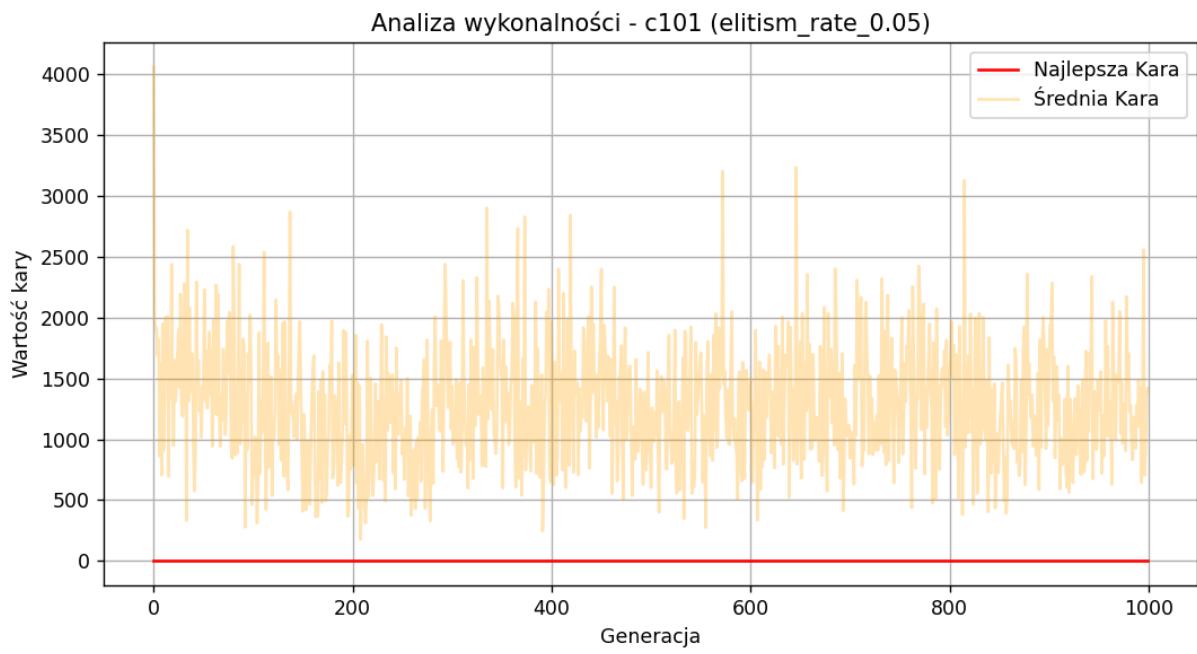
Elitaryzm (E)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
0.05	972.51	1013.56	29.67	11
0.20	981.10	1003.45	17.10	11
0.40	949.73	1009.22	33.15	11



**Wykres 61.** Wizualizacja rozwiązania początkowego.

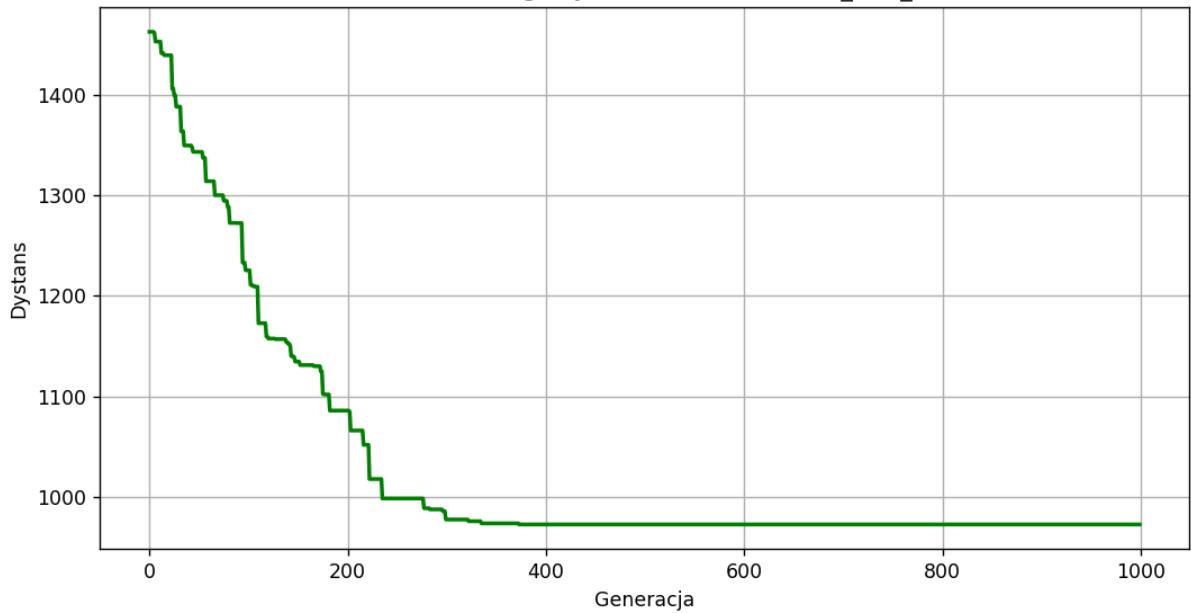


**Wykres 62.** Wizualizacja najlepszego rozwiązania.



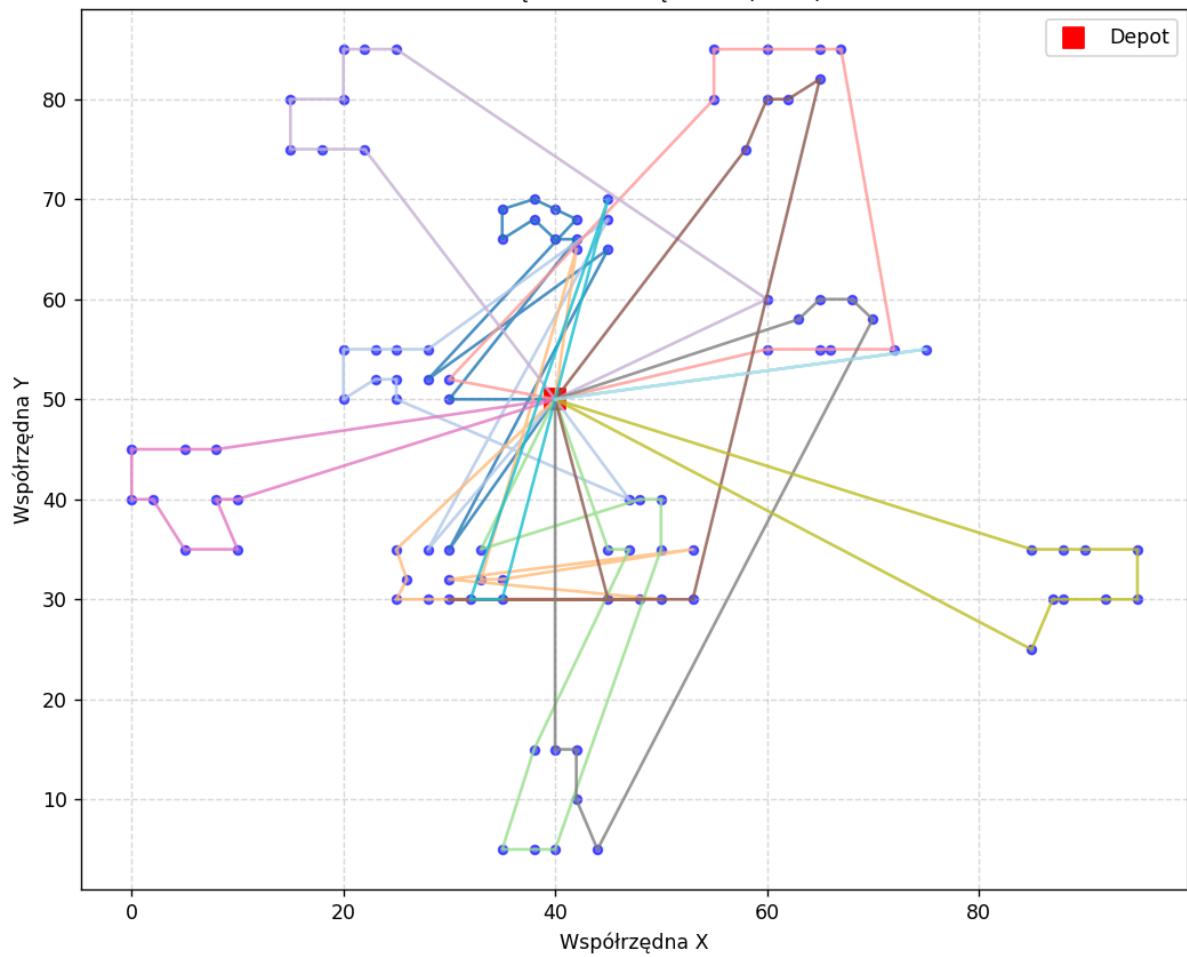
**Wykres 63.** Wizualizacja średniej kary populacji dla najlepszej trasy.

Zmiana całkowitego dystansu - c101 (elitism\_rate\_0.05)

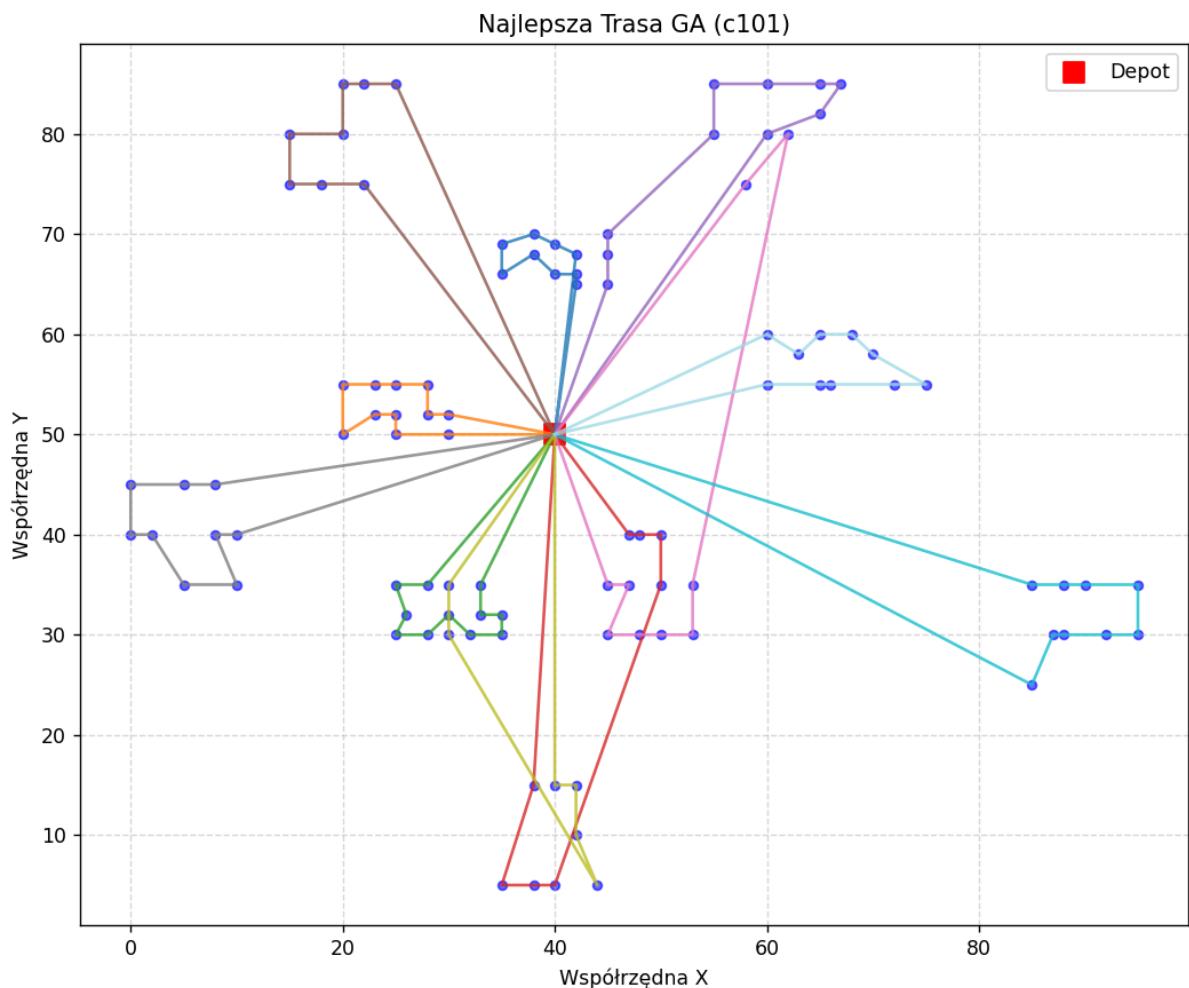


Wykres 64. Wizualizacja zmiany dystansu dla najlepszej trasy.

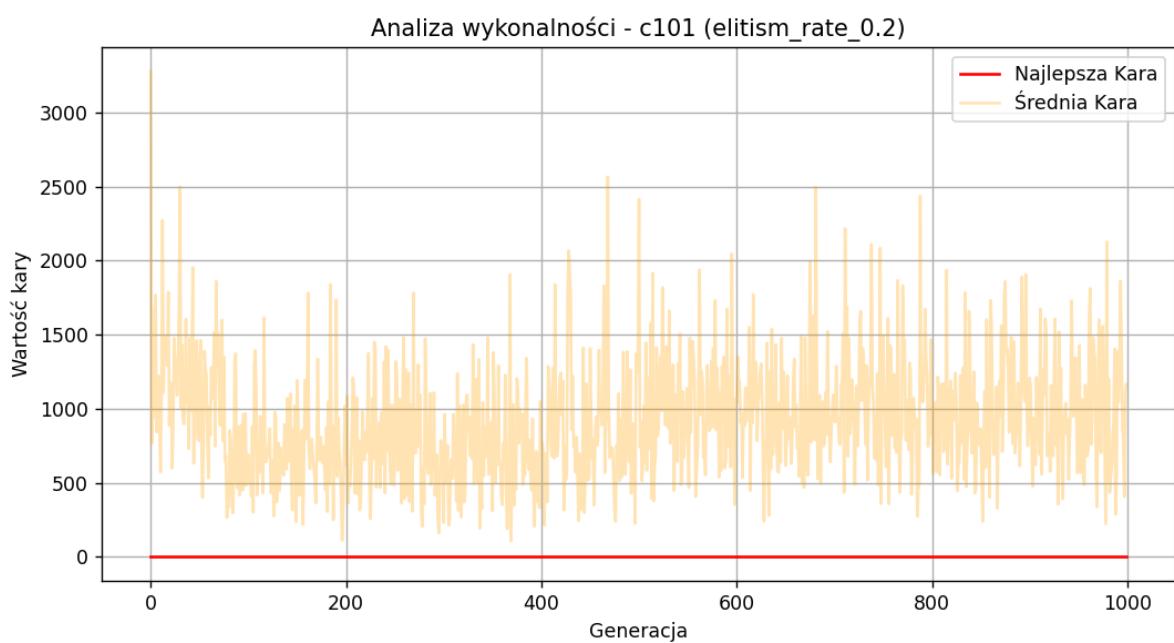
Rozwiążanie Początkowe (c101)



Wykres 65. Wizualizacja rozwiązania początkowego.

