



POLITECHNIKA KRAKOWSKA
WYDZIAŁ INŻYNIERII LĄDOWEJ
KATEDRA SYSTEMÓW TRANSPORTOWYCH
I LOGISTYCZNYCH

ROZPRAWA DOKTORSKA

**Synchronizacja rozkładów jazdy w systemie transportu
publicznego z wykorzystaniem algorytmów genetycznych**

Autor: **mgr inż. Daniel Kapica**
Promotor: **dr hab. inż. Vitalii Naumov, prof. PK**

Kraków, 2025

Streszczenie

W pierwszym rozdziale pracy przedstawiono definicję oraz prawne uwarunkowania funkcjonowania i organizowania publicznego transportu zbiorowego w Polsce. Opisano również proces i problemy napotymane podczas planowania oferty przewozowej. Przedstawiono czym jest synchronizacja rozkładów jazdy. Następnie przybliżono nomenklaturę, zasady działania oraz parametry algorytmów genetycznych. Rozdział kończy przegląd literatury przedmiotu wraz z jego podsumowaniem.

Drugi rozdział został poświęcony opisowi matematycznemu problemu synchronizacji. Zawarto w nim także szczegółowy opis użytej w pracy implementacji algorytmu genetycznego do synchronizacji rozkładów jazdy w publicznym transporcie zbiorowym. Opisano pakiety, klasy i metody wraz z ich argumentami, a także przedstawiono zachodzące pomiędzy nimi zależności.

Rozdział trzeci zawiera opis sieci publicznego transportu zbiorowego miasta Zielona Góra, z którego wybrane zostały linie do przeprowadzenia eksperymentu. Przedstawiono uwarunkowania przestrzenno-geograficzne miasta, opisano także główne generatory ruchu. Zaprezentowano również charakterystykę linii obsługujących tę sieć. Krótko opisano model miasta Zielona Góra wykonany w programie PTV Visum, który był źródłem danych do przeprowadzenia eksperymentu.

Rozdział czwarty rozpoczyna przełożenie terminologii algorytmów genetycznych na zagadnienia transportowe. Następnie opisano przeprowadzany eksperyment oraz szczegółowo przedstawiono jego plan i wykonywane czynności mające na celu znalezienie możliwie optymalnych parametrów algorytmu genetycznego wykorzystanego do synchronizacji rozkładów jazdy w publicznym transporcie zbiorowym. Zbadano również wpływ parametrów algorytmu na czas trwania obliczeń. Na podstawie wyników eksperymentu sformułowano wnioski i zasugerowano parametry, z jakimi powinien być uruchamiany algorytm genetyczny, a także przedstawiono dalsze kierunki badań związanych z tematyką synchronizacji rozkładów jazdy w publicznym transporcie zbiorowym.

Summary

In the first chapter, the definition and legal framework governing the functioning and organization of public transport in Poland are presented. The process of planning transport services and the challenges encountered therein are also discussed. The concept of timetable synchronization is introduced, followed by an outline of the terminology, operating principles, and parameters of genetic algorithms. The chapter concludes with a literature review and a summary of the state of research.

The second chapter is devoted to the mathematical formulation of the synchronization problem. It also provides a detailed description of the implementation of a genetic algorithm applied to timetable synchronization in public transport. The packages, classes, and methods used in the implementation are described along with their arguments, and the interdependencies between them are examined.

The third chapter contains a description of the public transport network in the city of Zielona Góra, from which selected bus lines were chosen for the experiment. The spatial and geographical characteristics of the city are presented, along with an account of its main traffic generators. The features of the lines operating within this network are also discussed. A brief description is given of the transport model of Zielona Góra developed using PTV Visum, which served as the data source for the experiment.

The fourth chapter begins with the translation of genetic algorithm terminology into transportation-related concepts. The experimental procedure is then described, with a detailed presentation of its design and the steps undertaken to determine the most suitable parameters of the genetic algorithm used for timetable synchronization in public transport. The impact of algorithm parameters on computational time is also analyzed. Based on the experimental results, conclusions are drawn and recommendations are made regarding the parameters with which the genetic algorithm should be executed. Finally, directions for further research on timetable synchronization in public transport are outlined.

Spis treści

Wstęp.....	9
Aktualność pracy.....	10
Cel i zadania.....	10
Struktura pracy.....	11
1. Rozdział I.....	13
1.1 Publiczny transport zbiorowy – definicja i uwarunkowania prawne jego funkcjonowania.....	13
1.1.1 Definicja publicznego transportu zbiorowego.....	13
1.1.2 Uwarunkowania prawne organizacji publicznego transportu zbiorowego.....	13
1.2 Planowanie oferty publicznego transportu zbiorowego.....	15
1.2.1 Planowanie układu sieci publicznego transportu zbiorowego.....	15
1.2.2 Przygotowanie rozkładu jazdy.....	17
1.2.3 Planowanie pracy pojazdów.....	18
1.2.4 Planowanie pracy personelu.....	20
1.2.5 Sprzężenia zwrotne w planowaniu oferty publicznego transportu zbiorowego....	22
1.3 Synchronizacja rozkładów jazdy.....	23
1.3.1 Typy synchronizacji rozkładów jazdy.....	23
1.3.2 Kryteria efektywności synchronizacji rozkładów jazdy.....	27
1.4 Algorytmy genetyczne.....	30
1.4.1 Działanie algorytmów genetycznych.....	30
1.4.2 Metody selekcji osobników.....	31
1.4.3 Metody krzyżowania osobników.....	32
1.4.4 Sposoby mutacji osobników.....	33
1.4.5 Warunek kończący działanie algorytmu.....	34
1.4.6 Parametry algorytmów genetycznych.....	34
1.4.7 Wady i zalety algorytmów genetycznych.....	35
1.5 Przegląd literatury przedmiotu.....	36

1.5.1 Synchronizacja rozkładów jazdy przy użyciu algorytmów genetycznych.....	37
1.5.2 Synchronizacja rozkładów jazdy za pomocą programowania całkowitoliczbowego i jego wariantów	42
1.5.3 Synchronizacja rozkładów jazdy za pomocą symulowanego wyżarzania	46
1.5.4 Synchronizacja rozkładów jazdy za pomocą innych metod.....	47
1.6 Podsumowanie przeglądu literatury przedmiotu	50
2. Rozdział II.....	55
2.1 Model sieci publicznego transportu zbiorowego	55
2.2 Ustalenie zapotrzebowania na przesiadki pasażerów	57
2.3 Sformułowanie problemu synchronizacji rozkładu jazdy	58
2.4 Implementacja modelu sieci publicznego transportu zbiorowego.....	59
2.4.1 Pakiet <i>transportnet</i>	59
2.4.2 Pakiet <i>genetics</i>	64
2.4.3 Pakiet <i>stochastic</i>	66
2.4.4 Relacje w pakiecie <i>transportnet</i>	67
2.4.5 Relacje w pakiecie <i>genetics</i>	70
2.4.6 Relacje pomiędzy pakietami	72
3. Rozdział III.....	75
3.1 Charakterystyka sieci transportu publicznego	75
3.1.1 Obsługiwany obszar	75
3.1.2 Uwarunkowania prawne funkcjonowania systemu transportu zbiorowego w Zielonej Górze.....	78
3.1.3 Charakterystyka systemu transportu zbiorowego w Zielonej Górze	78
3.2 Model miasta Zielona Góra w programie PTV Visum.....	81
3.3 Część sieci wytypowana do eksperymentu.....	81
4. Rozdział IV	84
4.1 Przełożenie terminologii algorytmów genetycznych na zagadnienia transportowe	84
4.2 Eksperyment obliczeniowy	89
4.2.1 Plan eksperymentu	89

4.2.2 Przebieg eksperymentu obliczeniowego.....	90
4.2.3 Ocena wpływu parametru odpowiedzialnego za prawdopodobieństwo wystąpienia mutacji na wartość funkcji celu	116
4.3 Wpływ parametrów algorytmu genetycznego na czas wykonania obliczeń	117
4.3.1 Wpływ rozmiaru populacji na czas trwania obliczeń	118
4.3.2 Wpływ liczby generacji na czas trwania obliczeń	119
4.3.3 Wpływ prawdopodobieństwa mutacji na czas trwania obliczeń	120
4.3.4 Wpływ rozmiaru populacji wraz z liczbą generacji na czas trwania obliczeń	121
4.3.5 Podsumowanie analizy wpływu parametrów algorytmu na czas obliczeń.....	126
Podsumowanie	127
Wnioski z pracy.....	127
Kierunki dalszych badań	128
Bibliografia	130
Spis tabel	141
Spis rysunków	143

Wstęp

Transport publiczny stanowi jeden z kluczowych elementów infrastruktury miejskiej, odgrywając istotną rolę w zapewnianiu mobilności mieszkańców, ograniczaniu negatywnych skutków ruchu samochodowego oraz w dążeniu do zrównoważonego rozwoju miast. W miarę intensyfikacji urbanizacji, rosnących potrzeb komunikacyjnych i presji na ograniczanie emisji zanieczyszczeń, jakość i efektywność systemów transportu zbiorowego staje się istotnym czynnikiem decydującym o jakości życia w mieście.

Jednym z podstawowych wyzwań w organizacji transportu publicznego jest zaprojektowanie oferty przewozowej, która z jednej strony odpowiada popytowi na przewozy w przestrzeni i czasie, a z drugiej minimalizuje koszty operacyjne oraz zapewnia satysfakcjonujący poziom komfortu i niezawodności dla pasażerów. W tym kontekście synchronizacja rozkładów jazdy, czyli skoordynowanie godzin odjazdów i przyjazdów między kursami różnych linii publicznego transportu zbiorowego jest jednym z czynników składających się na atrakcyjność oferty publicznego transportu zbiorowego w miastach.

Synchronizacja rozkładów jazdy publicznego transportu zbiorowego jest złożonym problemem. Wymaga kompromisu między wieloma celami: minimalizacją czasu oczekiwania pasażera, ograniczeniem opóźnień i ryzyka utraty możliwości przesiadki między pojazdami oraz uwzględnieniem ograniczeń technicznych i operacyjnych (m.in. prędkość pojazdów, czasy przejazdów, ograniczenia obsługi taborowej i personelu). Problem synchronizacji rozkładów jazdy jest zaliczany do problemów NP-trudnych, a liczba możliwych rozwiązań rośnie wykładniczo wraz z dodaniem każdej kolejnej linii, której rozkład ma być zsynchronizowany.

Z tego względu konieczne staje się poszukiwanie metod optymalizacyjnych, które pozwolą w rozsądnym czasie znaleźć rozwiązania bliskie optymalnym. Jednym z podejść szczególnie obiecujących w tym zakresie są algorytmy genetyczne, należące do grupy metaheurystyk inspirowanych procesami ewolucyjnymi. Ich zdolność do przeszukiwania dużych przestrzeni rozwiązań oraz elastyczność w dostosowywaniu do specyfiki badanego problemu sprawiają, że stanowią one wartościowe narzędzie wspierające proces synchronizacji rozkładów jazdy w systemach publicznego transportu zbiorowego.

Aktualność pracy

Problematyka synchronizacji rozkładów jazdy jest jednym z istotnych nurtów badań w obszarze planowania publicznego transportu zbiorowego, co potwierdzają liczne prace empiryczne i przeglądowe. Badania wykazały, że synchronizacja rozkładów jazdy może znacząco zredukować czas oczekiwania pasażerów i poprawić jakość przesiadek, co z wpływa na zwiększenie atrakcyjności komunikacji zbiorowej wobec transportu indywidualnego. W literaturze metaheurystyki, w szczególności algorytmy genetyczne, są wskazywane jako efektywne narzędzie do rozwiązywania problemów synchronizacyjnych o dużej złożoności.

Prace przeglądowe i badania porównawcze dostarczają dowodów na to, że algorytmy genetyczne są obiecującą metodą, gdy celem jest szybkie znalezienie rozwiązań dobrych jakościowo dla problemu o realistycznej skali. Jednocześnie literatura wskazuje na ograniczenia m.in. w postaci konieczności starannej kalibracji parametrów algorytmu która wymaga systematycznych badań eksperymentalnych.

Mając na uwadze powyższe uwarunkowania, niniejsza praca wpisuje się w aktualny nurt badań, podejmując problem synchronizacji rozkładów z perspektywy zarówno teoretycznej, jak i praktycznej implementacji. Wybór algorytmu genetycznego uzasadniony jest jego zdolnością do eksploracji złożonych przestrzeni rozwiązań oraz elastycznością w modelowaniu ograniczeń specyficznych dla sieci miejskich.

Cel i zadania

W literaturze naukowej, dotyczącej synchronizacji rozkładów jazdy transportu publicznego za pomocą algorytmów genetycznych, widoczna jest luka badawcza związana z parametryzacją algorytmu. W wielu pracach badacze nie podają z jakimi parametrami wykorzystywali algorytm genetyczny. Z kolei w tych pracach, w których to zrobili, brakuje argumentacji uzasadniającej wykorzystanie konkretnego zestawu parametrów. Ponieważ parametry algorytmu genetycznego wpływają na czas, w jakim znalezione zostanie rozwiązanie, bardzo istotnym z punktu widzenia praktycznego zastosowania tej heurystyki jest odpowiednie dobranie parametrów algorytmu.

Celem pracy jest znalezienie uzasadnienie wartości parametrów algorytmu genetycznego synchronizującego rozkłady jazdy. Weryfikację skuteczności stosowania parametrów algorytmu genetycznego w danej pracy wykonano na przykładzie synchronizacji rozkładów jazdy dla wybranych linii publicznego transportu zbiorowego miasta Zielona Góra.

Jako podstawowe **zadania badawcze**, niezbędne do osiągnięcia tak postawionego celu badania, w niniejszej pracy wyodrębniono:

- przegląd współczesnej literatury naukowej poświęconej tematyce synchronizacji rozkładów jazdy transportu publicznego,
- implementacja modelu symulacyjnego sieci transportu publicznego (dla wybranej części systemu transportu zbiorowego Zielonej Góry) oraz opracowanie metody synchronizacji rozkładu jazdy w sieci za pomocą algorytmu genetycznego,
- zaplanowanie i przeprowadzenie eksperymentu obliczeniowego w celu zbadania wpływu stosowanych parametrów algorytmu genetycznego na efektywność procedury synchronizacji rozkładów jazdy,
- wykonanie analizy rezultatów eksperymentu obliczeniowego oraz uzasadnienie wartości parametrów algorytmu genetycznego.

Struktura pracy

Niniejsza praca składa się z czterech rozdziałów.

Rozdział pierwszy zawiera definicję publicznego transportu zbiorowego oraz prezentuje prawne uwarunkowania jego organizacji i funkcjonowania w Polsce. Kolejne podrozdziały omawiają proces planowania oferty przewozowej i mechanizmy sprzężeń zwrotnych zachodzące podczas planowania. Opisano problem synchronizacji rozkładów jazdy, który jest analizowany z perspektywy różnych podejść do problemu synchronizacji i kryteriów efektywności. W dalszej części przedstawione są podstawy teoretyczne algorytmów genetycznych, w tym metody selekcji, krzyżowania, mutacji oraz parametry wpływające na ich działanie. Rozdział kończy szeroki przegląd literatury przedmiotu, obejmujący zarówno zastosowanie algorytmów genetycznych, jak i innych metod optymalizacyjnych (programowanie całkowitoliczbowe, symulowane wyżarzanie, metody heurystyczne) w synchronizacji rozkładów jazdy.

W rozdziale drugim opisano wykorzystaną implementację algorytmu genetycznego, obejmującej pakiety *transportnet*, *genetics* oraz *stochastic*. Omówione zostają pakiety, klasy i wykorzystywane metody oraz zachodzące pomiędzy nimi zależności.

W rozdziale trzecim skoncentrowano się na opisie systemu transportu publicznego w Zielonej Górze. Zostają przedstawione ogólne uwarunkowania prawne i organizacyjne funkcjonowania transportu publicznego w mieście oraz charakterystyka samej sieci

transportowej. Opisano również model symulacyjny miasta Zielona Góra stworzony w programie PTV Visum, który stanowił źródło danych do przeprowadzenia eksperymentu. W ostatniej części rozdziału przedstawiono fragment sieci komunikacyjnej, który posłużył jako materiał empiryczny do przeprowadzenia eksperymentu obliczeniowego.

Rozdział czwarty obejmuje właściwe badania eksperymentalne. Przedstawiono przełożenie terminologii algorytmów genetycznych na zagadnienia transportowe. Następnie szczegółowo opisano plan i przebieg eksperymentu obliczeniowego, w tym ocenę wpływu parametrów algorytmu na wartość funkcji celu. Kolejne podrozdziały analizują wpływ poszczególnych parametrów (rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji) na czas trwania obliczeń, co pozwala na określenie kompromisu pomiędzy ich efektywnością a jakością uzyskanych rozwiązań.

W końcowej części opisano podsumowanie eksperymentu obliczeniowego, przedstawiono wnioski, a także przedstawiono kierunki dalszych badań. Na końcu pracy znajduje się licząca 130 pozycji bibliografia, a także spis tabel i rysunków.

1. Rozdział I

1.1 Publiczny transport zbiorowy – definicja i uwarunkowania prawne jego funkcjonowania

1.1.1 Definicja publicznego transportu zbiorowego

Publiczny transport zbiorowy to powszechnie dostępny regularny przewóz osób wykonywany w określonych odstępach czasu i po określonej linii komunikacyjnej, liniach komunikacyjnych lub sieci komunikacyjnej (Ustawa z dnia 16 grudnia 2010 r. o publicznym transporcie zbiorowym, 2010). Istotą publicznego transportu zbiorowego jest zaspokajanie potrzeb przewozowych osób znajdujących się na obszarze, na którym jest on organizowany. Potrzeby te wiążą się ze społeczną i produkcyjną działalnością człowieka (Rydzykowski & Wojewódzka-Król, 2010). Przyczynami ich powstawania są m.in.:

- wykonywanie pracy,
- edukacja,
- korzystanie ze świadczeń medycznych,
- utrzymywanie więzi społecznych,
- zaspokajanie potrzeb kulturalnych,
- różne formy spędzania wolnego czasu.

1.1.2 Uwarunkowania prawne organizacji publicznego transportu zbiorowego

Uwarunkowania prawne organizacji publicznego transportu zbiorowego zostały opisane w Rozporządzeniu (WE) nr 1370/2007 Parlamentu Europejskiego i Rady dotyczącym usług publicznych w zakresie kolejowego i drogowego transportu pasażerskiego.

Aktem prawnym, który wdraża treść Rozporządzenia (WE) nr 1370/2007 w Polsce jest Ustawa o publicznym transporcie zbiorowym z dnia 16 grudnia 2010 r.

Zgodnie z Ustawą o publicznym transporcie zbiorowym z 2010 r. jego organizatorem może być:

- gmina,
- związek międzygminny,
- powiat,
- związek powiatów,

- związek powiatowo-gminny,
- związek metropolitalny,
- województwo,
- minister właściwy do spraw transportu.

Zgodnie z art. 8. wyżej wymienionej ustawy do głównych zadań organizatora należy:

- organizowanie publicznego transportu zbiorowego,
- zarządzanie publicznym transportem zbiorowym,
- planowanie rozwoju transportu.

Z kolei artykuł 15. ustawy definiuje pojęcie organizacji publicznego transportu zbiorowego i zgodnie z jego treścią organizacja polega m.in na:

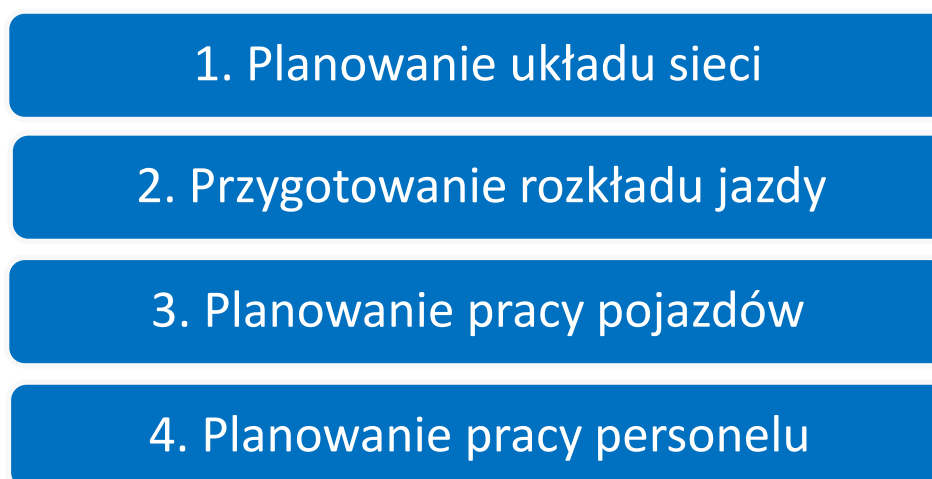
- badaniu i analizie potrzeb przewozowych w publicznym transporcie zbiorowym z uwzględnieniem osób niepełnosprawnych i osób o ograniczonej zdolności ruchowej,
- podejmowaniu działań zmierzających do realizacji istniejącego planu transportowego albo do aktualizacji tego planu,
- zapewnianiu odpowiednich warunków funkcjonowania publicznego transportu zbiorowego,
- określaniu sposobu oznakowania środków transportu wykorzystywanych w przewozach o charakterze użyteczności publicznej,
- ustalaniu stawek za korzystanie z przystanków i dworców przez operatorów i przewoźników,
- przygotowywaniu, przeprowadzaniu postępowania do zawarcia oraz zawieraniu umów o świadczenie usług w zakresie publicznego transportu zbiorowego,
- ustalaniu wysokości opłat za przewóz oraz innych opłat,
- ustalaniu sposobu dystrybucji biletów za usługę świadczoną przez operatora publicznego transportu zbiorowego.

Według Ustawy o publicznym transporcie zbiorowym operatorem publicznego transportu zbiorowego jest samorządowy zakład budżetowy oraz przedsiębiorca uprawniony do prowadzenia działalności gospodarczej w zakresie przewozu osób, który zawarł z organizatorem publicznego transportu zbiorowego umowę o świadczenie usług w zakresie publicznego transportu zbiorowego, na linii komunikacyjnej określonej w umowie.

Sposób wyboru operatora szczegółowo opisano w artykule 19. Ustawy o publicznym transporcie zbiorowym. Organizator dokonuje wyboru w trybie ustawy z dnia 11 września 2019 r. – Prawo zamówień publicznych lub w wyszczególnionych przypadkach na podstawie bezpośrednio zawartej umowy.

1.2 Planowanie oferty publicznego transportu zbiorowego

Planowanie oferty publicznego transportu zbiorowego jest wieloetapowym procesem. Ograniczone środki finansowe jednostek administracyjnych odpowiedzialnych za jego organizację wymuszają kompromis pomiędzy wydatkowanymi środkami a zaspokajaniem potrzeb pasażerów. Uproszczony schemat planowania oferty przewozowej zaprezentowano na rys. 1.



Rysunek 1. Uproszczony schemat planowania oferty przewozowej
Źródło: Opracowanie własne na podstawie (Ceder, 2007)

Zgodnie z porządkiem zaprezentowanym na rys. 1, w pierwszej kolejności należy zaplanować trasy linii obsługujących wybrany obszar. W drugim kroku ustala się rozkład jazdy zaplanowanych wcześniej linii. W trzecim zaplanowane kursy przydziela się do pojazdów, które mają je wykonywać. W czwartym planuje się pracę personelu obsługującego pojazdy.

1.2.1 Planowanie układu sieci publicznego transportu zbiorowego

Wydajna siatka publicznego transportu zbiorowego powinna być zaplanowana w taki sposób, aby w możliwie największym stopniu pokryć istniejący popyt na podróże w planowanym rejonie w ramach możliwości finansowych jego organizatora (Vuchic, 2005).

Punktem wyjścia planowania sieci powinny być badania dotyczące mobilności mieszkańców analizowanego obszaru (Mohammed & Jimi, 2023), dostarczające informacji o liczbie podróży, motywacji, czasie podróży, miejscach rozpoczęcia i zakończenia podróży oraz wybieranym środku transportu.

Dane o potokach pasażerskich, oszacowane na podstawie badań mobilności mieszkańców, są wykorzystywane do wytyczenia relacji, które będą obsługiwane w formie bezpośredniej, a które będą wymagały przesiadki. W celu umożliwienia przesiadek wyznacza się węzły przesiadkowe.

Ważnym aspektem jest również przyjęcie bazowego interwału czasowego z jakim będą realizowane kursy w planowanej sieci. Najczęściej z interwałem bazowym wykonywane są kursy w ramach linii tworzących szkielet sieci, a kursy w ramach pozostałych linii są wykonywane z jego wielokrotnością (Tzieropoulos, et al., 2010). To zagadnienie zostało przedstawione szerzej w podrozdziale 1.2.2.

Czynnikami wpływającymi na przebieg linii są także:

- infrastruktura punktowa i liniowa (Lu, et al., 2024),
- typ posiadanego taboru (Żak, 2011),
- zapewnienie odpowiedniego standardu obsługi (Zhang, et al., 2022).

Przebieg linii powinien być dostosowany do infrastruktury obsługiwanego obszaru i typu taboru jakim będzie ona obsługiwana. Trasa linii, która ma charakteryzować się dużymi potokami pasażerskimi powinna być obsługiwana przez większe pojazdy (np. przegubowe). Jednakże takie pojazdy mogą mieć problemy w poruszaniu się przez ciasne uliczki centrów miast. Takie miejsca powinny być obsługiwane przez mniejsze, ale kursujące częściej pojazdy. Ekonomicznie nieuzasadnione jest także wysyłanie taboru wielkopojemnego na obrzeża miast charakteryzujące się niską gęstością zaludnienia (Vansteenwegen, et al., 2022).

Jedną ze składowych standardu obsługi pasażerskiej jest zapewnienie minimalnej liczby odjazdów w ciągu godziny, nawet na obszarach o niskiej gęstości zaludnienia. W celu zapewnienia go i ograniczeniu kosztów ponoszonych przez organizatora należy kierować do ich obsługi możliwie jak najmniejsze pojazdy (Hassold & Ceder, 2012). Trasy linii obsługujące takie rejony należy kończyć przy najbliższym węźle przesiadkowym (Hassold & Ceder, 2014).

1.2.2 Przygotowanie rozkładu jazdy

Kolejnym krokiem w planowaniu oferty publicznego transportu zbiorowego jest stworzenie rozkładu jazdy. Na tym etapie ustala się czasy przejazdów pomiędzy przystankami i podejmuje decyzje czy mają być one zmienne w zależności od pory lub dnia tygodnia. W tym kroku określa się również godziny obsługi sieci przez każdą z linii.

Następna czynność to ułożenie szczegółowego rozkładu jazdy dla każdej z linii. W celu zwiększenia atrakcyjności oferty publicznego transportu zbiorowego rozkład ten powinien uwzględniać zarówno koordynację odjazdów w ciągach komunikacyjnych jak i synchronizację w węzłach przesiadkowych pozwalającą na przesiadki.

Skoordynowanym rozkładem jazdy jest taki rozkład, który zapewnia równe odstępy pomiędzy kursami różnych linii poruszających się po wspólnym ciągu komunikacyjnym (Madej, et al., 2017). W celu koordynacji rozkładu jazdy w ciągu komunikacyjnym koniecznym jest ustalenie wspólnego interwału odjazdów skoordynowanych linii. Aby go ustalić w zadanym przedziale czasowym należy podzielić ten przedział czasowy przez sumę liczbę wszystkich kursów na danym ciągu. Następnie należy tak ułożyć rozkłady jazdy branych pod uwagę w koordynacji linii, aby realizowały kursy z zadanym interwałem. W tabeli 1 jako przykład zaprezentowano liczbę kursów i interwały odjazdów trzech linii obsługujących wspólny ciąg komunikacyjny.

Tabela 1. Liczba kursów rozpatrywanych oddzielnie dla każdej linii i wspólnie w ciągu komunikacyjnym

	Indywidualnie		Na wspólnym ciągu komunikacyjnym	
	Interwał	Liczba kursów/h	Interwał	Liczba kursów/h
Linia A	10 min	6	5 min	12
Linia B	20 min	3		
Linia C	20 min	3		

Źródło: Opracowanie własne

Jak wynika z tabeli 1. linia A wykonuje kursy co 10 minut, co przekłada się na 6 kursów w ciągu godziny. Linie B i C wykonują kursy z dwukrotnością interwału f czyli co 20 minut. Na wspólnym odcinku wykonują łącznie 12 kursów w ciągu godziny, co przekłada się na 5-minutowy wspólny interwał odjazdów.

Fragment przykładowego skoordynowanego rozkładu jazdy wyżej wymienionych linii zaprezentowano w tabeli 2.

Tabela 2. Fragment przykładowego, skoordynowanego rozkładu jazdy na przystanku X

Godzina odjazdu	Linia	Czas od poprzedniego odjazdu z przystanku X	Czas od poprzedniego odjazdu tej samej linii z przystanku X
10:00	A	-	-
10:05	B	5 minut	-
10:10	A	5 minut	10 minut
10:15	C	5 minut	-
10:20	A	5 minut	10 minut
10:25	B	5 minut	20 minut
10:30	A	5 minut	10 minut
10:35	C	5 minut	20 minut
10:40	A	5 minut	10 minut
10:45	B	5 minut	20 minut
10:50	A	5 minut	10 minut
10:55	C	5 minut	20 minut

Źródło: Opracowanie własne

Jak zaprezentowano w tabeli 2, linia A wykonuje kursy co 10 minut, począwszy od 10:00. Linie B i C wykonują kursy naprzemiennie co 10 minut (każda z nich co 20 minut) 5 minut po odjeździe linii A. Oznacza to, że również 5 minut przed kursem linii A z przystanku X odjedzie pojazd linii B lub C.

Odjazdy planowane w równych odstępach czasu są łatwiejsze do przyswojenia i zapamiętania przez pasażerów co przekłada się na wzrost atrakcyjności oferty przewozowej.

Synchronizacja rozkładów jazdy w węzłach przesiadkowych została opisana w podrozdziale 1.3.

1.2.3 Planowanie pracy pojazdów

Trzeci etap planowania oferty to zaplanowanie pracy pojazdów. Mając ustalony już interwał odjazdów dla każdej z linii i czas przejazdu pomiędzy przystankami początkowymi i końcowymi w obu kierunkach można obliczyć teoretyczną liczbę pojazdów (P) potrzebną do obsługi każdej z linii, korzystając ze wzoru:

$$P \approx T/I \uparrow$$

(1)

gdzie:

P – liczba pojazdów potrzebnych do obsługi linii,

T – suma czasów jazdy w obu kierunkach,

I – założony interwał kursowania wybranej linii.

Czas T możemy przedstawić jako:

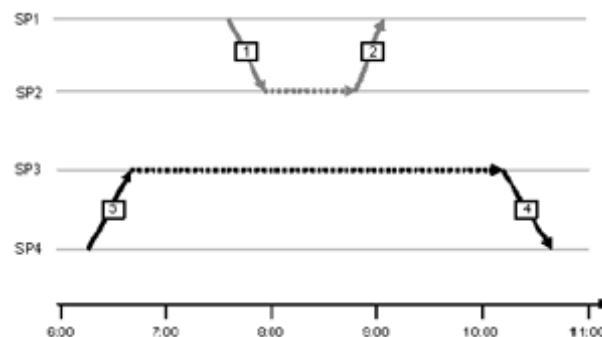
$$T = T_1 + T_2 \quad (2)$$

gdzie:

T_1 – czas jazdy wyrażony w minutach w jednym kierunku,

T_2 – czas jazdy wyrażony w minutach w kierunku powrotnym.

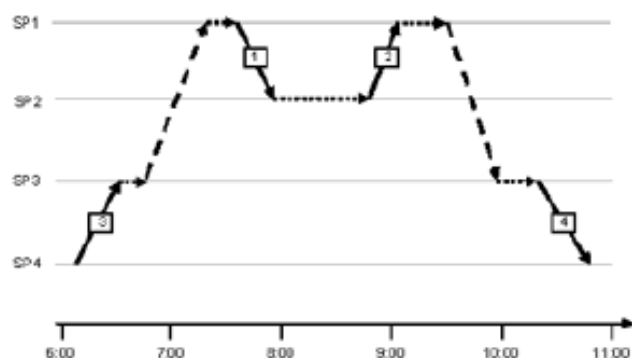
W celu usystematyzowania dalszych kroków, w tym planowania pracy personelu, kursy należy przypisać do konkretnych zadań przewozowych obsługiwanych przez poszczególne pojazdy. W takich zadaniach bardzo często pomocna okazuje się wizualizacja w postaci wykresu ruchu, w którym na osi X przedstawiony jest czas, a na osi Y kolejne lub wybrane przystanki na trasie linii. Przykład wykresu ruchu dla dwóch linii został zaprezentowany na rys. 2.



Rysunek 2. Przykład wykresu ruchu dwóch linii obsługiwanych przez dwa pojazdy
Źródło: (PTV, 2025)

Pierwsza z linii pokonuje trasę między przystankami SP1 i SP2, druga między przystankami SP3 i SP4. Łącznie wykonywane są cztery kursy, oznaczone linią ciągłą zakończoną strzałką. Linia przerywana symbolizuje przerwę. Pierwszy pojazd wykonuje kurs z przystanku SP1 do SP2, a następnie po przerwie kurs w odwrotnym kierunku. Drugi pojazd realizuje kurs od przystanku SP4 do SP3, a następnie pomiędzy przystankami SP3 i SP4. W takim wypadku do obsługi czterech zaprezentowanych kursów potrzebne są dwa pojazdy.

Dodawanie kolejnych, możliwych do wykonania, kursów w ramach tej samej linii do planu pracy pojazdu to najprostszy sposób planowania pracy pojazdów. Przestaje być on jednak efektywny już w przypadku niewielkich sieci. Wykorzystanie pojazdu do realizacji kursów na więcej niż jednej linii przedstawiono na rys. 3.



Rysunek 3. Przykład wykresu ruchu dwóch linii obsługiwanych przez jeden pojazd
Źródło: (PTV, 2025)

Jak pokazano na rys. 3, ten sam pojazd może realizować kursy różnych linii. Najpierw wykonuje kurs z przystanku SP4 do SP3. Następnie (przerywana linia zakończona strzałką) wykonuje kurs techniczny (bez pasażerów) z przystanku SP3 do SP1. Z przystanku SP1 wykonuje kurs do przystanku SP2, a następnie po przerwie (pozioma przerywana linia) wraca do przystanku SP1. Po kolejnej przerwie wykonuje kolejny kurs techniczny do przystanku SP3, z którego wykonuje kurs z pasażerami do przystanku SP4. Dodanie przejazdów technicznych i zrezygnowanie z zasady przypisania jednego do pojazdu tylko do jednej linii w czasie trwania dnia pozwoliło na zredukowanie potrzebnej liczby pojazdów z dwóch do jednego.

Problem przydziału pojazdów do kursów komplikuje się wraz ze wzrostem liczby kursów w obsługiwanej sieci, liczby zajezdni, ograniczeniami infrastruktury (np. ograniczoną liczbą miejsc postojowych na przystankach końcowych) czy też dopuszczeniem do obsługi wybranych kursów więcej niż jednego typu pojazdu.

1.2.4 Planowanie pracy personelu

Planowanie pracy personelu jest istotne z punktu widzenia operatora publicznego transportu zbiorowego. Pozwala na dobranie odpowiedniej liczby zatrudnionych prowadzących pojazdy, co bezpośrednio przekłada się na koszty operacyjne.

Na planowanie pracy personelu wpływają ograniczenia w postaci:

- miejsc, w których prowadzący mogą rozpoczynać i kończyć pracę (Zhao, 2006),
- ustawowo określonych norm czasu pracy,
- dodatkowych ustaleń wewnętrznych.

Choć teoretycznie prowadzący pojazd mogą zakończyć pracę i przekazać go kolejnemu prowadzącemu w dowolnym miejscu, w praktyce nie stosuje się takiego podejścia. Najczęściej takich zmian dokonuje się na przystankach początkowych/końcowych. Jeśli operator dopuszcza do zmian prowadzących w czasie trwania kursu najczęściej odbywają się one na przystankach w centrach miast, a nie na ich obrzeżach.

Ograniczenia ustawowe związane z czasem pracy kierowców dotyczą obowiązkowych przerw jakie muszą odbyć kierowcy w zależności od ich czasu pracy i czasu prowadzenia pojazdu. Zostały one przedstawione w tabeli 3.

Tabela 3. Obowiązkowe przerwy dla prowadzących w publicznym transporcie zbiorowym

Czas pracy	Czas prowadzenia pojazdu	Wymagana przerwa podczas pracy
Do 6 godzin	-	Brak wymaganej przerwy
Powyżej 6 godzin	Do 6 godzin	15 minut
Powyżej 6 godzin	Od 6 do 8 godzin	30 minut
Powyżej 6 godzin	Powyżej 8 godzin	45 minut

Źródło: Opracowanie własne na podstawie Ustawy z dnia 16 kwietnia 2004 r. o czasie pracy kierowców

Zgodnie z danymi przedstawionymi w tabeli 3. jeśli czas pracy prowadzącego jest krótszy niż sześć godzin (niezależnie od czasu prowadzenia pojazdu) nie przysługuje mu przerwa. Gdy czas pracy jest dłuższy niż 6 godzin, lecz łączny czas prowadzenia pojazdu jest krótszy niż 6 godzin prowadzącemu przysługuje 15 minut przerwy. Gdy czas prowadzenia pojazdu mieści się w przedziale od 6 do 8 godzin minimalny czas przerwy wynosi 30 minut, a jeśli jest dłuższy od 8 godzin minimalny czas przerwy to 45 minut. Przerwy te można dzielić na okresy nie krótsze niż 15 minut.

Wewnątrz zarządzenia każdego operatora publicznego transportu zbiorowego mogą narzucać kolejne ograniczenia w pracy personelu. Najczęściej jest to ograniczenie czasu pracy prowadzącego do 10 lub 11 godzin w ciągu doby.

1.2.5 Sprzężenia zwrotne w planowaniu oferty publicznego transportu zbiorowego

Chociaż planowanie oferty publicznego transportu zbiorowego odbywa się w określonej kolejności, to pomiędzy poszczególnymi etapami zachodzą sprzężenia zwrotne, które wpływają na cały proces i wymuszają korektę decyzji podjętych na wcześniejszych etapach. Zostało to przedstawione na rys. 4.



Rysunek 4. Sprzężenia zwrotne w planowaniu oferty publicznego transportu zbiorowego
Źródło: Opracowanie własne

Jeżeli na etapie podziału zadań przewozowych na porcje pracy kierowców okazują się, że niemożliwe jest na przykład spełnienie wymogów ustawy o czasie pracy kierowców lub wewnętrznych ustaleń przedsiębiorstwa należy powrócić do poprzedniego kroku. Rozpocząć proces przydzielania kursów do zadań przewozowych od nowa i ponownie sprawdzić porcje pracy dla prowadzących pod kątem przyjętych założeń.

W przypadku gdy na etapie trzecim okazuje się, że do obsługi zaplanowanego rozkładu jazdy wymagana jest większa liczba pojazdów niż ta którą dysponuje organizator należy zmodyfikować rozkład jazdy. W niektórych przypadkach wystarczającym działaniem będzie przesunięcie godziny odjazdu jednego bądź kilku kursów o kilka minut. Jednak te czynności mogą zaburzyć wykonaną wcześniej koordynację w ciągach komunikacyjnych i synchronizację w węzłach przesiadkowych. Jeśli nadal nie można uzyskać oczekiwanego rozwiązania należy zmienić godziny kursowania linii i przydzielony im interwał wykonywania kursów. W skrajnych przypadkach koniecznym może okazać się cofnięcie aż do pierwszego etapu i modyfikacja układu sieci.

1.3 Synchronizacja rozkładów jazdy

Synchronizacja rozkładów jazdy w systemach publicznego transportu zbiorowego ma na celu umożliwić pasażerom przesiadkę pomiędzy pojazdami obsługującymi różne linie z możliwie minimalnym czasem oczekiwania (Ceder, 2016). Jest to jedno z najtrudniejszych zadań związanych z planowaniem efektywnej sieci publicznego transportu zbiorowego (Ceder, 2007). Z matematycznego punktu widzenia znalezienie optymalnych czasów odjazdu poszczególnych linii jest problemem NP-trudnym (Ibarra-Rojas & Rios-Solis, 2012).

Rozkłady jazdy uwzględniające możliwość przesiadania się w wybranych węzłach wprowadzone zostały przez koleje niderlandzkie już przed drugą wojną światową. Ówczesni planiści polegać mogli jedynie na metodach heurystycznych (Jänsch & Reinhold, 1990). Pierwszy zsynchronizowany za pomocą metod matematycznych rozkład jazdy pojazdów publicznego transportu zbiorowego został wprowadzony 12 grudnia 2004 roku na sieci berlińskiego metra. Implementacja tak przygotowanego rozkładu pozwoliła na zwiększenie liczby relacji, w których czas oczekiwania na przesiadkę wynosił poniżej pięciu minut (Liebchen, 2008). Koleje Niderlandzkie w 2006 roku wprowadziły na swojej sieci zoptymalizowany rozkład jazdy pod kątem zmniejszenia oczekiwania czasu na przesiadkę, zapewnienia powtarzalnych godzin odjazdów, a także minimalizacji potrzebnego do jego obsługi personelu i taboru (Kron, et al., 2009).

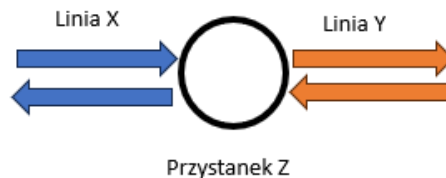
Ze względu na ograniczone środki finansowe organizatorów transportu w miastach i aglomeracjach oraz ich powierzchnie nie jest możliwe zapewnienie bezpośrednich połączeń pomiędzy wszystkimi ich rejonami. Organizatorzy są więc zmuszeni do zachowania równowagi pomiędzy oferowaną podażą a wydatkowanymi środkami. Liczba punktów, w których przesiadają się pasażerowie zależy od powierzchni obsługiwanego obszaru, jednak już w średniej wielkości miastach wyznaczenie tras wszystkich linii tak by miały chociaż jeden wspólny dla nich wszystkich przystanek nie jest możliwe.

1.3.1 Typy synchronizacji rozkładów jazdy

W literaturze można spotkać się z pracami dotyczącymi synchronizacji rozkładów jazdy w różnej skali od synchronizacji rozkładu jazdy kilku linii na jednym przystanku po synchronizację w całej sieci publicznego transportu zbiorowego.

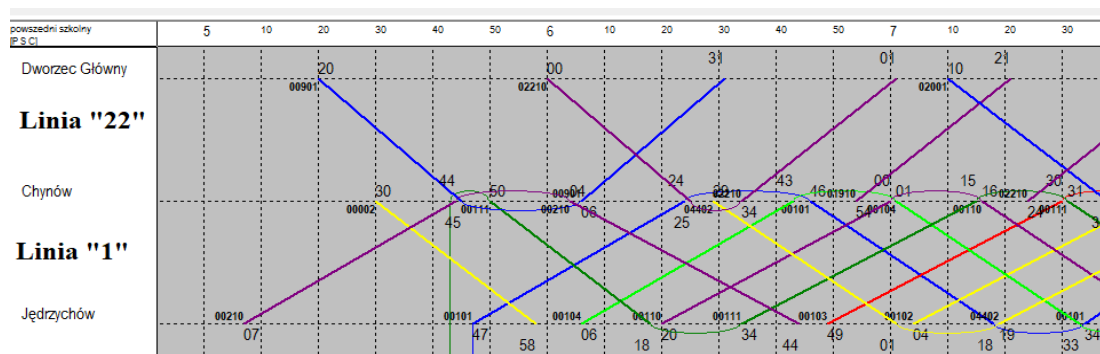
1.3.1.1 Synchronizacja rozkładu jazdy dwóch linii

Synchronizacja rozkładu jazdy dwóch linii polega na takim dopasowaniu godzin odjazdów i przyjazdów na wspólnym przystanku, w taki sposób, aby pasażerowie mogli przesiąść się do drugiego pojazdu w akceptowalnym dla nich czasie. Uproszczony schemat synchronizacji został przedstawiony na rys. 5.

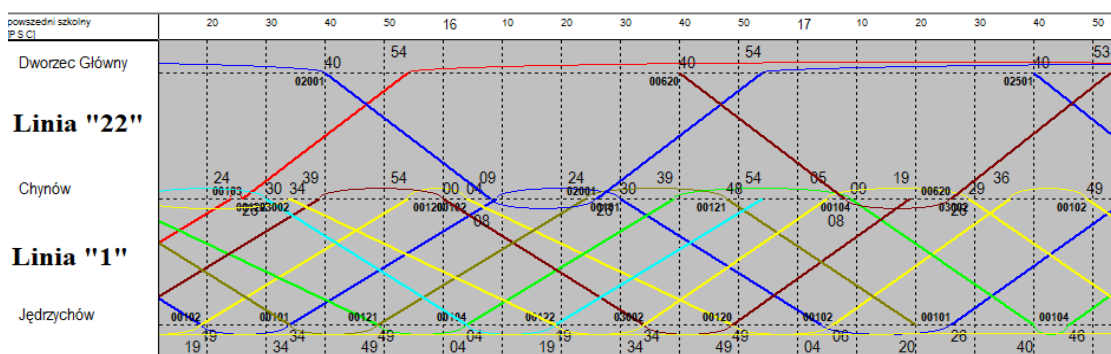


Rysunek 5. Uproszczony schemat synchronizacji rozkładu jazdy na jednym przystanku
Źródło: Opracowanie własne

Jak zaprezentowano na rys. 5, synchronizacja rozkładów jazdy linii X i Z występuje na przystanku końcowo-początkowym Z. Obie linie rozpoczynają tu i kończą swoje trasy. Linia X w szczycie porannym dowozi pasażerów do linii Y, którzy podróżują dalej w kierunku centrum miasta. W szczycie popołudniowym kierunek podróży pasażerów jest odwrócony. Przykład synchronizacji rozkładu dla dwóch linii na jednym przystanku pokazują rys. 6 i 7.



Rysunek 6. Synchronizacja rozkładów jazdy dwóch linii na jednym przystanku w godzinach szczytu porannego.
Źródło: (Dane Miejskiego Zakładu Komunikacji w Zielonej Górze, 2023)

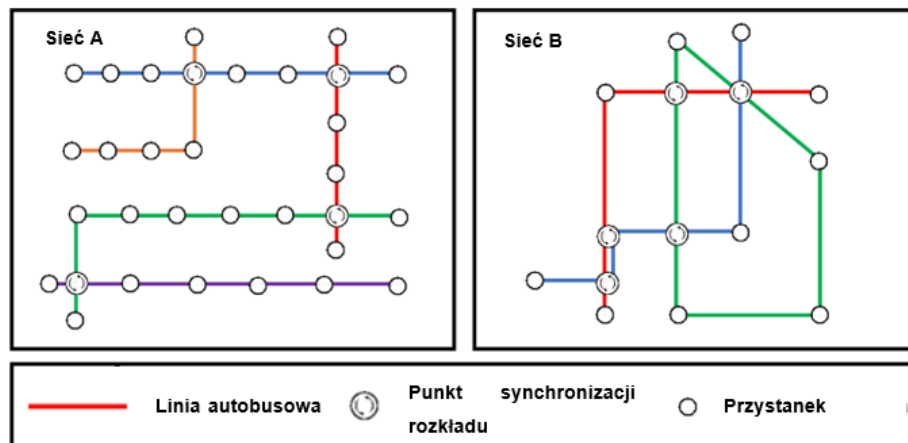


Rysunek 7. Synchronizacja rozkładów jazdy dwóch linii na jednym przystanku w godzinach szczytu popołudniowego.
Źródło: (Dane Miejskiego Zakładu Komunikacji w Zielonej Górze, 2023)

Na rys. 6 i 7 zaprezentowano rozkład jazdy w formie wykresu ruchu. Na rys. 6 zaprezentowano rozkład jazdy w szczycie porannym, kiedy to linia „22” dowozi pasażerów do linii „1”. Autobusy tej drugiej linii odjeżdżają z przystanku kilka minut po dotarciu pasażerów pierwszą linią na wspólny przystanek. Rys. 7 przedstawia rozkład jazdy w czasie popołudniowego szczytu. Autobusy linii „1” dowożą pasażerów do linii „22” przyjeżdżając na wspólny przystanek na kilka minut przed jej odjazdem.

1.3.1.2 Synchronizacja rozkładów jazdy kilku linii

Osobnym problemem jest równoczesna synchronizacja rozkładów jazdy kilku linii. Najczęściej takiej synchronizacji dokonuje się dla kilku wybranych linii obsługujących daną sieć. Uproszczony schemat synchronizacji rozkładów jazdy kilku linii autobusowych zaprezentowano na rys. 8.



Rysunek 8. Uproszczony schemat synchronizacji kilku linii autobusowych w sieci publicznego transportu zbiorowego

Źródło: Opracowanie własne na podstawie (Wang, et al., 2022)

Na rys. 8 zaprezentowano dwie przykładowe sieci A i B. W sieci A do synchronizacji wybrano cztery linie (oznaczone różnymi kolorami). Trasa każdej linii krzyżuje się z trasą co najmniej jednej innej linii w jednym określonym punkcie (przystanku). W sieci B do synchronizacji wybrano trzy linie. W tym przypadku trasy wyselekcjonowanych linii mają co najmniej dwa wspólne punkty (przystanki).

Synchronizacja rozkładów jazdy kilku linii jest trudniejsza od synchronizacji w jednym punkcie. Podczas jej wykonywania należy pamiętać, że ustalone godziny odjazdów jednej linii wpływają na godziny odjazdów pozostałych. Dodatkowo w przypadku sieci B, oprócz powyższego, należy najpierw zdecydować, w których wspólnych punktach rozkłady będą synchronizowane.

1.3.1.3 Synchronizacja rozkładów jazdy w całej sieci publicznego transportu zbiorowego

Synchronizacja rozkładów jazdy w całej sieci publicznego transportu zbiorowego jest rozszerzoną wersją synchronizacji rozkładów kilku linii.

Do synchronizacji rozkładów jazdy potrzebne są dane o interwale z jakim będą realizowane kursy na liniach, ogólnych godzinach kursowania każdej linii, czasach przejazdu między przystankami oraz o miejscach źródłowych i docelowych wykonywanych podróży. Na podstawie tych danych wybiera się linie z sieci, których rozkłady należy zsynchronizować.

Kolejnym etapem jest wybranie przystanków, na których rozkłady jazdy mają być ze sobą zsynchronizowane. W sytuacji, gdy trasy wybranych linii przecinają się w jednym punkcie wybór jest oczywisty. W sytuacji, w której wybrane linie mają więcej niż jeden wspólny punkt najczęściej wybiera się jeden z nich. Należy zdecydować o kolejności w jakiej będzie dokonywana synchronizacja rozkładów jazdy wybranych linii. W tym celu powstały modele służące do określenia, od którego punktu sieci zacząć synchronizację. Prostsze modele biorą pod uwagę tylko liczbę odjazdów oraz liczbę pasażerów przesiadających się pomiędzy liniami na danym przystanku. Z kolei te bardziej skomplikowane uwzględniają kierunek, w którym podróżują pasażerowie i łączą linie odjeżdżające w tym samym kierunku w grupy (Yap, et al., 2019).

1.3.1.4 Synchronizacja pierwszych i/lub ostatnich kursów

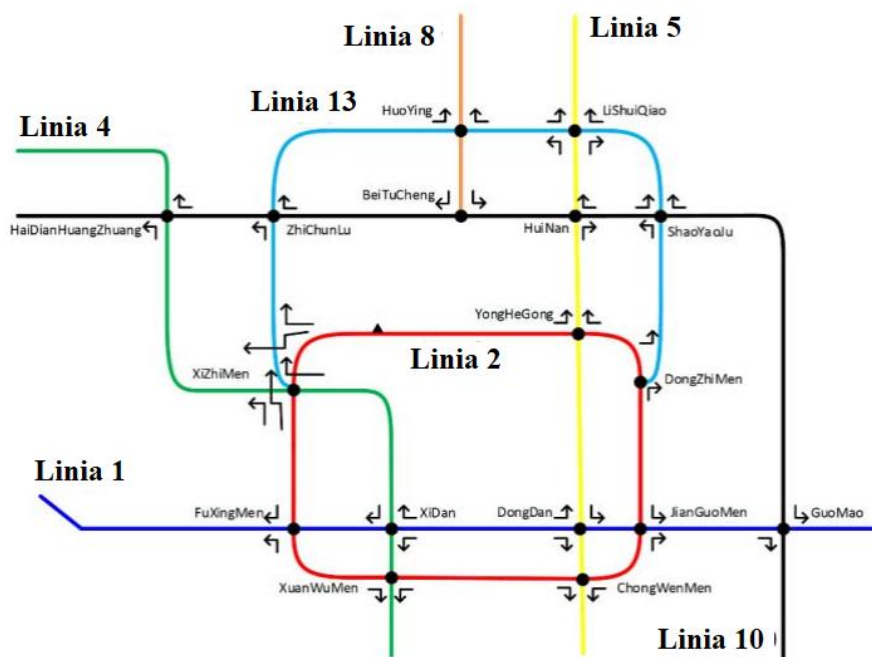
Specyficznym typem synchronizacji rozkładów jazdy w całej sieci jest synchronizacja pierwszych i/lub ostatnich kursów w sieci. Większość prac, w których opisywano ten problem skupiała się na liniach w sieci kolei miejskiej lub metra.

Problem synchronizacji pierwszych kursów jest istotny zarówno z perspektywy organizatora publicznego transportu zbiorowego i pasażera. Organizatorowi powinno zależeć na tym, aby pociągi nie traciły zbyt wiele czasu na oczekiwanie na skład obsługujący inną linię, pasażerowi zaś zależy na tym, aby jak najszybciej dotrzeć do celu i nie czekać długo na swoje połączenie (Xin, et al., 2016).

Celem synchronizacji ostatnich kursów jest takie ustalenie odjazdów pociągów ze stacji, aby wszyscy pasażerowie zdążyli się przesiąść do pociągów udających się w kierunku ich domu. O ile w ciągu dnia z uwagi na częste odjazdy pasażerowie nie mają problemów

z przesiadkami, w godzinach wieczornych najczęściej jest to dla nich jedyna szansa na powrót do domu za pomocą środków publicznego transportu zbiorowego (Liujiang, et al., 2015). Z tych względów są oni gotowi na dłuższe niż zwykle oczekiwanie na swój pociąg w godzinach pozaszczytowych, co pozwala na projektowanie rozkładu z większą swobodą.

Przykładem miasta, w którym dokonano synchronizacji ostatnich kursów jest Pekin, gdzie zsynchronizowano ostatnie kursy pociągów metra tak by umożliwić pasażerom wieczorne powroty do domu z centrum stolicy Chin. Na rys. 9 przedstawiono fragment schematu linii metra w Pekinie z 2012 roku.



Rysunek 9. Fragment schematu sieci pekińskiego metra
Źródło: Opracowanie własne na podstawie (Haodong, et al., 2018)

Kolorowe linie na rys. 9 reprezentują trasy linii metra, czarne kropki stacje węzłowe, a strzałki kierunki przesiadek pasażerów.

1.3.2 Kryteria efektywności synchronizacji rozkładów jazdy

Rezultat każdego działania powinien zostać oceniony. Również podczas synchronizowania rozkładów jazdy należy ocenić efektywność rozwiązań i zweryfikować, czy mają one pozytywny wpływ na funkcjonowanie i odbiór publicznego transportu zbiorowego. Do oceny efektywności można przyjąć różne kryteria.

Wśród miar efektywności można znaleźć:

- minimalizację czasu oczekiwania pasażerów na przesiadkę,
- minimalizację całkowitego czasu podróży pasażerów,
- maksymalizację łącznej liczby przyjazdów na wybrany przystanek w tym samym czasie,
- minimalną liczbę pojazdów potrzebną do obsłużenia sieci,
- minimalizację całkowitego czasu spędzonego w pojeździe przez pasażera,
- maksymalizację liczby możliwych przesiadek,
- minimalizację liczby podróży, dla których nie udało się zapewnić przesiadki w dogodnym czasie,
- maksymalizację satysfakcji pasażerów,
- minimalizację ponoszonych kosztów,
- minimalizację zmian w obecnym rozkładzie jazdy.

Miary te mogą być stosowane w funkcji celu (dopasowania) pojedynczo lub jako połączenie kilku z nich.

Minimalizacja łącznego czasu oczekiwania na przesiadkę przez pasażerów to działanie skupione na tym, aby wszyscy pasażerowie, którzy dokonują przesiadek oczekiwali na nie jak najkrócej. Czas oczekiwania pasażera na przesiadkę można opisać wzorem:

$$T^k = o_j^k - p_i^k \quad (3)$$

gdzie:

T^k – czas oczekiwania przez k -tego pasażera na przesiadkę,

o_j^k – czas odjazdu k -tego pasażera pojazdem j -tej linii,

p_i^k – czas przyjazdu k -tego pasażera pojazdem i -tej linii (Chakroborty, et al., 1995).

Zmniejszenie czasu oczekiwania pasażerów na przesiadkę wpływa również na skrócenie całkowitego czasu podróży.

Synchronizacja rozkładów jazdy pod kątem maksymalizacji łącznej liczby przyjazdów na wybrany przystanek w tym samym czasie skupia się na tym, aby jak najwięcej pojazdów obsługujących różne linie przyjechało na przystanek przesiadkowy w tym samym czasie (dopuszczalne są krótkie odchylenia). W rozkładzie jazdy należy zaplanować odpowiednio

długi postój na przystanku przesiadkowym tak aby pasażerowie zdążyli się przesiąść (Ceder, et al., 2001).

Synchronizacja rozkładów jazdy skupiona na minimalizacji liczby pojazdów do obsłużenia sieci dopuszcza nie tylko zmianę częstotliwości odjazdów poszczególnych linii w sieci transportu publicznego, ale także zmianę ich tras. Nadrzędnym jej celem jest zapewnienie przesiadek możliwie największej liczbie pasażerów przy jak najmniejszej liczbie zaangażowanych w to pojazdów (Bielli, et al., 2002).

Synchronizacja rozkładów jazdy zorientowana na minimalizację całkowitego czasu spędzonego w pojeździe przez pasażera zakłada, że pomiędzy źródłem podróży a celem jest dostępnych dla pasażera kilka tras, a także różne środki transportu, w tym transport na życzenie. Zakłada się, że istotnym dla pasażera czynnikiem jest łączny czas spędzony we wszystkich pojazdach, który powinien być jak najkrótszy (Lee, et al., 2022).

W synchronizacji rozkładów jazdy ukierunkowanej na maksymalizację liczby możliwych przesiadek konieczne jest zdefiniowanie okna czasowego, w którym możliwa jest przesiadka. Jeśli przyjazd pojazdu jednej linii i odjazd pojazdu innej linii znajdują się w tym oknie, wtedy przesiadkę między nimi uznaje się za możliwą (Ke, et al., 2020). Wykonanie takiej synchronizacji rozkładów jazdy odbywa się bez uwzględnienia danych o popycie w sieci publicznego transportu zbiorowego.

Synchronizacja rozkładów jazdy skupiona na minimalizacji liczby podróży, dla których nie jest możliwa przesiadka w dogodnym czasie, podobnie jak synchronizacja ukierunkowana na maksymalizację liczby możliwych przesiadek, zakłada istnienie okna czasowego, w którym można dokonać przesiadki (Nesheli & Ceder, 2014). W tym przypadku istotne są jednak dane o popycie w sieci publicznego transportu zbiorowego. Dąży się do minimalizacji liczby podróży bez możliwej przesiadki.

Celem synchronizacji rozkładów jazdy nastawionej na zwiększenie satysfakcji pasażera jest takie dopasowanie rozkładów jazdy, aby pasażer odczuwał jak największy komfort w czasie podróży. Na komfort pasażera składają się między innymi: czas oczekiwania na połączenie, długość jego trwania, czas wchodzenia do pojazdu, dostępność miejsc siedzących czy czas spędzony w pojeździe. Do oceny satysfakcji można wykorzystać jeden lub więcej wskaźników. Należy jednak uwzględnić, że ocena zależy od indywidualnych preferencji każdego pasażera (Shang, et al., 2019), (Cats, et al., 2016).

Podczas prac nad synchronizacją rozkładu jazdy zorientowaną na minimalizację całkowitych ponoszonych kosztów istotne jest, aby je właściwie zdefiniować. Kosztem może być na przykład: czas oczekiwania na przesiadkę, wydatki poniesione na wprowadzenie pojazdu do obsługi sieci (Shrivastava, et al., 2002) lub wprowadzenie do obsługi sieci większego pojazdu niż zazwyczaj (Chowdhury & Chien, 2011).

Podczas synchronizacji rozkładów jazdy, w której miarą efektywności jest minimalizacja zmian w rozkładzie, skupia się na tym, aby jak najmniej ingerować w rozkład jazdy, który jest już znany pasażerom (Liujiang, et al., 2015).

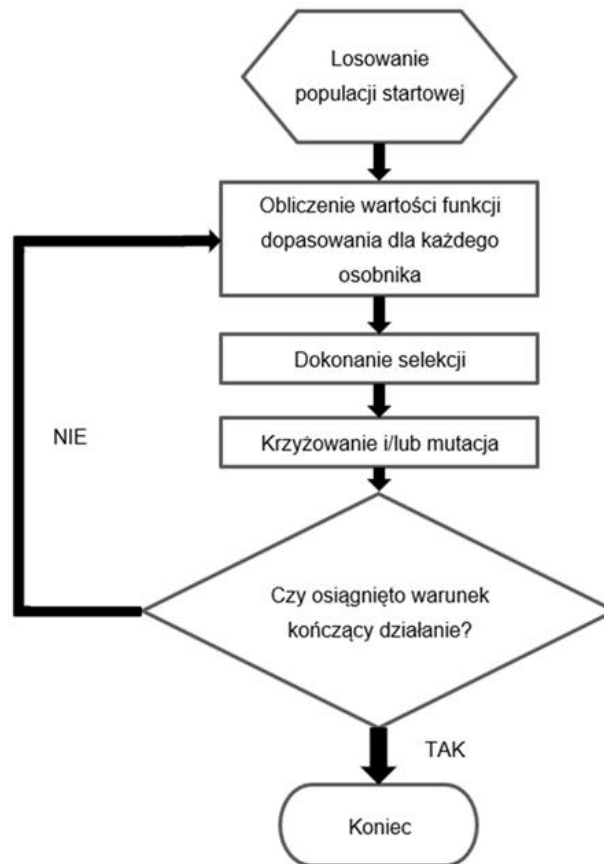
1.4 Algorytmy genetyczne

Twórcą koncepcji algorytmów genetycznych jest John Holland. Należą one do algorytmów ewolucyjnych i bazują na teorii doboru naturalnego Karola Darwina (Holland, 1975). Przydatność algorytmów genetycznych do rozwiązywania problemów optymalizacyjnych udowodnił De Jong (De Jong, 1975). Algorytm genetyczny wykorzystywany jest też do generowania puli rozwiązań wejściowych, z których korzystają następnie inne metody optymalizacyjne (Sivanandam & Deepa, 2007).

1.4.1 Działanie algorytmów genetycznych

Terminologia związana z algorytmami genetycznymi jest wprost zapożyczona z nauk biologicznych.

Populacja startowa jest generowanym losowo zbiorem osobników. Każdy osobnik jest scharakteryzowany za pomocą chromosomu. Chromosom jako element algorytmu genetycznego jest sekwencją niezależnych zmiennych nazywanych genami. Geny posiadają określoną wartość nazywaną allelem. Geny najczęściej kodowane są w formie binarnej, gdzie allel przyjmuje wartość 0 lub 1, ale mogą być też zakodowane w formie liczb całkowitych lub rzeczywistych. Dopasowanie poszczególnych osobników do środowiska określa wartość funkcji dopasowania, która najczęściej jest tożsama z funkcją celu (Ansari & Hou, 1997). Wartość funkcji celu jest podstawą do selekcji osobników, które w kolejnych etapach zostaną poddane krzyżowaniu i mutacjom. Krzyżowanie polega na wymianie części chromosomu między osobnikami (Wen-Yang, et al., 2003), a mutacja to zachodząca w losowy sposób zmiana wartości genu (Elsayed, et al., 2011). Każda z iteracji algorytmu genetycznego nazywana jest generacją, a zbiór wszystkich rozwiązań badanych podczas jej trwania to populacja. Schemat przedstawiający uproszczone działanie algorytmu genetycznego został zaprezentowany na rys. 10.



Rysunek 10. Schemat przedstawiający działanie algorytmu genetycznego
Źródło: Opracowanie własne

Jak zaprezentowano na rys. 10. w pierwszym kroku losowana jest populacja startowa. Następnie dla wszystkich znajdujących się w niej osobników obliczana jest wartość funkcji dopasowania. Po dokonaniu selekcji osobników na podstawie wartości funkcji dopasowania w kolejnym kroku następuje krzyżowanie osobników, a po nim zgodnie z parametrami podanymi przez użytkownika może dojść do mutacji poszczególnych genów. Ostatnią czynnością jest sprawdzenie czy osiągnięto warunek zakończenia działania algorytmu. Jeśli nie, algorytm powtarza proces na populacji osobników wygenerowanych w poprzedniej iteracji od etapu oceny dopasowania.

1.4.2 Metody selekcji osobników

Istnieją trzy główne sposoby wyłaniania osobników, które będą miały szansę przekazać swoje geny kolejnym pokoleniom. Są to:

1. Metoda ruletki. Osobnikom proporcjonalnie do wartości funkcji dopasowania przydziela się pole koła ruletki. Następnie losowane jest miejsce na tym kole i osobnik, który je zajmuje przechodzi do kolejnej iteracji. Sprawia to, że osobniki z najlepszą wartością funkcji dopasowania mają większe szanse na przejście do następnej iteracji (Lipowski & Lipowska, 2012).
2. Selekcja rankingowa. Osobniki klasyfikowane są pod względem najlepszej wartości funkcji dopasowania, a określona liczba najlepszych z nich przechodzi do kolejnej iteracji (Shukla, et al., 2015).
3. Selekcja turniejowa. Osobniki dzielone są na podgrupy. Z każdej podgrupy do kolejnej iteracji przechodzi osobnik z najlepszą wartością funkcji dopasowania (Sharapov, 2007).

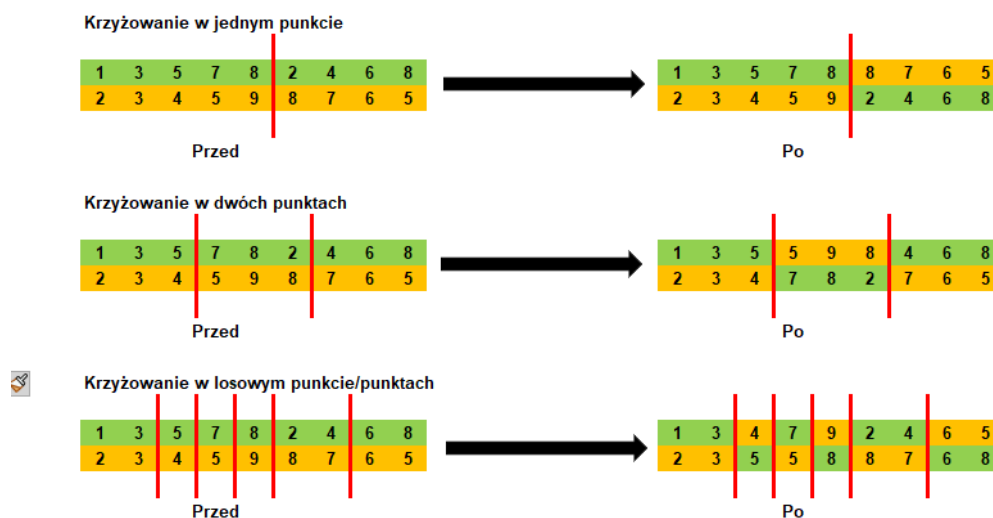
Dopuszcza się również dołączenie do metod ruletki i selekcji turniejowej elitaryzmu, polegającego na tym, że zawsze do następnej iteracji przejdzie określona przez użytkownika liczba osobników z najlepszą wartością funkcji dopasowania (Hancock, 1994).