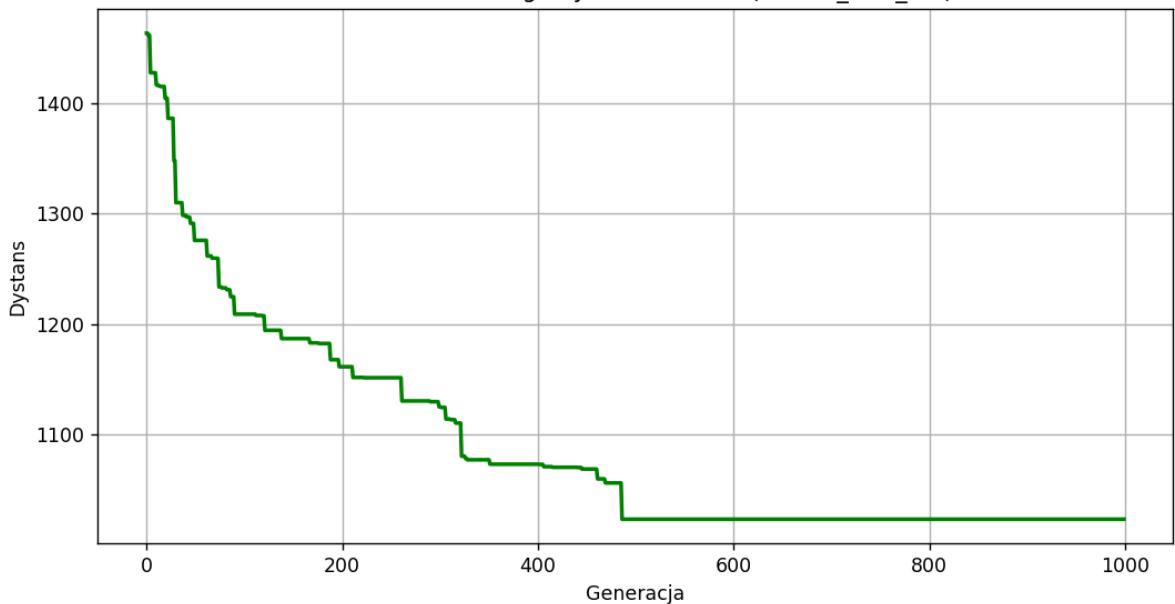


**Wykres 66.** Wizualizacja najlepszego rozwiązania.



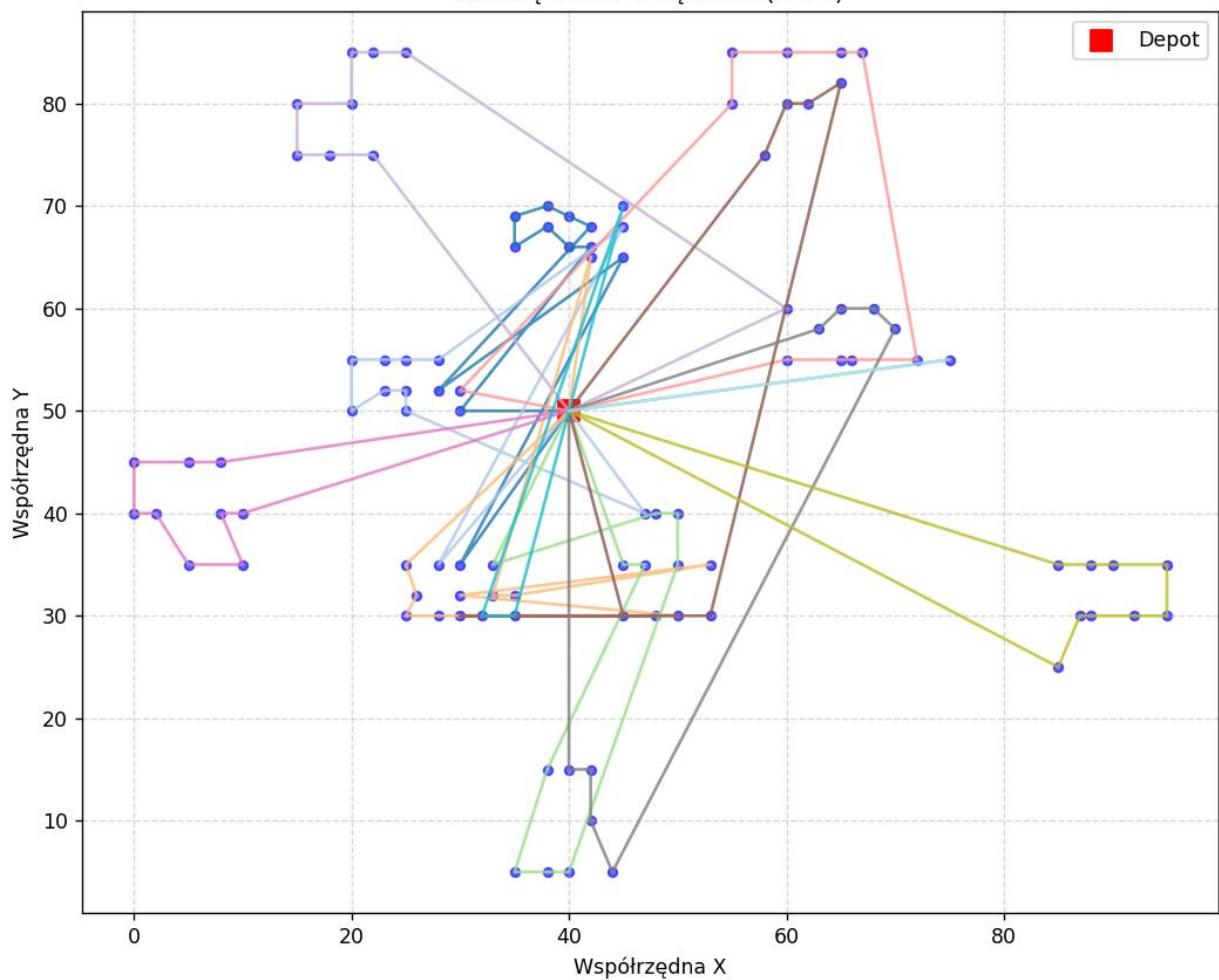
**Wykres 67.** Wizualizacja średniej kary populacji dla najlepszej trasy.

Zmiana całkowitego dystansu - c101 (elitism\_rate\_0.2)

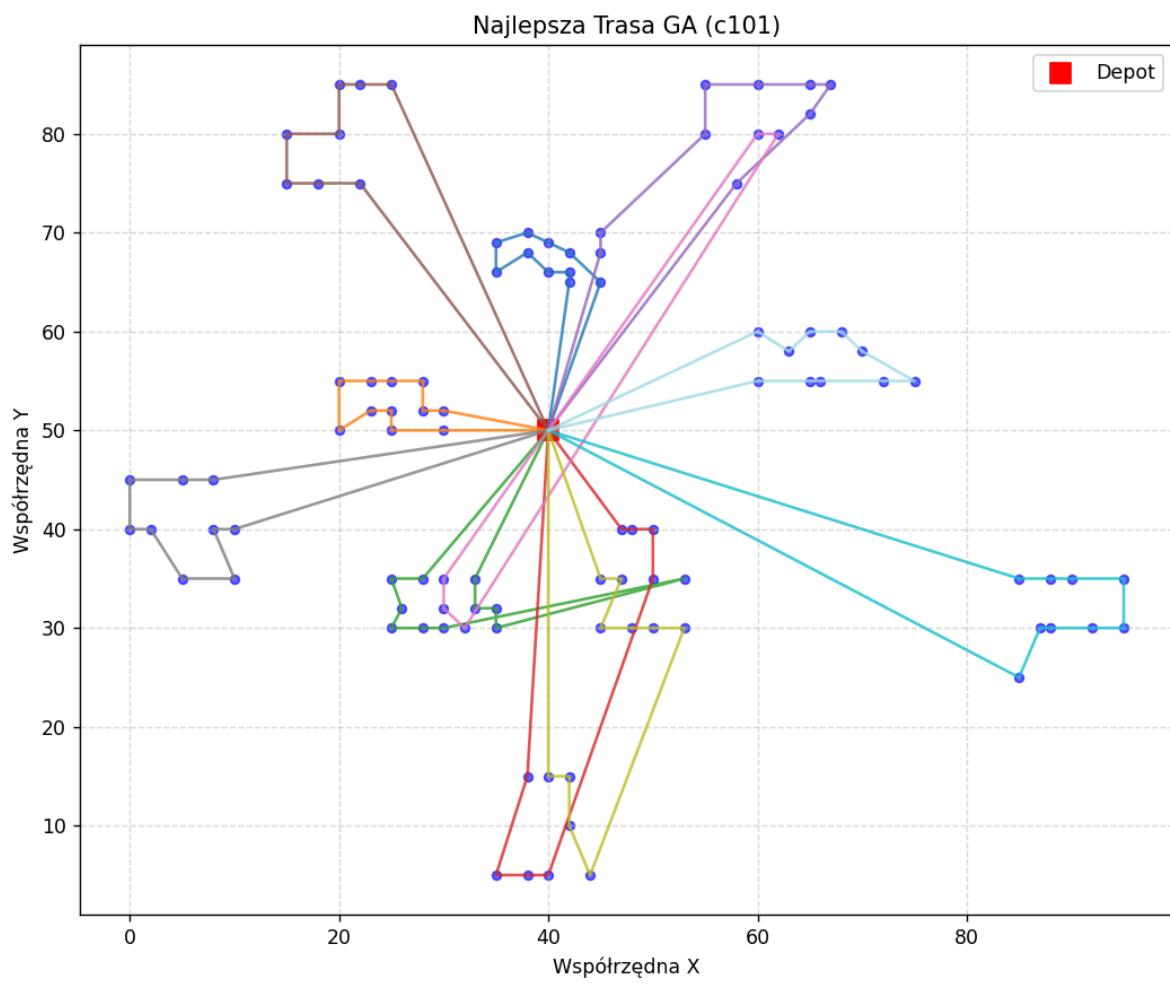


**Wykres 68.** Wizualizacja zmiany dystansu dla najlepszej trasy.

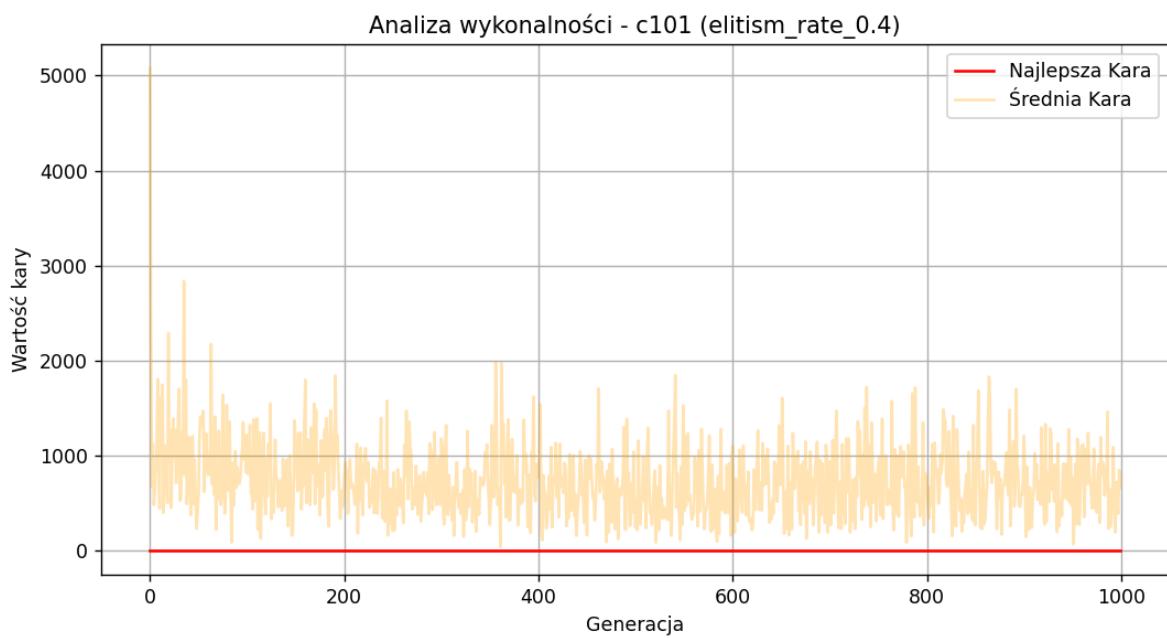
Rozwiązywanie Początkowe (c101)



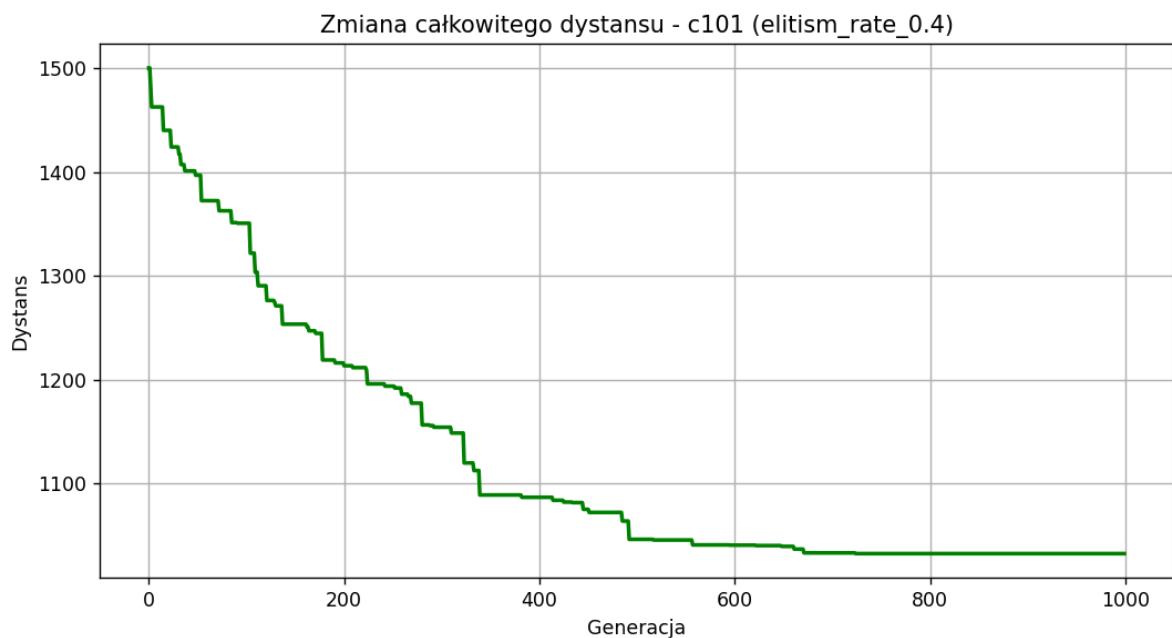
**Wykres 69.** Wizualizacja rozwiązania początkowego.



**Wykres 70.** Wizualizacja najlepszego rozwiązania.



**Wykres 71.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 72.** Wizualizacja zmiany dystansu dla najlepszej trasy.

### 6.3 Badanie dla pliku rc101.txt

#### 1. Badanie wpływu wielkości populacji

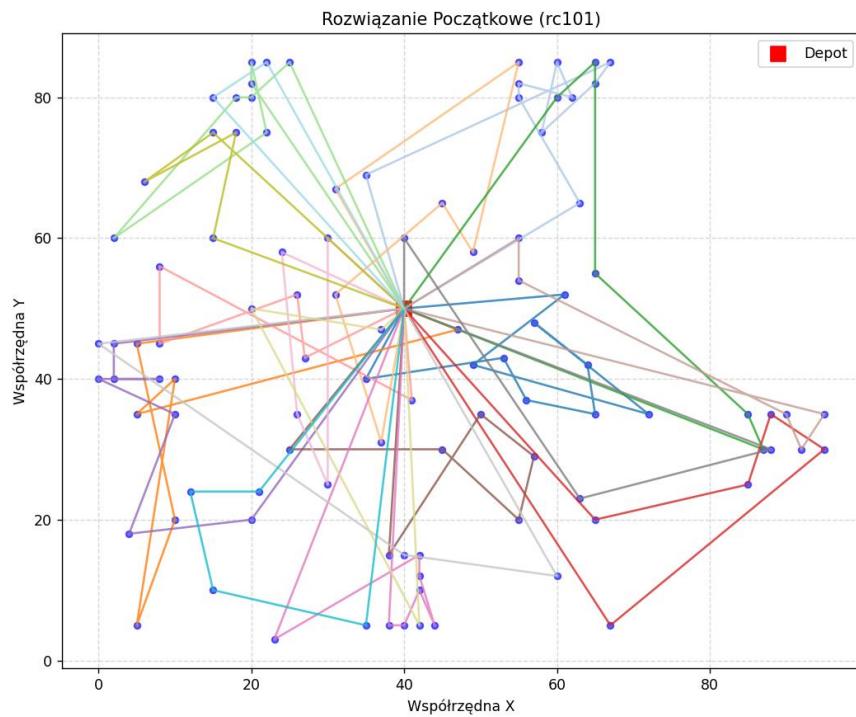
Badania wpływu wielkości populacji (N). Ten parametr decyduje o balansie między "szerokością" poszukiwań (eksploracją) a kosztem obliczeniowym. Wartości zmieniane będą wynosiły: N = [50,100,200].