1.4.3 Metody krzyżowania osobników

Istnieją trzy główne sposoby krzyżowania:

1. krzyżowanie w jednym punkcie,
2. krzyżowanie w dwóch punktach,
3. krzyżowanie w losowym punkcie/punktach.

Sposoby krzyżowania dwóch osobników przedstawia rys. 11.

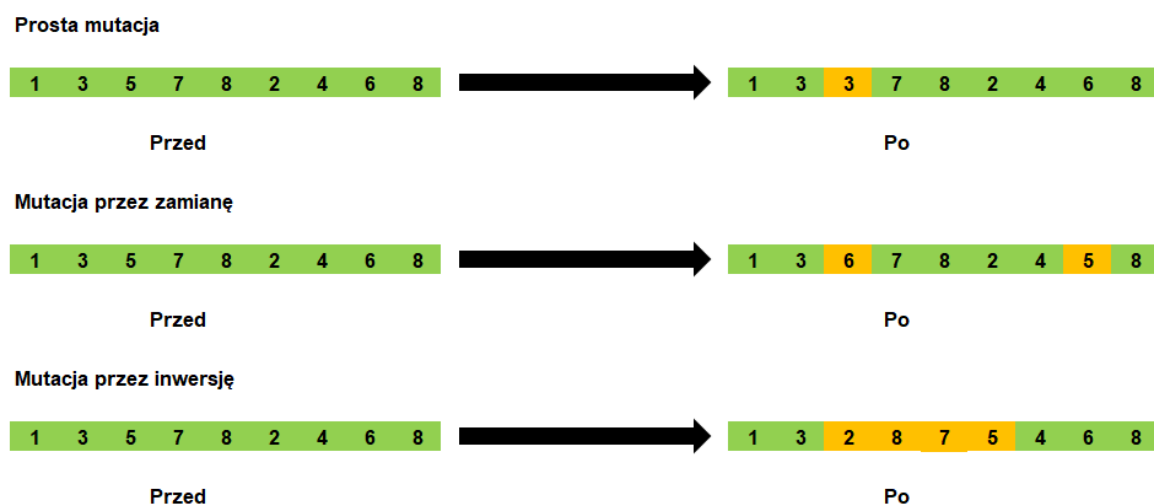


Rysunek 11. Metody krzyżowania
 Źródło: Opracowanie własne

Jak pokazano na rys. 11, podczas krzyżowania w jednym punkcie każdy z osobników „przecinany” jest w jednym miejscu. Pierwszy nowopowstały osobnik otrzymuje geny znajdujące się przed punktem krzyżowania i geny drugiego osobnika znajdujące się za punktem krzyżowania. Drugi nowopowstały osobnik otrzymuje geny w przeciwny sposób (Goldberg, 1989). W krzyżowaniu w dwóch punktach każdy osobnik zostaje podzielony na trzy części. Pierwszy nowopowstały osobnik otrzymuje pierwszą i trzecią część genów jednego osobnika oraz drugą część genów drugiego. Drugi nowopowstały osobnik otrzymuje pozostałe geny (Corne & Lones, 2018). W przypadku tych dwóch metod lokalizacja punktów krzyżowania może być z góry określona lub generowana losowo w każdej iteracji. W ostatniej z opisywanych metod krzyżowania lokalizacja punktów i liczba miejsc „przecięcia” osobników generowane są w sposób losowy (Eiben & Smith, 2015).

1.4.4 Sposoby mutacji osobników

Wraz z rozwojem metod poszukiwania rozwiązań bazujących na algorytmach genetycznych wykształciło się kilka sposobów dokonywania mutacji. Trzy najpopularniejsze sposoby przedstawiono na rys. 12.



Rysunek 12. Sposoby mutacji osobników
Źródło: Opracowanie własne

Jak pokazano na rys. 12, mutacja prosta to zmiana wartości jednego genu w losowym miejscu (Aibinu, et al., 2016). W mutacji poprzez zamianę dwa losowe geny zamieniają się miejscami (Kumar, et al., 2020). W mutacji przez inwersję losowane są dwie pozycje genów

w chromosomie, a następnie geny pomiędzy nimi zamieniają się pozycjami (Asadujjaman, et al., 2022).

1.4.5 Warunek kończący działanie algorytmu

Niezbędne jest zdefiniowanie warunku, po spełnieniu którego algorytm genetyczny ma zakończyć swoje działanie. Istnieją trzy główne warunki zatrzymania algorytmu:

- określona z góry liczba generacji (Fogel, 2012),
- brak jakiegokolwiek poprawy (rozumianej jako znalezienie rozwiązania z lepszą wartością funkcji celu) przez określoną liczbę generacji (Eiben & Smith, 2015),
- brak poprawy o określony przez użytkownika procent wartości funkcji celu przez określoną liczbę generacji (Grundler, 2001).

Podstawowym warunkiem zakończenia działania algorytmu jest wskazanie określonej z góry liczby iteracji algorytmu, po których ma on zakończyć swoje działanie. Wadą tej metody jest to, że algorytm będzie działał dalej nawet gdy znajdzie już rozwiązanie optymalne.

W metodzie drugiej użytkownik określa liczbę iteracji bez poprawy wartości funkcji celu, po której algorytm ma zakończyć działanie. Zaletą tej metody jest skrócenie czasu działania algorytmu.

Trzeci sposób jeszcze bardziej ogranicza czas działania algorytmu genetycznego. Zakłada on, że algorytm skończy swoje działanie nawet jeśli przez określoną liczbę iteracji dostarcza lepsze rozwiązania, ale wartość funkcji celu nie poprawia się w określonym przez użytkownika stopniu.

1.4.6 Parametry algorytmów genetycznych

Główne parametry algorytmów genetycznych to:

- liczba osobników w populacji,
- liczba generacji,
- przeżywalność osobników¹,
- prawdopodobieństwo krzyżowania²,

¹ ang. survivors rate

² ang. crossover rate

-
- prawdopodobieństwo mutacji³.

Liczba osobników w populacji określa, ile rozwiązań startowych ma zostać wylosowanych na początku działania algorytmu i ile rozwiązań będzie branych pod uwagę w każdej jego iteracji.

Liczba generacji określa przez ile iteracji algorytm genetyczny ma poszukiwać rozwiązania zadanego problemu.

Od wartości parametru przeżywalności zależy jak wiele osobników przejdzie do następnej iteracji algorytmu. Parametr przeżywalności przyjmuje wartości w zakresie od 0 do 1 gdzie 0 oznacza, że do kolejnej iteracji nie przejdzie żaden osobnik, a 1, że wszystkie.

Parametr nazwany prawdopodobieństwem krzyżowania określa szansę na posiadanie potomstwa przez osobniki składające się na populację. Jego wartość mieści się w przedziale od 0 do 1, gdzie 0 oznacza, że dany osobnik nie będzie miał potomstwa, a 1 – że na pewno będzie je miał.

Wartość parametru odpowiedzialnego za prawdopodobieństwo mutacji mieści się w przedziale od 0 do 1, gdzie 0 oznacza, że do mutacji nie dojdzie, a 1, że dojdzie do niej za każdym razem.

1.4.7 Wady i zalety algorytmów genetycznych

Algorytmy genetyczne posiadają dwie podstawowe zalety (Xin-She, 2014). Pierwszą jest umiejętność radzenia sobie ze złożonymi problemami. Drugą jest możliwość równoległego przeszukiwania przestrzeni możliwych rozwiązań w różnych kierunkach. Związane jest to z jego charakterystyką, w każdej przeprowadzonej iteracji liczba badanych rozwiązań równa jest założonej z góry przez użytkownika liczbie osobników w populacji. Rozwiązania te za każdym razem są porównywane między sobą i losowo krzyżowane co pozwala na ucieczkę od lokalnego optimum (Chung, et al., 2001).

Jako wadę algorytmów genetycznych traktować należy trudność ich parametryzacji. Należy odpowiednio dobrać wartości dotyczące wielkości populacji, prawdopodobieństwo krzyżowania osobników oraz jego sposób, kryteria przejścia osobników do następnej

³ ang. mutation rate

generacji oraz warunki, po spełnieniu których działanie algorytmu zostanie zatrzymane. Każdy błąd na tym etapie może utrudnić osiągnięcie zbieżności algorytmu lub być przyczyną otrzymania wyniku dalekiego od optimum (Xin-She, 2014).

1.5 Przegląd literatury przedmiotu

Przełgądu literatury dokonano z wykorzystaniem baz danych Scopus, Web of Science oraz Google Scholar, posługując się słowami kluczowymi takimi jak: synchronizacja rozkładów jazdy, koordynacja przesiadek, optymalizacja przesiadek i koordynacja transportu publicznego. Dodatkowo, niektóre publikacje zostały odnalezione poprzez analizę źródeł cytowanych w istniejących opracowaniach. Przegląd zakończono w kwietniu 2023 roku, obejmując zakres badań opublikowanych w latach 1988–2022. Co istotne, ponad połowa uwzględnionych prac (około 60%) ukazała się w 2013 roku lub później, co odzwierciedla rosnące zainteresowanie tym obszarem badań.

Podczas gdy wcześniejsze prace koncentrowały się na kryteriach efektywności takich jak minimalizacja czasu oczekiwania pasażerów czy maksymalizacja możliwości przesiadek, nowsze badania coraz częściej podkreślają optymalizację bardziej złożonych i wieloaspektowych kryteriów. Na przykład współczesne analizy często starają się równoważyć czas oczekiwania pasażerów z kosztami operacyjnymi lub dążyć do minimalizacji czasu przejazdu oraz liczby potrzebnych pojazdów, a także do maksymalizacji liczby podróży bezpośrednich.

Publikacje sklasyfikowano w cztery odrębne grupy, w zależności od zastosowanych głównych technik optymalizacji:

- algorytmy genetyczne – najliczniejsza grupa, w której stosuje się algorytmy genetyczne do rozwiązywania problemów synchronizacji rozkładów jazdy,
- programowanie całkowitoliczbowe – grupa wykorzystująca różne warianty programowania całkowitoliczbowego do rozwiązania problemu synchronizacji,
- symulowane wyżarzanie – mniejsza grupa publikacji, w której podstawowym podejściem do rozwiązania problemu synchronizacji jest metoda symulowanego wyżarzania,
- inne metody – kategoria obejmująca różnorodne alternatywne techniki opisane w literaturze.

Dobór metody w istotny sposób wpływa na jakość i efektywność obliczeniową rozwiązań problemu synchronizacji. Różne metody heurystyczne mogą prowadzić do odmiennych wyników i wymagać różnych czasów obliczeń, aby osiągnąć zbieżność. Optymalny wybór metody heurystycznej w dużej mierze zależy od specyficznych cech systemu transportowego, takich jak popyt, liczba linii i pojazdów czy układ sieci.

Aby uwzględnić tę zmienność przegląd literatury szczegółowo analizuje rodzaj środka transportu, lokalizację geograficzną oraz funkcję celu użytą w danym badaniu. Celem analizy jest dostarczenie wglądu w sposób wykorzystywania różnych metod w odniesieniu do zróżnicowanych problemów rzeczywistych.

1.5.1 Synchronizacja rozkładów jazdy przy użyciu algorytmów genetycznych

Prace napisane przed 2013 roku skupiały się na optymalizacji tylko jednego parametru. Najczęściej była to minimalizacja czasu oczekiwania pasażerów na przesiadkę lub minimalizacja ponoszonych kosztów przez organizatorów. W pracach napisanych później funkcja celu składała się z większej liczby kryteriów. Były to: minimalizacja czasu oczekiwania pasażerów na przystanku, minimalizacja całkowitego czasu podróży, minimalizacja zmian w wyjściowym rozkładzie jazdy, minimalizacja liczby pojazdów potrzebnych do realizacji rozkładu jazdy oraz zwiększenie możliwości przesiadek pomiędzy pojazdami.

Badacze najczęściej korzystali z populacji liczących mniej niż 100 osobników, przy prawdopodobieństwie krzyżowania pomiędzy 50% a 60% oraz prawdopodobieństwie mutacji równym 10%. Prace nad synchronizacją rozkładów jazdy przedstawiono w tabeli 4.

Tabela 4. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą algorytmu genetycznego w dokonanych przeglądzie literatury

Autorzy (rok publikacji)	Funkcja celu	Typ sieci	Część optymalizowanej sieci	Miejsce	Parametry algorytmu genetycznego
Chakroborty Deb i Subrahmanyam (1995)	Minimalizacja czasu oczekiwania na przystanku początkowym i czasu oczekiwania na przesiadkę	Autobusowa	Węzeł (sieć testowa)	-	M – 0,5%, K – 95%, Pop – 350, Gen – 200
Nachtigall i Voget (1996)	Minimalizacja czasu oczekiwania na przesiadkę	Kolejowa	Sieć testowa	-	Nie opisano
Biell, Caramia i Carotenuto (2002)	Minimalizacja liczby pojazdów obsługujących sieć	Autobusowa	Sieć	Parma (Włochy)	K – 80%, M – 10%

Autorzy (rok publikacji)	Funkcja celu	Typ sieci	Część optymalizo- wanej sieci	Miejsce	Parametry algorytmu genety- cznego
Shrivastava, Dhingra i Gundaliya (2002)	Minimalizacja całkowitego kosztu obsługi	Autobusowa	Wybrane linie	Mumbaj (Indie)	M – 1%, K – 80% Pop – 420
Shrivastava i Dhingra (2002)	Minimalizacja całkowitego kosztu obsługi	Autobusowa	Wybrane linie	Mumbaj (Indie)	M – 1%, K – 80% Pop – 420
Ngamchai i Lovell (2003)	Minimalizacja całkowitego kosztu obsługi	Autobusowa	Sieć testowa	-	Nie opisano
Cevallos i Zhao (2006a)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Sieć	Hrabstwo Brodward (USA)	K – 50%, M – 10% Pop – 20
Cevallos i Zhao (2006b)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Sieć	Hrabstwo Brodward (USA)	K – 50%, M – 10% Pop – 20
Shrivastava i O'Mahony (2006)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Wybrane linie	Dublin (Irlandia)	K – 95%, M – 10%
Shafahi i Khani (2010)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Sieć	Meszhed (Iran)	K – 50%, M – 50% Pop – 20
Yu, Yang i Yao (2010)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Sieć	Dalian (Chiny)	Nie opisano
Niu i Zhou (2013)	Minimalizacja czasu oczekiwania na przesiadkę	Kolejowa	Linia	Kanton (Chiny)	K – 98%, M – 15% Pop – 40
Aksu i Yılmaz (2014)	Minimalizacja czasu oczekiwania na przesiadkę i minimalizacja liczby utraconych możliwości przesiadek.	Kolejowa	Sieć	Stambuł (Turcja)	K – 90%, M – 8% Pop – 2000
Wu, Liu, Sun, Li, Gao i Wang (2015)	Minimalizacja całkowitego kosztu obsługi	Metro	Sieć	Pekin (Chiny)	K – 80%, M – 10% Pop – 100
Kang, Wu, Sun, Zhu i Gao (2015)	Maksymalizacja możliwych przesiadek	Metro (Ostatni kurs)	Sieć	Pekin (Chiny)	Nie opisano
Kang, Wu, Sun, Zhu i Wang (2015)	Minimalizacja zmian w rozkładzie, całkowitego czasu podróży i czasu oczekiwania na przesiadkę	Kolej miejska (Ostatni kurs)	Sieć	Pekin (Chiny)	Nie opisano
Wu, Tang, Yu i Pan (2015)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Sieć testowa	-	K – 80% M – 5% Pop – 100
Wu, Yang, Tang I Yu (2016)	Maksymalizacja liczby pasażerów, którzy się przesiadają oraz minimalizacja zmian w rozkładzie	Autobusowa	Sieć	Chiny	Nie opisano

Autorzy (rok publikacji)	Funkcja celu	Typ sieci	Część optymalizowanej sieci	Miejsce	Parametry algorytmu genetycznego
Shang, Li, Liu, Xian I Guo (2018)	Minimalizacja całkowitego czasu podróży	Metro	Sieć	Shenzhen (Chiny)	K – 80% M – 15% Pop – 2000
Cao, Ceder, Li and Zhang (2019)	Maksymalizacja możliwości przesiadek	Kolejowa	Sieć	Pekin (Chiny)	Nie opisano
Wu, Sun, Kang i Liu (2019)	Maksymalizacja efektywności sieci	Kolej miejska (Ostatni kurs)	Sieć	Pekin (Chiny)	Nie opisano
Chen, Mao, Bai, Ho I Li (2019)	Maksymalizacja liczby przesiadek	Kolej miejska (Ostatni kurs)	Sieć	Shenzhen (Chiny)	Pop – 300 Gen – 200
Naumov (2020)	Minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Węzeł	Kraków	K – 50% M – 10% Pop – 100 Gen – 20
Wang, Li i Cao (2020)	Minimalizacja czasu oczekiwania na przesiadkę	Kolejowa	Wybrane linie	Shenyang (Chiny)	K – 80% M – 15% Pop – 30
Cao, Tang i Gao (2020)	Minimalizacja czasu oczekiwania na przesiadkę	Kolejowa	Węzeł	Pekin (Chiny)	K – 70% M – 0,5%
Guo, Wu, Sun, Yang, Jin I Wang (2020)	Minimalizacja czasu oczekiwania na przesiadkę	Kolej miejska (Ostatni kurs)	Sieć	Pekin (Chiny)	Nie opisano
Ataeian, Solimanpur, Amiripour i Shankar (2021)	Maksymalizacja przyjazdów w tym samym czasie oraz minimalizacja liczby pojazdów potrzebnych do obsługi sieci	Autobusowa	Sieć	Teheran (Iran)	Nie opisano
Wang, Zhou i Yan (2022)	Maksymalizacja liczby przesiadek i minimalizacja czasu oczekiwania na przesiadkę	Autobusowa	Wybrane linie	Singapur (Singapur)	K – 80% M – 10% Pop 100
Naeini, Shafahi i Taherkhani (2022)	Minimalizacja czasu oczekiwania pasażerów i czasu podróży pasażerów nieprzesiadających się oraz maksymalizacja liczby pasażerów, którzy mogą dotrzeć do celu podróży	Kolejowa	Węzeł	Teheran (Iran)	K – 60% M – 35% Pop – 90

Pop – liczba osobników w populacji, Gen – liczba generacji, K – prawdopodobieństwo krzyżowania,

M – prawdopodobieństwo mutacji

Źródło: (Kapica, et al., 2025)

Zgodnie z danymi przedstawionymi w tabeli 4. Chakroborty, Deb K i Subrahmanyam (1995) zastosowali algorytm genetyczny, aby zminimalizować sumę czasu okiwania pasażerów na przystanku, z którego zaczynają podróż i czasu oczekiwania na przesiadkę.

Nachtigall i Voged (1996) porównali wyniki jakie otrzymywali za pomocą algorytmu genetycznego w zależności od puli rozwiązań startowych by zminimalizować czasu oczekiwania pasażerów. W pierwszym podejściu algorytm genetyczny zaczynał pracę od losowo wygenerowanych rozwiązań, w drugim od rozwiązań uzyskanych algorytmem zachłannym⁴, osiągając lepsze wyniki. Bielli, Caramia i Carotenuto (2002) wykorzystali algorytm genetyczny do optymalizacji sieci autobusowej w Parmie we Włoszech. Autorzy zwiększyli wartość wielokryterialnej funkcji dopasowania o około 90%. Shrivastava i inni (2002a, 2002b) zsynchronizowali sieć autobusową z kolejową. Wprowadzili karę za czas oczekiwania na przesiadkę dłuższy niż 10 minut. Shafahi i Khani (2010) zsynchronizowali sieć autobusową w Maszadzie w Iranie i udowodnili, że algorytm genetyczny osiąga lepsze wyniki niż metoda podziału i ograniczeń⁵. Wu i inni (2014) zsynchronizowali sieć metra w Pekinie w Chinach. Skoncentrowali się na zminimalizowaniu łącznego czasu oczekiwania pasażerów na przesiadkę i wyrównaniu średniego czasu oczekiwania na wszystkich stacjach. Aksu i Yilmaz (2014) sformułowali funkcję celu, aby zminimalizować czas oczekiwania pasażerów i zminimalizować liczbę utraconych możliwości przesiadek.

Kang i in. (2015a) zajęli się problemem synchronizacji ostatnich pociągów metra w Pekinie (Chiny), aby zminimalizować różnicę pomiędzy czasem odjazdu ostatniego pociągu przesiadkowego a czasem przybycia pasażerów ostatniego pociągu dowozowego. Obliczenia wykazały, że funkcja celu minimalizująca ten czas może zagwarantować bardziej dopasowane połączenia niż funkcja minimalizująca całkowity czas oczekiwania pasażerów przy takim samym czasie oczekiwania na przesiadkę. W drugim badaniu Kang i in. (2015b) sformułowali funkcję celu minimalizującą czas podróży pasażerów (bez czasu oczekiwania na przesiadki) i minimalizującą zmiany w rzeczywistym rozkładzie jazdy w chińskiej sieci metra w Pekinie. Porównali symulowane wyżarzanie⁶, przeszukiwanie tabu⁷, metodę podziału i ograniczeń oraz algorytm genetyczny. Algorytm genetyczny znalazł najwyższą wartość funkcji celu w ciągu 6 sekund, podczas gdy pozostałe potrzebowały od pięciu (przeszukiwanie tabu) do stu (symulowane wyżarzanie) razy więcej czasu,

⁴ ang. Greedy Algorithm

⁵ ang. Branch and Bound

⁶ ang. Simulated Annealing

⁷ ang. Tabu Search,

aby znaleźć taki sam lub zbliżony wynik. Niu i in. (2015), aby zmniejszyć dyskomfort pasażerów dodali do funkcji celu minimalizującej czas oczekiwania, element powiązany z zatłoczeniem w pociągu.

Cao i in. (2018) zajęli się problemem maksymalizacji zsynchronizowanych przyjazdów na stacjach metra w Pekinie. Ich algorytm genetyczny do optymalizacji zsynchronizowanego i skoordynowanego rozkładu jazdy⁸ działał szybciej i znalazł lepsze rozwiązanie niż model CPLEX. Yin i in. (2019) zauważyli, że wydłużenie postojów w celu oczekiwania na inny pojazd generuje koszty, które nie zawsze są adekwatne do zwiększonych wpływów. Jak już wspomniano transport zbiorowy wymaga dotacji rządowych lub lokalnych i musi zachować równowagę między kosztami operacyjnymi a poziomem usług. Sformułowali więc funkcję celu, która pozwoliła utrzymać tę równowagę, zmniejszyć liczbę nieudanych przesiadek z 20 do 14, skrócić średni czas oczekiwania o ponad 50% oraz zwiększyć liczbę pasażerów przesiadających się o ponad 90% w metodzie synchronizacji ostatnich pociągów.

Chen i in. (2019) zastosowali algorytm genetyczny w celu maksymalizacji liczby dostępnych par źródło-cel dla ostatnich pociągów. Wang, Li i Cao (2020) porównali algorytm genetyczny i algorytm szarych wilków⁹. Ich funkcja celu składała się z trzech części: minimalizacji czasu oczekiwania pasażerów na stacji początkowej, rzeczywistego czasu oczekiwania na przesiadkę oraz wartości kary, dodawanej w przypadku nieudanej przesiadki. Algorytm genetyczny znalazł rozwiązanie charakteryzujące się lepszą wartością funkcji celu o 0,35% w porównaniu do algorytmu szarych wilków, przy czym należy zauważyć, że znalazł je trzykrotnie szybciej.

Ataeian i in. (2021) zastosowali algorytm genetyczny w celu maksymalizacji liczby jednoczesnych przyjazdów oraz minimalizacji liczby pojazdów wymaganych do obsługi sieci szybkiego transportu autobusowego¹⁰ w stolicy Iranu, Teheranie. Naeini, Shafahi i Taherkhani (2022) zaproponowali synchronizację miejskiej sieci metra w Teheranie z wykorzystaniem strategii skip-stop. Strategia ta pozwala pojazdom na pomijanie losowo

⁸ ang. Synchronized and Coordinated Scheduling Optimization Genetic Algorithm (SCSO-GA)

⁹ ang. Grey Wolf Optimizer (GWO)

¹⁰ ang. Bus Rapid Transit (BRT) – system szybkiego transportu oparty na autobusach o dużej pojemności, cechami charakterystycznymi BRT są wydzielone pasy znajdujące się pomiędzy jezdniami dla ruchu ogólnego, platformy dla wsiadających umieszczone na podwyższeniu, bezwzględne pierwszeństwo pojazdów na skrzyżowaniach, wysoka prędkość komunikacyjna i częste odjazdy (min. raz na 10 minut) (Institute for Transportation & Development Policy, 2024).

wybranych stacji w celu minimalizacji ważonej sumy trzech czynników: całkowitego czasu oczekiwania pasażerów na przystanku, całkowitego czasu podróży w pojeździe oraz całkowitych kosztów operacyjnych. Sformułowali wielokryterialną funkcję celu, obejmującą minimalizację czasu oczekiwania pasażerów na stacjach początkowych, czasu oczekiwania na przesiadkę, czasu podróży pasażerów przesiadających się i nieprzesiadających się oraz maksymalizację liczby pasażerów, którzy skutecznie dotrą do celu.

Wang, Zhou i Yan (2022) wykorzystali algorytm genetyczny do synchronizacji rozkładów jazdy autonomicznych autobusów w sieci transportu publicznego w Singapurze.

1.5.2 Synchronizacja rozkładów jazdy za pomocą programowania całkowitoliczbowego¹¹ i jego wariantów

W znacznej części prac dotyczących synchronizacji rozkładów jazdy synchronizację wykonano przy użyciu programowania całkowitoliczbowego i jego wariantów. Rozwiązywano problemy dotyczące synchronizacji rozkładów jazdy sieci autobusowych i kolejowych, w jednym węźle, w wybranych węzłach, a także w całej sieci. Prace z wykorzystaniem programowania całkowitoliczbowego zaprezentowano w tabeli 5.

Tabela 5. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą programowania całkowitoliczbowego i jego wariantów w dokonanym przeglądzie literatury

Autorzy (rok publikacji)	Funkcja celu	Metoda	Typ sieci	Część optymalizowanej sieci	Miejsce
Ceder, Golany i Tal (2001)	Maksymalizacja liczby przyjazdów w oknie czasowym	MIP	Autobusowa	Sieć testowa	-
Eranksi (2004)	Maksymalizacja liczby przyjazdów w oknie czasowym	MIP	Autobusowa	Sieć testowa	-
Vansteenwegen i van Oudheusden (2007)	Minimalizacja kosztu pasażerów związanego z oczekiwaniem na przesiadkę	LP	Kolejowa	Sieć	Belgia
Liebchen (2008)	Minimalizacja czasu oczekiwania na przesiadkę	IP	Kolej miejska	Sieć	Berlin (Niemcy)

¹¹ ang. Integer Programming (IP)

Autorzy (rok publikacji)	Funkcja celu	Metoda	Typ sieci	Część optymalizowanej sieci	Miejsce
Wong, Yuen, Fung i Leung (2008)	Minimalizacja czasu oczekiwania na przesiadkę	MIP	Kolejowa	Sieć	Hongkong (Chiny)
Bruno, Improta i Sgalambro (2009)	Minimalizacja kosztów operacyjnych i czasu oczekiwania na przesiadkę	MIP	Autobusowa	Węzeł	Włochy
Nesheli i Ceder (2014)	Minimalizacja czasu oczekiwania na przesiadkę i liczby utraconych przesiadek	MIP	Autobusowa	Wybrane linie	Auckland (Nowa Zelandia)
Dou, Meng i Guo (2015)	Maksymalizacja możliwości przesiadek	MILP	Autobusowa do kolejowej (ostatni kurs)	Wybrane linie	Singapur (Singapur)
Ibarra-Rojas, López-Irarragorri i Rios-Solis (2015)	Maksymalizacja możliwości przesiadek	MILP	Autobusowa	Sieć	Monterrey (Meksyk)
Guo, Wu, Sun, Liu i Gao (2016)	Minimalizacja kosztu oczekiwania na przesiadkę	MILP	Pierwszy kurs	Sieć	Pekin (Chiny)
Wu, Liu i Jin (2016)	Minimalizacja kosztu całkowitego	MINLP	Kolejowa	Sieć testowa	-
Gschwender, Jara- Díaz i Bravo (2016)	Minimalizacja kosztu pasażerów związane z oczekiwaniem na przesiadkę i uruchomienia pojazdów	MILP	Autobusowa (Bus Rapid Transit)	Sieć testowa	-
Dou i Guo (2017)	Minimalizacja liczby nieudanych przesiadek	MILP	Kolej miejska (Ostatni kurs)	Sieć	Singapur (Singapur)
Liu, Ceder i Chowdhury (2017)	Maksymalizacja liczby przyjazdów w oknie czasowym i minimalizacja liczby pojazdów	MIP+DF	Autobusowa	Wybrane linie	Auckland (Nowa Zelandia)
Kang, Zhu, Sun, Wu, Gao i Hu (2019)	Maksymalizacja możliwości przesiadek	MILP	Ostatni kurs	Sieć	Wiedeń (Austria)
Shang, Huang i Wu (2019)	Równowaga pomiędzy satysfakcją pasażerów i efektywnością realizacji przewozów	NLIP	Autobusowa	Ciąg komunikacyjny	Pekin (Chiny)

Autorzy (rok publikacji)	Funkcja celu	Metoda	Typ sieci	Część optymalizowanej sieci	Miejsce
Wang, Wei, Zhang, Shi i Shang (2019)	Minimalizacja czasu oczekiwania na przesiadkę i liczby nieudanych przesiadek	MILP	Metro (Ostatni kurs)	Sieć	Pekin (Chiny)
Takamatsu i Taguchi (2020)	Minimalizacja kosztu pasażerów związanego z oczekiwaniem na przesiadkę.	MIP	Autobusowa do kolejowej	Ciąg komunikacyjny	Dystrykt Tohoku (Japonia)
Ke, Nie, Liebchen, Yuan i Wu (2020)	Maksymalizacja możliwości przesiadek	MIP	Kolejowa do lotniczej	Linia	Lotnisko Shijiazhuang Zhengding (Chiny)
Lee, Jiang, Ceder, Dauwels, Su i Nielsen (2022)	Minimalizacja czasu oczekiwania na przesiadkę i czasu przebywania w pojeździe	MILP	Autobusowa	Wybrane linie	Kopenhaga (Dania)

MIP – Mieszane programowanie całkowitoliczbowe¹², IP – programowanie całkowitoliczbowe, LP – Programowanie liniowe¹³, MILP – Mieszane całkowitoliczbowe programowanie liniowe¹⁴, MINLP - Mieszane całkowitoliczbowe programowanie nieliniowe¹⁵, NLIP – Nieliniowe programowanie całkowitoliczbowe¹⁶, DF – Funkcja deficytu¹⁷

Źródło: (Kapica, et al., 2025)

Zgodnie z danymi zaprezentowanymi w tabeli 5. w 2001 roku Ceder, Golany i Tal zaproponowali synchronizację rozkładów jazdy opartą na możliwie największej liczbie przyjazdów do wybranego węzła w jednym czasie, w celu umożliwienia pasażerom dokonania przesiadek w czasie kilkuminutowego postoju pojazdów. Po trzech latach do modelu dodano drugą składową funkcji celu, która dążyła do minimalizacji czasu oczekiwania przez pasażerów na przesiadkę (Eranksi, 2004).

W 2007 roku Vansteenwegen i Van Oudheusden zaproponowali minimalizację czasu oczekiwania pasażerów pociągów międzymiastowych w Holandii poprzez minimalizację „wartości czasu”¹⁸. Najczęściej w pracach przyjmuje się wartość tego wskaźnika równą 2,5

¹² ang. Mixed Integer Programming

¹³ ang. Linear Programming

¹⁴ ang. Mixed-Integer Linear Programming

¹⁵ ang. Mixed-Integer Non-Linear Programming

¹⁶ ang. Non-Linear Integer Programming

¹⁷ ang. Deficit Function

¹⁸ ang. Value of Time (VOT)

(Wardman, 2004). Vansteenwegen i Van Oudheusden przyjęli jego wartość równą 2,7 dla pasażerów oczekujących na przesiadkę powyżej 15 minut na stacji oraz równą 2 dla pasażerów znajdujących się w składzie oczekującym na stacji. W 2009 roku Bruno, Importa i Sgalambro zaproponowali model dla wybranych węzłów transportowych zapewniający równowagę pomiędzy kosztami operacyjnymi a czasem oczekiwania pasażerów.

W 2014 roku Nesheli i Ceder używając mieszanego programowania całkowitoliczbowego skrócili całkowity czas trwania podróży oraz zwiększyli liczbę dokonywanych przesiadek przez pasażerów wybranych linii autobusowych wchodzących w skład publicznego transportu zbiorowego w Auckland w Nowej Zelandii. W modelu wykorzystali dwie strategie: wstrzymywania pojazdów w wybranych miejscach w celu umożliwienia przesiadki pasażerom oraz pomijania niektórych przystanków przez pojazd, jeśli żaden ze znajdujących się w nim pasażerów nie planował na nim wysiąść. Przy zastosowaniu pierwszej strategii liczba przesiadek wzrosła o 100%, w przypadku drugiej o 150%. Całkowity czas podróży pasażerów zmniejszył się odpowiednio o 2,14% i 4,1%.