**Tabela 13.** Wyniki 5 eksperymentów dla badanego współczynnika.

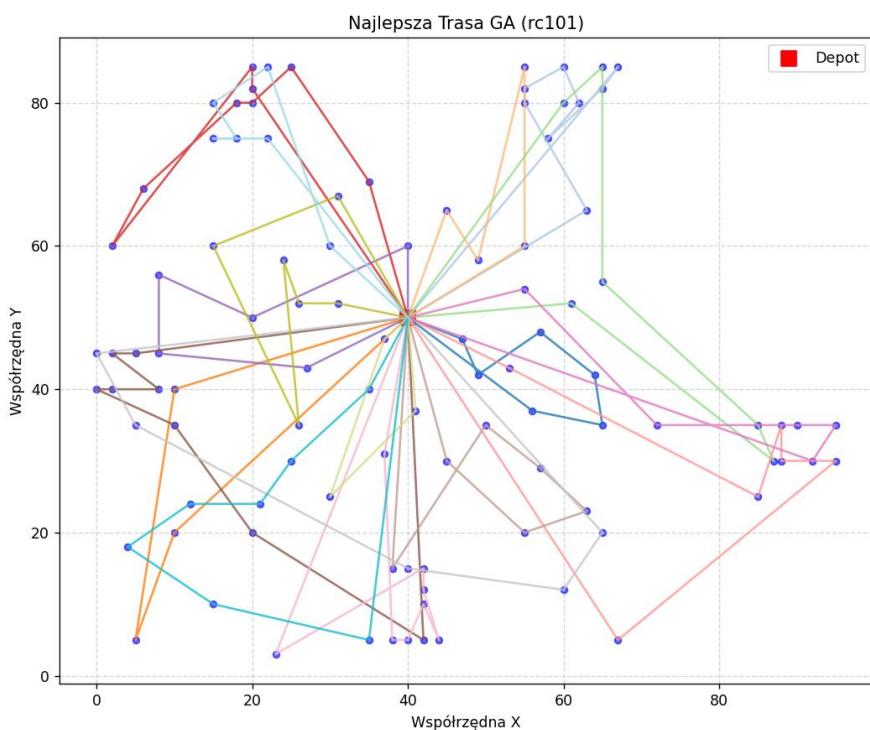
Lp.	Populacja (N)	Próba	Pojazdy (Veh)	Dystans (Dist)	Kara (Penalty)
1	50	1	17	2064.48	0.00
2		2	18	2004.34	0.00
3		3	18	1944.51	0.00
4		4	18	1957.84	0.00
5		5	18	2057.55	0.00
6	100	1	17	1932.77	0.00
7		2	17	2009.67	0.00
8		3	17	1874.95	0.00
9		4	17	1955.92	0.00
10		5	18	1984.66	0.00
11	200	1	17	1845.14	0.00
12		2	18	1951.32	0.00
13		3	18	2021.72	0.00
14		4	18	1941.61	0.00
15		5	18	1794.55	0.00

**Tabela 14.** Wyniki podsumowujące.

Populacja (N)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
50	1944.51	2005.74	49.36	17
100	1874.95	1951.59	46.29	17
200	1794.55	1910.87	80.89	17

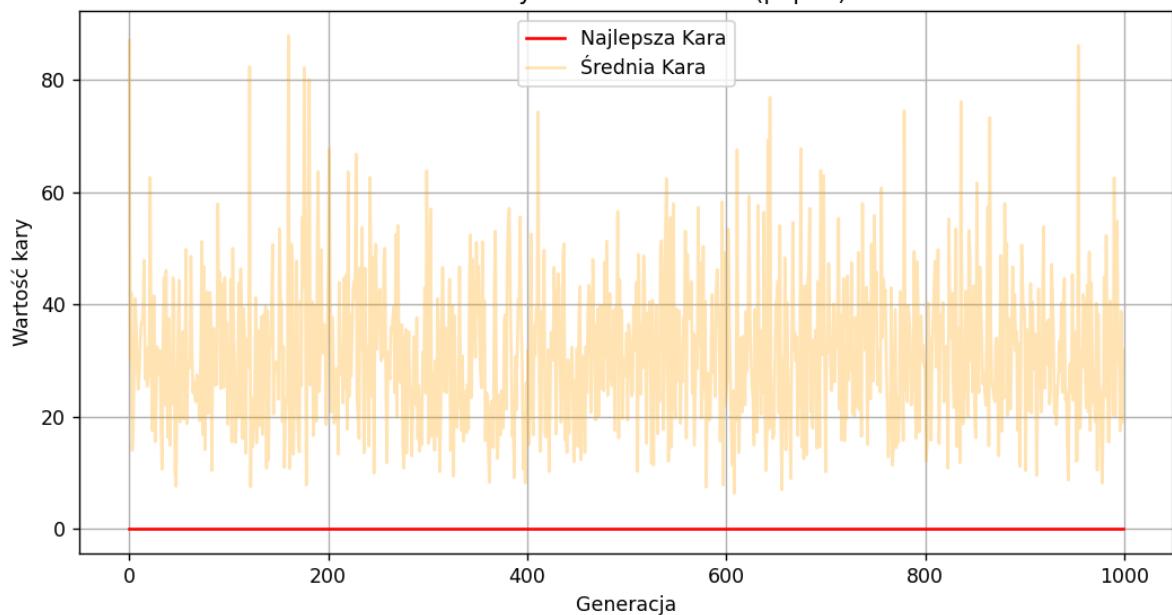


**Wykres 73.** Wizualizacja rozwiązania początkowego.



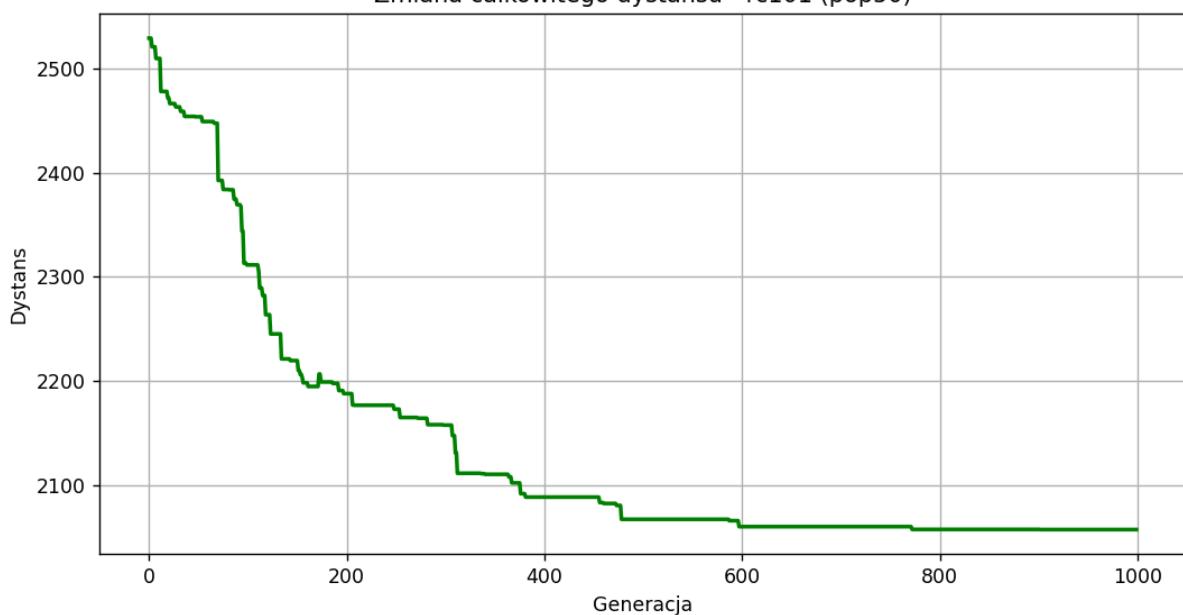
**Wykres 74.** Wizualizacja najlepszego rozwiązania.

Analiza wykonalności - rc101 (pop50)

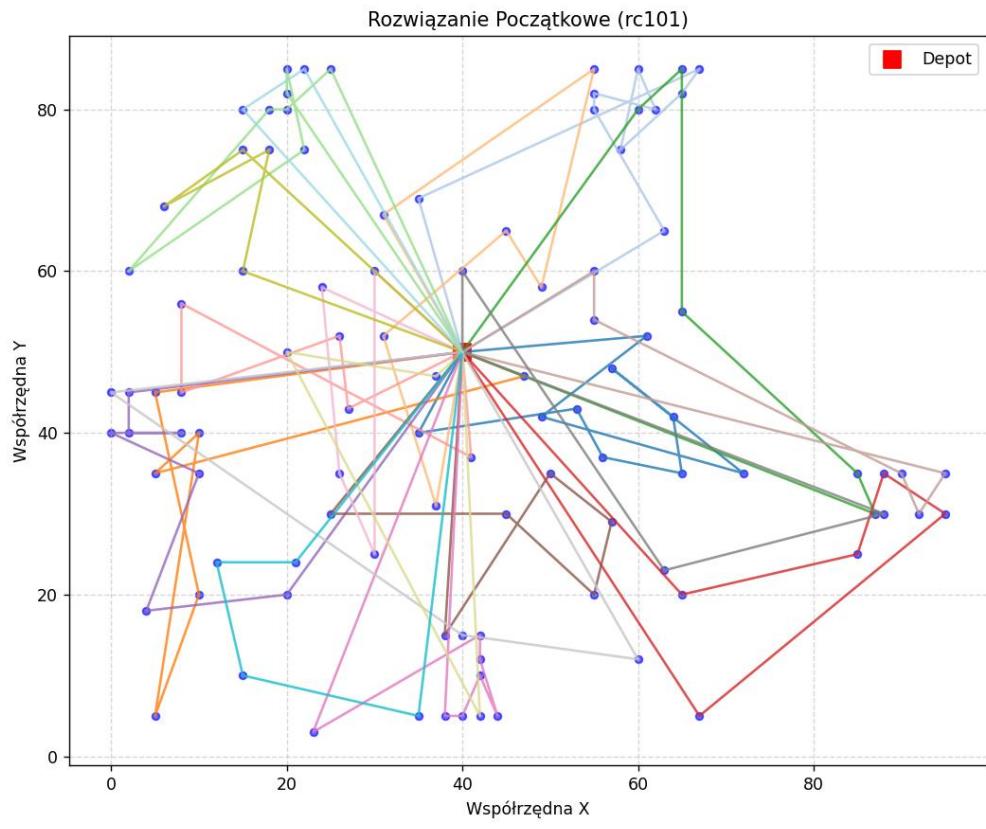


Wykres 75. Wizualizacja średniej kary populacji dla najlepszej trasy.

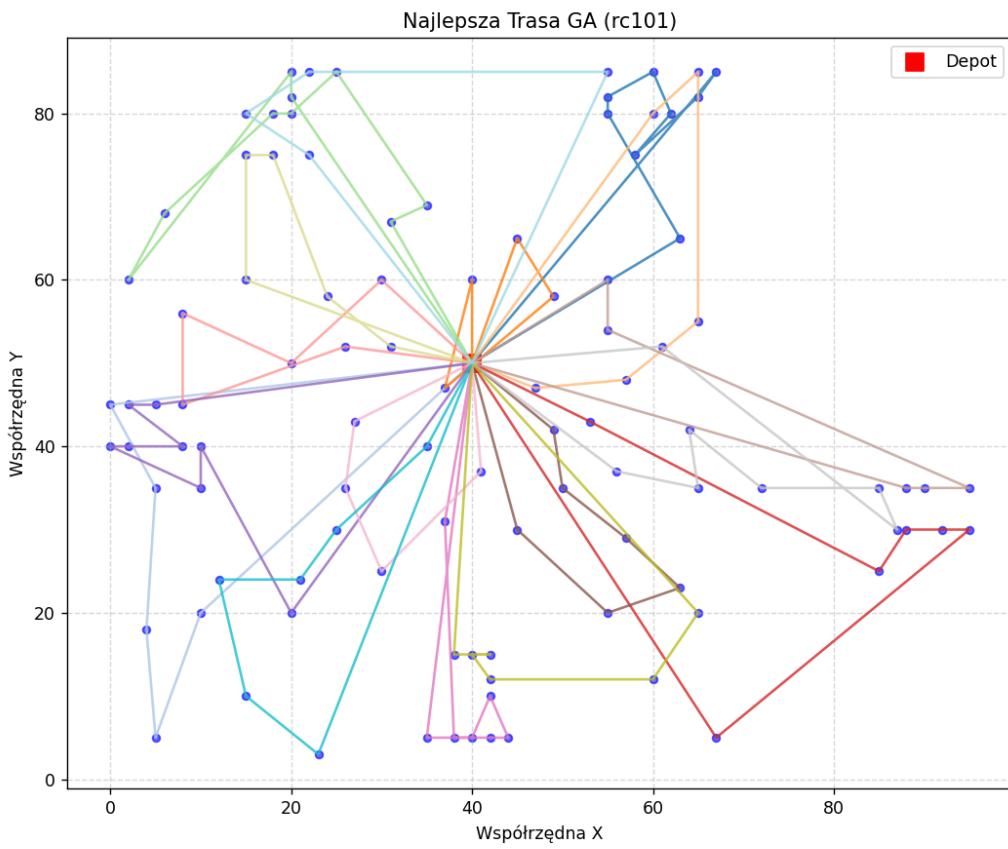
Zmiana całkowitego dystansu - rc101 (pop50)



Wykres 76. Wizualizacja zmiany dystansu dla najlepszej trasy.

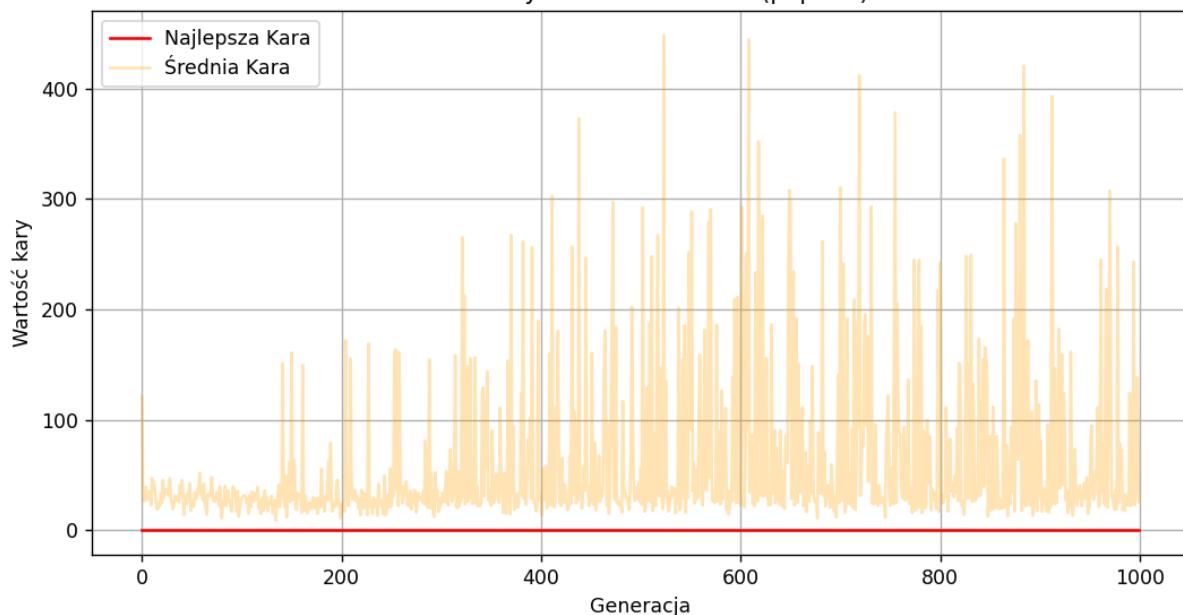


**Wykres 77.** Wizualizacja rozwiązań początkowego.



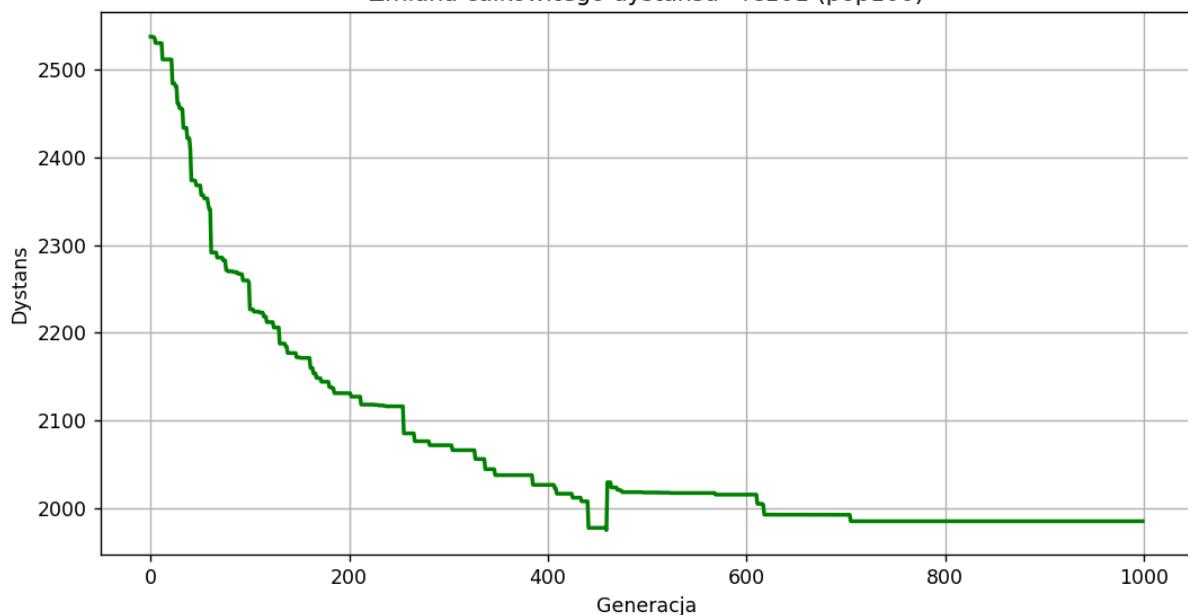
**Wykres 78.** Wizualizacja najlepszego rozwiązania.

Analiza wykonalności - rc101 (pop100)

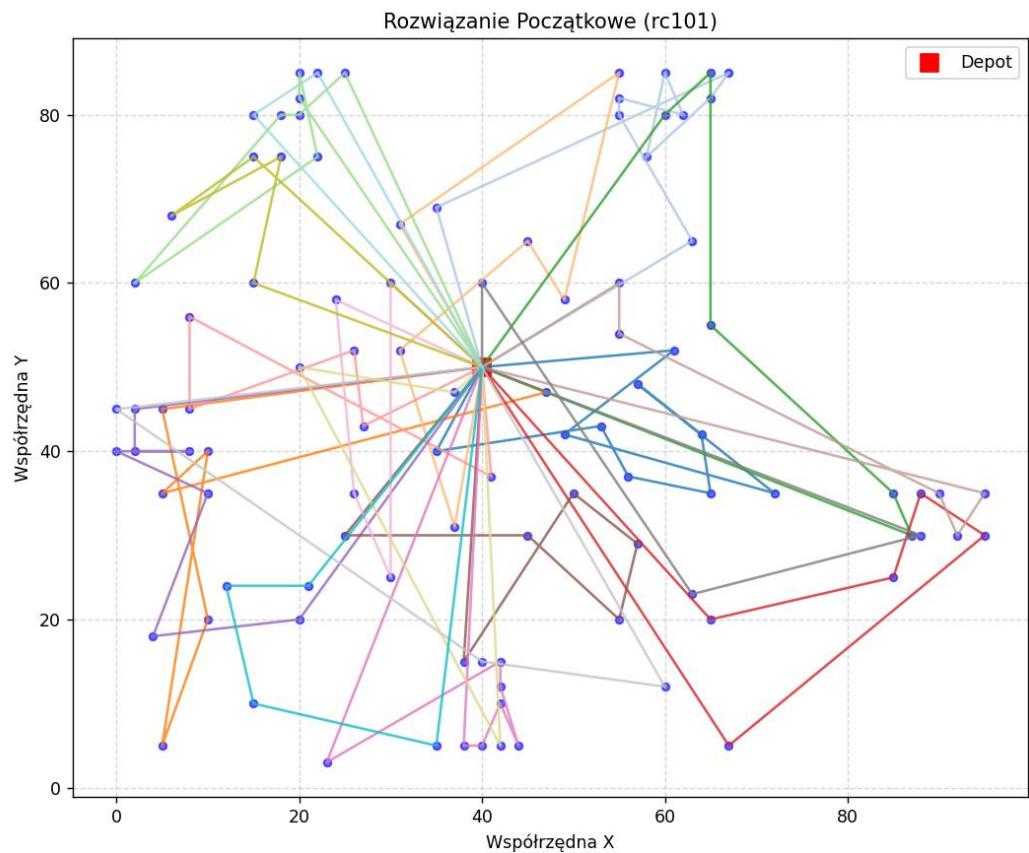


Wykres 79. Wizualizacja średniej kary populacji dla najlepszej trasy.

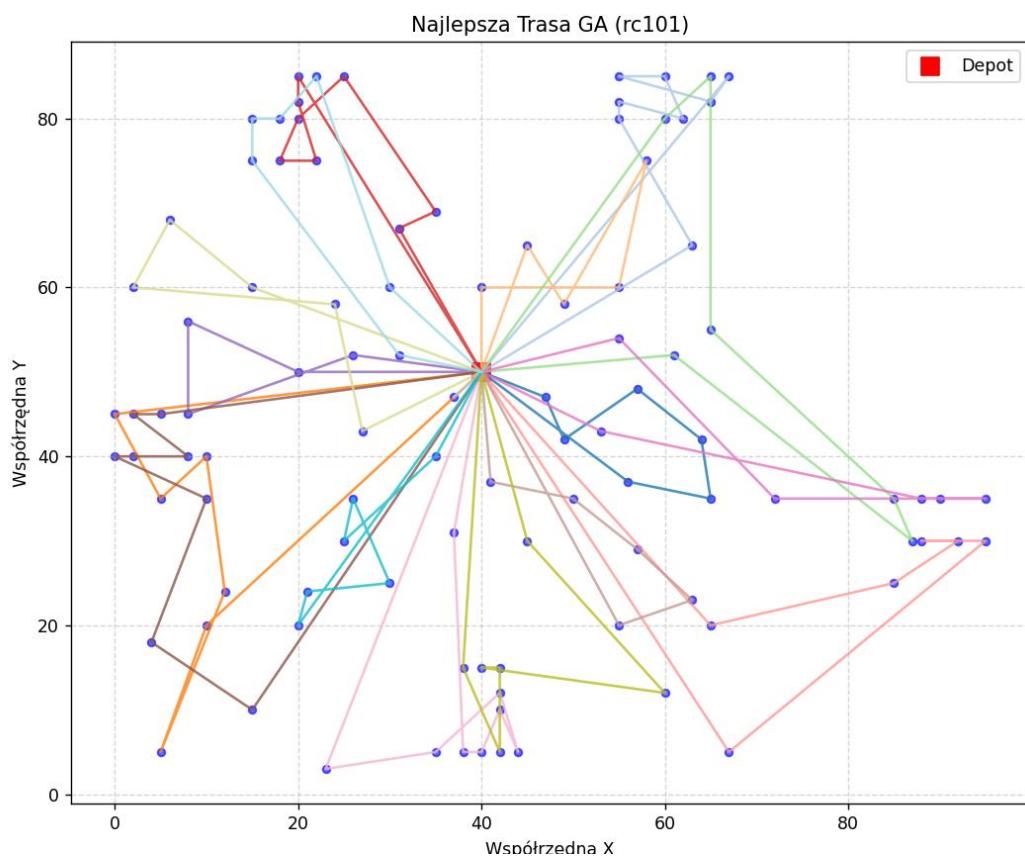
Zmiana całkowitego dystansu - rc101 (pop100)



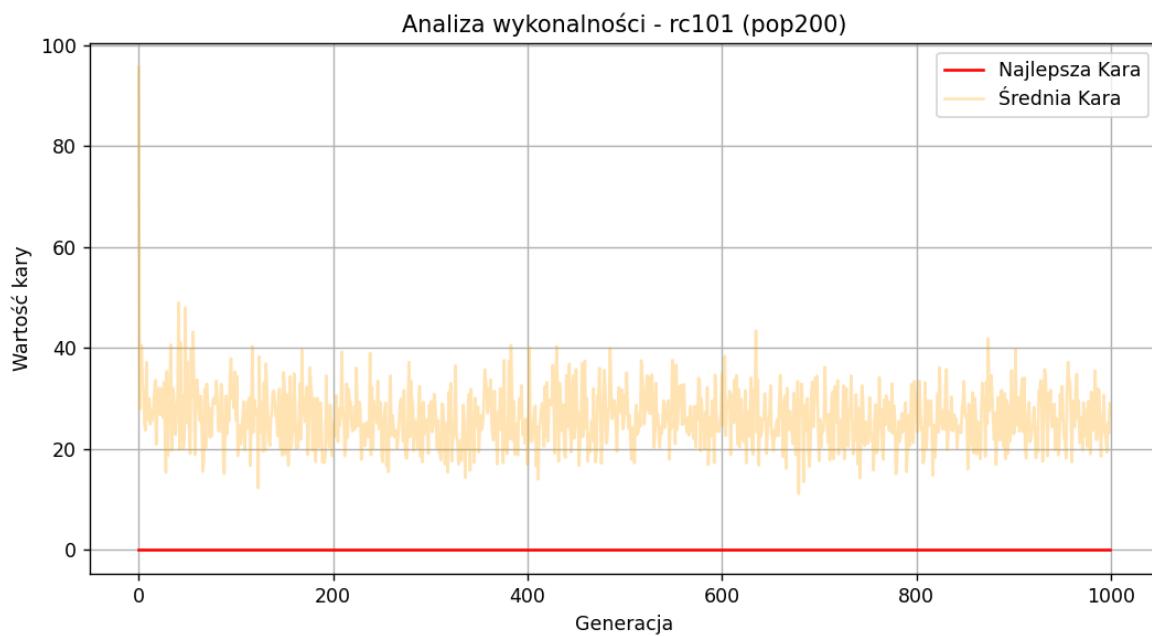
Wykres 80. Wizualizacja zmiany dystansu dla najlepszej trasy.



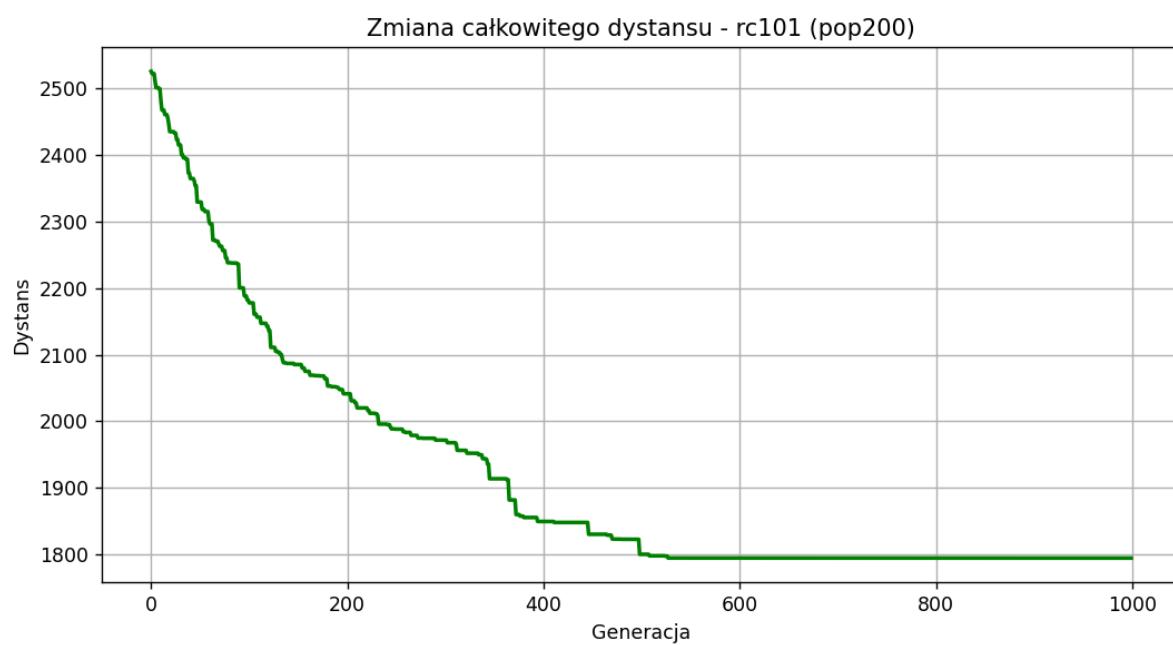
**Wykres 81.** Wizualizacja rozwiązań początkowego.



**Wykres 82.** Wizualizacja najlepszego rozwiązania.



**Wykres 83.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 84.** Wizualizacja zmiany dystansu dla najlepszej trasy.

## 2. Badanie wpływu współczynnika mutacji

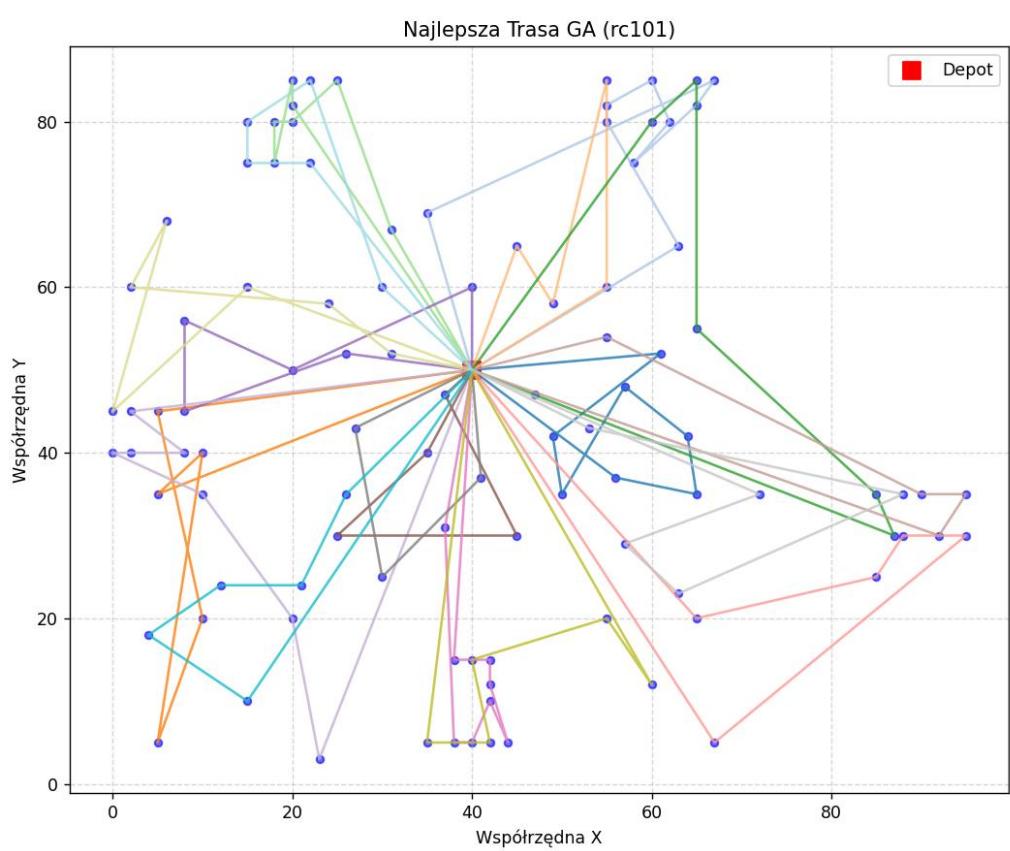
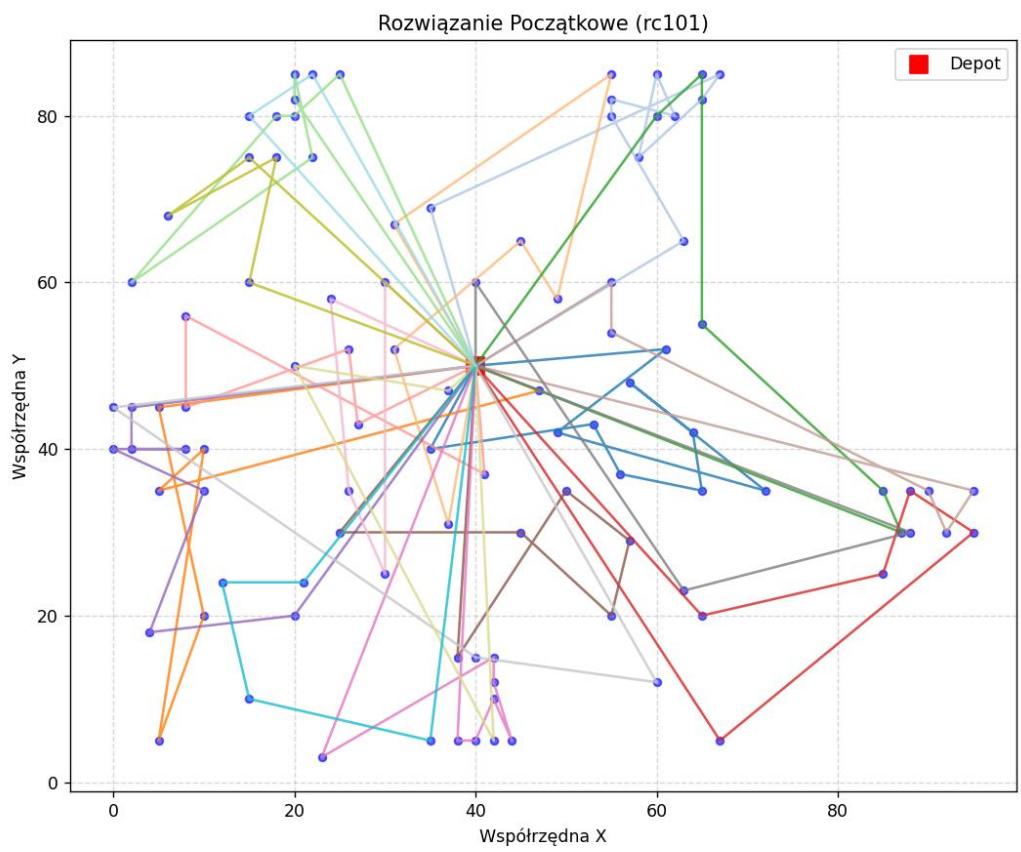
Współczynnik mutacji jest kluczowym parametrem kontrolującym balans między eksploracją (przeszukiwaniem nowych obszarów przestrzeni rozwiązań) a eksploatacją (poprawianiem aktualnie najlepszych tras). Wartości zmieniane będą wynosiły:  $Pm = [0.05, 0.1, 0.4]$ . Eksperyment zostanie wykonany po 5 razy dla każdej badanej wartości dla 3 plików.

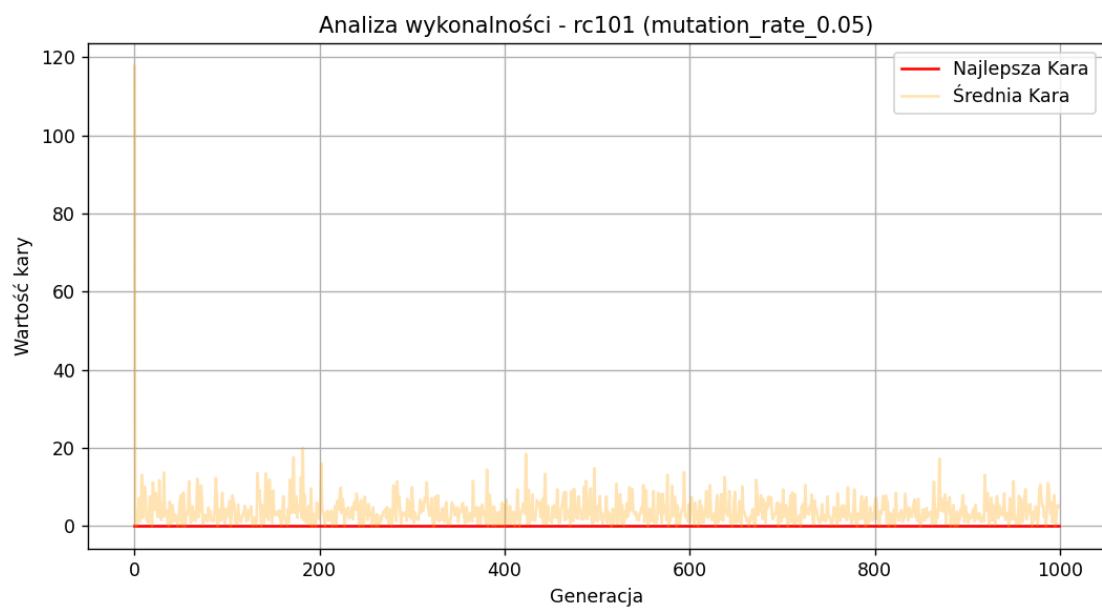
**Tabela 15.** Wyniki 5 eksperymentów dla badanego współczynnika.