Wu, Liu i Jin w 2016 roku zaproponowali synchronizację rozkładów jazdy w dwóch krokach. Pierwszym było dodanie postojów na wybranych przystankach w trakcie realizacji kursu, w celu zmniejszenia prawdopodobieństwa propagacji odchylenia od rozkładu jazdy. W drugim kroku zaproponowali kontrolę realizacji rozkładu jazdy w czasie rzeczywistym i bieżące reagowanie na sytuację. W skład funkcji celu w proponowanym modelu wchodziły: minimalizacja kosztów operacyjnych, minimalizacja kosztów związanych z oczekiwaniem pasażerów, minimalizacja utraconych przesiadek oraz minimalizacja opóźnień.

Kang i inni w 2019 roku przy użyciu mieszanego całkowitoliczbowego programowania liniowego zaproponowali koordynację ostatnich kursów metra i kolei miejskiej w Wiedniu. Zaproponowali oni dynamiczne tworzenie tymczasowych autobusowych połączeń pomostowych w celu przywrócenia możliwości przesiadki pomiędzy liniami, jeśli na jednej z nich doszło do jakichkolwiek zakłóceń. Pozwoliło to na zwiększenie liczby pasażerów przesiadających się o 20%. W tym samym roku Shang, Huang i Wu użyli nieliniowego programowania całkowitoliczbowego do optymalizacji rozkładów jazdy pomiędzy węzłem przesiadkowym przy moście Guomao a przedmieściami Pekinu. Ich celem było znalezienie równowagi pomiędzy zadowoleniem pasażerów a efektywnością usług transportowych. Zadowolenie pasażerów zdefiniowali jako suma czasu oczekiwania na pojazd oraz wartość

wskaźnika komfortu w pojeździe, zależnego od napełnienia pojazdu. Efektywność została zdefiniowana jako stosunek liczby pasażerów w pojeździe do jego całkowitej pojemności.

1.5.3 Synchronizacja rozkładów jazdy za pomocą symulowanego wyżarzania

Symulowane wyżarzanie zostało wykorzystane do synchronizacji rozkładów jazdy metra i autobusów. Do tej pory wykonano zaledwie kilka prób, jednak ta metoda daje szansę rozwiązania problemu synchronizacji rozkładu jazdy w czasie podobnym do czasu jaki potrzebuje algorytm genetyczny. Prace z wykorzystaniem symulowanego wyżarzania zaprezentowano w tabeli 6.

Tabela 6. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą symulowanego wyżarzania w dokonanym przeglądzie literatury

Autorzy (rok publikacji)	Funkcja celu	Typ sieci	Część optymalizowanej sieci	Miejsce
Zhao i Zeng (2008)	Minimalizacja całkowitego kosztu	Autobusowa	Sieć testowa	-
Poorjafari, Yue i Holyoak (2014)	Minimalizacja czasu oczekiwania na przesiadkę	Wybrane węzły	Sieć testowa	-
Guo, Sun, Wu, Jin, Zhou i Gao (2017)	Maksymalizacja możliwości przesiadek	Metro	Sieć	Pekin (Chiny)

Źródło: (Kapica, et al., 2025)

W 2008 roku Zhao i Zeng zaproponowali model wykorzystujący symulowane wyżarzanie, algorytm zachłanny oraz przeszukiwane Tabu¹⁹ do zaplanowania nowej sieci i stworzenia dla niej rozkładu jazdy z określoną liczbą linii i pojazdów. W nowej sieci liczba podróży bez przesiadek wzrosła, spadła liczba podróży z jedną przesiadką, a liczba podróży z dwoma przesiadkami spadła do zera przy niezmienionej liczbie linii i pojazdów względem scenariusza bazowego.

Poorjafari, Yue i Holyoak w 2014 roku wykorzystali symulowane wyżarzanie do minimalizacji całkowitego czasu oczekiwania pasażerów na przesiadkę. Guo i in. w 2017 roku skupili się na optymalizacji rozkładu jazdy sieci metra w czasie zmiany częstotliwości kursowania pojazdów ze szczytowych na międzyszczytowe. Użyli oni kombinacji symulowanego wyżarzania i metody optymalizacji rojem cząstek²⁰ do maksymalizacji liczby możliwych przesiadek. Efektywność wybranej metody autorzy porównali do algorytmu

¹⁹ ang. Tabu Search

²⁰ ang. Particle Swarm Optimization

genetycznego oraz metody podziału i ograniczeń, za pomocą których udało im się uzyskać podobne wyniki, ale w dłuższym czasie od pierwotnej metody.

1.5.4 Synchronizacja rozkładów jazdy za pomocą innych metod

Poza powyższymi metodami badacze korzystali również z innych metod optymalizacyjnych do rozwiązania problemu synchronizacji rozkładów jazdy. Zostały one przedstawione w tabeli 7.

Tabela 7. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą innych metod w dokonanym przeglądzie literatury

Autorzy (rok publikacji)	Funkcja celu	Metoda	Typ sieci	Część optymalizowanej sieci	Miejsce
Klemt, Stemme (1988)	Minimalizacja czasu oczekiwania na przesiadkę	Heurystyczna	Metro	Sieć	Berlin (Niemcy)
Daduna, Voß (1995)	Minimalizacja czasu oczekiwania na przesiadkę	QSAP ²¹ i przeszukiwanie Tabu	Autobusowa	Sieć testowa	-
Jansen, Pedersen, Nielsen (2002)	Minimalizacja czasu oczekiwania na przesiadkę	Przeszukiwanie Tabu	Nie podano	Sieć	Kopenhaga (Dania)
Teodorović, Lučić (2005)	Minimalizacja czasu oczekiwania na przesiadkę	FAS ²²	Nie podano	Sieć testowa	-
Schröder, Solchenbach (2006)	Zwiększenie jakości przesiadek	QSAP	Autobusowa do kolejowej	Wybrane węzły	Kaiserslautern (Niemcy)
Wang, Shen (2007)	Minimalizacja liczby pojazdów obsługujących sieć	Algorytm mrówkowy ²³	Autobusowa	Sieć testowa	-
Liu, Shen, Wang, Yang (2007)	Minimalizacja czasu oczekiwania na przesiadkę	Przeszukiwanie Tabu	Autobusowa	Wybrane węzły	Nie podano
Guihaire, Jin-Kao (2008)	Minimalizacja liczby pojazdów obsługujących sieć i maksymalizacja możliwości przesiadek	LS ²⁴	Autobusowa	Nie podano	Nie podano

²¹ ang. Quadratic Semi-Assignment Problem

²² ang. Fuzzy Ant System

²³ ang. Ant Colony System

²⁴ ang. Local Search

Autorzy (rok publikacji)	Funkcja celu	Metoda	Typ sieci	Część optymalizowanej sieci	Miejsce
Hadas, Ceder (2010)	Minimalizacja czasu oczekiwania na przesiadkę i średniego czasu oczekiwania na przesiadkę	Programowanie dynamiczne ²⁵	Autobusowa	Wybrane linie	Nie podano
Chowdhury, Chien (2011)	Minimalizacja kosztów ponoszonych przez operatora i pasażerów	Metoda Powellsa ²⁶	Autobusowa do kolejowej	Wybrane linie	New Jersey Coast Line (USA)
Parbo, Nielsen Prato (2014)	Minimalizacja czasu oczekiwania na przesiadkę	Przeszukiwanie Tabu	Autobusowa	Sieć	Dania
Shen, Wang (2015)	Maksymalizacja liczby przyjazdów w oknie czasowym	Optymalizacja rojem cząstek	Autobusowa do metra	Węzeł	Wuhan (Chiny)
Liu, Ceder, (2017)	Minimalizacja liczby pojazdów obsługujących sieć i minimalizacja czasu oczekiwania na przesiadkę	Funkcja deficytu	Autobusowa	Wybrane węzły	Auckland (Nowa Zelandia)
Fonseca, van der Hurk, Roberti, Larsen (2018)	Minimalizacja kosztów ponoszonych przez operatora i pasażerów	Metaheurystyka	Autobusowa	Wybrane linie	Kopenhaga (Dania)
Gkiotsalitis, Maslekar (2018)	Minimalizacja czasu oczekiwania na przesiadkę	SHC ²⁷	Autobusowa	Wybrane linie	Sztokholm (Szwecja)
Shang, Liu (2019)	Minimalizacja kosztów ponoszonych przez operatora i pasażerów	Funkcja deficytu	Autobusowa	Wybrane linie	Pekin (Chiny)
Shang, Liu, Huang, Guo (2019)	Minimalizacja kosztów ponoszonych przez operatora i pasażerów	Funkcja deficytu	Autobusowa	Wybrane linie	Pekin (Chiny)
Abdolmalek, Masoud, Yin (2020)	Minimalizacja całkowitego czasu oczekiwania na przesiadkę	LS	Autobusowa	Sieć	Meshhed (Iran)

Źródło: (Kapica, et al., 2025)

²⁵ ang. Dynamic Programming

²⁶ ang. Powell's Method

²⁷ ang. Sequential Hill Climbing

Jak wskazano w tabeli 7. w 1988 roku Klemt i Stemme zaproponowali metodę heurystyczną do optymalizacji rozkładu jazdy zachodniobrzeńskiego metra. Sieć z tysiącem możliwości przesiadek była optymalizowana pod kątem zmniejszenia czasu oczekiwania na przesiadkę przez pasażerów przez ich algorytm w około minutę. Siedem lat później zbudowali model matematyczny oparty o QSAP, symulowane wyzarcanie i przeszukiwanie Tabu.

W 2005 roku Teodorović i Lučić wykorzystali kombinację algorytmu mrówkowego i logiki rozmytej²⁸, którą nazwali rozmytym algorytmem mrówkowym do minimalizacji całkowitego czasu oczekiwania na przesiadkę. Takie rozwiązanie pozwoliło na osiągnięcie o 2% lepszej wartości funkcji celu niż klasyczny algorytm mrówkowy. Schröder i Solchenbach wykorzystali QSAP do zwiększenia jakości przesiadek odczuwanej przez pasażerów. Zaproponowali klasyfikację przesiadek zależną od czasu oczekiwania.

Hadas i Ceder w 2010 roku użyli programowania dynamicznego do minimalizacji czasu przesiadki oraz średniego czasu oczekiwania na nią. Zaproponowali kilka strategii postępowania wdrażanych dynamicznie w zależności od aktualnej sytuacji w ruchu ulicznym. Wśród nich znalazły się: opóźnianie odjazdu pojazdu z przystanku węzłowego, zmiana prędkości pojazdu podczas przejazdu między przystankami, pomijanie przystanków na trasie lub skrócenie trasy. Pozwoliło to na zmniejszenie czasu oczekiwania pasażerów na przesiadkę o około 10% oraz wzrost liczby podróży bezpośrednich o kilkaset procent.

Shen i Wang w 2015 roku użyli metody PSO do maksymalizacji liczby synchronizacji przyjazdów autobusów dowożących pasażerów do metra w Wuhan w Chinach. Zaproponowali modyfikację metody poprzez przechowywanie ostatnich 10% najlepszych rozwiązań. W przypadku gdy po określonej liczbie iteracji wartość funkcji celu nie poprawiała się algorytm wybierał losowo jedno z przechowywanych rozwiązań i pracował dalej.

Liu i Ceder w 2017 roku, Shang i Liu oraz Shang wraz z zespołem w 2019 roku użyli funkcji deficytu do zachowania równowagi pomiędzy kosztami operacyjnymi operatora a czasem oczekiwania pasażerów na przesiadkę w Auckland w Nowej Zelandii²⁹ oraz w Pekinie

²⁸ ang. Fuzzy Logic

²⁹ Praca Liu i Cedera

w Chinach ³⁰ . Udowodnili oni, że zaproponowana przez nich metoda może pozwolić jednocześnie na zmniejszenie liczby zaangażowanych pojazdów do obsługi rozkładu jazdy jak i zmniejszenie czasu oczekiwania pasażerów na przesiadkę.

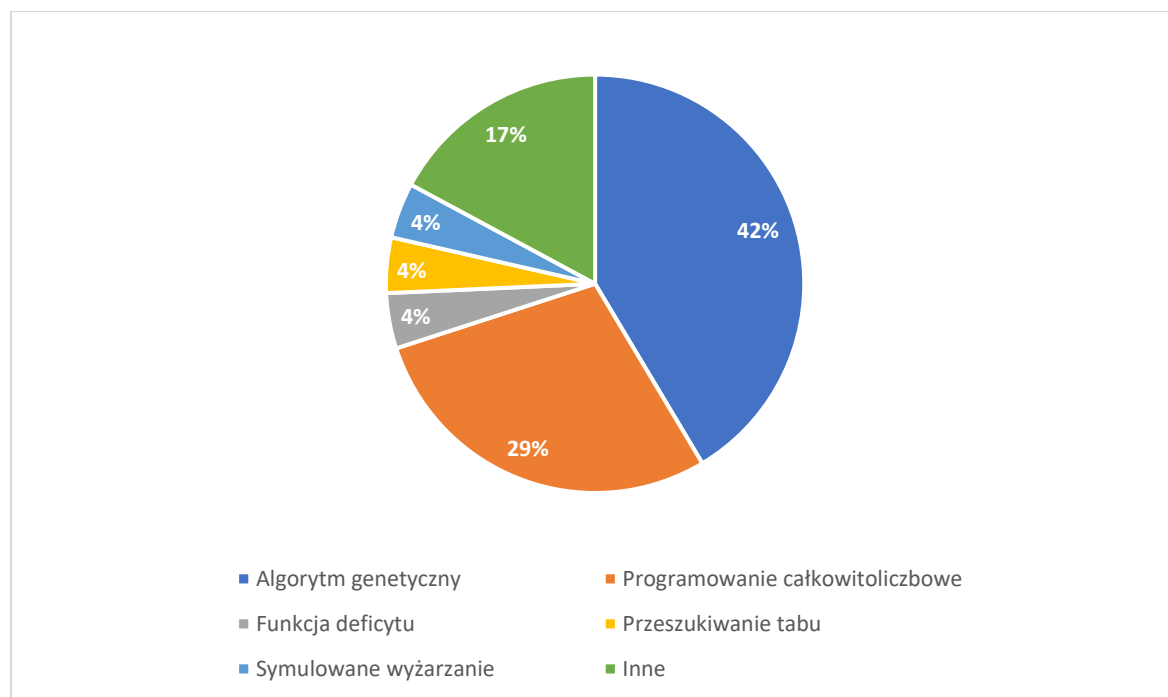
1.6 Podsumowanie przeglądu literatury przedmiotu

Do synchronizacji rozkładów jazdy badacze używali algorytmu genetycznego, programowania całkowitoliczbowego, funkcji deficytu, metody przeszukiwania tabu, symulowanego wyżarzania oraz innych wymienionych wcześniej metod. Procentowy udział poszczególnych metod w całej analizowanej literaturze przedstawiono na rys. 13. Zestawienie liczbowe publikacji przedstawiono w tabeli 8.

Tabela 8. Metody oraz liczba prac, w których wykorzystano je do synchronizacji rozkładów jazdy.

Metoda	Liczba prac
Algorytm genetyczny	29
Programowanie całkowitoliczbowe	20
Funkcja Deficytu	3
Przeszukiwanie Tabu	3
Symulowane wyżarzanie	3
Inne	12
Razem	70

Źródło: (Kapica, et al., 2025)



Rysunek 13. Procentowy udział metod prac dotyczących synchronizacji rozkładów jazdy w dokonanym przeglądzie literatury

³⁰ Prace Shanga i Liu oraz Shanga z zespołem

Źródło: (Kapica, et al., 2025)

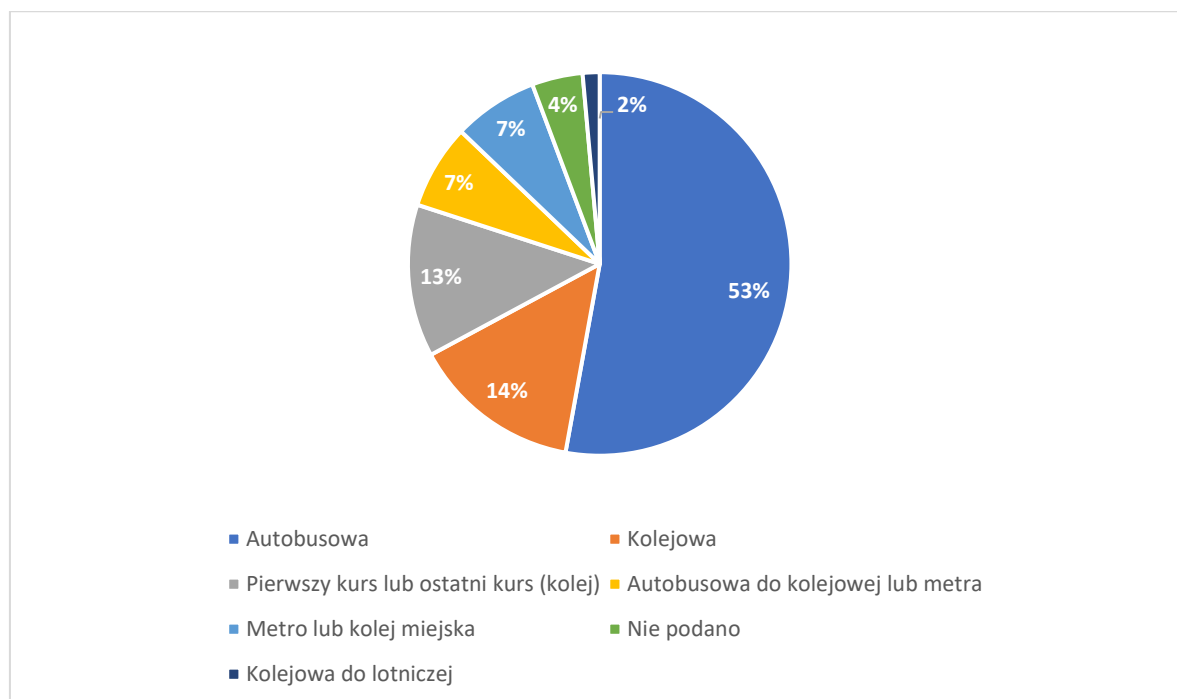
Zgodnie z rys. 13 i tabelą 8, w 42% prac (29 prac) do synchronizacji rozkładów jazdy wykorzystano algorytm genetyczny, w 29% (20 prac) sięgnięto po programowanie całkowitoliczbowe. Czteroprocentowym (po 3 prace) udziałem charakteryzują się: funkcja deficytu, przeszukiwanie Tabu oraz symulowane wyżarzanie. W pozostałych 17% prac (12 prac) wykorzystano inne metody.

Autorzy analizowanych prac podejmowali się synchronizacji sieci autobusowych, kolejowych, metra i kolei miejskich. Szczegółowe dane znajdują się w tabeli 9 oraz na rys. 14.

Tabela 9. Typy synchronizowanych sieci w dokonanym przeglądzie literatury

Typ sieci	Liczba prac
Autobusowa	37
Kolejowa	10
Pierwszy kurs lub ostatni kurs (kolej)	9
Autobusowa do kolejowej lub metra	5
Metro lub kolej miejska	5
Nie podano	3
Kolejowa do lotniczej	1
Suma	70

Źródło: (Kapica, et al., 2025)



Rysunek 14. Procentowy udział typów synchronizowanych sieci w dokonanym przeglądzie literatury

Źródło: (Kapica, et al., 2025)

Jak zaprezentowano w tabeli 9 i rys. 14, 37 analizowanych prac dotyczyło synchronizacji sieci autobusowych. Stanowi to 53% wszystkich prac. Dziesięć prac dotyczyło synchronizacji sieci kolejowej, a dziewięć dotyczyło specyficznej synchronizacji pierwszych lub ostatnich kursów w danej sieci.

W tabeli 10 w pierwszej kolumnie przedstawiono szczegółowe zestawienie liczby prac w zależności od wybranej metody i części synchronizowanej sieci. W drugiej i trzeciej kolumnie zestawiono prace z synchronizacją wykonaną za pomocą algorytmu genetycznego i programowania całkowitoliczbowego. Druga kolumna przedstawia liczebność prac w zależności od części synchronizowanej sieci, a trzecia w zależności od typu synchronizowanej sieci.

Tabela 10. Liczba prac w zależności od metody, części oraz typu synchronizowanej sieci w dokonanym przeglądzie literatury

Algorytm genetyczny	29	Algorytm genetyczny	29	Algorytm genetyczny	29
Sieć	19	Sieć	19	Autobusowa	15
Sieć	19	Autobusowa	9	Sieć	9
Wybrane linie	5	Pierwszy lub ostatni kurs	5	Wybrane linie	4
Węzeł	4	Kolejowa	3	Węzeł	2
Linia	1	Metro lub kolej miejska	2	Kolejowa	7
Programowanie całkowitoliczbowe	20	Wybrane linie	5	Sieć	3
Sieć	12	Autobusowa	4	Węzeł	2
Wybrane linie	4	Kolejowa	1	Wybrane linie	1
Ciąg komunikacyjny	2	Węzeł	4	Linia	1
Linia	1	Kolejowa	2	Pierwszy lub ostatni kurs	5
Węzeł	1	Autobusowa	2	Sieć	5
Inne	12	Linia	1	Metro lub kolej miejska	2
Sieć	5	Kolejowa	1	Sieć	2
Wybrane linie	4	Programowanie całkowitoliczbowe	20	Programowanie całkowitoliczbowe	20
Wybrane węzły	1	Sieć	12	Autobusowa	9
Węzeł	1	Autobusowa	4	Sieć	4
Nie opisano	1	Pierwszy lub ostatni kurs	4	Wybrane linie	3
Funkcja deficytu	3	Kolejowa	3	Ciąg komunikacyjny	1
Wybrane linie	2	Metro lub kolej miejska	1	Węzeł	1
Wybrane węzły	1	Wybrane linie	4	Pierwszy lub ostatni kurs	4
Przeszukiwanie Tabu	3	Autobusowa	3	Sieć	4
Sieć	2	Autobusowa do kolejowej lub metra	1	Kolejowa	3
Wybrane węzły	1	Ciąg komunikacyjny	2	Sieć	3
Symulowane wyżarzanie	3	Autobusowa do kolejowej lub metra	1	Autobusowa do kolejowej lub metra	2
Sieć	3	Autobusowa	1	Wybrane linie	1
		Linia	1	Ciąg komunikacyjny	1
		Kolejowa do lotniczej	1	Kolejowa do lotniczej	1
		Węzeł	1	Linia	1
		Autobusowa	1	Metro lub kolej miejska	1
				Sieć	1

Źródło: (Kapica, et al., 2025)

Zgodnie z danymi w tabeli 10. dziewiętnaście z dwudziestu dziewięciu prac, w których badacze użyli algorytmu genetycznego dotyczyło synchronizacji całych sieci. Pięć prac dotyczyło synchronizacji wybranych linii, cztery rozkładów jazdy w wybranym węźle, a jedna praca dotyczyła synchronizacji jednej linii. Piętnaście z dwudziestu dziewięciu prac dotyczyło synchronizacji rozkładów jazdy autobusów, siedem kolei, pięć synchronizacji pierwszych lub ostatnich kursów, a dwie metra lub kolei miejskiej.

Można zatem stwierdzić, że algorytm genetyczny był najczęściej wybieraną metodą przez badaczy do synchronizacji rozkładów jazdy, a najczęściej za jego pomocą synchronizowano rozkłady jazdy autobusów kursujących po całej sieci analizowanego obszaru.

2. Rozdział II

2.1 Model sieci publicznego transportu zbiorowego

Na najwyższym poziomie sieć publicznego transportu zbiorowego Ω można przedstawić jako zbiór zawierający zbiór Λ zawierający w sobie linie publicznego transportu zbiorowego oraz zbiór \mathbf{D} zawierający w sobie pasażerów korzystających z publicznego transportu zbiorowego w celu zaspokojenia swoich potrzeb transportowych.

$$\Omega = \{\Lambda, \mathbf{D}\} \quad (4)$$

Linia publicznego transportu zbiorowego λ_i , gdzie $\lambda_i \in \Lambda$, w modelu matematycznym jest przedstawiana jako zbiór, gdzie elementami są zbiór \mathbf{L}_i zawierający odcinki, z których składa się i -ta linia oraz zbiór \mathbf{V}_i zawierający pojazdy obsługujące i -tą linię:

$$\lambda_i = \{\mathbf{L}_i, \mathbf{V}_i\}, i = 1 \dots N_A, \quad (5)$$

gdzie:

N_A to liczba linii publicznego transportu zbiorowego.

Elementy zbioru \mathbf{L}_i określają punkty początkowe n i końcowe m , którymi są przystanki oraz ich waga w , która odpowiada długości każdego odcinka l .

$$l_{ij} = \{n_{ij}, m_{ij}, w_{ij}\}, l_{ij} \in \mathbf{L}_i, j = 1 \dots N_{L(i)}, \quad (6)$$

gdzie:

l_{ij} jest j -tym odcinkiem i -tej linii;

n_{ij} oraz m_{ij} są początkowym i końcowym przystankiem j -tego odcinka, $n_{ij}, m_{ij} \in \mathbf{N}_i$;

w_{ij} jest wagą każdego j -tego odcinka trasy linii wyrażoną w kilometrach;

$N_{L(i)}$ to numer odcinka międzyprzystankowego i -tej linii;

\mathbf{N}_i jest zbiorem wszystkich przystanków i -tej linii.

Pojazd v_{ij} jest elementem zbioru \mathbf{V}_i ($v_{ij} \in \mathbf{V}_i, i = 1 \dots N_{(V)i}$, gdzie $N_{(V)i}$ jest liczbą pojazdów obsługujących i -tą linię), który jest charakteryzowany poprzez pojemność oraz rozkład jazdy na i -tej linii:

$$v_{ij} = \{c_{ij}, s_{ij}\},$$

(7)

gdzie:

c_{ij} jest pojemnością j -tego pojazdu wyrażoną w liczbie pasażerów, a s_{ij} rozkładem jazdy j -tego pojazdu.

Rozkład jazdy może być opisany (jego k -ta pozycja) dla j -tego pojazdu obsługującego i -tą linię jako krotka $s_{ijk} = \langle p_{ijk}, t_{ijk} \rangle$, gdzie p_{ijk} to pozycja w rozkładzie jazdy (przystanek na trasie linii), a t_{ijk} to moment przyjazdu j -tego pojazdu na przystanek p_{ijk} . W związku z tym rozkład s_{ij} j -tego pojazdu na i -tej linii można zaprezentować jako zbiór krotek:

$$s_{ij} = \bigcup_{k=1}^{N_{s(ij)}} s_{ijk} = \bigcup_{k=1}^{N_{s(ij)}} \langle p_{ijk}, t_{ijk} \rangle, t_{ijk} < t_{ij(k+1)}, \quad (8)$$

gdzie:

$N_{s(ij)}$ to numer pozycji w sekwencji przystanków, na których aktualnie znajduje się j -ty pojazd wykonujący przewozy na i -tej linii.

Moment przyjazdu na pierwszy przystanek w sekwencji przystanków w rozkładzie jazdy j -tego pojazdu na i -tej linii jest opisany następująco:

$$t_{ij(1)} = t_0 + \Psi_i + \Delta t_{ij}, \quad (9)$$

gdzie:

t_0 to czas, w którym system transportu zbiorowego rozpoczyna obsługę pasażerów [min];

Ψ_i to przesunięcie rozpoczęcia kursowania i -tej linii [min];

Δt_{ij} to przesunięcie rozpoczęcia kursowania j -tego pojazdu na i -tej linii [min].

Popyt na usługi przewozowe przedstawiany jest jako zbiór elementów opisujących pasażerów zamierzających skorzystać z systemu publicznego transportu zbiorowego. Element tego zbioru można opisać na podstawie kilku parametrów:

$$\pi_i = \{\eta_i, \mu_i, \mathbf{P}_i, \tau_i\}, \pi_i \in \mathbf{D}, i = 1 \dots N_D, \quad (10)$$

gdzie:

π_i to i -ty pasażer, η_i oraz μ_i to przystanki, na których i -ty pasażer rozpoczyna i kończy podróż,

$\eta_i \in \mathbf{N}, \mu_i \in \mathbf{N};$

\mathbf{P}_i jest zbiorem przystanków, na których i -ty pasażer przesiada się między liniami w czasie podróży;

τ_i jest czasem, w którym i -ty pasażer pojawia się na przystanku η_i w celu wykonania podróży [min];

N_D jest liczbą wszystkich pasażerów korzystających z systemu publicznego transportu zbiorowego [pas].

W celu symulowania popytu na przewozy podzielono wszystkie elementy zbioru \mathbf{D} na grupy według przystanków, na których rozpoczynana jest podróż:

$$\mathbf{D} = \bigcup_{j=1}^{N_L+1} \mathbf{D}_j, \quad (11)$$

gdzie:

\mathbf{D}_j jest grupą pasażerów rozpoczynających podróż na j -tym przystanku i -tej linii:

$$\mathbf{D}_j = \{\pi_i: \eta_i = j\}. \quad (12)$$

Odstęp czasu ξ_j pomiędzy pojawieniem się pasażerów na j -tym przystanku jest zmienną losową i w modelu traktowany jest jako cecha przystanku. Następnie dla każdej grupy \mathbf{D}_j parametry τ_i poszczególnych elementów zbioru mogą być zdefiniowane na podstawie realizacji zmiennej losowej ξ_j odstępów czasu pomiędzy pojawieniem się pasażerów na j -tym przystanku:

$$\tau_i = \begin{cases} \xi_j, & i = 1, \eta_i = j, \\ \tau_{i-1} + \xi_j, & i > 1, \eta_i = j, \end{cases} \quad (13)$$

gdzie:

ξ_j jest realizacją zmiennej losowej odstępów czasu pomiędzy pojawieniem się pasażerów na j -tym przystanku [min] (Naumov & Samchuk, 2017).

2.2 Ustalenie zapotrzebowania na przesiadki pasażerów

Zapotrzebowanie na dokonanie przesiadki w węzłach przesiadkowych z wykorzystaniem zbioru \mathbf{D} można określić w następujący sposób:

- dla każdego π_i określana jest trasa od przystanku początkowego do przystanku końcowego; wybór trasy może być dokonany za pomocą różnych kryteriów i metod, najprostszą metodą jest wyznaczenie najkrótszej trasy za pomocą algorytmu Dijkstry,
- dla otrzymanej trasy wyznacza się zbiór przystanków P_i , na których możliwa jest przesiadka; wyznacza się część wspólną zbioru wszystkich przystanków, na których można się przesiąść w sieci i zbioru przystanków na trasie pasażera z pominięciem sekwencji przystanków zależnych od jednej linii,
- dla elementów zbioru π_i z niepustym zbiorem P_i obliczana jest liczba możliwych przystanków przesiadkowych.

Ostatecznie, każdy przystanek przesiadkowy w sieci publicznego transportu zbiorowego można scharakteryzować liczbą pasażerów, którzy dokonują na nim przesiadki (Naumov, 2019).

2.3 Sformułowanie problemu synchronizacji rozkładu jazdy

Zmiennymi wejściowymi w problemie synchronizacji rozkładu jazdy w ramach modelu wykorzystanego w pracy są: liczba pojazdów obsługujących linie publicznego transportu zbiorowego, pojemność pojazdów oraz parametry numeryczne rozkładu jazdy - przesunięcia czasowe Ψ_i oraz Δt_{ij} .

Synchronizację rozkładów jazdy można osiągnąć poprzez ustawienie wartości przesunięć rozpoczęcia obsługi linii publicznego transportu zbiorowego w istniejącym systemie publicznego transportu zbiorowego. Przesunięcia te można przedstawić jako wektor Ψ wyrażony liczbami całkowitymi:

$$\Psi = \langle \Psi_1, \Psi_2 \dots \Psi_{N_A} \rangle.$$

(14)

Podstawowym kryterium efektywności oceniającym wynik synchronizacji rozkładu jazdy jest łączny czas oczekiwania pasażerów na przystankach. Celem synchronizacji rozkładu jazdy jest minimalizacja łącznego czasu oczekiwania pasażerów na przystankach.

$$T_W(\Psi) = \sum_{i=1}^{N_D} t_{wi} \rightarrow \min,$$

(15)

gdzie:

t_{wi} to czas oczekiwania i -tego pasażera publicznego transportu zbiorowego [min] (Naumov, 2017).

2.4 Implementacja modelu sieci publicznego transportu zbiorowego

Do modelowania funkcjonowania procesów publicznego transportu zbiorowego w niniejszej pracy wykorzystano ogólnodostępną bibliotekę napisaną w języku Python, stworzoną przez dr. hab. Vitalia Naumova, i udostępnioną w repozytorium GitHub (Naumov, 2021). Biblioteka składa się z trzech pakietów, zawierających klasy bazowe. Oryginalne oprogramowanie bazowe zostało zmodyfikowane na potrzeby niniejszego doktoratu. Wykorzystane w ramach danej pracy pakiety, klasy oraz zależności zachodzące pomiędzy nimi zostały opisane poniżej.

2.4.1 Pakiet *transportnet*

Najobszerniejszym pakietem jest pakiet *transportnet*. Zostało w nim zdefiniowane sześć klas tożsamyh z elementami sieci publicznego transportu zbiorowego: *Net*, *Node*, *Link*, *Line*, *Vehicle* i *Passenger*.

2.4.1.1 Klasa *Net*

Klasa *Net* stanowi centralny komponent modelu symulacyjnego odpowiedzialny za reprezentację i zarządzanie całością sieci transportowej. Jest ona agregatem wszystkich elementów infrastruktury (węzłów, odcinków, linii), pasażerów, pojazdów oraz mechanizmów kontroli i analizy symulacji. Przechowuje globalne parametry symulacji, steruje cyklem jej działania, umożliwia wprowadzanie zmian w infrastrukturze symulowanej sieci oraz monitorowanie wskaźników wydajności i jakości obsługi użytkowników systemu.

2.4.1.1.1 Atrybuty klasy *Net*

Atrybutami klasy *Net* są:

- *time* – reprezentuje aktualny czas trwania symulacji,
- *duration* – reprezentuje całkowity czas trwania symulacji,
- *nodes* – przechowuje listę obiektów typu *Node*, tworzących graf przestrzenny sieci,
- *links* – przechowuje listę obiektów *Link*, łączących poszczególne węzły w jedno- lub dwukierunkowy graf,
- *lines* – przechowuje listę obiektów *Line*, definiujących przebiegi tras linii publicznego transportu zbiorowego,

- *demand* jest zbiorem wszystkich pasażerów generowanych w trakcie symulacji,
- *total_wait_time* – reprezentuje łączny czas oczekiwania wszystkich pasażerów na przystankach,
- *sum_vehicles_time* – reprezentuje łączny czas pracy wszystkich pojazdów,
- *num_serviced_passengers* – reprezentuje liczbę pasażerów, którzy odbyli podróż w symulacji.

2.4.1.1.2 Metody klasy *Net*

W klasie *Net* wykorzystywane są następujące metody:

- *contains_node* – sprawdza, czy w sieci istnieje węzeł o określonym kodzie. Umożliwia weryfikowanie istnienia elementów sieci przed wykonaniem operacji na nich,
- *get_node* – zwraca obiekt węzła (*Node*) o wskazanym kodzie, jeżeli istnieje w kolekcji węzłów, w przeciwnym razie zwraca *None*. Stanowi narzędzie do pobierania referencji do elementów grafu,
- *contains_link* – sprawdza, czy w sieci istnieje połączenie (obiekt *Link*) skierowane od zadanego węzła początkowego do węzła docelowego,
- *get_link* – zwraca obiekt połączenia (*Link*) skierowanego od danego węzła początkowego do zadanego węzła docelowego,
- *add_link* – dodaje nową krawędź do grafu sieci, łącząc ze sobą istniejące lub automatycznie tworząc nowe węzły o zadanych kodach. Jeśli połączenie już istnieje, aktualizuje jego wagę,
- *dijkstra* – implementuje algorytm najkrótszej ścieżki Dijkstry. Dla zadanego węzła startowego oblicza najkrótsze ścieżki do wszystkich pozostałych węzłów na podstawie wagi połączeń,
- *define_path* – na podstawie wyników metody *dijkstra* konstruuje sekwencję węzłów tworzącą najkrótszą trasę od węzła początkowego do docelowego,
- *define_destinations* – zwraca listę węzłów pełniących funkcję przesiadek w trasie z węzła rozpoczynającego do docelowego. Bierze pod uwagę punkty przesiadkowe oraz istniejący układ linii,
- *gen_demand* – generuje zapotrzebowanie na podróże w sieci w zadanym okresie. Przy opcji stochastycznej momenty pojawiania się pasażerów są losowe (zgodnie z rozkładem losowym określonym w każdym węźle), a cele podróży dobierane

losowo spośród wszystkich węzłów. Przy trybie deterministycznym wykorzystywana jest alternatywna logika oparta na parach źródło-cel,

- *simulate* – metoda realizująca symulację ruchu w sieci. W jej ramach:
 - generowany jest popyt wśród pasażerów,
 - ustalane są rozkłady jazdy dla wszystkich linii i pojazdów,
 - inicjowane i wykonywane są cykliczne operacje ruchu pojazdów oraz obsługi pasażerów we wszystkich przystankach,
 - agregowane są główne wskaźniki efektywności, takie jak łączny czas oczekiwania, liczba obsłużonych pasażerów oraz liczba nieobsłużonych pasażerów.

2.4.1.2 Klasa *Node*

Klasa *Node* służy do modelowania pojedynczego węzła sieci transportowej. W niniejszej pracy węzeł odwzorowuje przystanek. Klasa *Node* odpowiedzialna jest za utrzymanie informacji o relacjach topologicznych oraz dynamicznych aspektach funkcjonowania sieci (np. obecności pasażerów).

2.4.1.2.1 Atrybuty klasy *Node*

Atrybutami klasy *Node* są:

- *code* – przechowuje unikalny identyfikator węzła,
- *name* – przechowuje opisową nazwę węzła (np. nazwę przystanku),
- *out_links* – lista wychodzących linków, pozwalających prowadzić ruch z bieżącego węzła do innych,
- *in_links* – lista linków prowadzących do danego węzła z innych lokalizacji,
- *s_interval* – wartość generowana przez klasę *Stochastic* niezbędna do wygenerowania losowo pojawiających się pasażerów w sieci,
- *pass_in* – lista pasażerów oczekujących na odjazd z danego węzła,
- *pass_out* – lista pasażerów, którzy wysiedli w danym węźle.

2.4.1.3 Klasa *Link*

Obiekty typu *Link* stanowią formalną reprezentację krawędzi w grafie sieci. Odzwierciedlają one infrastrukturę transportową umożliwiając przemieszczanie się pojazdów i pasażerów pomiędzy dwoma węzłami.

2.4.1.3.1 Atrybuty klasy *Link*

Atrybutami klasy *Link* są:

- *in_node* – zawiera informację o węźle, do którego prowadzi obiekt typu *Link*,
- *out_node* – zawiera informację o węźle, z którego prowadzi obiekt typu *Link*,
- *weight* – zawiera informację o wadze krawędzi w grafie, w przypadku niniejszej pracy wagę reprezentuje odległość pomiędzy dwoma obiektami typu *Node*.

2.4.1.4 Klasa *Line*

Klasa *Line* odzwierciedla linię publicznego transportu zbiorowego.

2.4.1.4.1 Atrybuty klasy *Line*

Atrybutami klasy *Line* są:

- *net* – odniesienie do obiektu *Net* zapewniające powiązanie z siecią nadrzędną,
- *nodes* – przechowuje listę wszystkich węzłów obsługiwanych przez linię, ułożonych w kolejności ich obsługi,
- *vehicles* – przechowuje listę pojazdów obsługujących daną linię,
- *schedule_shift* – parametr określający przesunięcie czasu rozpoczęcia kursu,
- *end_stop_duration* – parametr reprezentujący minimalny czas postoju od zakończenia kursu do rozpoczęcia następnego,
- *intermediate_stop_duration* – parametr reprezentujący czas postoju na każdym z przystanków,
- *velocity* – parametr reprezentujący średnią prędkość pojazdów obsługujących linię.

2.4.1.4.2 Metody klasy *Line*

W klasie *Line* wykorzystywane są następujące metody:

- *trace_string* – generuje ciąg tekstowy opisujący trasę linii poprzez listę kodów kolejnych obiektów typu *Node*,
- *nodes_reversed* – zwraca listę przystanków linii w porządku odwrotnym. Metoda jest wykorzystywana do symulacji kursów powrotnych,
- *define_sequence* – tworzy pełną kolejność obsługiwanych przystanków (składającą się z kursów w obu kierunkach),
- *define_schedule* – definiuje rozkład jazdy każdego pojazdu obsługującego daną linię,
- *add_vehicles* – przypisuje pojazdy do obsługi linii,

- *run* - symuluje ruch po sieci wszystkich pojazdów przypisanych do linii zgodnie z przygotowanym rozkładem jazdy.

2.4.1.5 Klasa *Vehicle*

Obiekty klasy *Vehicle* reprezentują pojazdy poruszające się po sieci na przypisanych do nich liniach. Pozwalają przemieszczać się po sieci pasażerom.

2.4.1.5.1 Atrybuty klasy *Vehicle*

Atrybutami klasy *Vehicle* są:

- *line* – przechowuje numer linii, której kursy obsługuje dany pojazd,
- *schedule* – lista zaplanowanych do obsługi przystanków,
- *last_move* – przechowuje czas w jakim pojazd wykonał ostatni ruch w sieci,
- *passengers* – lista przewożonych aktualnie pasażerów,
- *serviced_passengers* – lista zawierająca przewiezionych już pasażerów,
- *capacity* – przechowuje informację o pojemności pojazdu.

2.4.1.5.2 Metody klasy *Vehicle*

Metody zawarte w tej klasie pozwalają na obsługę pasażerów w węzłach przystankowych oraz na poruszanie się pojazdów po sieci. W klasie *Vehicle* wykorzystywane są następujące metody:

- *occupancy* – metoda zwraca aktualną liczbę pasażerów w pojeździe,
- *current_position* – metoda zwraca bieżącą pozycję względem zaplanowanej trasy,
- *stops_left* – metoda zwraca listę pozostałych do obsługi przystanków,
- *set_passengers* – metoda realizując obsługę pasażerów wysiadających na przystankach,
- *get_passengers* – metoda realizując obsługę pasażerów wsiadających na przystankach,
- *move* – metoda pozwala na symulację procesów poruszania się pojazdu po sieci.

2.4.1.6 Klasa *Passenger*

Obiekty klasy *Passenger* odzwierciedlają pasażerów poruszających się w sieci publicznego transportu zbiorowego.

2.4.1.6.1 Atrybuty klasy *Passenger*

Atrybutami klasy *Passenger* są:

- *m_appearance* – reprezentuje czas pojawienia się pasażera w sieci, jeszcze przed wejściem po pojeździe,
- *m_boarding* – zawiera listę czasów wejścia do pojazdów,
- *m_disembarkation* – zawiera listę czasów wyjścia z pojazdów,
- *origin_node* – reprezentuje numer węzła, w którym pasażer rozpoczął podróż,
- *destination_nodes* – zawiera listę celów podróży tj. obiektów klasy *Node*,
- *current_destination_node* – reprezentuje aktualny węzeł docelowy pasażera,
- *used_vehicles* – zawiera listę pojazdów wykorzystanych przez pasażera podczas podróży.

2.4.2 Pakiet *genetics*

2.4.2.1 Klasa *Chromosome*

Klasa *Chromosome* reprezentuje pojedynczego osobnika w populacji algorytmu genetycznego. Jest zbiorem genów - każda sekwencja tych genów definiuje potencjalne rozwiązanie analizowanego problemu synchronizacji rozkładów jazdy w publicznym transporcie zbiorowym.

2.4.2.1.1 Atrybuty klasy *Chromosome*

Atrybutami w klasie *Chromosome* są:

- *genes* – zawiera listę z aktualnymi wartościami genów kodujących potencjalne rozwiązania,
- *size* – krotka, zawierająca liczbę linii oraz liczbę analizowanych parametrów,
- *length* – reprezentuje całkowitą liczbę genów w analizowanym rozwiązaniu.

2.4.2.1.2 Metody klasy *Chromosome*

Metody używane w klasie *Chromosome* to:

- *decode* – przekształca wewnętrzną reprezentację chromosomu tj. z postaci binarnej na postać odpowiadającą rzeczywistemu rozwiązaniu problemu, czyli przesunięcie rozpoczęcia kursowania wyrażone w pełnych minutach,
- *evaluate* – oblicza wartość funkcji celu na podstawie aktualnej wartości genów,
- *mutate* – wprowadza losowe zmiany w pojedynczych genach,
- *replicate* – tworzy nowy chromosom na podstawie dwóch innych.

2.4.2.2 Klasa *Population*

Klasa *Population* reprezentuje zbiór chromosomów w danej generacji algorytmu genetycznego. Zarządza inicjalizacją populacji, ewolucją (selekcją, krzyżowaniem, mutacją) oraz przechowuje bieżące oceny dopasowania.

2.4.2.2.1 Atrybuty klasy *Population*

Atrybutami w klasie *Population* są:

- *size* – reprezentuje liczbę osobników w populacji,
- *chromosome_size* – reprezentuje rozmiar tj. liczbę genów w pojedynczym chromosomie,
- *fitness_function* – wartość funkcji celu obliczona dla każdego osobnika w populacji,
- *maximize* – zmienna typu logicznego określająca czy wartość funkcji celu ma dążyć do jak najwyższych lub jak najniższych wartości,
- *surivors_rate* – reprezentuje odsetek osobników jaki ma przeżyć i przejść do kolejnej generacji algorytmu,
- *crossover_probability* – reprezentuje prawdopodobieństwo krzyżowania,
- *mutation_probability* – reprezentuje prawdopodobieństwo wystąpienia mutacji,
- *mutation_turns* – reprezentuje maksymalną liczbę genów pojedynczego osobnika jakie mogą zostać poddane mutacji,
- *chromosomes* – zawiera listę obiektów typu *Chromosome* należących do aktualnej populacji.

2.4.2.2.2 Metody klasy *Population*

Metody używane w klasie *Population* to:

- *initiate* – wykorzystywana do wygenerowania losowej populacji startowej,
- *evolve* – cyklicznie ocenia osobniki w populacji pod względem wartości funkcji celu, dokonuje selekcji najlepszych osobników, steruje procesem krzyżowania i mutacji osobników.

2.4.2.3 Klasa *GA (Genetic Algorithm)*

Klasa *GA* jest obiektem sterującym pełnym procesem działania algorytmu genetycznego. Zarządza rozmiarem populacji, liczbą generacji, ewaluacją rozwiązań oraz szczegółowymi parametrami mutacji i krzyżowania.

2.4.2.3.1 Atrybuty klasy GA

Atrybutami obiektów klasy GA są:

- *chromosome_size* – reprezentuje rozmiar chromosomu,
- *population_size* – reprezentuje liczbę chromosomów w czasie każdej iteracji,
- *generations* – reprezentuje liczbę generacji algorytmu,
- *fitness_function* – funkcja celu dla rozważanego problemu,
- *maximize* – zmienna typu logicznego określająca czy wartość funkcji celu ma dążyć do jak najwyższych lub jak najniższych wartości,
- *surivors_rate* – reprezentuje odsetek osobników jaki ma przeżyć i przejść do kolejnej generacji algorytmu,
- *crossover_probability* – reprezentuje prawdopodobieństwo krzyżowania,
- *mutation_probability* – reprezentuje prawdopodobieństwo wystąpienia mutacji,
- *mutation_turns* – reprezentuje maksymalną liczbę genów pojedynczego osobnika jakie mogą zostać poddane mutacji.

Wartości powyższych atrybutów przekazywane są do klasy *Population*.

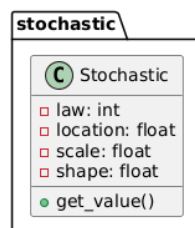
2.4.2.3.2 Metody klasy GA

W klasie GA używana jest jedna metoda:

- *run* – kontroluje i uruchamia działanie algorytmu poprzez cykliczne wywołanie metody *evolve* z klasy *Population* oraz pojedyncze wywołanie metody *initate* z klasy *Population*.

2.4.3 Pakiet *stochastic*

W ramach pakietu *stochastic* zamodelowana została tylko jedna klasa o tej samej nazwie, która jest implementacją generatora zmiennych losowych. Atrybuty i metody pakietu i klasy zostały przedstawione na rys. 15.



Rysunek 15. Atrybuty i metody w pakiecie i klasie *Stochastic*
Źródło: Opracowanie własne

2.4.3.1 Atrybuty klasy *Stochastic*

Atrybutami obiektów klasy *Stochastic* są:

- *law* – reprezentuje typ wybranego rozkładu prawdopodobieństwa,
- *location* – reprezentuje położenie rozkładu prawdopodobieństwa,
- *scale* – reprezentuje rozproszenie rozkładu prawdopodobieństwa,
- *shape* – reprezentuje parametr kształtu pozwalający na symulowanie bardziej złożonych rozkładów losowych.

2.4.3.2 Metody klasy *Stochastic*

W klasie *Stochastic* używana jest tylko jedna metoda, którą jest:

- *get_value* – pozwala na uzyskanie wartości losowej zgodnie z zadaniem typem rozkładu prawdopodobieństwa i parametrami.

2.4.4 Relacje w pakiecie *transportnet*

Relacje pomiędzy klasami pakietu *transportnet* tworzą rozbudowaną strukturę, która pozwala na całościowe i dynamiczne modelowanie systemu transportu zbiorowego jako grafu, wraz z symulacją ruchu w jego obrębie, obsługą popytu i ewolucją stanu podczas symulacji. Rolą klasy *Net* jest zarządzanie całą siecią publicznego transportu zbiorowego. Zgodnie z rys. 16, przechowuje ona i zarządza głównymi komponentami:

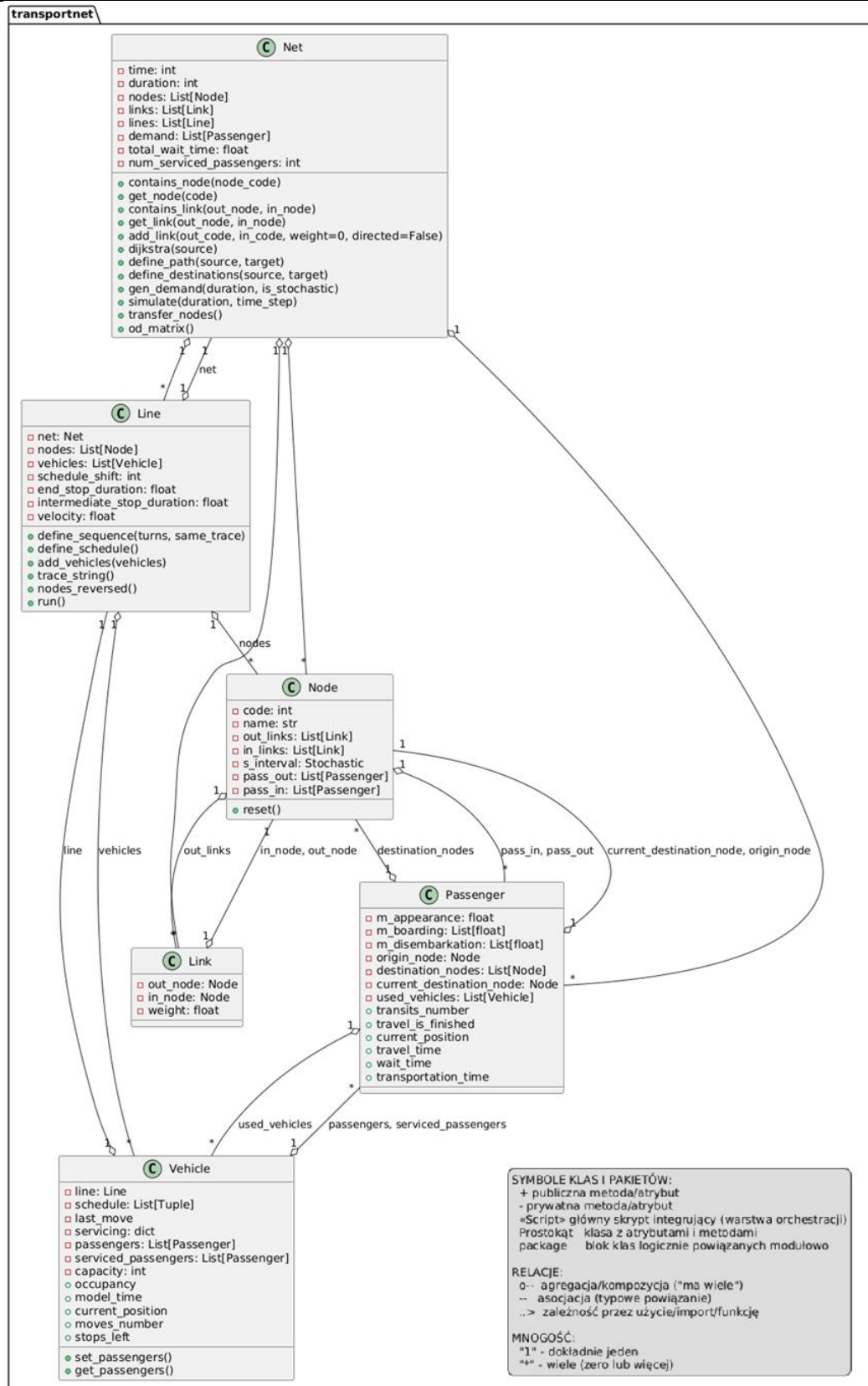
- listą węzłów (*nodes* – obiekty klasy *Node*),
- listą połączeń (*links* – obiekty klasy *Link*),
- trasami (*lines* – obiekty klasy *Line*),
- popytem (*demand* – czyli obiekty klasy *Passenger*).

Klasa *Net* agreguje również wyniki symulacji (czas oczekiwania pasażerów, liczbę obsłużonych pasażerów) oraz przekazuje do podrzędnych obiektów parametry globalne.

Klasa *Node* reprezentuje węzeł przystankowy, którego argumentem są listy wychodzących i dochodzących do niego linków (*out_links*, *in_links* - obiekty klasy *Link*). Węzeł jest także miejscem do obsługi popytu. Do niego przypisane są listy pasażerów oczekujących na odjazd (*pass_out*) i pasażerów, którzy do danego węzła przybyli (*pass_in*). Dzięki temu przepływ pasażerów przez sieć może być analizowany i śledzony. Do każdego węzła przypisana jest

także cecha stochastyczna (*s_interval*), dyktująca interwały w jakich pojawiają się w nim kolejni rozpoczynający swoją podróż pasażerowie.

Obiekty klasy *Link* są połączeniem pomiędzy dwoma konkretnymi węzłami sieci (*out_node*, *in_node*). Link istnieje wyłącznie w kontekście tej pary węzłów. Atrybutem linku jest jego waga (*weight*), czyli odległość pomiędzy węzłami, które są dzięki niemu połączone.



Rysunek 16. Atrybuty i metody w klasach należących do pakietu *transportnet* oraz relacje pomiędzy nimi
 Źródło: Opracowanie własne

Każdy obiekt klasy *Line* (linia publicznego transport zbiorowego) jest przypisany do danej sieci (obiekt klasy *Net*) i przechowuje uporządkowaną listę węzłów (obiekty klasy *Node*), które wyznaczają jej trasę. Do linii przypisana jest lista pojazdów (obiekty klasy *Vehicle*), które kursują na jej trasie. Każdy pojazd ma przypisaną linię (obiekty klasy *Line*) w ramach, której wykonuje kursy. Dzięki temu zachowany jest silny związek: linia agreguje pojazdy, a pojazd posiada referencję do linii.

Pojazdy (obiekty klasy *Vehicle*) przechowują szereg własności związanych z poruszaniem się (harmonogram – *schedule*, czasowy zapis ostatniego ruchu – *last_move*) oraz obsługą pasażerów. Do obiektów klasy *Vehicle* przypisane są dwie listy: aktualnie obsługiwanych pasażerów (*passengers*) i pasażerów, których przejazd pojazdem już się zakończył (*serviced_passengers*). W każdym przypadku są to obiekty klasy *Passenger*. Pojazd jest zawsze związany z konkretną linią, ale jedna linia może mieć wiele pojazdów.

Pasażer (obiekt klasy *Passenger*) przypisany jest do węzła początkowego podróży (*origin_node*), ma zaplanowaną sekwencję odwiedzanych węzłów (*destination_nodes*) oraz zawsze wie, do jakiego węzła obecnie zmierza (*current_destination_node*). W trakcie podróży pasażer gromadzi również listę pojazdów, którymi się przemieszczał (*used_vehicles*).

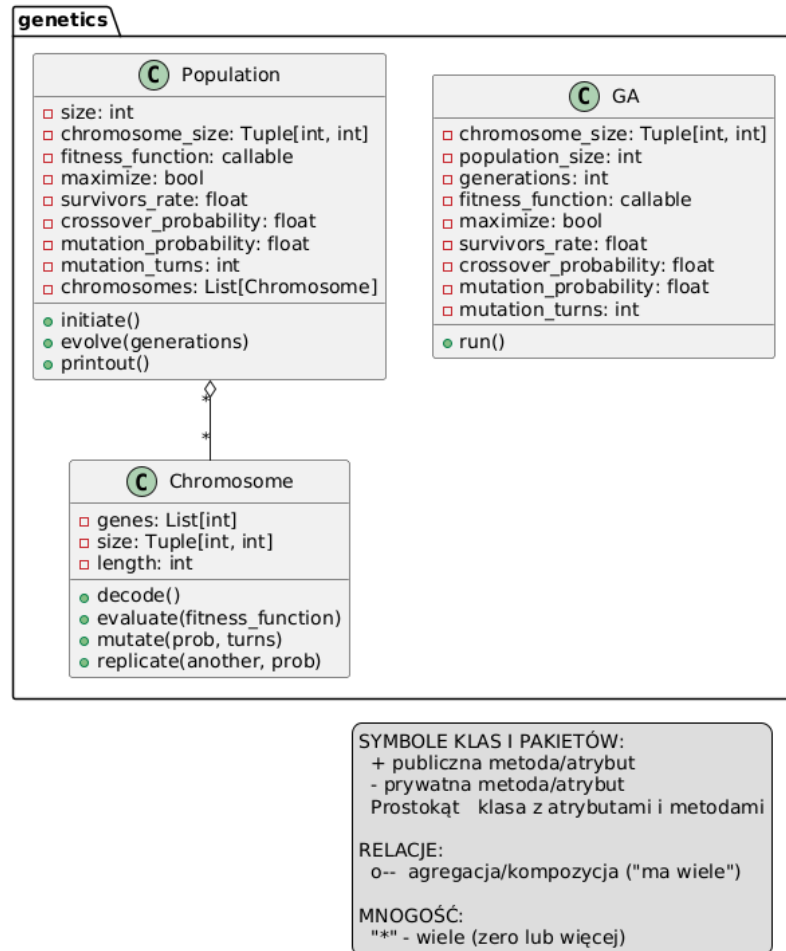
2.4.4.1 Kierunki i rodzaje zależności

Większość relacji pomiędzy obiektami ma charakter agregacyjny np. lista pojazdów w linii czy lista pasażerów w węźle. Kluczowe są także następujące wzajemne zależności:

- obiekty klasy *Vehicle* odnoszą się do obiektów klasy *Line*,
- obiekty klasy *Line* odnoszą się do obiektów klasy *Net*,
- obiekty klasy *Node* oraz obiekty klasy *Link* odnoszą się do siebie nawzajem,
- obiekty klasy *Passenger* odnoszą się do obiektów klasy *Node* i klasy *Vehicle*.

2.4.5 Relacje w pakiecie *genetics*

Pakiet *genetics* stanowi spójną strukturę hierarchiczną, która modeluje procesy ewolucyjne algorytmu genetycznego. Relacje w tym pakiecie zostały przedstawione na rys. 17.



Rysunek 17. Atrybuty i metody w klasach należących do pakietu Genetics oraz relacje pomiędzy nimi
 Źródło: Opracowanie własne

Jak zaprezentowano na rys. 17, obiekty klasy *Chromosome* kodują jedno potencjalne rozwiązanie problemu w postaci sekwencji genów. Chromosomy są obiektami autonomicznymi, przygotowanymi do operacji takich jak: mutacja, krzyżowanie czy ocena wartości funkcji celu. Przedstawiona na rys.17 relacja klas *Chromosome–Population* ma charakter agregacji: chromosomy są tworzone, przetwarzane i usuwane przez klasę *Population*.

Klasa *Population* pełni rolę agregatora i zarządcy zbioru chromosomów, czyli obiektów klasy *Chromosome*. To w jej obrębie przechowywana jest cała populacja oraz poddawana jest ocenie, selekcji i replikacji. Klasa *Population* odpowiada za inicjalizację chromosomów, wielokrotne wywoływanie ich metod ewaluacyjnych, zarządzanie procesami krzyżowania i mutacji oraz tworzenie kolejnych pokoleń. Ponadto klasa *Population* przechowuje referencję do globalnej funkcji celu (*fitness_function*) i zarządza jej przekazywaniem do pojedynczych chromosomów w ramach operacji oceny.

Klasa *GA* znajduje się najwyżej w hierarchii i jest odpowiedzialna za koordynację całego procesu ewolucji. Tworzy populację, zarządza jej inicjalizacją oraz kolejnymi etapami ewolucji, wywołując metody cyklicznie. Klasa *GA* decyduje o globalnych parametrach takich jak liczebność populacji, liczba pokoleń, rozmiar chromosomów, prawdopodobieństwo mutacji i krzyżowania. Klasa *GA* nie ingeruje bezpośrednio w pojedyncze chromosomy, wpływa na nie pośrednio poprzez operacje na populacji.

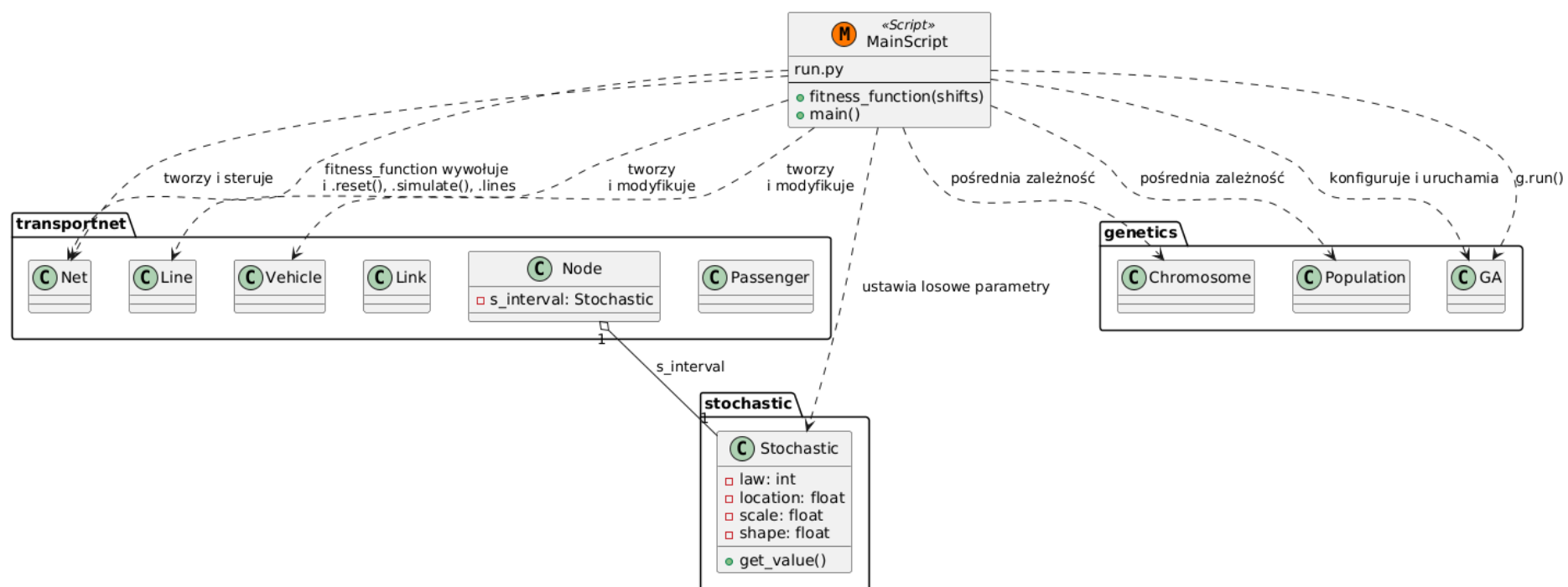
2.4.5.1 Kierunki i rodzaje zależności

Relacje w pakiecie *genetics* charakteryzują się hierarchią. Klasa *GA* agreguje i zarządza pojedynczą populacją, klasa *Population* grupuje i zarządza chromosomami. Komunikacja z niższym poziomem polega na przekazywaniu parametrów i funkcji, natomiast na poziomie wyższym przekazywane są wyniki w postaci najlepszych znalezionych rozwiązań (obiektów klasy *Chromosome* o najniższej wartości funkcji celi). Każda z klas koncentruje się wyłącznie na własnym zakresie działań, nie ingerując w logikę czy stan wewnętrzny innych.

2.4.6 Relacje pomiędzy pakietami

Każdy z pakietów pełni wyspecjalizowaną rolę, a powiązania między nimi mają charakter zarówno koncepcyjny, jak i techniczny, umożliwiając efektywną integrację mechanizmów losowości, symulacji oraz optymalizacji heurystycznej. Relacje pomiędzy pakietami zostały przedstawione na rys. 18.

Zgodnie ze schematem zaprezentowanym na rys. 18, pakiet *transportnet* stanowi rdzeń modelu. Odpowiada za pełne odwzorowanie realnego systemu transportowego: zarządza strukturą sieci w postaci węzłów, połączeń i linii komunikacyjnych, symuluje ruch pojazdów, generuje popyt oraz przepływ i obsługę pasażerów w przebiegu symulacji. W nim występują klasy *Net*, *Node*, *Link*, *Line*, *Vehicle* i *Passenger*, które wchodzą ze sobą w relacje agregacyjne i asocjacyjne. Model transportowy jest jednocześnie strukturą statyczną (topologia) i dynamiczną (realizacja ruchu i obsługa popytu).



Rysunek 18. Relacje i zależności pomiędzy pakietami
 Źródło: Opracowanie własne

Bezpośrednim narzędziem do modelowania nieregularności oraz realizacji charakterystyk losowych w systemie jest pakiet *stochastic*. Udostępnia on generator zmiennych losowych (klasa *Stochastic*). Generator ten wykorzystywany jest przez klasę *Node* (atrybut *s_interval*), umożliwiając stochastyczne wyznaczanie interwałów generowania popytu. Pakiet *Transportnet* wykorzystuje funkcjonalności *Stochastic* do zwiększenia realizmu symulacji, przy czym *Stochastic* nie posiada wiedzy o strukturze czy stanie modelu sieci. Działanie generatora losowego jest sprowadzone do mechanizmu usługodawczego i pozostaje całkowicie niezależne od kontekstu modelu, przez co zachowana jest możliwość łatwej rozbudowy systemu o inne źródła losowości.

Warstwa optymalizacji realizowana jest przez pakiet *genetics*, który zawiera komplet metod do przeprowadzania ewolucyjnych procesów poszukiwania najlepszych rozwiązań. *Genetics* działa na zasadach silnej niezależności. Jego klasy (*Chromosome*, *Population*, *GA*) nie znają struktury pakietu *transportnet*, operują wyłącznie na przekazanych im wartościach funkcji celu (*fitness_function*) i parametrach chromosomów.