Lp.	Mutacja (Pm )	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	0.05	1	19	2097.39	0.00
2		2	19	2260.83	0.00
3		3	18	2222.27	0.00
4		4	18	2108.37	0.00
5		5	18	2166.84	0.00
6	0.10	1	18	2058.47	0.00
7		2	19	2101.05	0.00
8		3	18	2129.05	0.00
9		4	18	1996.23	0.00
10		5	19	2079.62	0.00
11	0.40	1	17	1879.25	0.00
12		2	17	1988.80	0.00
13		3	18	2050.47	0.00
14		4	18	1878.36	0.00
15		5	18	1966.29	0.00

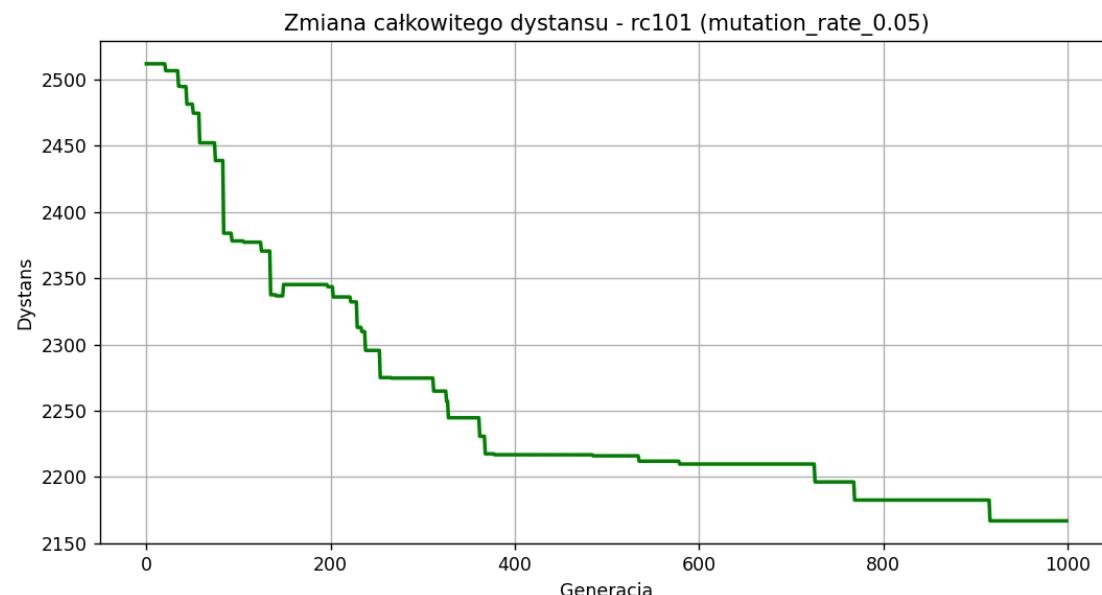
**Tabela 16.** Wyniki podsumowujące.

Mutacja (Pm )	Najlepszy Dystans	Średnia (Mean)	Odcchylenie (Std)	Najlepsza liczba aut
0.05	2097.39	2171.14	63.33	18
0.10	1996.23	2072.88	44.89	18
0.40	1878.36	1952.63	66.29	17

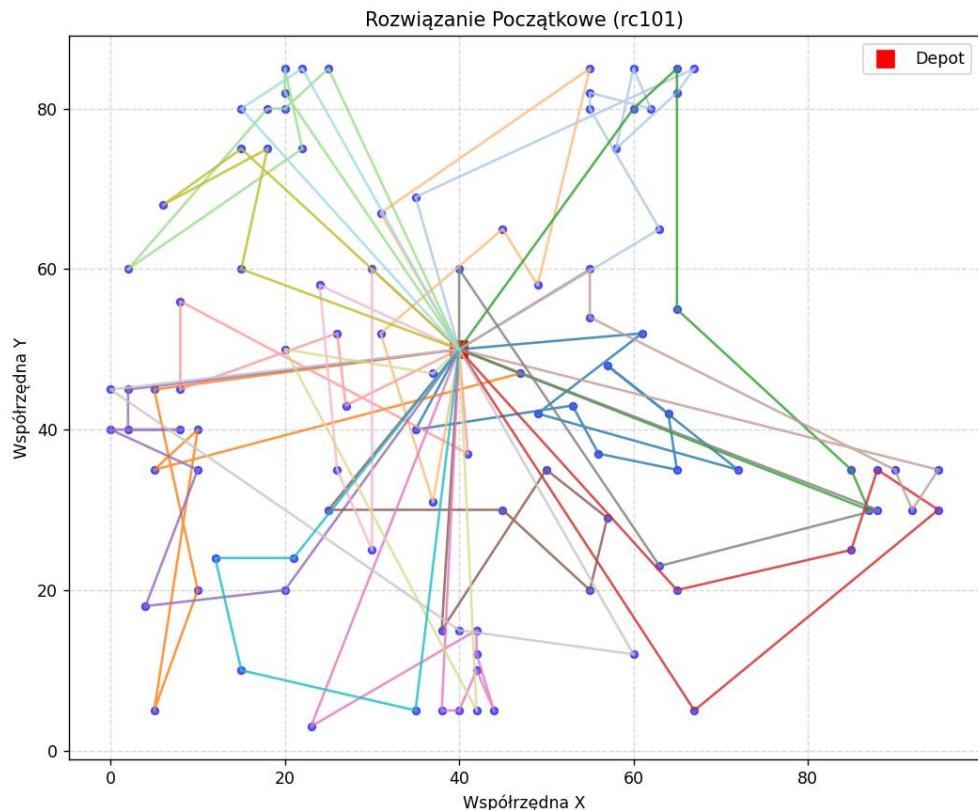




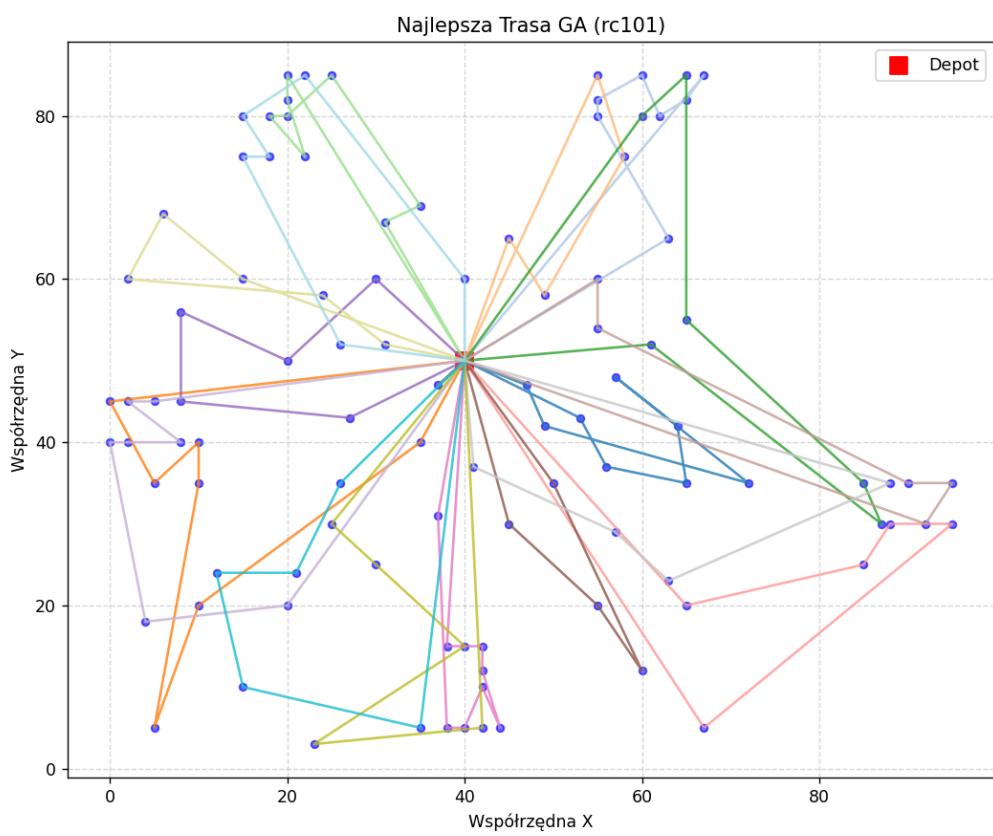
**Wykres 87.** Wizualizacja średniej kary populacji dla najlepszej trasy.



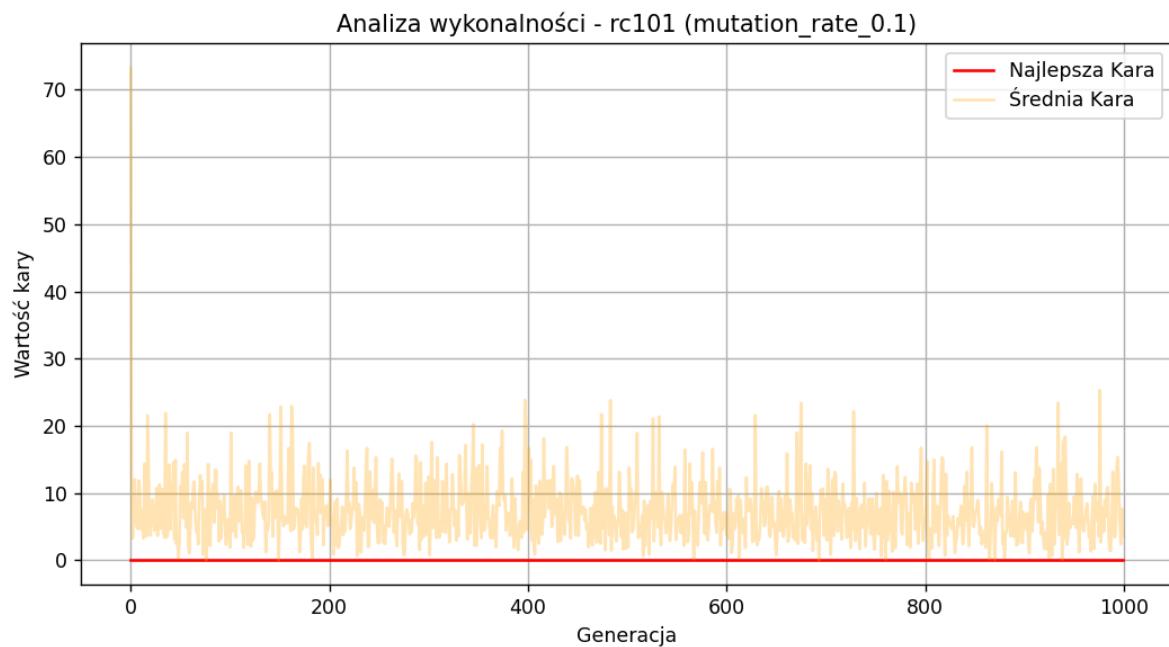
**Wykres 88.** Wizualizacja zmiany dystansu dla najlepszej trasy.



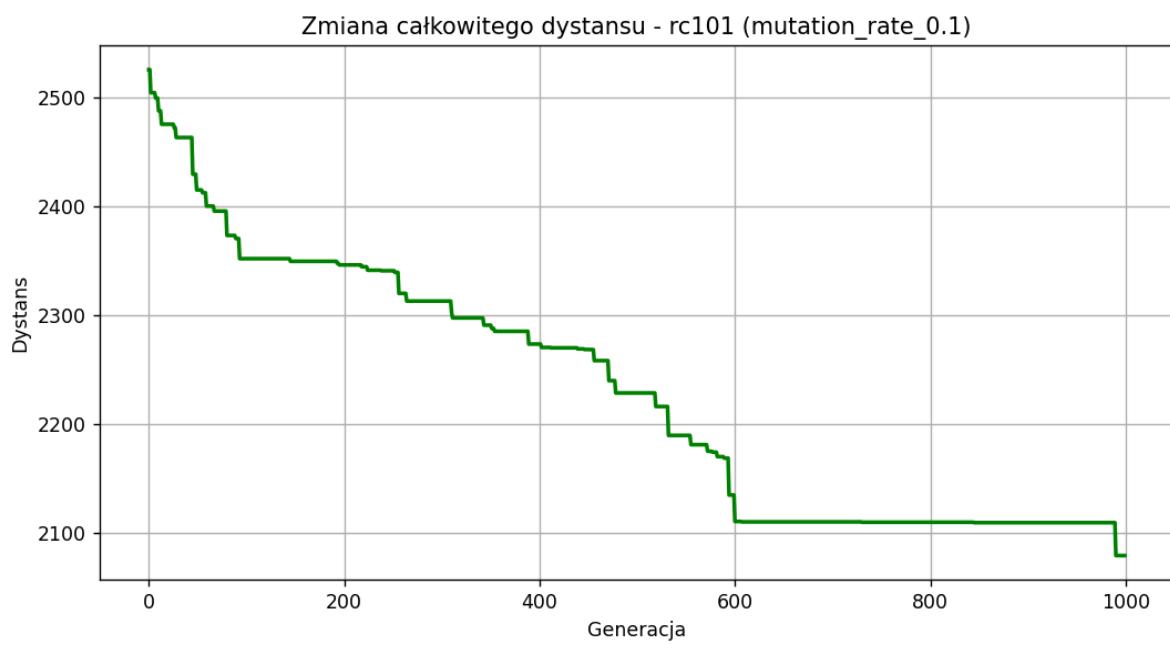
**Wykres 89.** Wizualizacja rozwiązania początkowego.



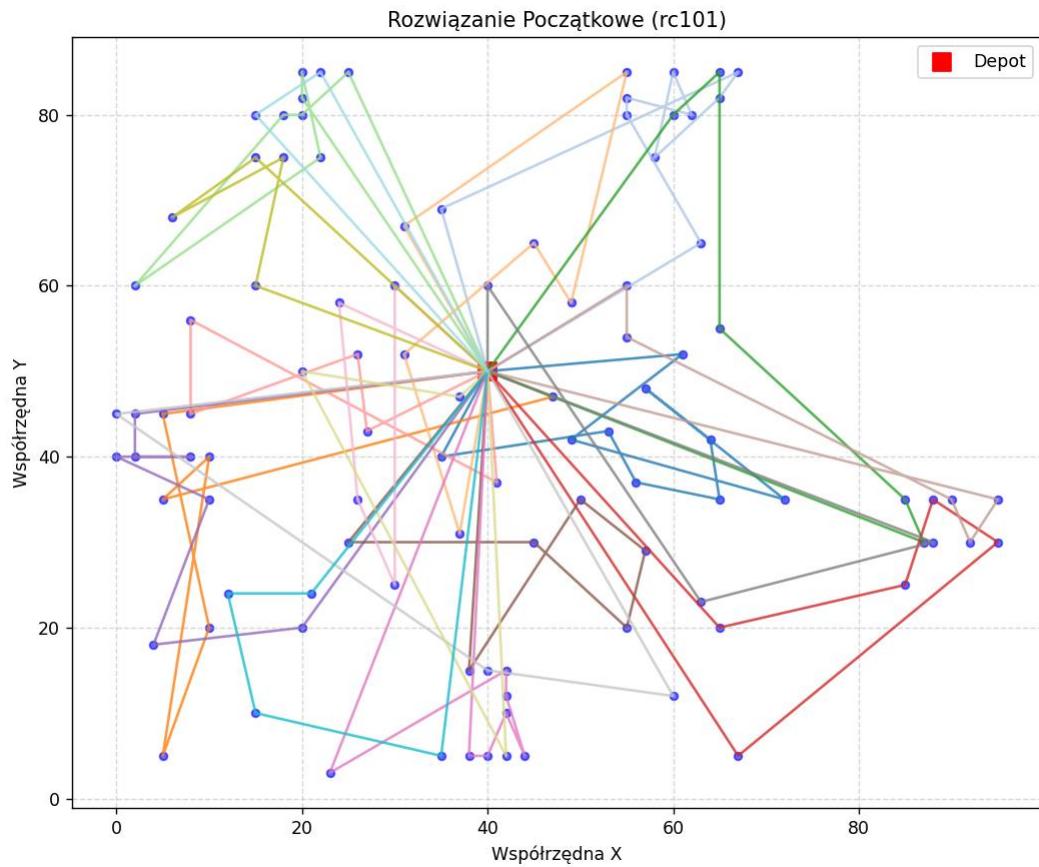
**Wykres 90.** Wizualizacja najlepszego rozwiązania.



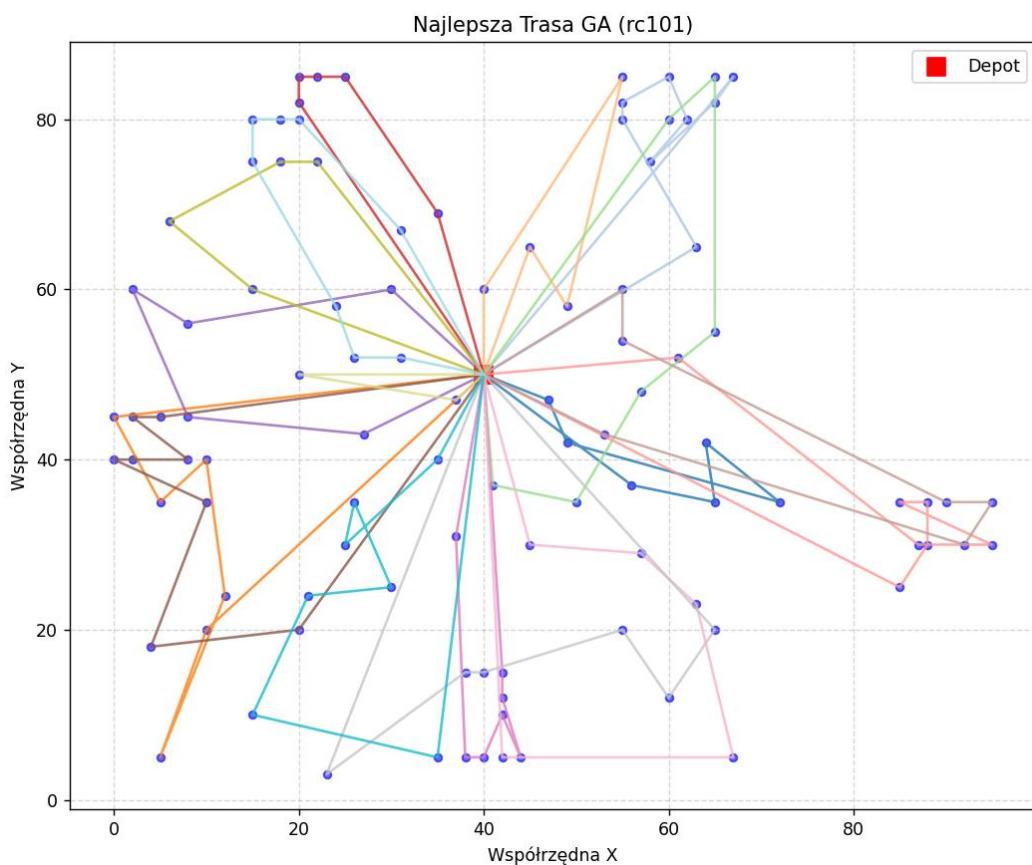
**Wykres 91.** Wizualizacja średniej kary populacji dla najlepszej trasy.



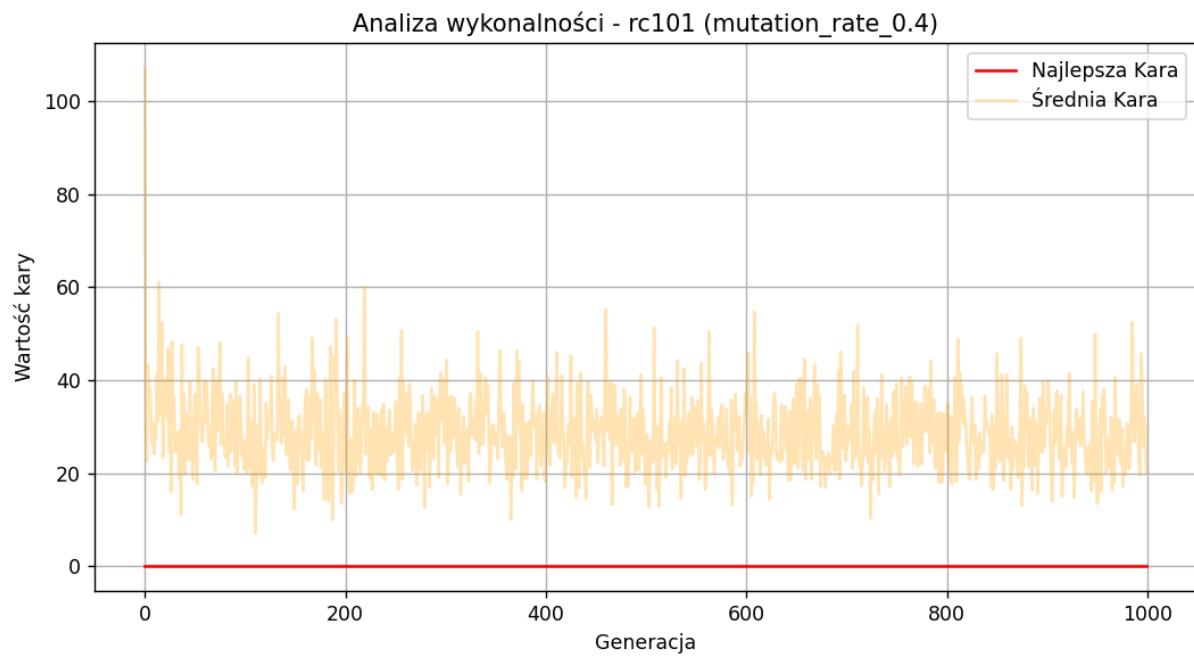
**Wykres 92.** Wizualizacja zmiany dystansu dla najlepszej trasy.



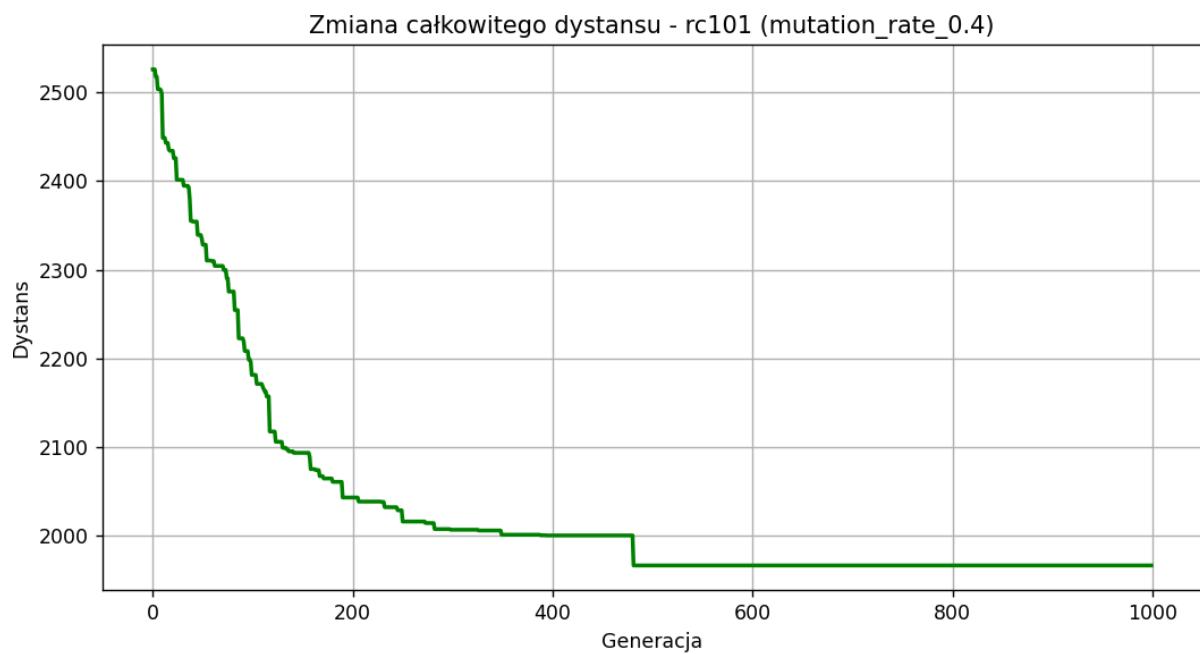
**Wykres 93.** Wizualizacja rozwiązania początkowego.



**Wykres 94.** Wizualizacja najlepszego rozwiązania.



*Wykres 95. Wizualizacja średniej kary populacji dla najlepszej trasy.*



*Wykres 96. Wizualizacja zmiany dystansu dla najlepszej trasy.*

### 3. Badanie wpływu współczynnika elitaryzmu.

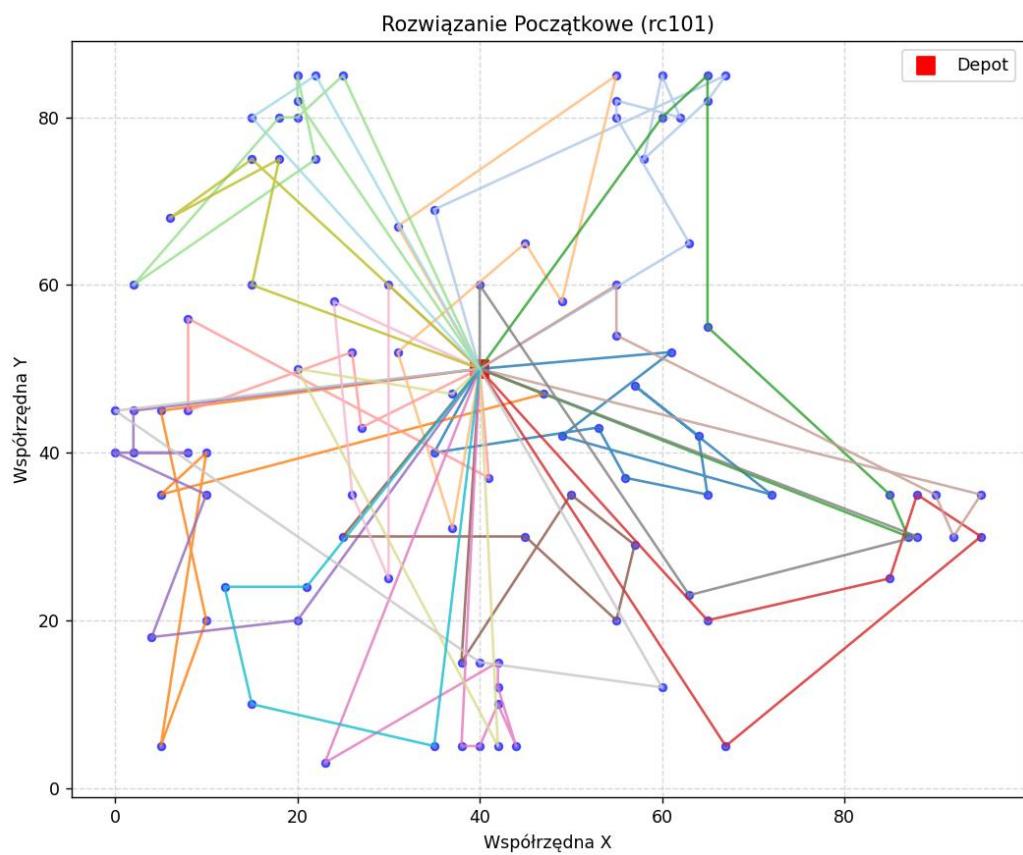
Elitaryzm chroni najlepsze osobniki przed negatywnymi skutkami losowości. W problemie VRPTW, gdzie znalezienie trasy spełniającej wszystkie okna czasowe (Penalty = 0) jest trudne, elitaryzm pełni rolę "bezpiecznika". Niski elitaryzm: Pozwala na dużą rotację w populacji, ale ryzykuje utratę rekordowego wyniku ("regresja"). Wysoki elitaryzm: Gwarantuje zachowanie najlepszych cech, ale może prowadzić do zbyt szybkiej dominacji jednego osobnika i utraty różnorodności (stagnacja w minimum lokalnym). Przeprowadzane będzie badanie wartości E = [0.05,0.2,0.4].

**Tabela 17.** Wyniki 5 eksperymentów dla badanego współczynnika.

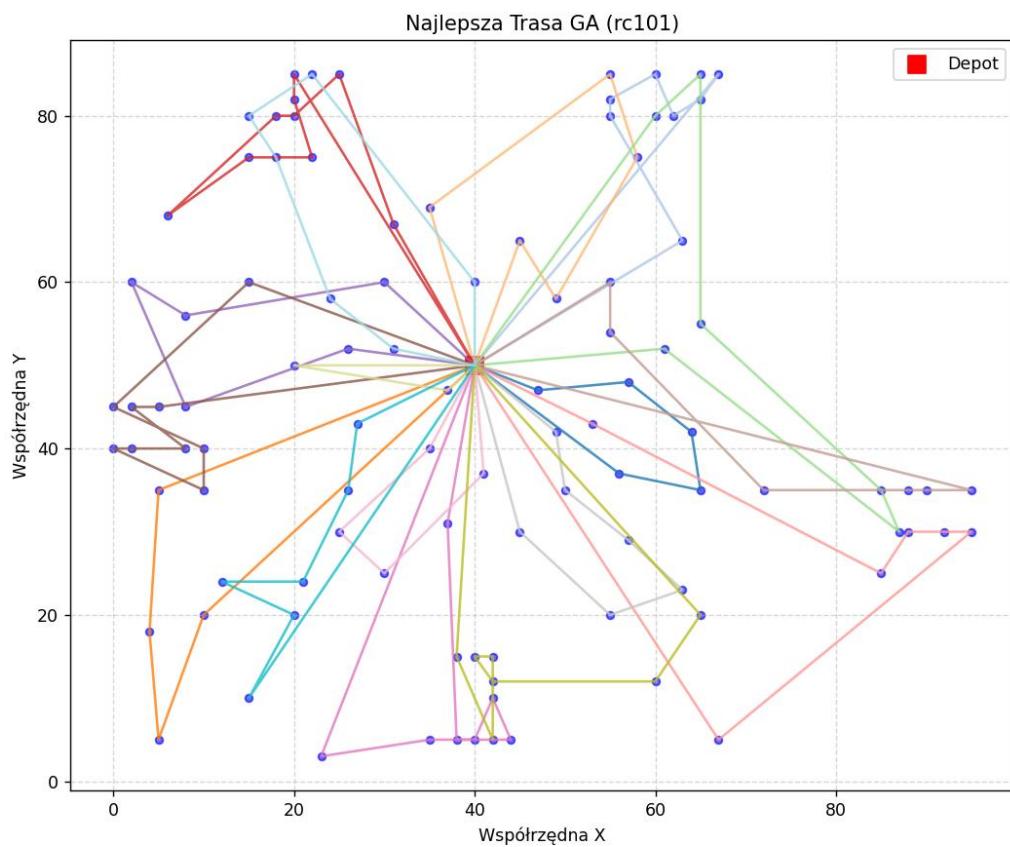
Lp.	Elitaryzm (E)	Próba	Liczba aut (Veh)	Dystans (Dist)	Kara (Penalty)
1	0.05	1	17	1931.37	0.00
2		2	19	2056.73	0.00
3		3	19	2001.50	0.00
4		4	18	1974.21	0.00
5		5	17	1843.45	0.00
6	0.20	1	19	1922.65	0.00
7		2	17	1901.54	0.00
8		3	18	2045.51	0.00
9		4	18	2026.42	0.00
10		5	18	2047.78	0.00
11	0.40	1	18	2008.74	0.00
12		2	18	1971.12	0.00
13		3	19	2034.54	0.00
14		4	19	2003.12	0.00
15		5	17	1826.86	0.00

**Tabela 18.** Wyniki podsumowujące.

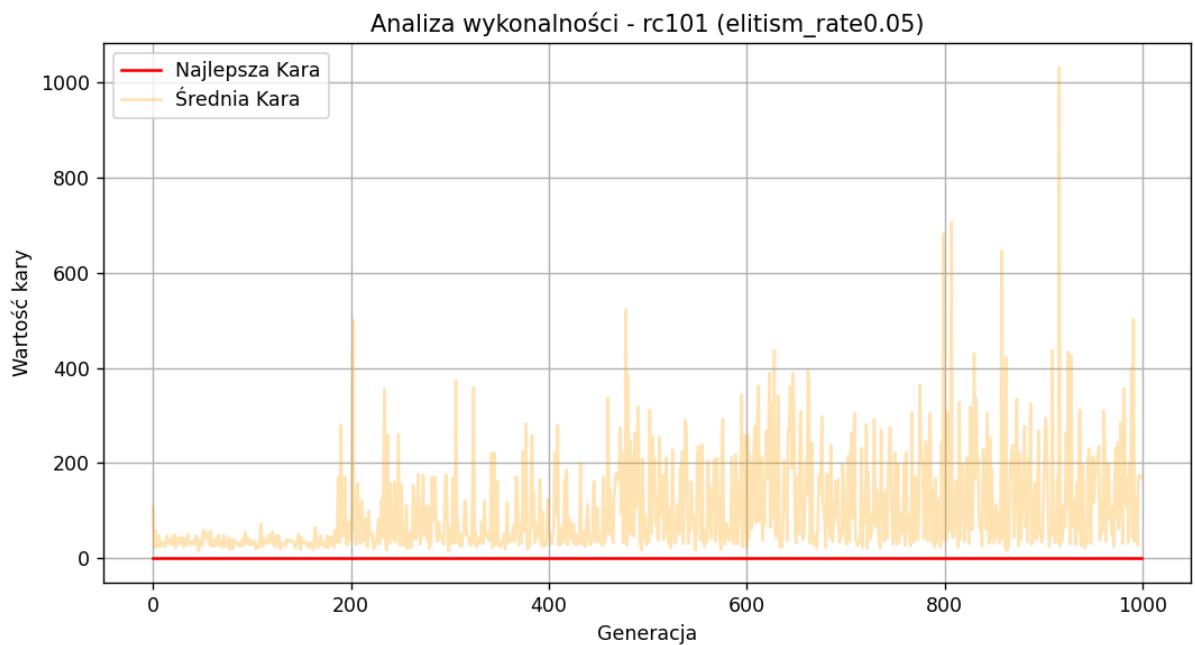
Elitaryzm (E)	Najlepszy Dystans	Średnia (Mean)	Odchylenie (Std)	Najlepsza liczba aut
0.05	1843.45	1961.45	71.66	17
0.20	1901.54	1988.78	63.41	17
0.40	1826.86	1968.87	73.82	17