Powiązania między pakietami mają wyraźnie charakter warstwowy i jednokierunkowy. *transportnet* wykorzystuje pakiet *stochastic* by wzbogacać model o losowość, nie naruszając niezależności tego generatora. Pakiet *genetics* pozostaje neutralny wobec modelu sieci. Jego kontakt z warstwą symulacyjną odbywa się wyłącznie poprzez funkcję celu przekazaną przez warstwę sterującą (główny skrypt uruchamiający – *run.py*). Taka organizacja relacji zapewnia niezależność i pozwala na dowolną rekonfigurację dowolnego pakietu bez ryzyka negatywnego wpływu na pozostałe. Użytkownik lub warstwa sterująca integruje wszystkie pakiety, decydując, które parametry podlegają optymalizacji, które atrybuty w modelu mają być losowe oraz jakie kryteria jakościowe są używane podczas ewolucji rozwiązań.

3. Rozdział III

W części badawczej pracy wykorzystano rozkłady jazdy wybranych sześciu linii należących do zielonogórskiej sieci publicznego transportu zbiorowego w 2019 roku.

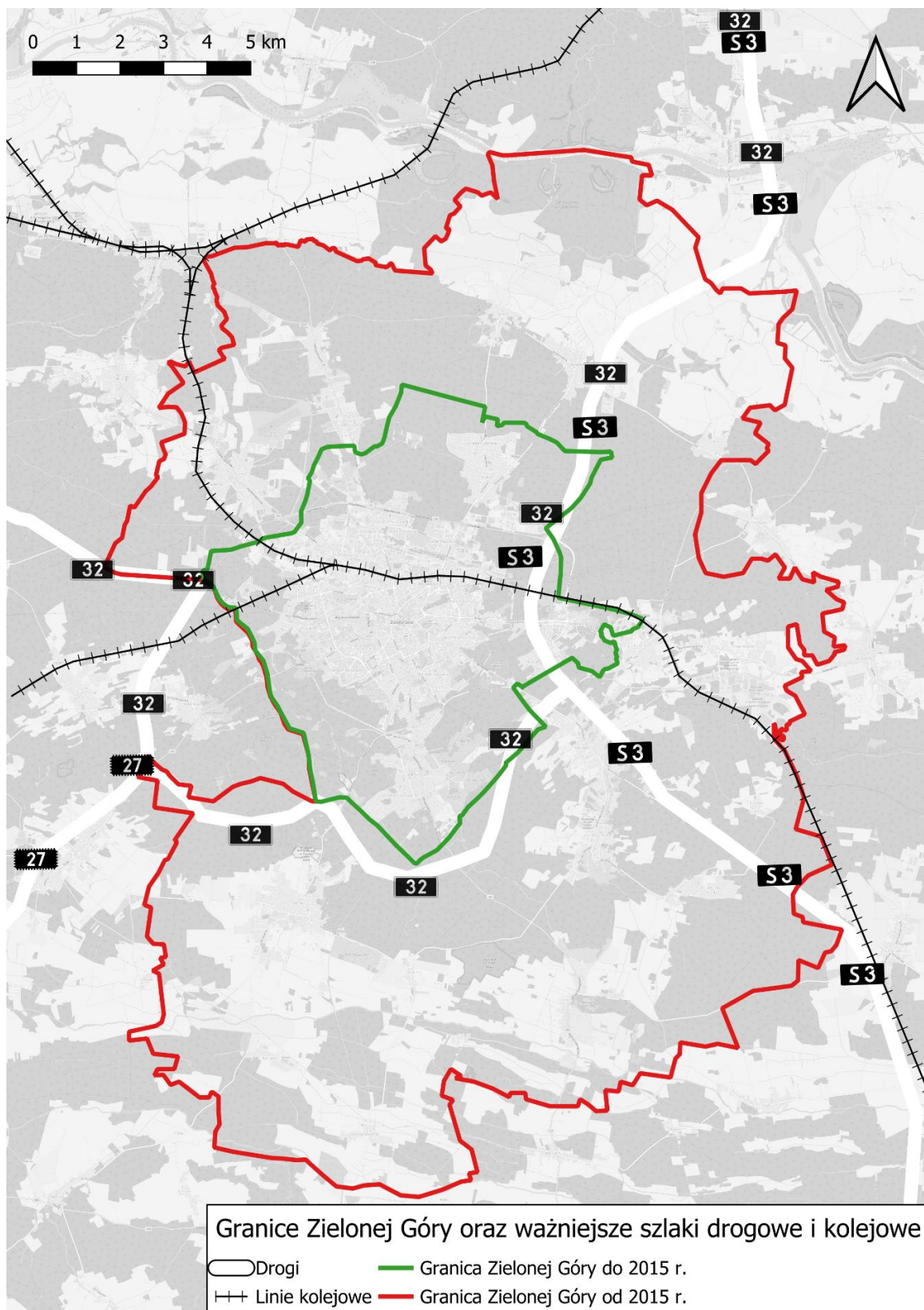
3.1 Charakterystyka sieci transportu publicznego

3.1.1 Obsługiwany obszar

Sieć publicznego transportu zbiorowego w Zielonej Górze obsługuje obszar Miasta Zielona Góra oraz jedną miejscowość (Droszków) w sąsiadującej z nim od wschodu gminie wiejskiej Zabór.

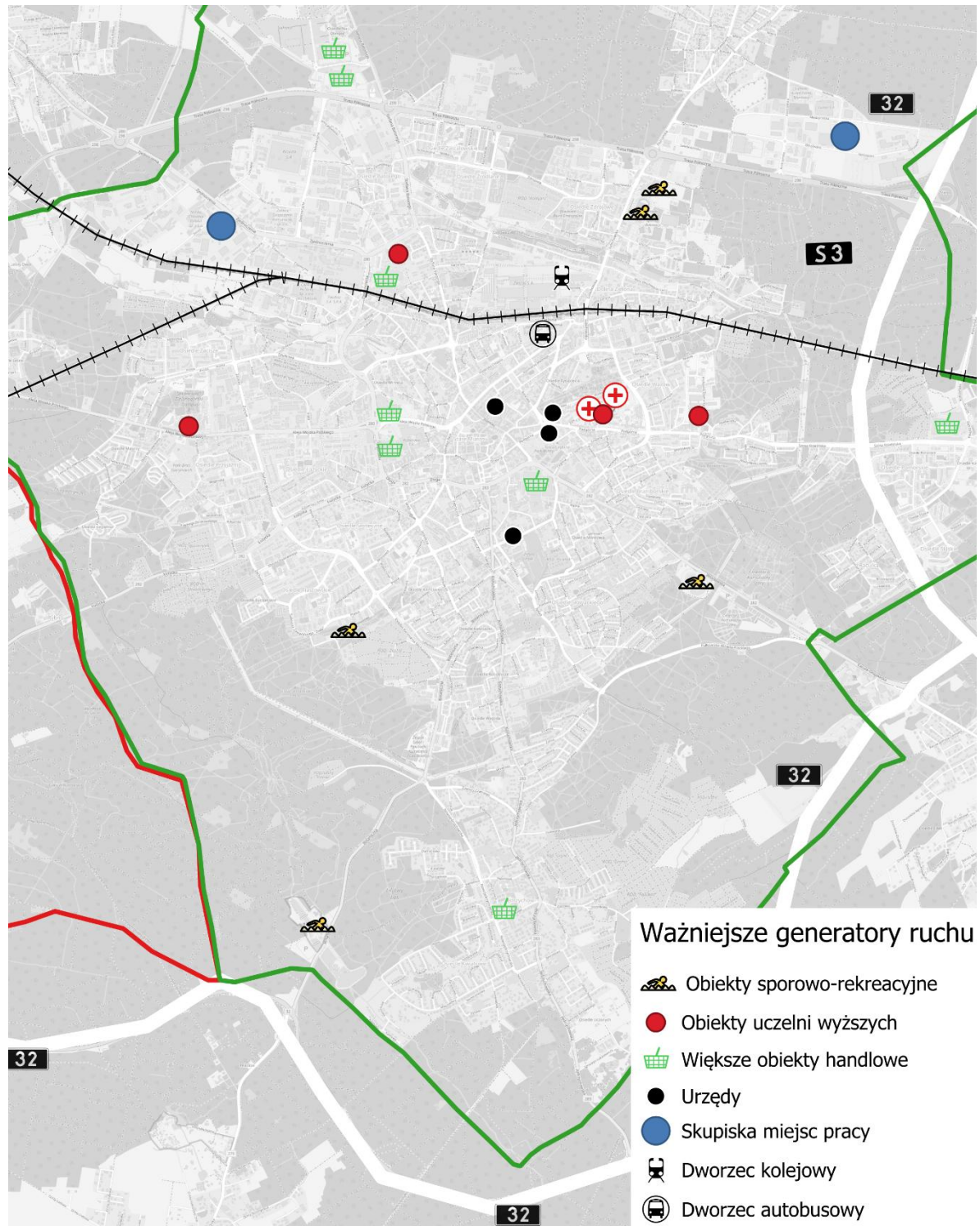
Zielona Góra jest miastem na prawach powiatu, zajmuje powierzchnię 277 km², mieszkają w nim 138 932 osoby, a gęstość zaludnienia wynosi 499,3 osób na km² (GUS, 2025) wg stanu na ostatni dzień grudnia 2023 roku. Powierzchnia miasta powiększyła się z dniem 1 stycznia 2015 roku, kiedy to dołączono do niego obszar gminy Zielona Góra. Miasto Zielona Góra jest wraz z Gorzowem Wielkopolskim jedną ze stolic Województwa Lubuskiego. Znajduje się w nim siedziba Urzędu Marszałkowskiego Województwa Lubuskiego, Delegatura Lubuskiego Urzędu Wojewódzkiego oraz regionalne oddziały instytucji rządowych.

Przez miasto przebiega droga ekspresowa S3 łącząca port w Świnoujściu z granicą z Czechami w Lubawce, będąc jednocześnie elementem europejskiego korytarza transportowego E65, łączącego szwedzkie Malmö z grecką wyspą Kreta. Przez Zieloną Górę przebiega także droga krajowa nr 32 łącząca podpoznański Stęszew z granicą państwa z Niemcami w Gubinie. W mieście swój bieg rozpoczyna także droga krajowa nr 27, która łączy je z granicą z Niemcami w Przewozie, w południowo-zachodniej części województwa. Przez Zieloną Górę przebiega linia kolejowa nr 273 łącząca Szczecin z Wrocławiem, swój początek w niej ma także linia nr 370 biegnąca do położonych w południowej części województwa Żar. Od 1 stycznia 2015 roku miasto częścią swojej północnej granicy sięga do rzeki Odry, gdzie na drugim jej brzegu znajduje się port rzeczny w Cigacicach. W odległości ok. 30 km od miasta znajduje się międzynarodowy port lotniczy Zielona Góra-Babimost. Układ dróg i linii kolejowych oraz granice miasta przed i po powiększeniu zaprezentowano na rys. 19.



Rysunek 19. Granice Zielonej Góry oraz ważniejsze szlaki drogowe i kolejowe
 Źródło: Opracowanie własne

Zgodnie z rys. 19, droga ekspresowa S3 przebiega przez wschodnią część miasta, tworząc tym samym jej wschodnią obwodnicę. Przebieg drogi krajowej nr 32 tworzy zachodnią i południową obwodnicę miasta. Na rysunku 20. przedstawiono ważniejsze generatory ruchu w mieście.



Rysunek 20. Ważniejsze generatory ruchu w mieście
Źródło: Opracowanie własne

Zgodnie z rys. 20, urzędy ulokowane są w centralnej części miasta. W mieście funkcjonuje jedna uczelnia wyższa - Uniwersytet Zielonogórski, którego budynki zlokalizowane są w różnych częściach miasta.

W centrum miasta znajdują się dwa szpitale oraz dworce kolejowy i autobusowy. Obiekty sportowo-rekreacyjne rozmieszczone są w zachodniej (amfiteatr), wschodniej (stadion żużlowy), północnej (Centrum Rekreacyjno-Sportowe) oraz południowej (kąpielisko H2Ochla) części miasta. Większe skupiska miejsc pracy znajdują się we wschodniej (Lubuski Park Przemysłowo Technologiczny), północnej (Strefa Aktywności Gospodarczej „Spalony Las”) i zachodniej części miasta. W rdzeniu miasta dominuje zabudowa wielorodzinna, poza nią jednorodzinna.

3.1.2 Uwarunkowania prawne funkcjonowania systemu transportu zbiorowego w Zielonej Górze

Zgodnie z ustawą o publicznym transporcie zbiorowym organizatorem transportu na terenie Miasta Zielona Góra jest Prezydent Miasta Zielona Góra. W jego imieniu obowiązki organizatora pełni Departament Przedsiębiorczości i Gospodarki Komunalnej Urzędu Miasta Zielona Góra. Funkcję operatora pełni Miejski Zakład Komunikacji spółka z ograniczoną odpowiedzialnością³¹.

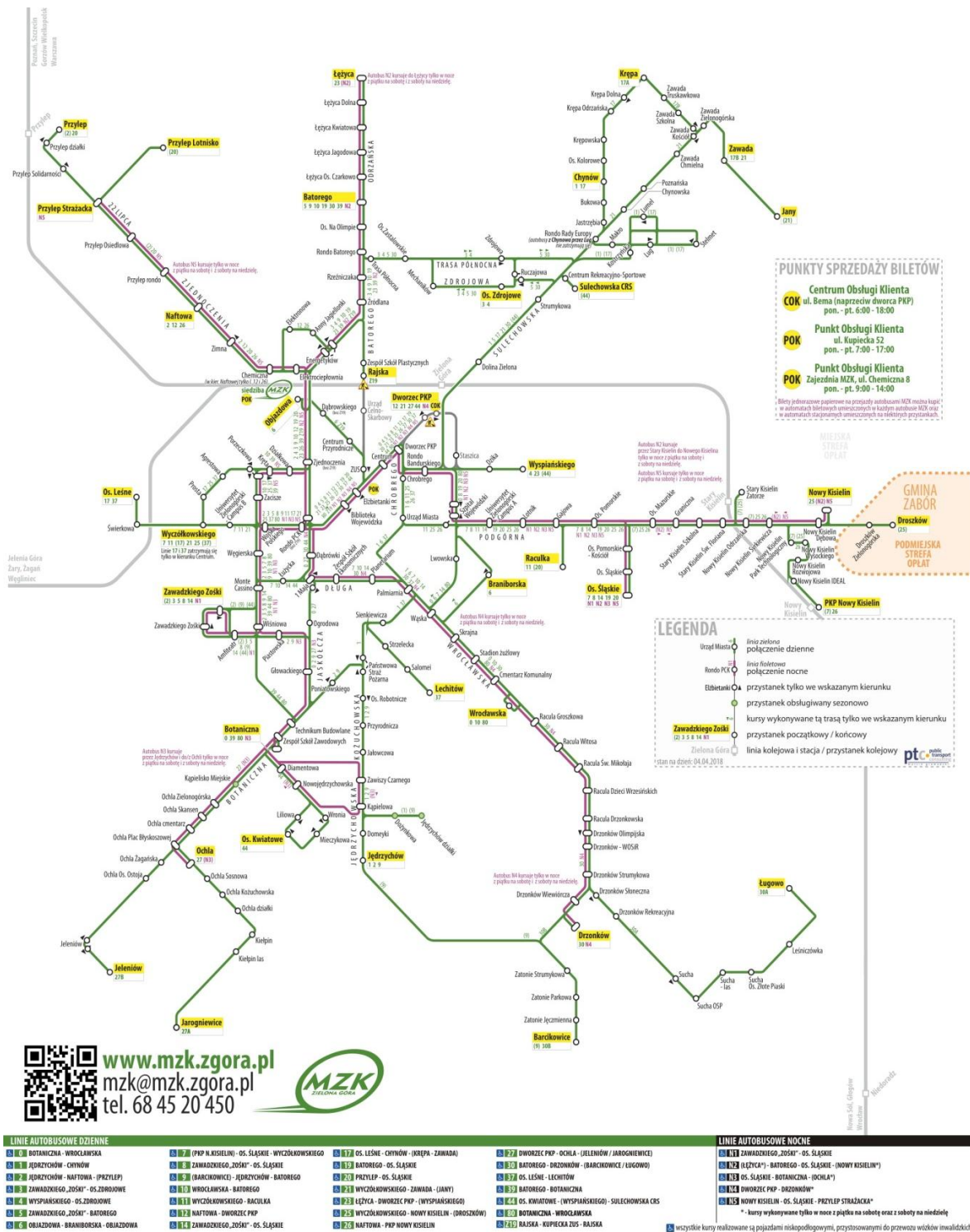
3.1.3 Charakterystyka systemu transportu zbiorowego w Zielonej Górze

Na system transportu zbiorowego w Zielonej Górze w 2019 roku składały się 32 linie autobusowe (27 dziennych i 5 nocnych). Spośród nich tylko dwie linie wyjeżdżały poza granice miasta – linia nr 25, która wybranymi kursami obsługiwała miejscowość Droszków, w gminie Zabór oraz linia nr 21, która wybranymi kursami obsługiwała miejscowość Przytok, również w gminie Zabór. Linie dzienne kursowały w dzień powszedni od 4:16 do 23:52. Długość sieci transportu zbiorowego przekraczała 151 kilometrów, w jej skład wchodziły 203 przystanki, a długość wszystkich linii wynosiła 419 kilometrów.

Spośród linii dziennych 20 wykonywało kursy przez cały tydzień. Siedem z nich kursowało tylko w dzień powszedni, jedna - linia nr 3 wykonywała kursy tylko w niedziele. Dwie linie (nr 4 i 39), kursujące tylko w dni powszednie, funkcjonowały tylko w dni nauki szkolnej. Trzy linie nocne N1, N2 i N3 wykonywały kursy przez wszystkie noce w tygodniu,

³¹ Do 30 czerwca 2021 roku funkcjonowała jako zakład budżetowy Urzędu Miasta Zielona Góra

dwie kolejne N4 i N5 tylko w noc z piątku na sobotę i z soboty na niedzielę. Schemat sieci transportu zbiorowego w Zielonej Górze w 2019 roku zaprezentowano na rys. 21.



Rysunek 21. Plan sieci zielonogórskiego systemu transportu zbiorowego w 2019 roku
Źródło: Miejski Zakład Komunikacji w Zielonej Górze

Według badań popytu przeprowadzonych w 2018 roku, w dzień powszedni z systemu transportu zbiorowego w mieście Zielona Góra korzystało 60 740 osób, w soboty 28 356 osób, w niedzielę korzystało z niego 18 773 osoby. Przekładało się to na szacunkowy wynik 17 milionów 830 tysięcy pasażerów w ciągu roku. Szczegółowe dane na temat popytu przedstawiono w tabeli 11.

Tabela 11. Liczba pasażerów korzystających z poszczególnych linii w dzień powszedni w systemie publicznego transportu zbiorowego w Zielonej Górze

Nr linii	Liczba pasażerów w dzień powszedni	Procentowy udział w sumie wszystkich przewiezionych pasażerów
0	2 953	4,86%
1	2 588	4,26%
2	899	1,48%
4	59	0,10%
5	2 995	4,93%
6	557	0,92%
7	770	1,27%
8	8 560	14,09%
9	3 336	5,49%
10	735	1,21%
11	427	0,70%
12	257	0,42%
14	2 988	4,92%
17	2 937	4,84%
19	2 989	4,92%
20	3 357	5,53%
21	1 025	1,69%
23	2 296	3,78%
25	3 707	6,10%
26	415	0,68%
27	2 220	3,65%
30	4 199	6,91%
37	2 411	3,97%
39	171	0,28%
44	3 052	5,02%
80	3 755	6,18%
Z19	967	1,59%
N1	46	0,08%
N2	51	0,08%
N3	18	0,03%
Σ	60 740	100%

Źródło: (Kapica, 2021)

Z danych w tabeli 11 wynika, że najwięcej pasażerów w dzień powszedni realizowanych korzystało z linii nr 8. Było ich 8560, co stanowiło 14,09% wszystkich pasażerów tego dnia. Drugą pod względem udziału pasażerów była linia nr 80, z której skorzystało 6,18% wszystkich pasażerów. Najmniej (pomijając linie nocne N1, N2 i N3) pasażerów skorzystało

z linii nr 4. Było ich 0,1% ze wszystkich w ciągu dnia. Wynika to jednak z faktu, że na linii tej wykonywany był tylko jeden kurs (Kapica, 2021).

3.2 Model miasta Zielona Góra w programie PTV Visum

Liczbę kursów oraz dane o popycie na przewozy z wykorzystaniem publicznego transportu zbiorowego odczytano z modelu symulacyjnego miasta Zielona Góra wykonanego w PTV Visum. Model ten został udostępniony przez Urząd Miasta Zielona Góra w ramach ustawy z dnia 6 września 2001 o dostępie do informacji publicznej.

Model został wykonany na zlecenie Urzędu Miasta Zielona Góra a dane wejściowe do modelu uzyskano w ramach kompleksowych badań ruchu i ankiet przeprowadzanych wśród mieszkańców w 2019 roku. Model składa się z 396 rejonów komunikacyjnych i swoim zasięgiem obejmuje cały obszar funkcjonalny miasta Zielona Góra.

3.3 Część sieci wytypowana do eksperymentu

Do przeprowadzenia eksperymentu wytypowano sześć linii z zielonogórskiej sieci publicznego transportu zbiorowego. Są to linie „0”, „1”, „5”, „8”, „19” i „20”. Kryterium wyboru tych linii był fakt, że są to linie szkieletowe, kursujące w godzinach szczytu najczęściej tj. co 15 minut. Informacje na temat liczby przewożonych pasażerów przez te linie zawarto w tabeli 12.

Tabela 12. Liczba pasażerów korzystających z wybranych sześciu linii w dzień powszedni w systemie publicznego transportu zbiorowego w Zielonej Górze

Nr linii	Liczba pasażerów w dzień powszedni	Procentowy udział w sumie wszystkich przewiezionych pasażerów
0	2 953	4,86%
1	2 588	4,26%
5	2 995	4,93%
8	8 560	14,09%
19	2 989	4,92%
80	3 755	6,18%
Σ	23 840	39,24%
Pozostałe	36 900	60,76%
Σ	60 740	100%


Źródło: (Kapica, 2021)


Zgodnie z danymi z tabeli 12. wybrane do eksperymentu linie przewiozły 23 840 pasażerów co stanowi blisko 40% pasażerów całej sieci. Uproszczony schemat przebiegu linii wykonany w programie PTV Visum zaprezentowano na rys. 22.

Legenda

Węzły przystankowe

StoplCode

 3733 Dozwolone przesiadki

 3733 Przystanki końcowe

Linie

Distinct:Vehicle journeys (undirected)\Line name

 0

 1

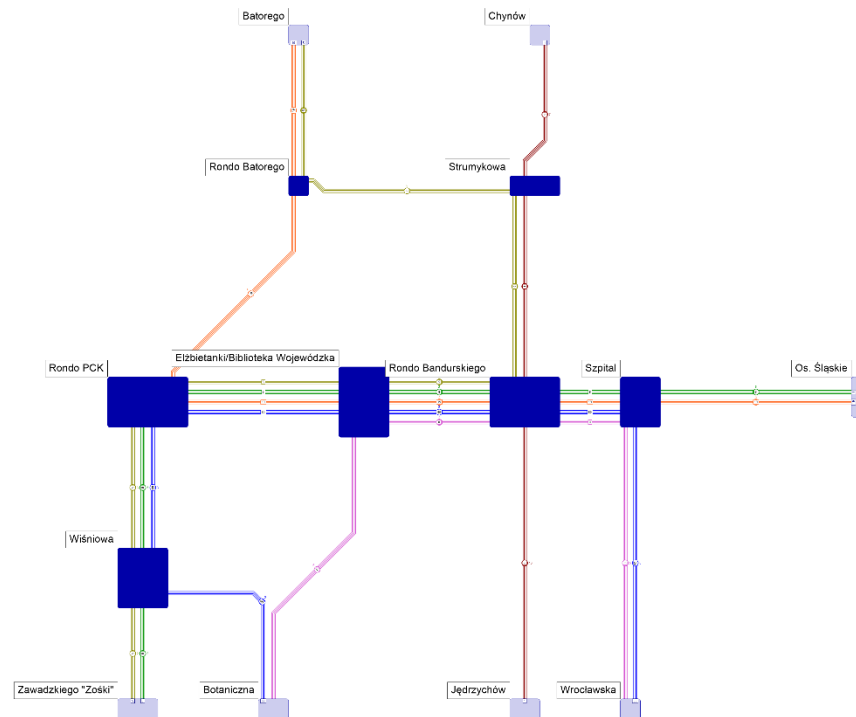
 19

 5

 8

 80

 *



Rysunek 22. Uproszczony schemat tras sześciu wytypowanych linii
Źródło: Opracowanie własne

Na rys. 22 przedstawiono uproszczony schemat tras sześciu wytypowanych do przeprowadzenia eksperymentu linii. Schemat zawiera przystanki końcowe oraz przystanki, na których możliwe są przesiadki pomiędzy liniami. Jest ich siedem. Ich wykaz wraz z numerami liniami, które się na nich zatrzymują przedstawiono w tabeli 13.

Tabela 13. Lista przystanków, na których możliwe są przesiadki

Nazwa przystanku	Zatrzymujące się linie
Elżbietanki/Biblioteka Wojewódzka	0, 5, 8, 19, 80
Rondo Bandurskiego	0, 1, 5, 8, 19, 80
Rondo Batorego	5,19
Rondo PCK	5, 8, 19, 80
Strumykowa	1,5
Szpital	0, 8, 19, 80
Wiśniowa	5, 8, 80

Źródło: Opracowanie własne

Jak przedstawiono w tabeli 13. tylko jeden przystanek jest wspólny dla wszystkich linii. Jest to przystanek Rondo Bandurskiego, znajdujący się w pobliżu dworca kolejowego w Zielonej Górze. Jeden przystanek jest wspólny dla pięciu linii – Elżbietanki/Biblioteka Wojewódzka. Dwa przystanki są wspólne dla czterech linii – Rondo PCK i Szpital. Dla trzech linii wspólny przystanek to Wiśniowa, a dla dwóch linii wspólne są dwa przystanki – Rondo Batorego i Strumykowa.

4. Rozdział IV

Celem niniejszej pracy jest znalezienie możliwie optymalnych parametrów algorytmu genetycznego rozwiązującego problem synchronizacji rozkładów jazdy. W pracy zsynchronizowano rozkłady jazdy sześciu linii wchodzących w skład sieci publicznego transportu zbiorowego w Zielonej Górze. Wykorzystanie rzeczywistego rozkładu jazdy i popytu na przewozy umożliwiło ocenę rozwiązań wygenerowanych przez algorytm genetyczny. Dodatkowo przeanalizowano wpływ wybranych parametrów wykorzystanego w pracy algorytmu genetycznego na czas trwania obliczeń.

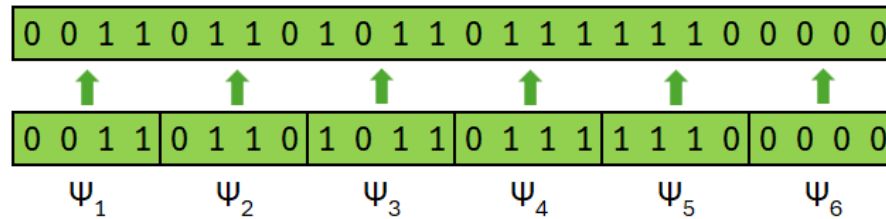
4.1 Przełożenie terminologii algorytmów genetycznych na zagadnienia transportowe

Jak wcześniej wspomniano w terminologii algorytmów genetycznych funkcjonują pojęcia m.in. takie jak: chromosom, gen i populacja. W teorii algorytmów genetycznych chromosomem nazywane jest jedno rozwiązanie badanego problemu. W przełożeniu tej terminologii na zagadnienia transportowe w niniejszej pracy chromosom odpowiada rozkładowi jazdy. W wykorzystanej implementacji algorytmu genetycznego chromosom jest sekwencją bitów (genów). Godziny odjazdów pojazdów obsługujących linie wyznacza się za pomocą przesunięć czasowych Ψ i interwału kursowania. Oznacza to, że algorytm genetyczny wyznacza czas rozpoczęcia pierwszego kursu każdej z linii, a kolejne generowane są poprzez dodanie do tego czasu interwału kursowania. Linie zsynchronizowane w niniejszej pracy charakteryzują się stałym 15-minutowym interwałem kursowania. Ponieważ przesunięcie czasowe nie może być większe od interwału kursowania zapisano je za pomocą czterech bitów, które są wystarczające do odwzorowania przesunięcia czasowego w zakresie od 0 do 15 minut.

Zbiorem rozkładów jazdy, na których algorytm genetyczny przeprowadza operacje jest populacja.

W niniejszej pracy funkcją celu jest minimalizacja czasu oczekiwania pasażerów na przesiadkę i pod tym kątem oceniane były rozwiązania dostarczane przez algorytm genetyczny.

Przykładowy chromosom, zakodowany w formie binarnej jako sekwencja zer i jedynek, z przesunięciami czasowym sześciu linii został zaprezentowany na rys. 23.



Rysunek 23. Przykładowy chromosom
Źródło: Opracowanie własne

Zgodnie z rys. 23, przesunięcia czasowe zostały zakodowane w 24 genach (po 4 geny na linię). Zapis binarny przedstawiony na rysunku 23. oznacza, że przesunięcie czasowe dla pierwszej linii Ψ_1 jest równe trzem minutom, dla drugiej Ψ_2 jest równe sześciu minutom, dla trzeciej Ψ_3 jedenastu minutom, dla czwartej Ψ_4 siedmiu minutom, dla piątej Ψ_5 czternastu minutom, a dla szóstej Ψ_6 przesunięcie czasowe równa się 0 minut. Oznacza to, że szósta linia rozpocznie kursowanie jako pierwsza wraz ze startem każdego 15-minutowego interwału. Fragment przykładowego rozkładu jazdy sześciu linii z wygenerowanymi przesunięciami i interwałem odjazdów wynoszącym 15 minut został przedstawiony w tabeli 14.

Tabela 14. Fragment przykładowego rozkładu jazdy zgodnego z chromosomem przedstawionym na rysunku 23.

Linia	Godziny odjazdów
Ψ_1	10:03 , 10:18, 10:33, 10:48
Ψ_2	10:06 , 10:21, 10:36, 10:51
Ψ_3	10:11 , 10:26, 10:41, 10:56
Ψ_4	10:07 , 10:22, 10:37, 10:52
Ψ_5	10:14 , 10:29, 10:44, 10:59
Ψ_6	10:00 , 10:15, 10:30, 10:45

Źródło: Opracowanie własne

W wyniku działania algorytmu genetycznego powstają rozwiązania zawierające tylko pierwsze (pogrubione) odjazdy każdej z linii. Kolejne odjazdy tworzone są poprzez dodanie do nich przyjętego interwału czasowego dla linii. W tym przypadku było to 15 minut.

W trakcie swojego działania algorytm genetyczny przeprowadza m.in. operacje krzyżowania i mutacji na chromosomach. Przykładowy efekt operacji krzyżowania w przełożeniu na rozkłady jazdy przedstawiają tabele 15-18.

Tabela 15. Zapis binarny i dziesiętny dwóch rozwiązań

	Ψ_1				Ψ_2				Ψ_3				Ψ_4				Ψ_5				Ψ_6			
Rozwiązanie 1 (system binarny)	0	0	1	1	0	1	1	0	1	0	1	1	0	1	1	1	1	1	1	0	0	0	0	0
Rozwiązanie 1 (system dziesiętny)	3				6				11				7				14				0			
Rozwiązanie 2 (system binarny)	1	0	0	1	0	0	1	1	1	0	0	0	0	1	0	0	1	0	1	1	0	1	1	0
Rozwiązanie 2 (system dziesiętny)	9				3				8				4				11				6			

Źródło: Opracowanie własne

Tabela 16. Fragmenty dwóch rozkładów jazdy

Rozwiązanie 1.		Rozwiązanie 2.	
Linia	Godziny odjazdów	Linia	Godziny odjazdów
Ψ_1	10:03, 10:18, 10:33, 10:48	Ψ_1	10:09, 10:24, 10:39, 10:54
Ψ_2	10:06, 10:21, 10:36, 10:51	Ψ_2	10:03, 10:18, 10:33, 10:48
Ψ_3	10:11, 10:26, 10:41, 10:56	Ψ_3	10:08, 10:33, 10:38, 10:53
Ψ_4	10:07, 10:22, 10:37, 10:52	Ψ_4	10:04, 10:19, 10:34, 10:49
Ψ_5	10:14, 10:29, 10:44, 10:59	Ψ_5	10:11, 10:26, 10:41, 10:56
Ψ_6	10:00, 10:15, 10:30, 10:45	Ψ_6	10:06, 10:21, 10:36, 10:51

Źródło: Opracowanie własne

Tabela 15 przedstawia dwa rozwiązania wejściowe w formach binarnej i dziesiętnej. Tabela 16 przedstawia fragment stworzonego na ich podstawie rozkładu jazdy sześciu linii z 15-minutowym interwałem kursowania.

Tabela 17. Zapis binarny i dziesiętny dwóch rozwiązań po operacji krzyżowania

	Ψ_1				Ψ_2				Ψ_3				Ψ_4				Ψ_5				Ψ_6			
Rozwiązanie 3 (system binarny)	0	0	1	1	0	1	1	0	1	0	1	1	0	1	0	0	1	0	1	1	0	1	1	0
Rozwiązanie 3 (system dziesiętny)	3				6				11				4				11				6			
Rozwiązanie 4 (system binarny)	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
Rozwiązanie 4 (system dziesiętny)	9				3				8				7				14				0			

Źródło: Opracowanie własne

Tabela 18. Fragmenty dwóch rozkładów jazdy po operacji krzyżowania

Rozwiązanie 3.		Rozwiązanie 4.	
Linia	Godziny odjazdów	Linia	Godziny odjazdów
Ψ_1	10:03, 10:18, 10:33, 10:48	Ψ_1	10:09, 10:24, 10:39, 10:54
Ψ_2	10:06, 10:21, 10:36, 10:51	Ψ_2	10:03, 10:18, 10:33, 10:48
Ψ_3	10:11, 10:26, 10:41, 10:56	Ψ_3	10:08, 10:33, 10:38, 10:53
Ψ_4	10:04, 10:19, 10:34, 10:49	Ψ_4	10:07, 10:22, 10:37, 10:52
Ψ_5	10:11, 10:26, 10:41, 10:56	Ψ_5	10:14, 10:29, 10:44, 10:59
Ψ_6	10:06, 10:21, 10:36, 10:51	Ψ_6	10:00, 10:15, 10:30, 10:45

Źródło: Opracowanie własne

Tabela 17 przedstawia dwa rozwiązania w formach binarnej i dziesiętnej po dokonaniu na nich operacji krzyżowania. Tabela 18 przedstawia fragment stworzonego na ich podstawie rozkładu jazdy sześciu linii z 15-minutowym interwałem kursowania. Rozwiązania 3 i 4 powstały z części rozwiązań 1 i 2 z tabeli 15. Rozwiązanie 3 z pierwszej części rozwiązania 1 i drugiej części rozwiązania 2, natomiast rozwiązanie 4 powstało z pierwszej części rozwiązania 2 i drugiej części rozwiązania 1.

Zgodnie z kolejnością działań algorytmu genetycznego po procesie krzyżowania może wystąpić mutacja. Przykładowy wpływ zastosowania mutacji na rozwiązanie 4 z tabeli 17 w przełożeniu na rozkłady jazdy przedstawiają tabele 19 i 20.

Tabela 19. Zapis binarny i dziesiętny rozwiązania przed i po wystąpieniu mutacji

	Ψ_1				Ψ_2				Ψ_3				Ψ_4				Ψ_5				Ψ_6			
Rozwiązanie 4 (system binarny)	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
Rozwiązanie 4 (system dziesiętny)	9				3				8				7				14				0			
Rozwiązanie 4 po mutacji (system binarny)	1	0	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1	1	1	0	0	1	0	0
Rozwiązanie 4 po mutacji (system dziesiętny)	9				3				8				7				14				4			

Źródło: Opracowanie własne

Tabela 20. Fragmenty rozkładów jazdy przed i po wystąpieniu mutacji

Rozwiązanie 4.		Rozwiązanie 4. po mutacji	
Linia	Godziny odjazdów	Linia	Godziny odjazdów
Ψ_1	10:09, 10:24, 10:39, 10:54	Ψ_1	10:09, 10:24, 10:39, 10:54
Ψ_2	10:03, 10:18, 10:33, 10:48	Ψ_2	10:03, 10:18, 10:33, 10:48
Ψ_3	10:08, 10:33, 10:38, 10:53	Ψ_3	10:08, 10:33, 10:38, 10:53
Ψ_4	10:07, 10:22, 10:37, 10:52	Ψ_4	10:07, 10:22, 10:37, 10:52
Ψ_5	10:14, 10:29, 10:44, 10:59	Ψ_5	10:14, 10:29, 10:44, 10:59
Ψ_6	10:00, 10:15, 10:30, 10:45	Ψ_6	10:04, 10:19, 10:34, 10:49

Źródło: Opracowanie własne

Tabela 19 przedstawia rozwiązanie w formach binarnej i dziesiętnej przed i po wystąpieniu mutacji. Mutacja polegająca w tym przypadku na zmianie wartości jednego bitu (genu) wystąpiła we fragmencie genów kodujących przesunięcie czasowe szóstej linii. Zmieniony jeden bit oraz jego odwzorowanie w formie dziesiętnej oznaczono niebieską czcionką. Tabela 20 przedstawia fragment rozkładu jazdy sześciu linii z 15-minutowym interwałem kursowania stworzonego na podstawie rozwiązań przed i po wystąpieniu mutacji. Zmienione w wyniku wystąpienia mutacji godziny odjazdów szóstej linii zostały oznaczone niebieską czcionką. Rozpocznie ona swoje kursowanie z 15-minutowym interwałem nie o 10:00, a o 10:04.

4.2 Eksperyment obliczeniowy

4.2.1 Plan eksperymentu

W pracy próbowano znaleźć możliwie optymalne parametry algorytmu genetycznego rozwiązującego problem synchronizacji rozkładów jazdy. W tym celu zaplanowano eksperyment składający się z następujących kroków:

1. Uruchomienie algorytmu genetycznego z kombinacjami różnych parametrów dotyczących rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji oraz z popytem generowanym losowo przez klasę *Stochastic*. Szczegółowe zestawienie użytych parametrów w seriach eksperymentu przedstawia tabela 21.

Tabela 21. Parametry algorytmu genetycznego użyte w eksperymencie

Parametry zmienne	
Rozmiar populacji	[30, 50, 100, 150, 200]
Liczba generacji	[50, 100, 150, 200]
Prawdopodobieństwo mutacji	[0,01; 0,05; 0,1; 0,2]
Parametry stałe	
Prawdopodobieństwo krzyżowania	0,25
Prawdopodobieństwo przeżycia	0,1
Liczba prób dokonania mutacji chromosomu	3
Selekcja najlepszych rozwiązań	Selekcja rankingowa
Krzyżowanie	Losowe w jednym punkcie

Źródło: Opracowanie własne

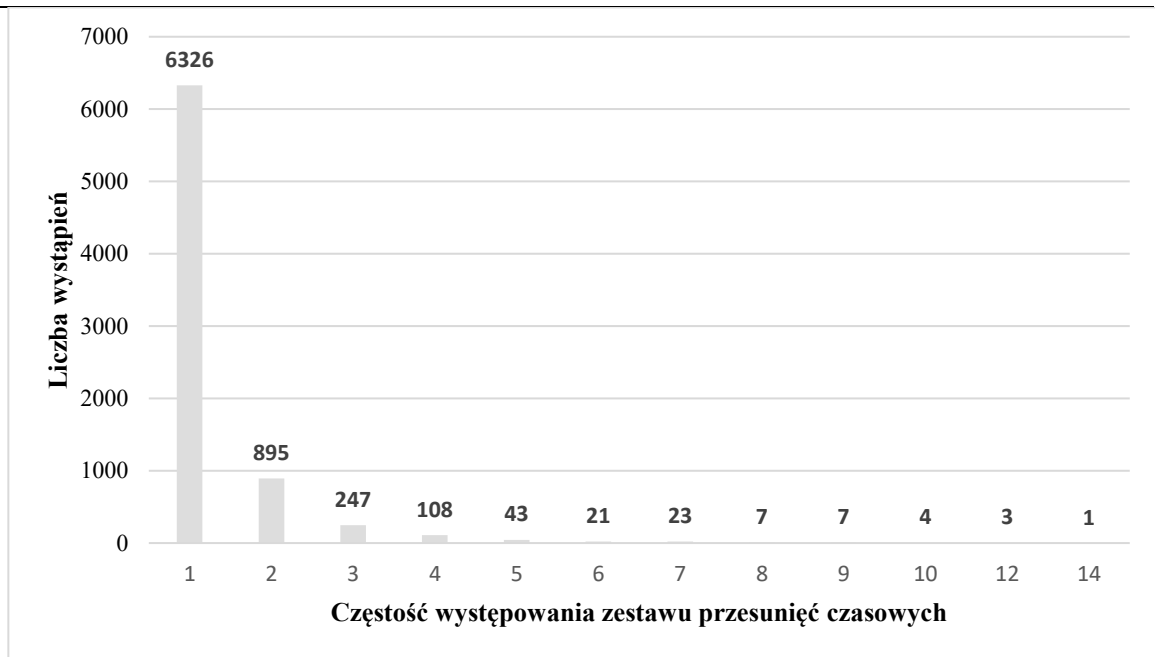
2. Zapisanie rezultatów wygenerowanych przez algorytm genetyczny w postaci zestawu sześciu przesunięć czasowych i wyliczonej dla nich wartości funkcji celu.
3. Selekcja wyników - odrzucenie rozwiązań z zestawami przesunięć czasowych, które pojawiły się mniej niż trzy razy. Następnie wyselekcjonowanie spośród rozwiązań z takimi samymi zestawami przesunięć czasowych jednego z najniższą wartością funkcji celu.
4. Selekcja wyników poprzez analizę statystyczną – uporządkowanie rozwiązań na podstawie wartości wybranej funkcji celu (najmniejszego średniego czasu oczekiwania pasażerów na przesiadkę) w kolejności rosnącej i wybór rezultatów

do kolejnego punktu eksperymentu za pomocą punktu odcięcia w postaci 51. percentyla.

5. Ocena wyników – uruchomienie części zaimplementowanego modelu odpowiedzialnej za obliczenie wartości funkcji celu dla stu popytów wygenerowanych losowo oddzielnie dla każdego zestawu przesunięć czasowych wyselekcjonowanego w poprzednim kroku. Następnie selekcja wyników na podstawie istotności statystycznej wartości funkcji celu.
6. Odczytanie i analiza parametrów serii eksperymentu, z których uzyskano wyselekcjonowane zestawy przesunięć czasowych.
7. Odczytanie danych o popycie na podróże i rzeczywistego rozkładu jazdy sześciu linii z modelu symulacyjnego miasta w PTV Visum.
8. Trzystukrotne uruchomienie części zaimplementowanego modelu odpowiedzialnej za obliczenie wartości funkcji celu dla rzeczywistego popytu na podróże i rozkładu jazdy sześciu linii odczytanych w kroku siódmym.
9. Trzystukrotne uruchomienie części zaimplementowanego modelu odpowiedzialnej za obliczenie wartości funkcji celu dla rzeczywistego popytu na podróże i wyselekcjonowanych w kroku piątym zestawów przesunięć czasowych.
10. Analiza porównawcza rezultatów uzyskanych w kroku ósmym i dziewiątym.

4.2.2 Przebieg eksperymentu obliczeniowego

W pierwszym kroku uruchomiono algorytm genetyczny ze wszystkimi kombinacjami parametrów zawartych w tabeli 21 i losowo wygenerowanym przez klasę *Stochastic* popytem. Dało to 80 różnych zestawów parametrów. Uzyskano w sumie 10 000 rezultatów w postaci zestawu sześciu przesunięć czasowych i wyliczonej dla nich wartości funkcji celu. Charakterystykę otrzymanych zestawów przesunięć czasowych pod względem częstości występowania przedstawiono na rys. 24.



Rysunek 24. Charakterystyka zestawu przesunięć czasowych pod względem częstości występowania
Źródło: Opracowanie własne

Zgodnie z danymi przedstawionymi na rys. 24, 6326 zestawów przesunięć czasowych pojawiło się wśród rozwiązań jeden raz. Najczęściej (14 razy) wśród rozwiązań pojawił się jeden zestaw przesunięć czasowych. Jak można obliczyć z danych przedstawionych na rys. 27, zestawów przesunięć czasowych, które wystąpiły przynajmniej trzy razy było 1884.

Zgodnie z trzecim krokiem eksperymentu do dalszej analizy wybrano wspomniane 1884 rezultatów. Wśród rezultatów z takim samym zestawem przesunięć czasowych do kolejnego etapu wybrano jeden z najniższą wartością funkcji celu, co zredukowało liczbę analizowanych rezultatów do 464.

Następnie zgodnie z czwartym krokiem eksperymentu dokonano analizy statystycznej zbioru rezultatów i posortowano je rosnąco według wartości funkcji celu. Wyniki analizy zaprezentowano w tabeli 22.

Wyniki analizy zbioru danych przedstawione w tabeli 22 wykazały istotne cechy świadczące o jego różnorodności oraz asymetrii prawostronnej. Średnia arytmetyczna wyniosła 5,476, natomiast odchylenie standardowe 3,498, co świadczy o rozproszeniu danych wokół średniej. Wysoki współczynnik zmienności (63,8%) potwierdza brak jednorodności rozkładu. Niewielka różnica między pierwszym kwantylem a medianą oraz znacznie większy odstęp między medianą a trzecim kwantylem sugerują asymetrię.

Tabela 22. Wyniki analizy statystycznej zbioru rezultatów

	Wartość funkcji celu		
Średnia arytmetyczna	5,476	Współczynnik zmienności	63,8%
Wartość minimalna	2,894	Odchylenie standardowe	3,498
Kwantyl rzędu 0.25	3,059	Współczynnik asymetrii	0,865
Kwantyl rzędu 0.5	3,137		
Percentyl rzędu 0.51	3,141		
Kwantyl rzędu 0.75	9,411		
Wartość maksymalna	13,178		

Źródło: Opracowanie własne

W związku z użyciem w eksperymencie selekcji danych wykonanej za pomocą 51. percentyla obliczono jego wartość, która wyniosła 3,141.

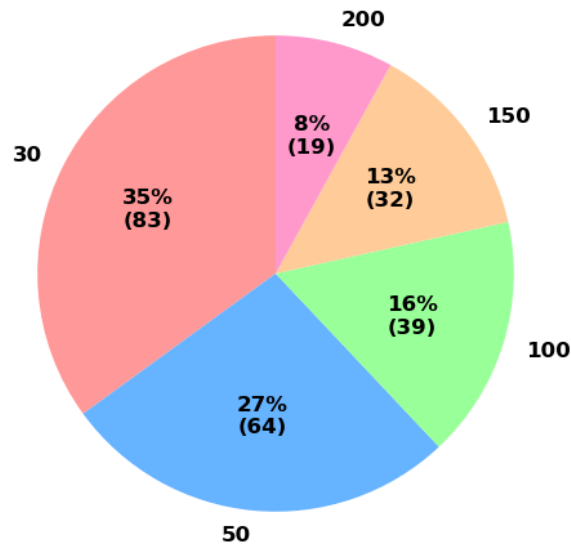
W kolejnej części analizowano uzyskane rezultaty, dla których wartość średniego czasu oczekiwania pasażera była niższa od wartości wyżej wymienionego percentyla. Dalszej analizie poddano zatem 237 rezultatów.

W następnym kroku do każdego zestawu przesunięć czasowych wygenerowano losowo sto popytów. Dla każdej pary złożonej z zestawu przesunięć czasowych i popytu wyliczono wartość funkcji celu za pomocą metody *fintess_function*. Otrzymano 23 700 wartości funkcji celu, które poddano ocenie istotności statystycznej. Dla 177 z 237 (75%) zestawów przesunięć czasowych liczba próbek równa 100 okazała się wystarczająca, aby osiągnąć przedział ufności równy 0,95. Pozostałe 60 zestawów przesunięć czasowych mieściło się w przedziale ufności równym 0,93.

W związku z tym, że wszystkie zestawy przesunięć czasowych mieściły się w przedziale ufności 0,93 zdecydowano o dopuszczeniu wszystkich 237 zestawów przesunięć czasowych z kroku czwartego do dalszej analizy.

W szóstym kroku odczytano i przeanalizowano parametry serii eksperymentu, z których pochodziły dopuszczone do tego etapu zestawy przesunięć czasowych. Udział poszczególnych parametrów w analizowanej puli rozwiązań zaprezentowano na rys. 25-28.

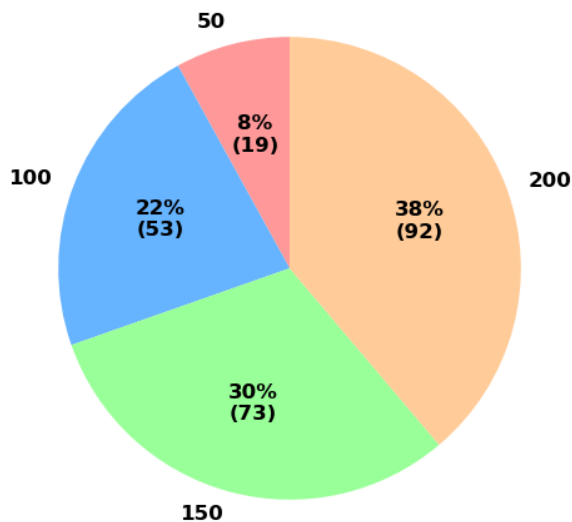
Rozmiar populacji



Rysunek 25. Udział procentowy rozmiarów populacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

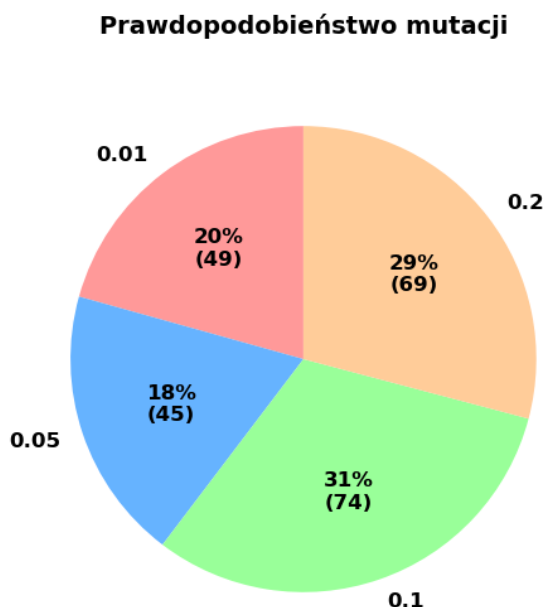
Jak wynika z rys. 25, najczęściej występującym rozmiarem populacji była populacja licząca 30 osobników. Następną grupą pod względem liczebności była populacja składająca się z 50 osobników. Najrzadziej występowała populacja o rozmiarze 200 osobników, a populacje o rozmiarach 150 i 100 osobników stanowiły odpowiednio 13% i 16%.

Liczba generacji



Rysunek 26. Udział procentowy liczby generacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

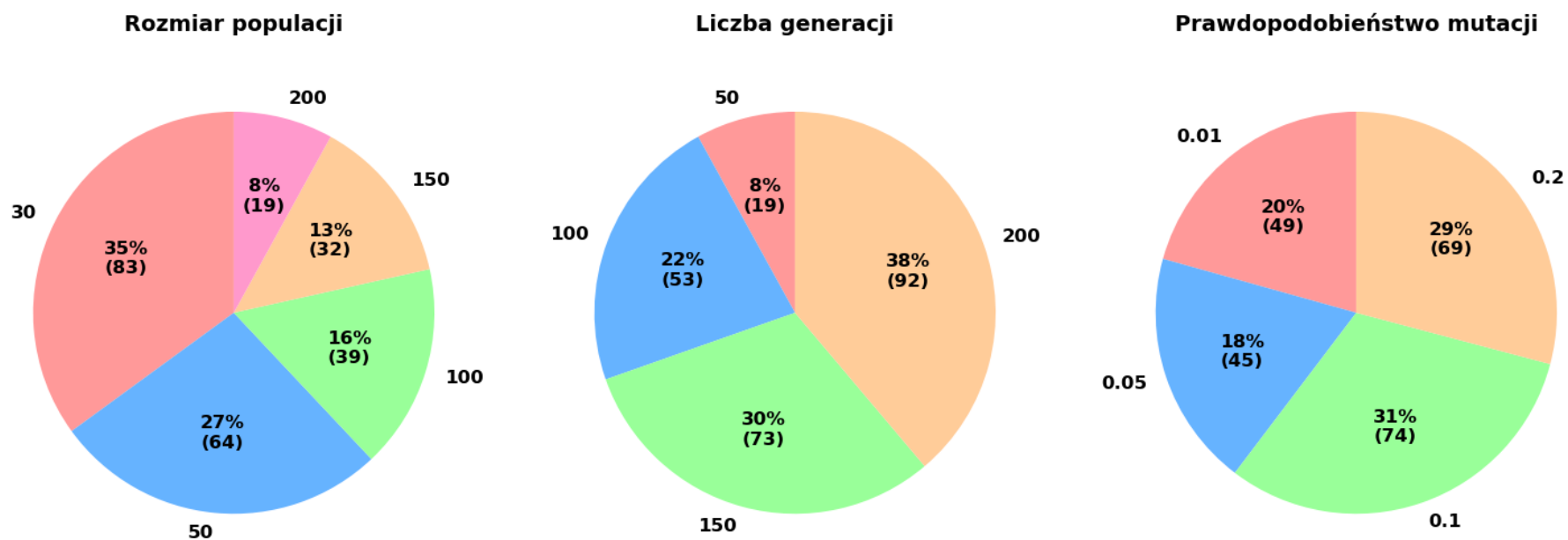
Jak przedstawia rys. 26, najwięcej rozwiązań (w sumie 68%) pochodziło z serii eksperymentów, w których parametr liczby generacji przyjmował wartość 150 lub 200 (odpowiednio 30% i 38%). Liczba generacji 100 wystąpiła w 22% rozwiązań, a liczba generacji równa 50 w 8%.



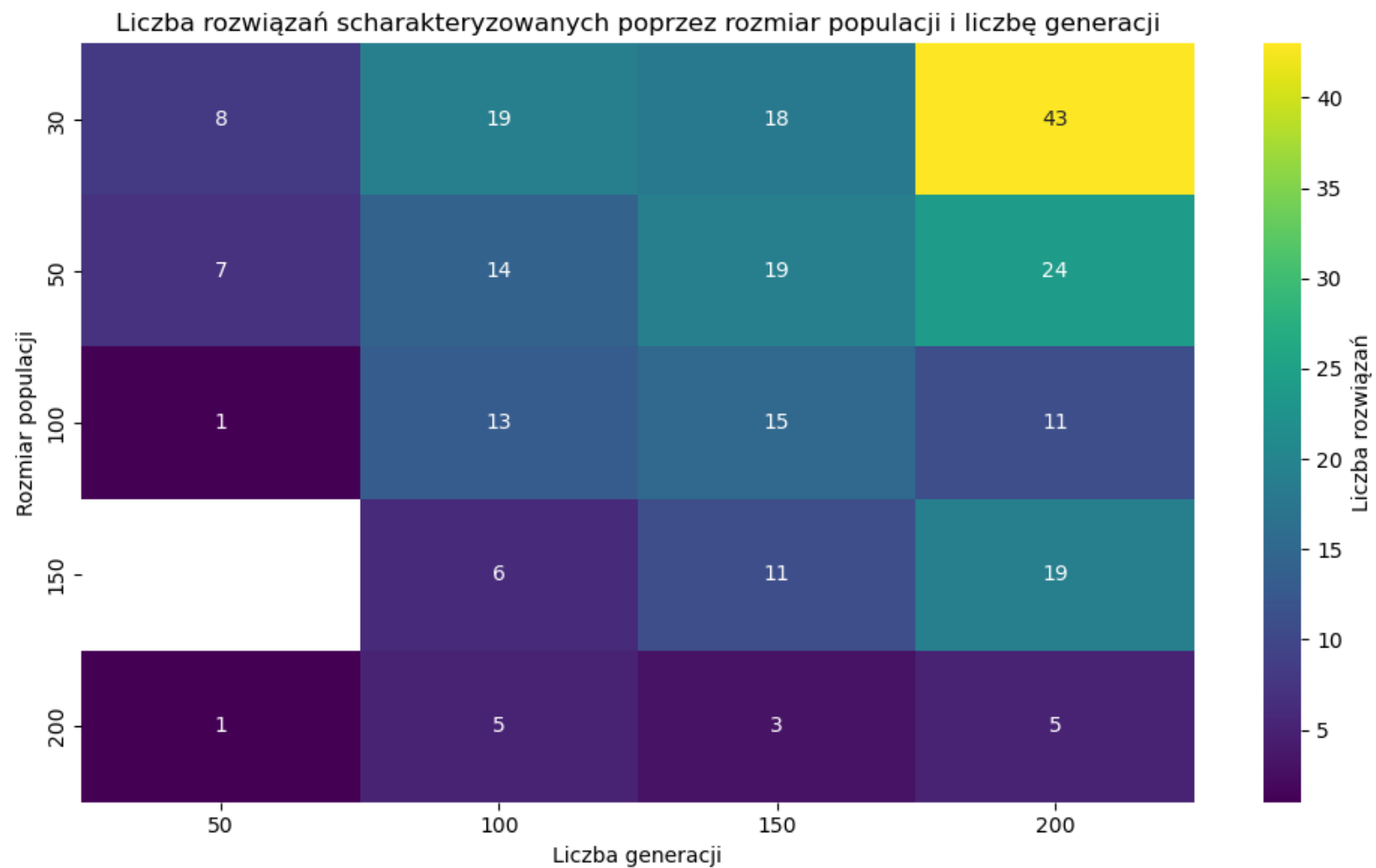
Rysunek 27. Udział procentowy prawdopodobieństwa mutacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

Zgodnie z wykresem przedstawionym na rys. 27, najwięcej rozwiązań pochodziło z serii eksperymentu z parametrem prawdopodobieństwa mutacji równym 0,1 (31%). W podobnej liczbie rozwiązań (29%) parametr prawdopodobieństwa mutacji przyjmował wartość 0,2. Mniejszym udziałem charakteryzowały się rozwiązania z prawdopodobieństwem mutacji 0,01 i 0,05 odpowiednio 20% i 18%.

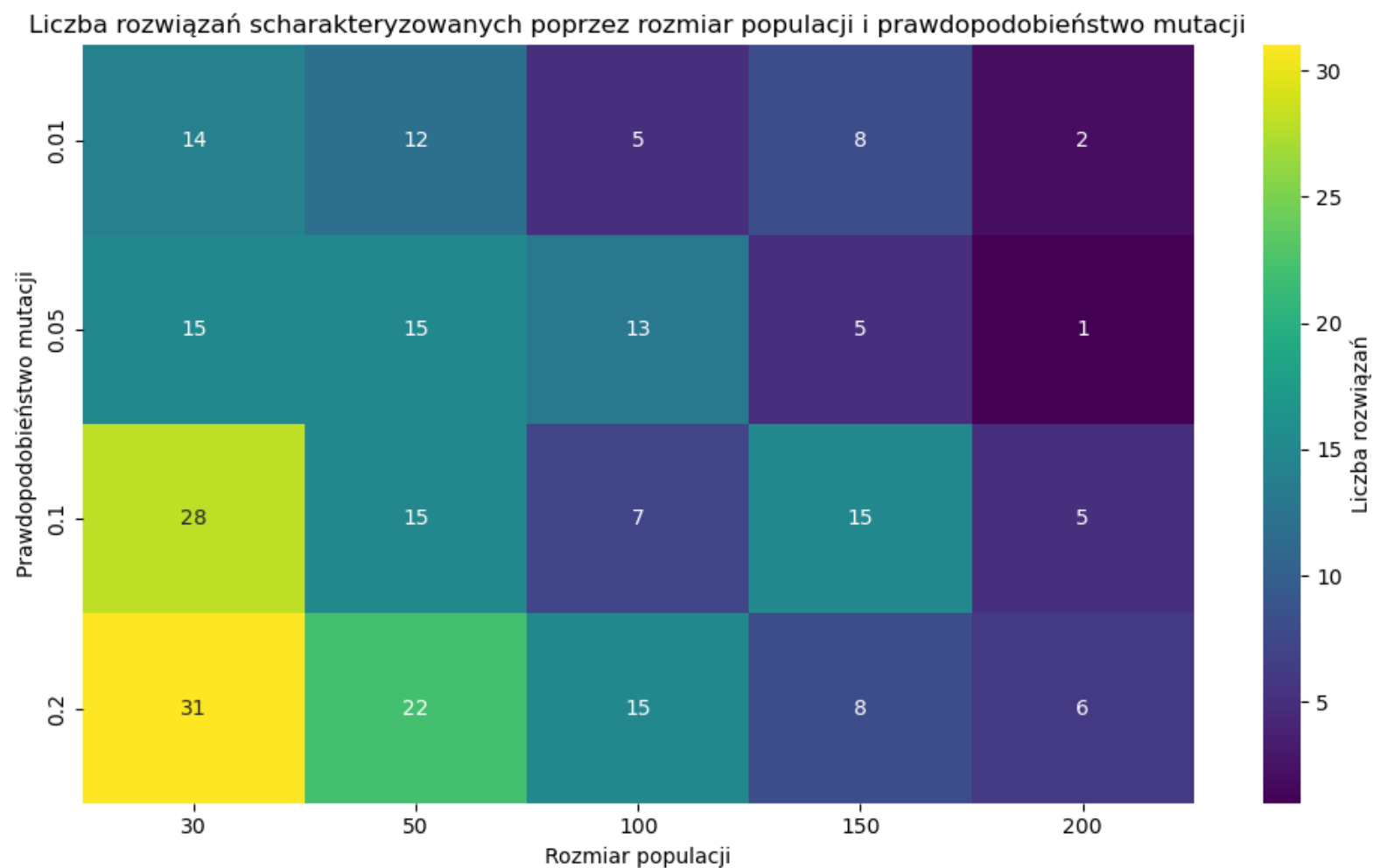
Scharakteryzowano również otrzymane rozwiązania za pomocą zestawów parametrów. Charakterystykę rozwiązań wykonaną za pomocą par parametrów: liczba osobników w populacji i liczba generacji, liczba osobników w populacji i prawdopodobieństwo mutacji oraz liczba generacji i prawdopodobieństwo mutacji zaprezentowano na rys. 29-31.



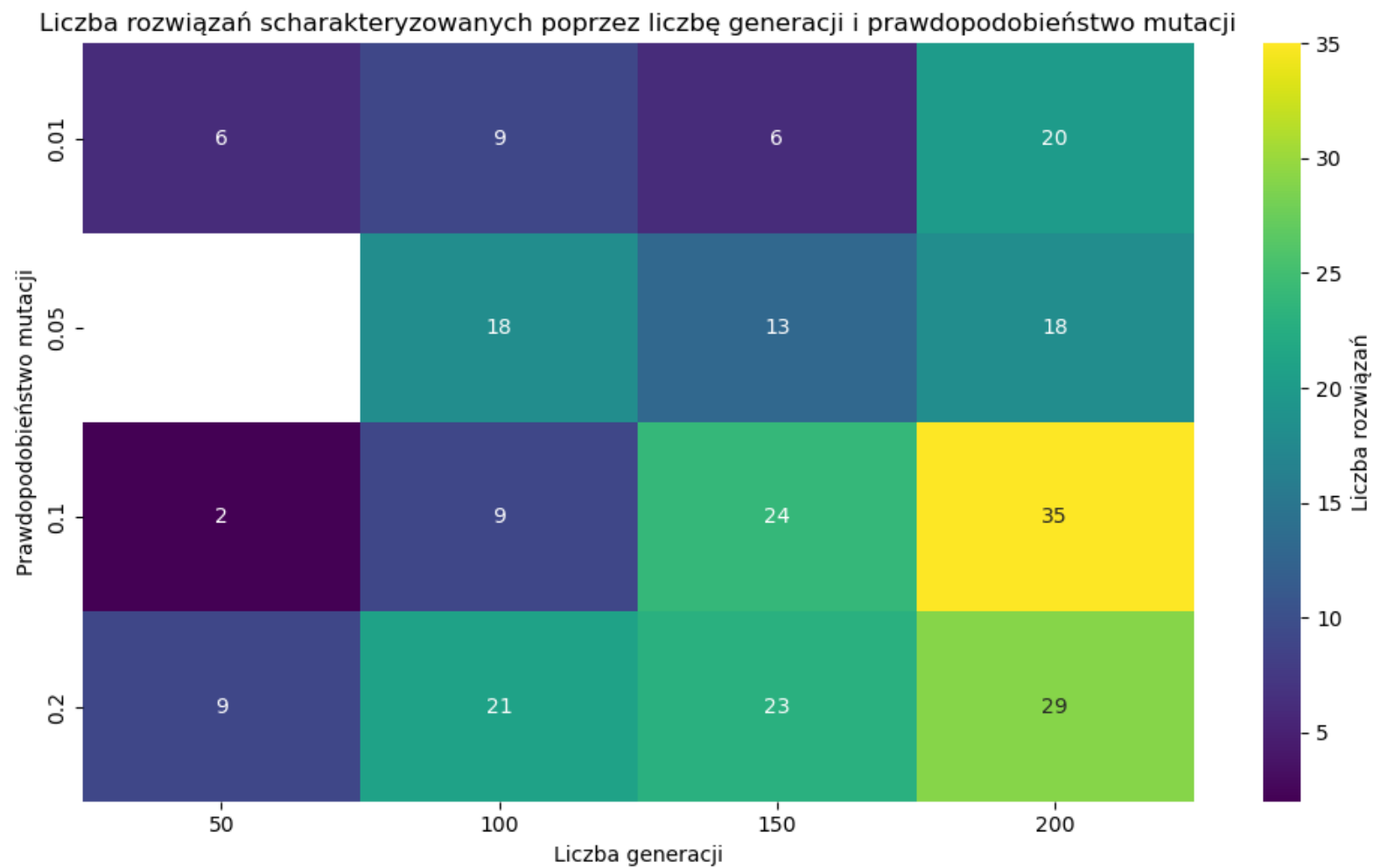
Rysunek 28. Zbiorcze zestawienie udziałów procentowych poszczególnych parametrów w analizowanej puli rozwiązań
Źródło: Opracowanie własne



Rysunek 29. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i liczby generacji w analizowanej puli rozwiązań
 Źródło: Opracowanie własne



Rysunek 30. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne



Rysunek 31. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

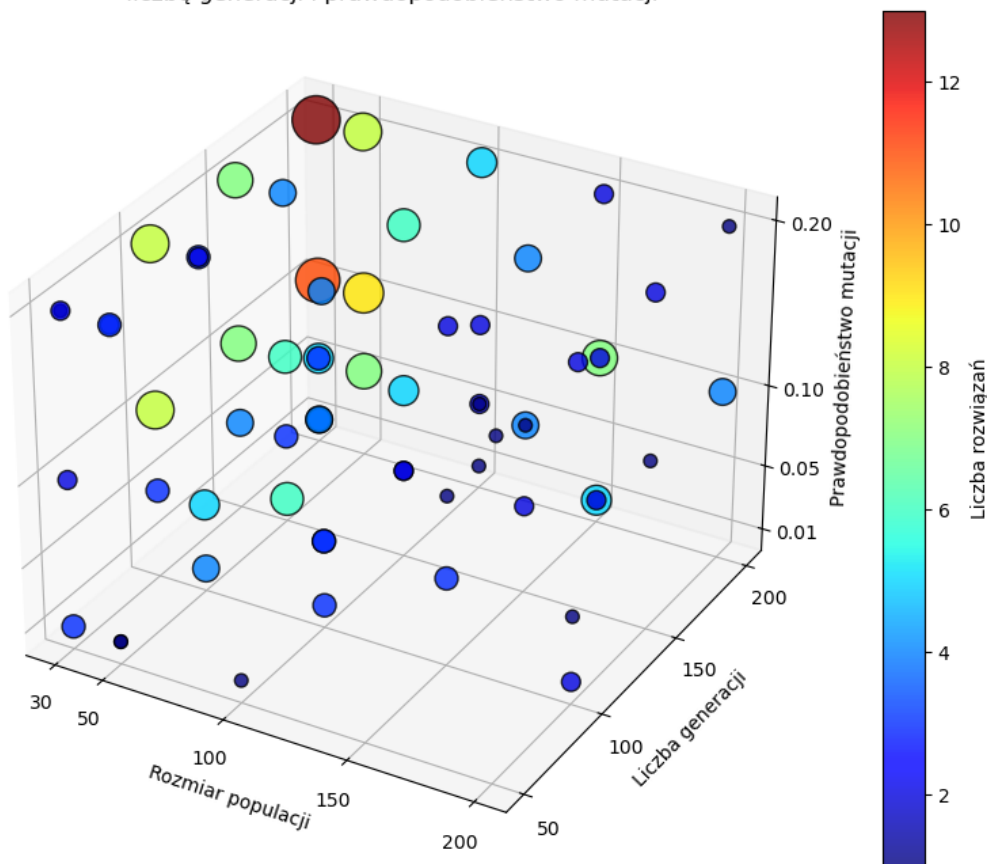
Dane na rys. 29 wskazują na to, że częściej występowały rozwiązania powstałe z par parametrów charakteryzujących się niską liczbą osobników w populacji (30 i 50) i dużą liczbą generacji (150 i 200).