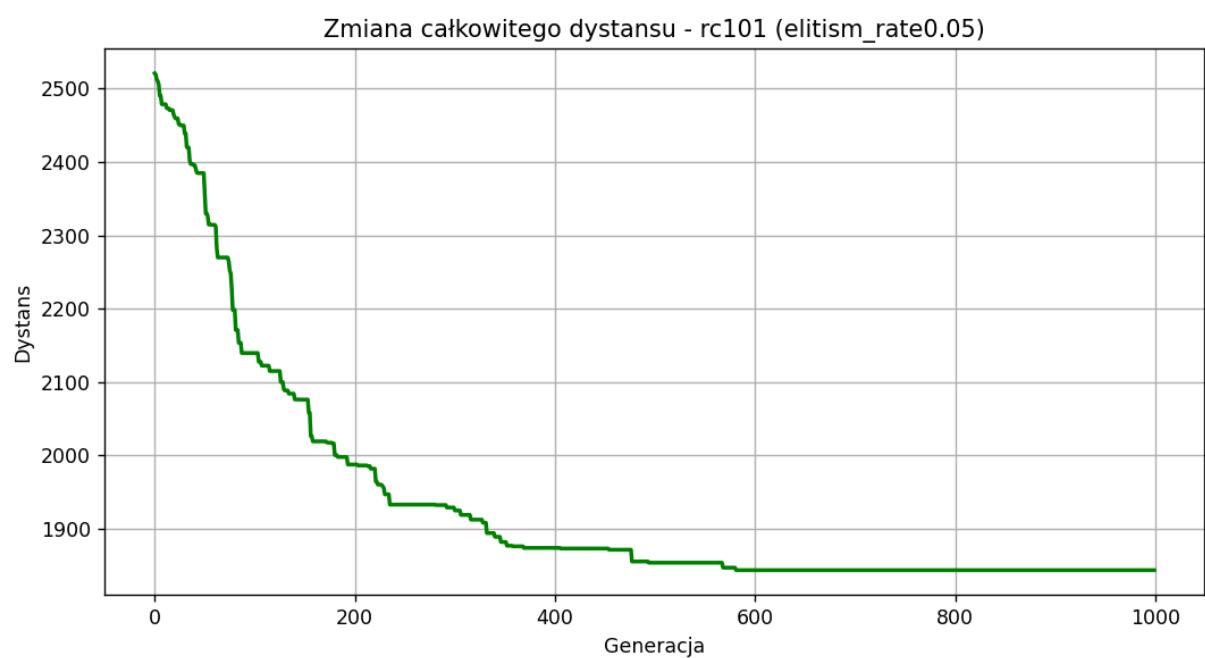
**Wykres 97.** Wizualizacja rozwiązania początkowego.



**Wykres 98.** Wizualizacja najlepszego rozwiązania.

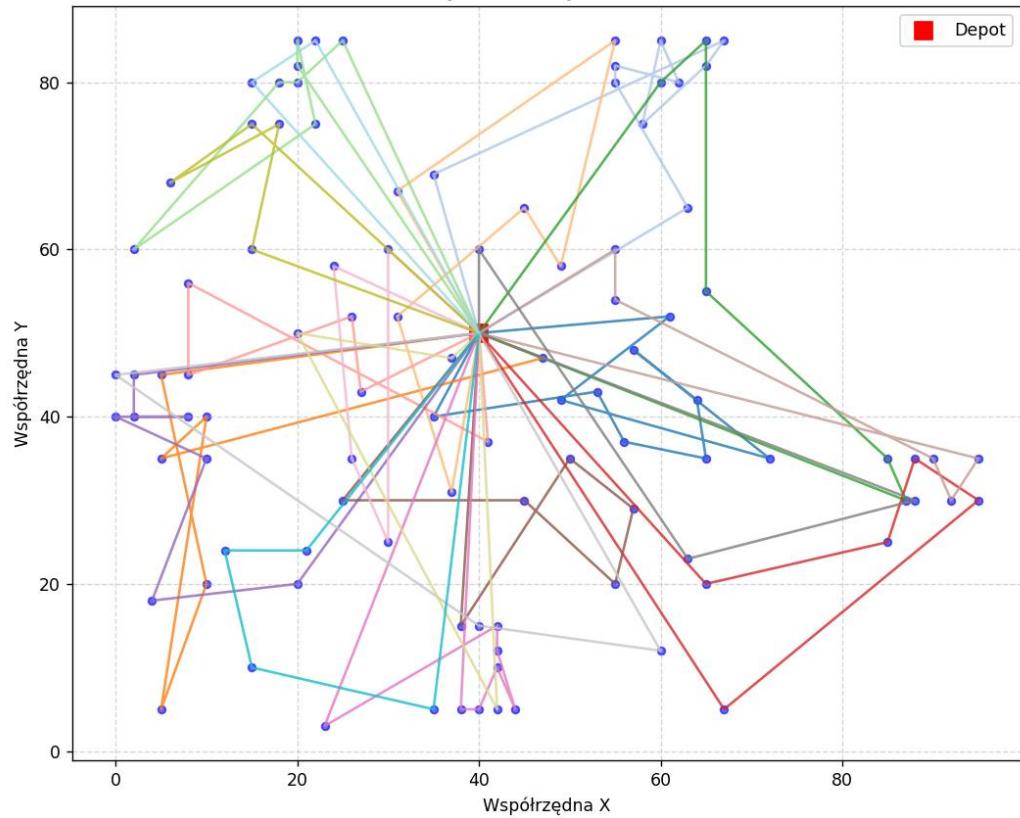


**Wykres 99.** Wizualizacja średniej kary populacji dla najlepszej trasy.



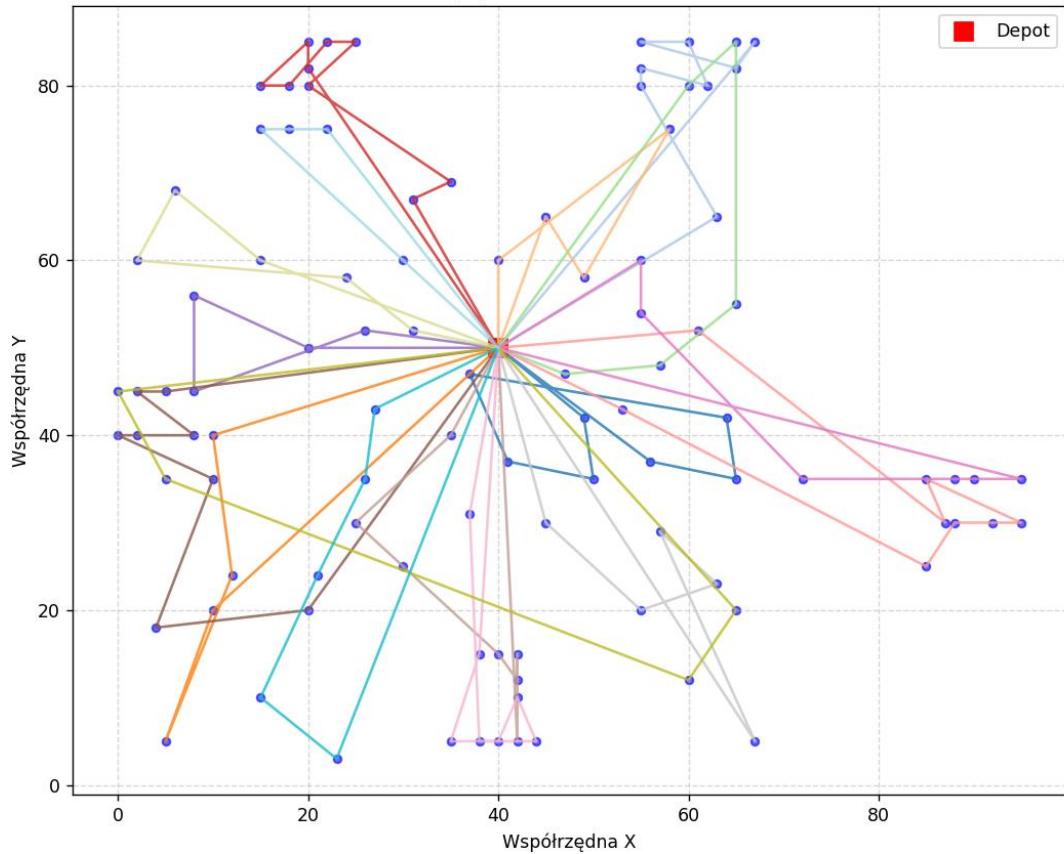
**Wykres 100.** Wizualizacja zmiany dystansu dla najlepszej trasy.

Rozwiązań Początkowe (rc101)



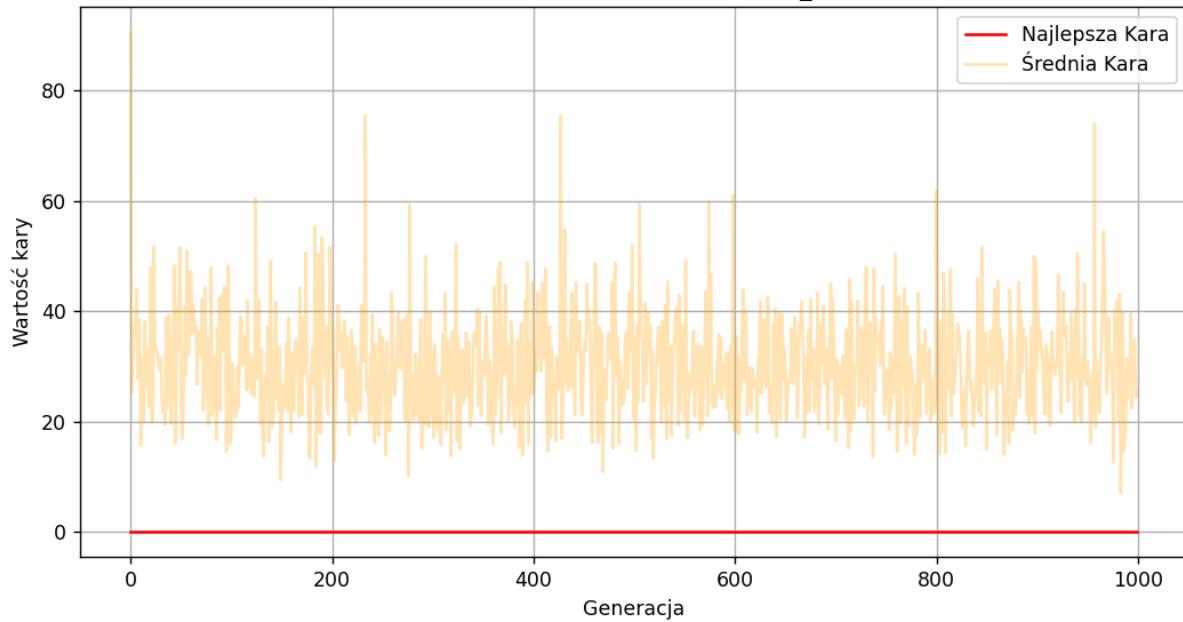
Wykres 101. Wizualizacja rozwiązania początkowego.

Najlepsza Trasa GA (rc101)



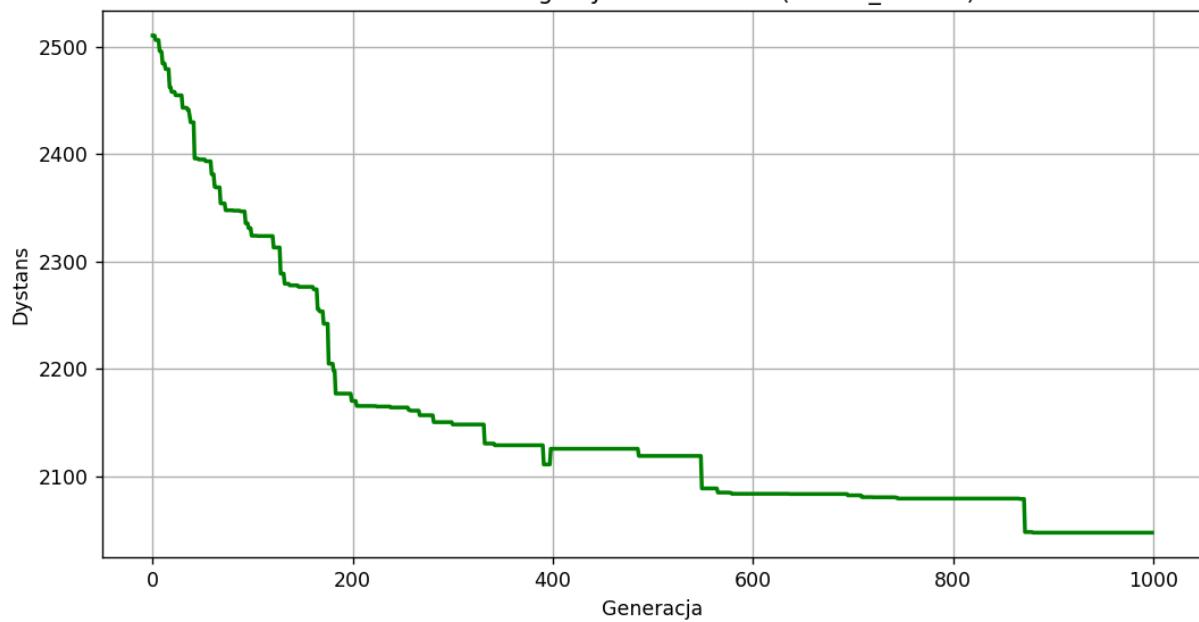
Wykres 102. Wizualizacja najlepszego rozwiązania.

Analiza wykonalności - rc101 (elitism\_rate0.2)

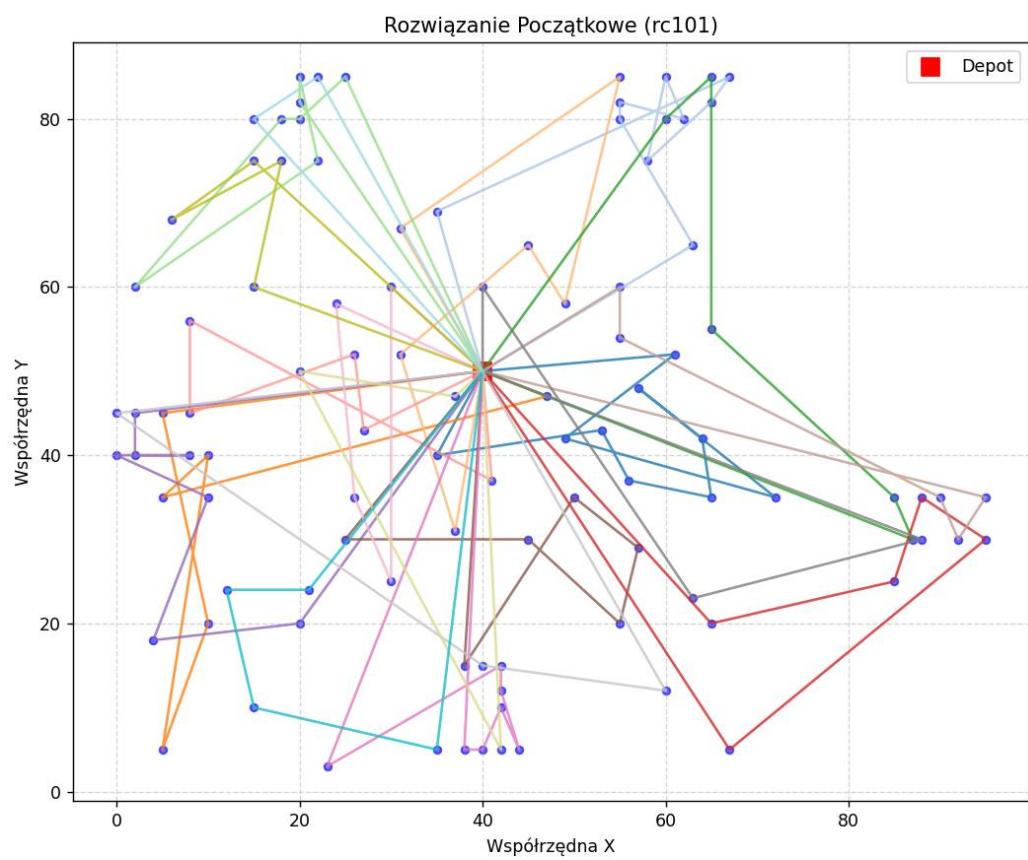


Wykres 103. Wizualizacja średniej kary populacji dla najlepszej trasy.

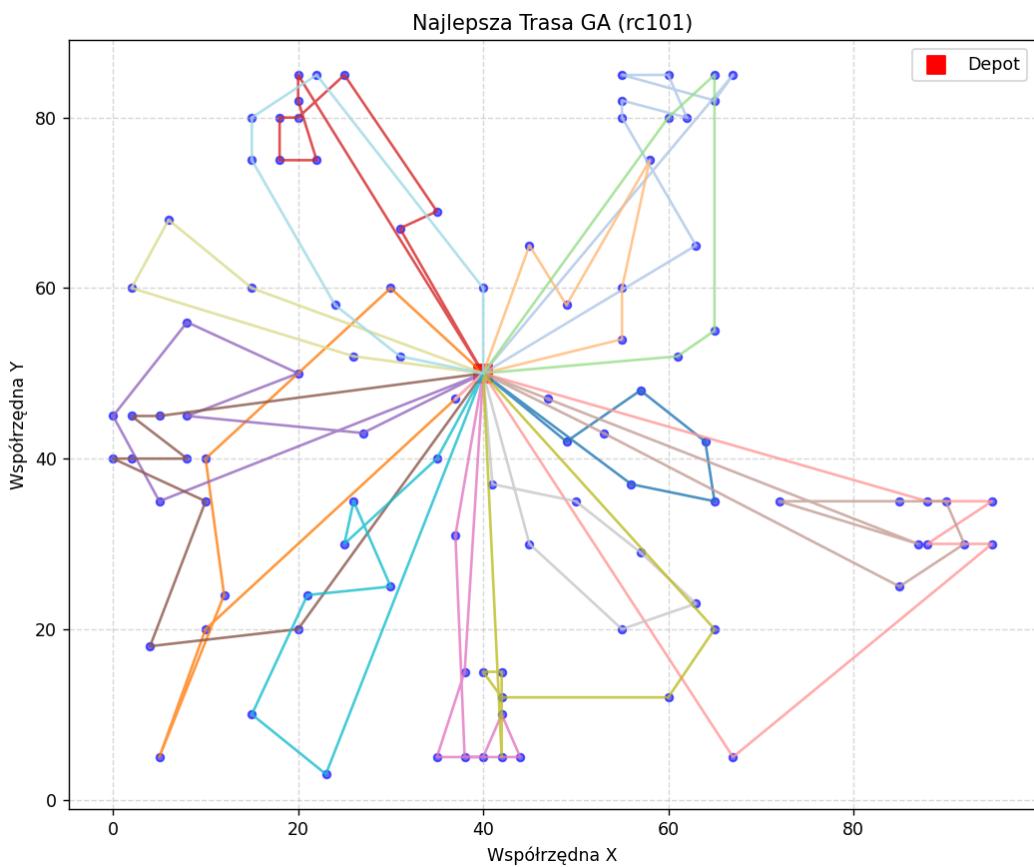
Zmiana całkowitego dystansu - rc101 (elitism\_rate0.2)



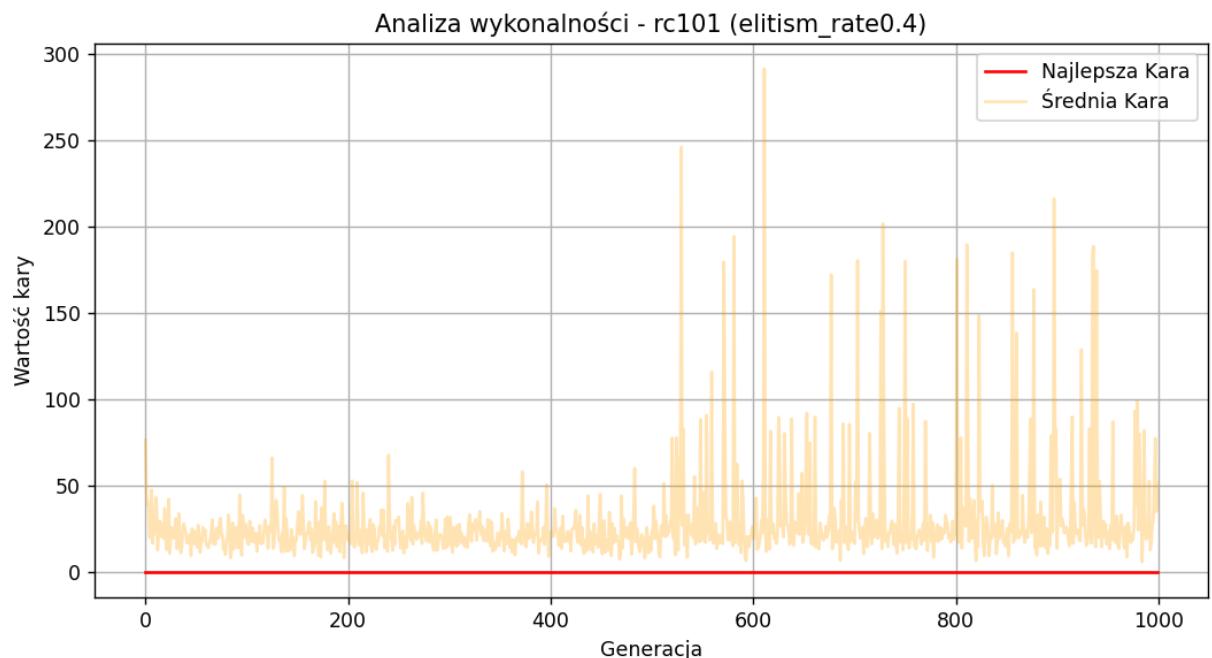
Wykres 104. Wizualizacja zmiany dystansu dla najlepszej trasy.



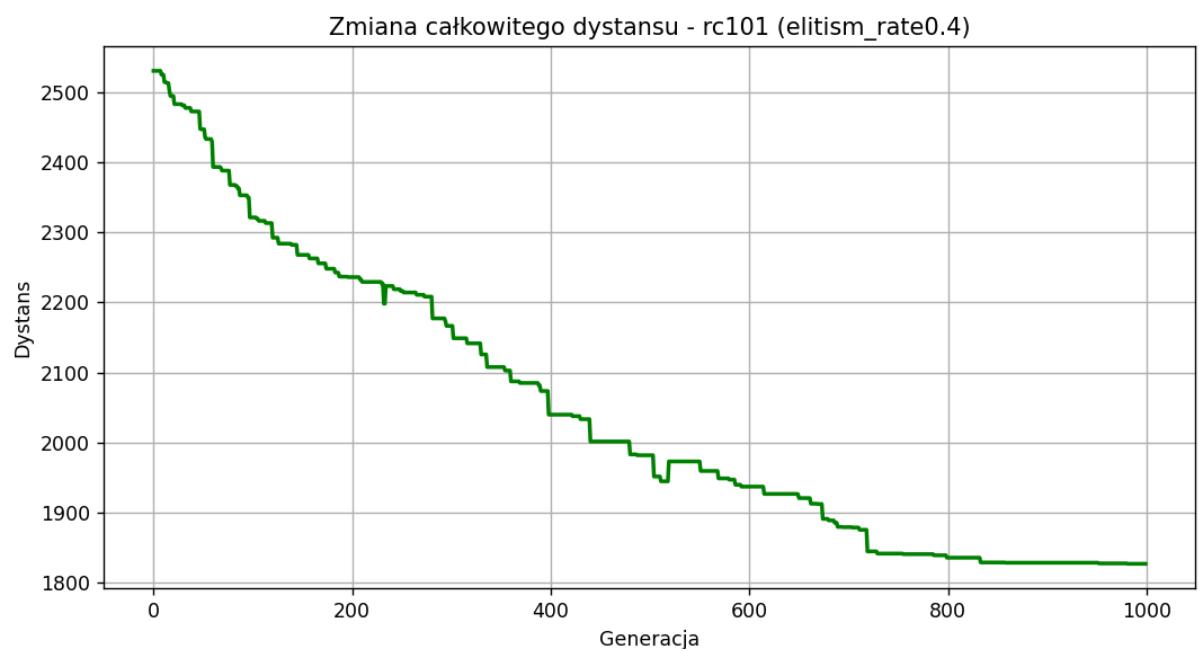
**Wykres 105.** Wizualizacja rozwiązania początkowego.



**Wykres 106.** Wizualizacja najlepszego rozwiązania.



**Wykres 107.** Wizualizacja średniej kary populacji dla najlepszej trasy.



**Wykres 108.** Wizualizacja zmiany dystansu dla najlepszej trasy.

## 7 Analiza Wyników Eksperymentalnych

### 7.1 Wpływ poszczególnych parametrów na jakość rozwiązania

#### 7.1.1 Wielkość populacji (N)

Zwiększenie wielkości populacji z 50 do 200 osobników miało pozytywny wpływ na jakość znajdowanych rozwiązań. Dystans dla wszystkich badanych plików znacznie się zmniejszył na przestrzeni tej samej ilości generacji algorytmu.

- Dystans: W instancji R101 przejście z N=50 na N=200 zaowocowało spadkiem średniego dystansu z wartości 2046.47 do 1859.78, w C101 zwiększenie spowodowało spadek z N = 972.70 do N = 929.49, w RC101 z N = 1944.51 do N = 1794.55.
- Liczba pojazdów: Większa populacja umożliwiła algorytmowi skuteczniejsze przeszukiwanie "wąskich gardeł" w oknach czasowych, co pozwoliło na stabilne osiąganie niższej liczby pojazdów np. 21 zamiast 22/23 w R101. W innych plikach nie znaleziono różnic w ilości aut.
- Wniosek: Większa populacja zapewnia lepszą eksplorację, ale po przekroczeniu N=100 zysk jakościowy maleje względem liniowego wzrostu kosztu obliczeniowego.

#### 7.1.2 Współczynnik mutacji (Pm)

- We wszystkich trzech plikach (r101, c101, rc101) zaobserwowano wyraźną korelację dodatnią między wzrostem współczynnika mutacji a jakością końcowego rozwiązania:
- Przełamywanie minimów lokalnych: Najniższa mutacja (0.05) konsekwentnie dawała najgorsze wyniki. Algorytm przy tak małej zmienności zbyt szybko tracił różnorodność i "utykał" w suboptimalnych rozwiązaniach.
- Optymalizacja floty: Wysoka mutacja (0.40) okazała się niezbędna do redukcji liczby pojazdów w trudniejszych instancjach. W przypadku c101 pozwoliła zejść z 12 do 11 aut, a w rc101 z 18 do 17 aut. Sugeruje to, że agresywne mutacje są kluczowe dla znalezienia rzadkich kombinacji ruchów, które umożliwiają całkowite usunięcie jednej trasy.
- Zysk dystansowy: Największą poprawę dystansu przy wzroście mutacji odnotowano dla instancji c101 oraz r101.

#### 7.1.3 Elitaryzm

Elitaryzm pełni funkcję "pamięci" algorytmu, chroniąc najlepsze osobniki przed destrukcyjnym wpływem mutacji. Z danych wynika, że jego optymalny poziom zależy od charakteru instancji:

- Instancja r101: Zaobserwowano korelację liniową – wraz ze wzrostem elitaryzmu jakość rozwiązania rosła. Najlepszy wynik (1769.41) uzyskano przy najwyższym badanym wskaźniku E = 0.40. Sugeruje to, że w środowisku rozproszonym silna ochrona dobrych wzorców jest kluczowa dla sukcesu optymalizacji.

- Instancja c101: Tutaj najwyższy elitaryzm również przyniósł najlepszy dystans (949.73), jednak różnica między E=0.05 a E=0.20 była niewielka. Klastry klientów są naturalnie stabilne dla operatorów lokalnych, więc algorytm szybciej znajduje dobre rozwiązania nawet przy mniejszej elicie.
- 4. Instancja rc101: Wyniki są najbardziej zróżnicowane. Najlepszy dystans (1826.86) uzyskano przy E = 0.40, ale wartość E = 0.20 dała gorszy wynik niż E = 0.05. Świadczy to o dużej złożoności tej instancji, gdzie zbyt duża lub zbyt mała elita może prowadzić do utknięcia w różnych minimach lokalnych.

## 7.2 Stabilność działania algorytmu

- Wzrost populacji z 50 na 200 zazwyczaj poprawia stabilność. Większa liczba osobników pozwala na bardziej systematyczne przeszukanie przestrzeni, co redukuje ryzyko przypadkowego, słabego wyniku w pojedynczym biegu.
- Współczynnik mutacji (Pm): Agresywna mutacja (0.4) obniża stabilność. Na przykład w C101 wzrost mutacji z 0.1 do 0.4 spowodował skok Std z 14.85 do 66.18. Jest to koszt "szukania" lepszych rozwiązań – algorytm częściej znajduje rekordowe wyniki, ale rzadziej je powtarza.
- Elitaryzm (E): Wskaźnik na poziomie 0.20 okazał się najbardziej stabilny dla wszystkich instancji. Zbyt niska elita (0.05) wprowadza chaos (utrata dobrych wyników), a zbyt wysoka (0.40) prowadzi do przedwcześnie zbieżności do różnych suboptimalnych punktów, co również zwiększa rozrzut statystyczny.

## 7.3 Trudność grup danych (R vs C vs RC)

### Grupa C (Clustered) – Najłatwiejsza optymalizacja dystansu

Instancje typu C okazały się najbardziej podatne na działanie hybrydowego GA, szczególnie dzięki integracji operatora 2-opt.

- Klienci są pogrupowani w gęste skupiska, co sprawia, że optymalna kolejność odwiedzin wewnętrz klastra jest geometrycznie oczywista.
- Grupa ta uzyskała najniższe wyniki dystansu (nawet poniżej 900 przy optymalnych parametrach) oraz najniższą zmienność (Std).

### Grupa R (Random) – Wyzwanie dla eksploracji

Instancje typu R stanowią test dla czystej wydajności algorytmu genetycznego w przeszukiwaniu rozproszonej przestrzeni.

- Brak skupisk wymusza na algorytmie budowanie tras poprzez łączenie odległych punktów, co zwiększa liczbę możliwych kombinacji.
- Średnie dystanse są tu znaczco wyższe niż w grupie C (oscylują wokół 1800-2000).
- Grupa ta najlepiej reagowała na zwiększenie wielkości populacji – im więcej osobników, tym lepiej algorytm "wyczuwał" optymalne korytarze przejazdowe między rozproszonymi klientami.

## Grupa RC (Mixed) – Najwyższy stopień trudności

Instancje RC łączą w sobie cechy obu poprzednich grup, co czyni je najbardziej wymagającymi dla metaheurystyk.

- Część klientów tworzy klastry, podczas gdy pozostali są rozlokowani losowo. Wymaga to od algorytmu jednoczesnego szlifowania tras lokalnych i planowania długich tras łącznikowych.
- Ta grupa wygenerowała najwyższe odchylenia standardowe (nawet powyżej 80), co świadczy o dużej liczbie minimów lokalnych, w których algorytm może utknąć.
- RC101 wymagała najbardziej agresywnej mutacji (0.4), aby uzyskać jakąkolwiek redukcję liczby pojazdów, co potwierdza jej złożoność strukturalną.

## 8 PORÓWNANIE Z ROZWIĄZANIAMI LITERATUROWYMI (BKS)

Instancja	Najlepszy wynik literaturowy (BKS)	Liczba aut	Najlepszy wynik eksperymentu	Najlepsza liczba aut (Eksperiment)
R101	1650.80	19	1769.41	21
C101	828.94	10	861.70	11
RC101	1696.95	14	1794.55	17

- Najbliższy wynik wynika z współpracy algorytmu genetycznego z lokalnym operatorem 2-opt. W grupach klastrowych optymalizacja ścieżek "wewnętrz" skupisk jest geometrycznie prostsza, co pozwoliło niemal zblizić się do rekordów światowych. Jednak przy większej flocie.
- W instancji R algorytm musi przeszukiwać gigantyczną liczbę kombinacji połączeń między rozproszonymi punktami, co przy limicie 1000 generacji utrudnia znalezienie idealnego "szwu" łączącego wszystkie trasy.
- W instancji typu RC konieczność jednoczesnej obsługi klastrów i pojedynczych punktów tworzy bardzo wiele minimów lokalnych. Algorytm skupił się tu na priorytetowej redukcji liczby aut, co odbyło się kosztem optymalizacji dystansu.

## 9 WNIOSKI I PODSUMOWANIE

### 9.1 Skuteczność algorytmu

Algorytm wykazał się wysoką efektywnością, szczególnie w aspekcie priorytetyzacji celów.

- W każdej próbie udało się wyeliminować kary za spóźnienia (Penalty = 0.0).
- Optymalizacja floty: Algorytm skutecznie redukował liczbę pojazdów, osiągając wyniki bardzo bliskie optymalnym.
- Struktura danych: Najlepiej sprawdził się na danych klastrowych (C101).

## 9.2 Najlepsze ustawienia parametrów

- Populacja (N): 100–200 osobników. Większa populacja poprawia eksplorację, ale zwiększa czas obliczeń.
- Mutacja (Pm): 0.4 (40%). Wysoki wskaźnik jest niezbędny do przełamywania struktur klastrowych i redukcji floty.
- Elitaryzm (E): 0.2 (20%). Wartość ta zapewniała najniższe odchylenie standardowe (Std) i stabilną akumulację dobrych rozwiązań.

## 9.3 Ograniczenia rozwiązania

- Ze względu na hybrydyzację (GA + 2-opt), czas wykonania 1000 generacji dla dużych populacji staje się znaczący.
- W instancjach typu RC, algorytm wykazywał największą zmienność wyników (wysokie Std), co sugeruje trudność w ucieczce z głębokich minimów lokalnych przy obecnych operatorach.
- Jakość końcowa jest częściowo determinowana przez rozwiązywanie początkowe, co może ograniczać "świeżość" populacji w pierwszych fazach ewolucji.

## 9.4 Kierunki dalszej poprawy

- Zastąpienie prostego 2-opt algorytmem 3-opt lub Lin-Kernighan dla jeszcze lepszego szlifowania dystansu.
- Wprowadzenie mechanizmu *Adaptive Mutation Rate – dynamiczne parametry*, który zwiększałby mutację w momencie wykrycia stagnacji populacji.