Rys. 30 ukazuje, że częściej występowały rozwiązania powstałe z par parametrów charakteryzujących się niską liczbą osobników w populacji (30 i 50) i prawdopodobieństwem mutacji równym 0,1 i 0,2.

Rys. 31 ilustruje częstszą obecność rezultatów powstałych z par parametrów charakteryzujących się dużą liczbą generacji (150 i 200) i prawdopodobieństwem mutacji równym 0,1 i 0,2.

Rys. 32 przedstawia liczbę rozwiązań w zależności od zestawienia trzech parametrów (rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji) w analizowanej puli rozwiązań.

Liczba rozwiązań scharakteryzowanych poprzez rozmiar populacji, liczbę generacji i prawdopodobieństwo mutacji



Rysunek 32. Wykres 3D przedstawiający liczbę rozwiązań dla różnych wartości rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań

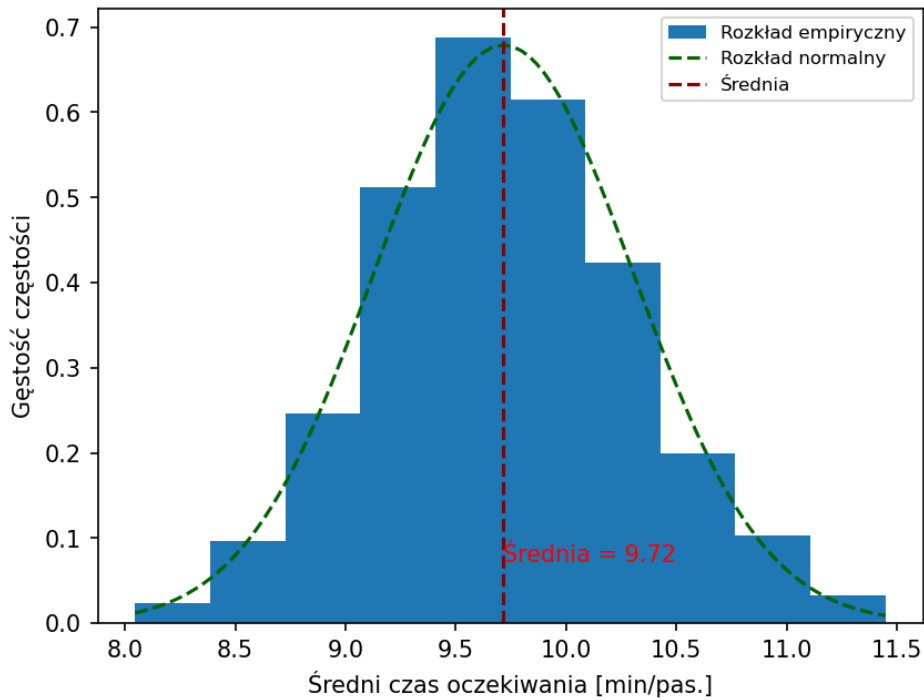
Źródło: Opracowanie własne

Dane na rys. 32 wskazują, że najczęściej występowały rozwiązania, w których rozmiar populacji była niski (30 i 50), liczba generacji była wysoka (150 i 200) a prawdopodobieństwo mutacji wynosiło 0,1 i 0,2.

W ostatnim kroku eksperymentu wyselekcjonowane wcześniej zestawy przesunięć czasowych porównano z rzeczywistym rozkładem jazdy sześciu linii z zielonogórskiego systemu publicznego transportu zbiorowego. W tym celu odczytano rozkład jazdy badanych linii z modelu symulacyjnego miasta Zielona Góra wykonanego w programie PTV Visum i zapisano go w postaci zestawu przesunięć czasowych. Z modelu symulacyjnego miasta uzyskano również dane o popycie na podróże w ciągu jednej godziny wyrażone w postaci listy pasażerów analizowanych linii z informacją o przystanku początkowym i końcowym ich podróży.

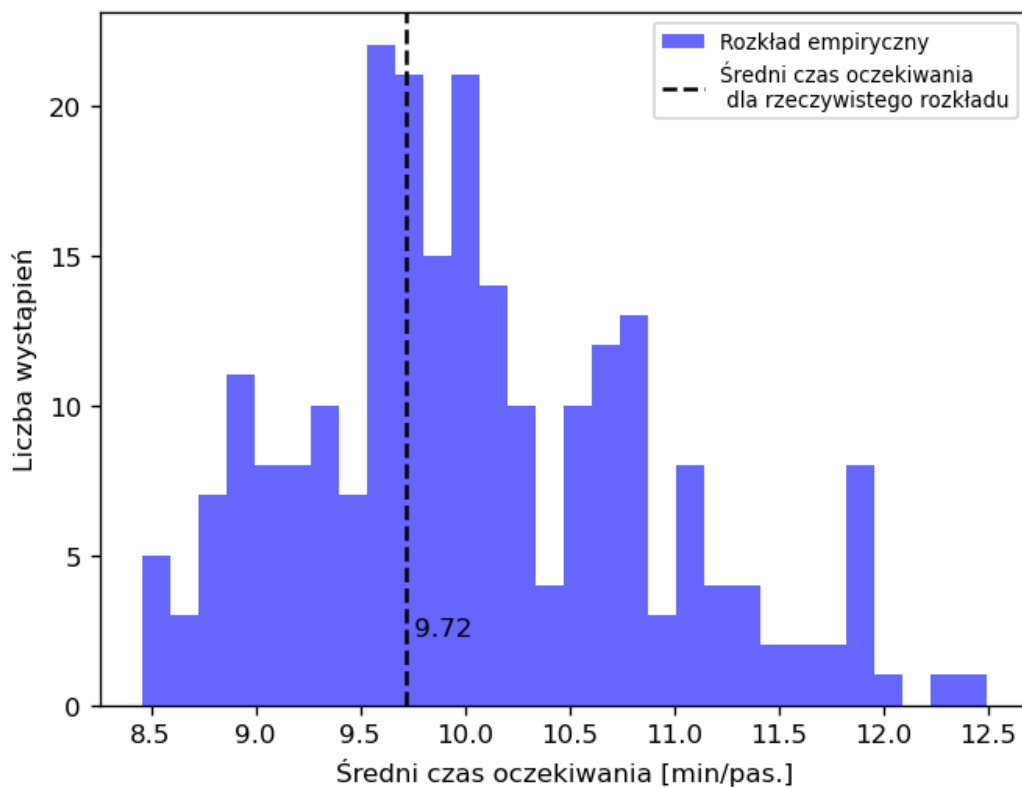
Część modelu odpowiedzialną za wyliczenie wartości funkcji celu (metoda *fitness_function*) uruchomiono z rzeczywistymi danymi o popycie na podróże i rozkładem jazdy sześciu linii trzysta razy. W celu zwiększenia realizmu symulacji dane o popycie rzeczywistym uzupełniono o czas pojawienia się pasażera na przystanku początkowym, który dla każdego pasażera generowany był losowo przez klasę *Stochastic* przy każdym uruchomieniu. Dla trzystu otrzymanych wartości funkcji celu (średnich czasów oczekiwania na przesiadkę) obliczono ich średnią arytmetyczną, który wyniosła 9,72 minuty na pasażera. Histogram rozkładu średnich czasów oczekiwania na przesiadkę dla rzeczywistego popytu i rozkładu jazdy przedstawia rys. 33.

Analogicznie część modelu odpowiedzialną za wyliczenie wartości funkcji celu (metoda *fitness_function*) uruchomiono trzysta razy z rzeczywistymi danymi o popycie na podróże dla każdego z 237 wyselekcjonowanych w kroku piątym eksperymentu zestawu przesunięć czasowych. Ponownie dane o popycie rzeczywistym uzupełniono o czas pojawienia się pasażera na przystanku początkowym, który dla każdego pasażera generowany był losowo przez klasę *Stochastic* przy każdym uruchomieniu. Obliczono średnią arytmetyczną trzystu otrzymanych wartości funkcji celu dla każdego z 237 zestawów przesunięć czasowych. Histogram rozkładu obliczonych średnich arytmetycznych przedstawiono na rys. 34.



Rysunek 33. Histogram rozkładu średnich czasów oczekiwania na przesiadkę dla rzeczywistego popytu i rozkładu jazdy

Źródło: Opracowanie własne



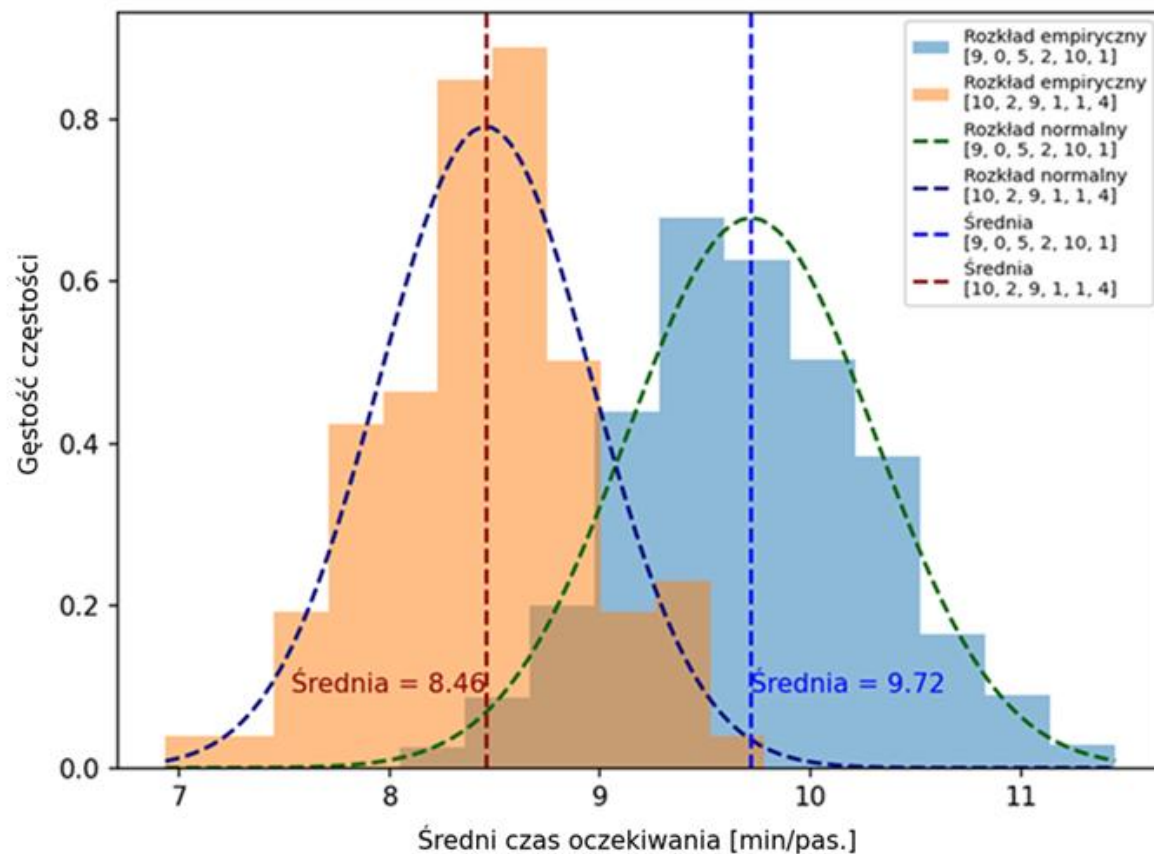
Rysunek 34. Histogram rozkładu średnich czasów oczekiwania na przesiadkę dla zestawów przesunięć czasowych zaproponowanych przez algorytm genetyczny

Źródło: Opracowanie własne

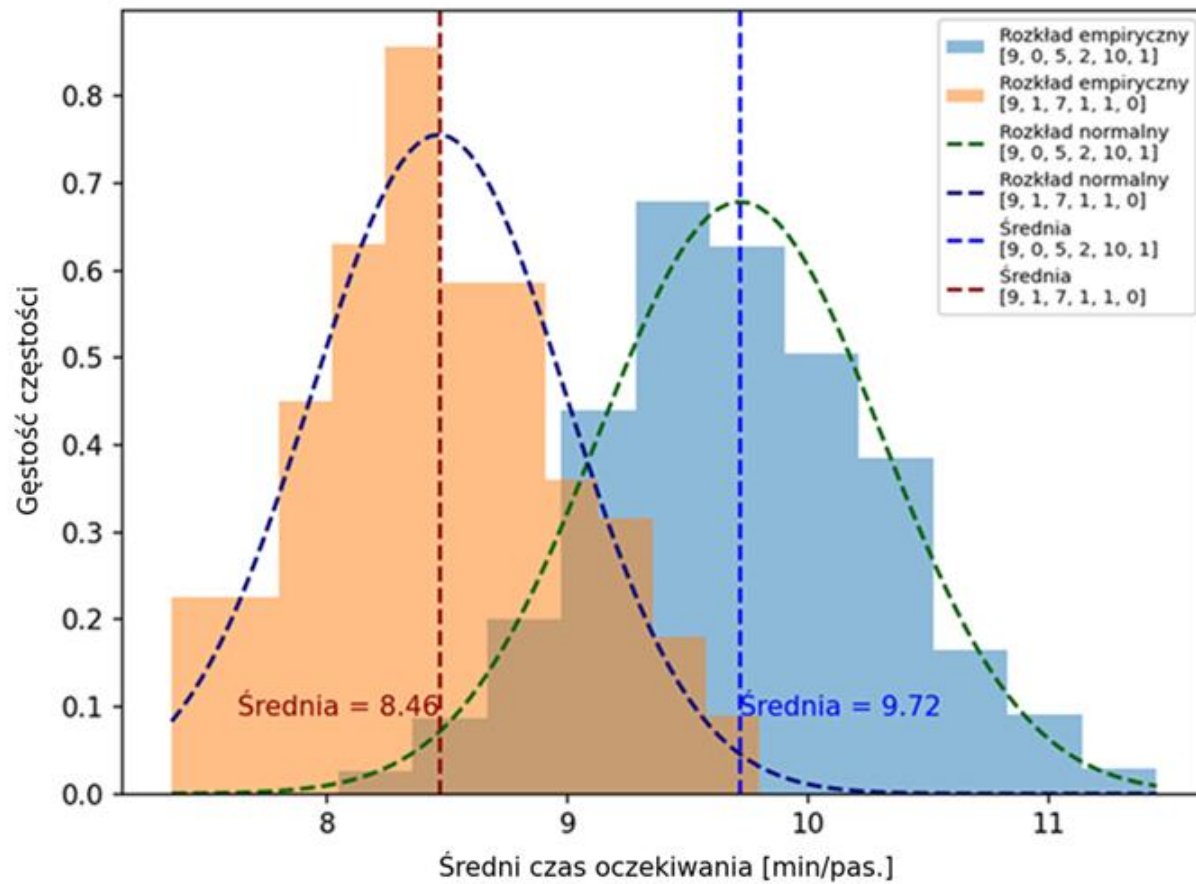
Na rys. 34 dodatkowo oznaczono czarną przerywaną linią średnią arytmetyczną średniego czasu oczekiwania dla rzeczywistego rozkładu jazdy i popytu obliczonego w kroku ósmym. Wśród 237 porównywanych zestawów przesunięć czasowych 90 z nich (37,97%) charakteryzowało się krótszym średnim czasem oczekiwania na przesiadkę od średniego czasu oczekiwania dla rzeczywistego rozkładu jazdy. Pozostałe 147 charakteryzowały się dłuższym średnim czasem oczekiwania na przesiadkę.

Rys. 35-39 przedstawiają porównanie histogramów pięciu zestawów przesunięć czasowych uzyskanych w dziewiątym kroku eksperymentu wybranych na podstawie najkrótszego średniego czasu oczekiwania na przesiadkę z histogramem rozkładu średnich czasów oczekiwania na przesiadkę dla rzeczywistego popytu i rozkładu jazdy.

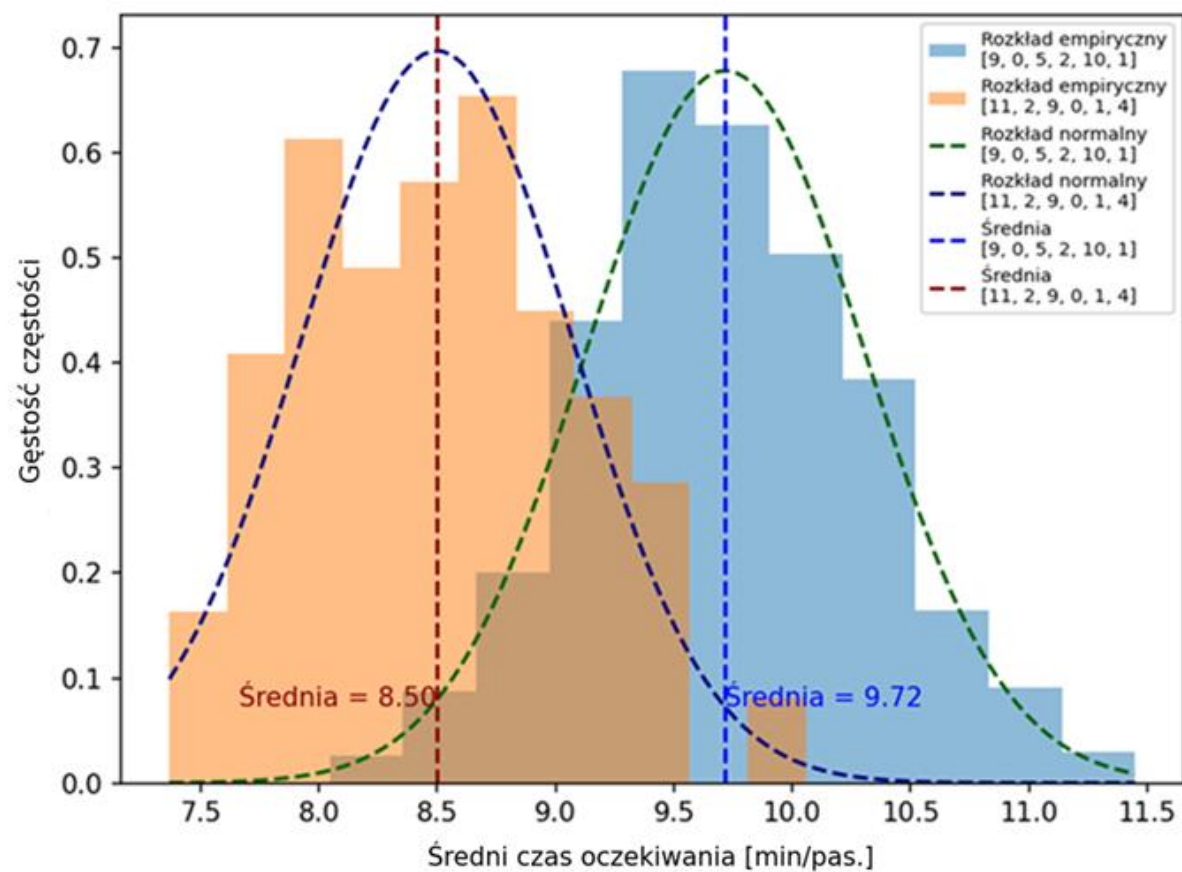
Jak przedstawiono na rys. 35 i 36, najkrótszym średnim czasem oczekiwania charakteryzowały się dwa zestawy przesunięć czasowych $\Psi = [10, 2, 9, 1, 1, 4]$ oraz $\Psi = [9, 1, 7, 1, 1, 0]$. Wynosił on 8,46 minuty na pasażera. Dla kolejnych trzech zestawów przesunięć czasowych $\Psi = [11, 2, 9, 0, 1, 4]$ (rys. 37), $\Psi = [8, 1, 4, 6, 0, 1]$ (rys. 38) oraz $\Psi = [7, 2, 6, 0, 2, 3]$ (rys. 39) wynosił on odpowiednio 8,5, 8,52 i 8,57 minuty na pasażera.



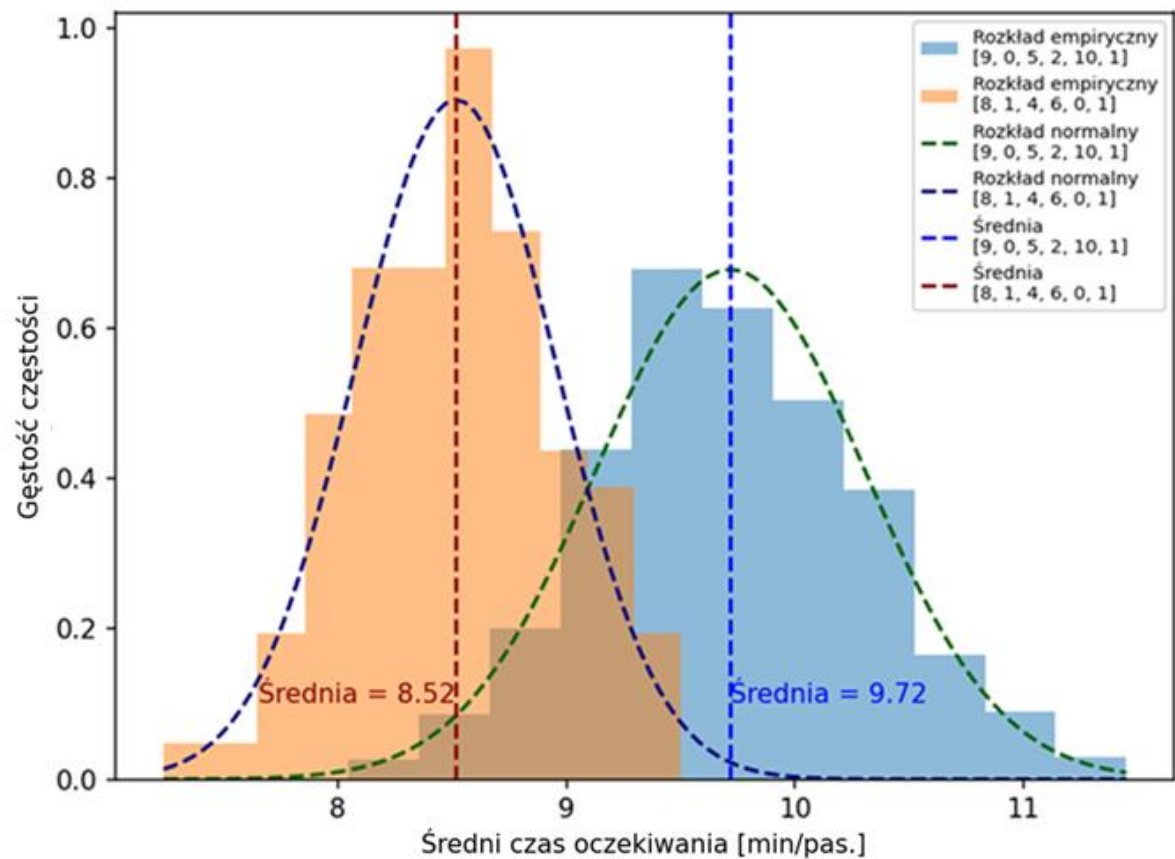
Rysunek 35. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [10, 2, 9, 1, 1, 4]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy
 Źródło: Opracowanie własne



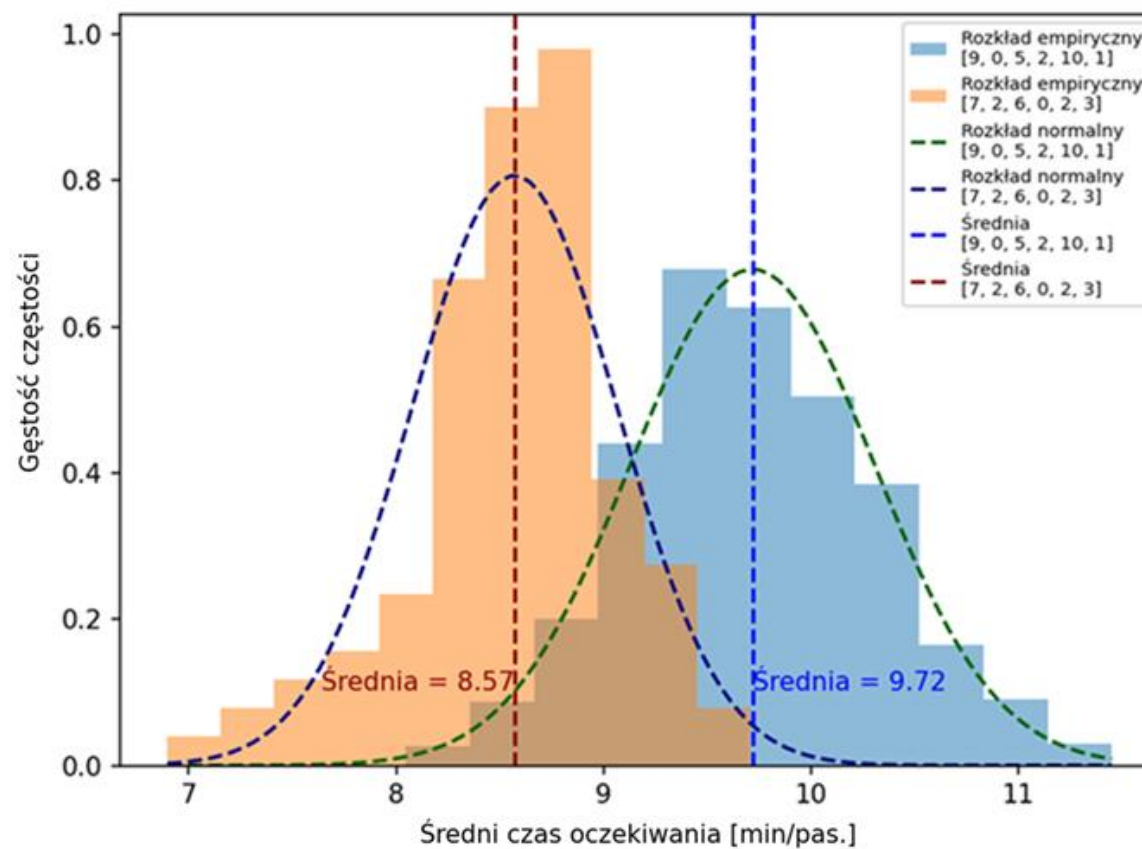
Rysunek 36. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [9, 1, 7, 1, 1, 0]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy
 Źródło: Opracowanie własne



Rysunek 37. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [11, 2, 9, 0, 1, 4]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy
Źródło: Opracowanie własne

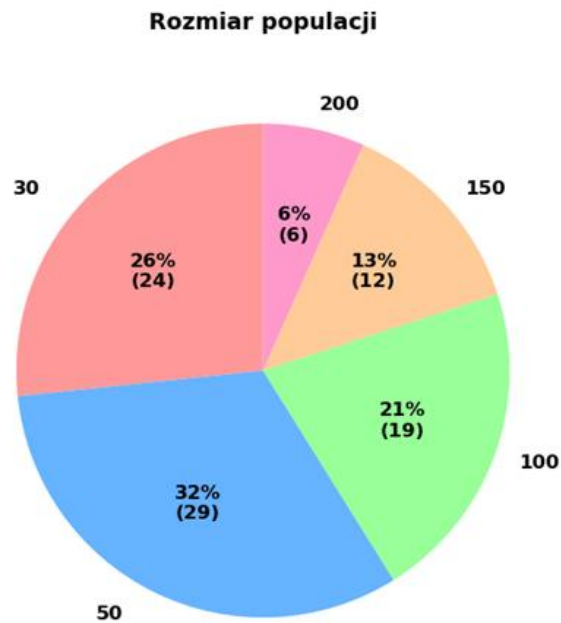


Rysunek 38. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [8, 1, 4, 6, 0, 1]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy
 Źródło: Opracowanie własne



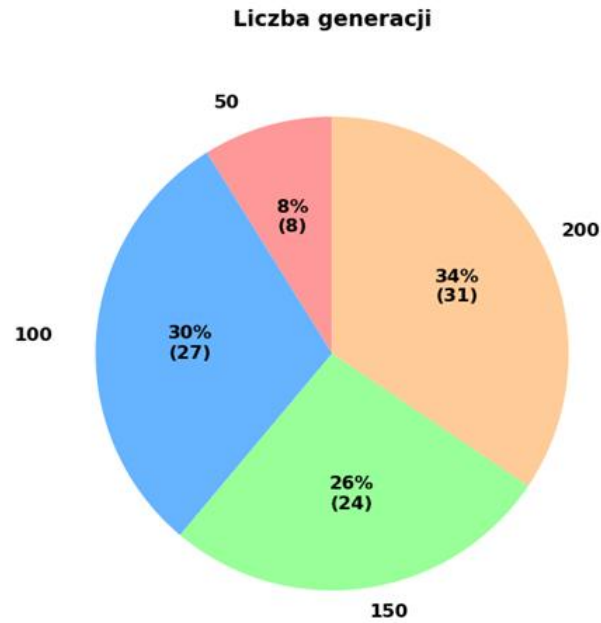
Rysunek 39. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesuńć $\Psi = [7, 2, 6, 0, 2, 3]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy
Źródło: Opracowanie własne

Na następnym etapie eksperymentu obliczeniowego odczytano parametry algorytmu genetycznego, przy których otrzymano 90 zestawów przesunięć czasowych (rys. 33), które charakteryzowały się krótszym średnim czasem oczekiwania na przesiadkę od średniego czasu oczekiwania dla rzeczywistego rozkładu jazdy. Udział poszczególnych parametrów wśród 90 wspomnianych rozwiązań zaprezentowano na rys. 40-43.



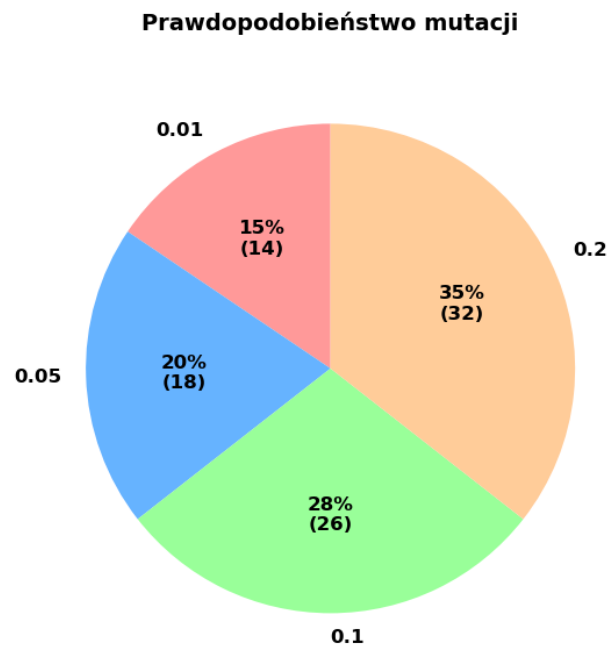
Rysunek 40. Udział procentowy rozmiaru populacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

Jak wynika z rysunku 40. najczęściej występującym rozmiarem populacji była populacja licząca 50 osobników (32%). Następną grupą pod względem liczebności był rozmiar populacji liczący 30 osobników (26%), podobnym udziałem (21%) charakteryzowała się grupa z populacją 100 osobników. Najrzadziej występowała populacja o rozmiarze 200 osobników (6%), a populacja o rozmiarze 150 osobników stanowiła 13%.



Rysunek 41. Udział procentowy liczby generacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

Jak przedstawia rys. 41, najrzadziej wśród rozwiązań występowały te otrzymane z parametrem liczby generacji równym 50 (8%). Pozostałe wartości parametrów tj. 100, 150 i 200 występowały z podobną częstością, odpowiednio 30%, 26% i 34%.



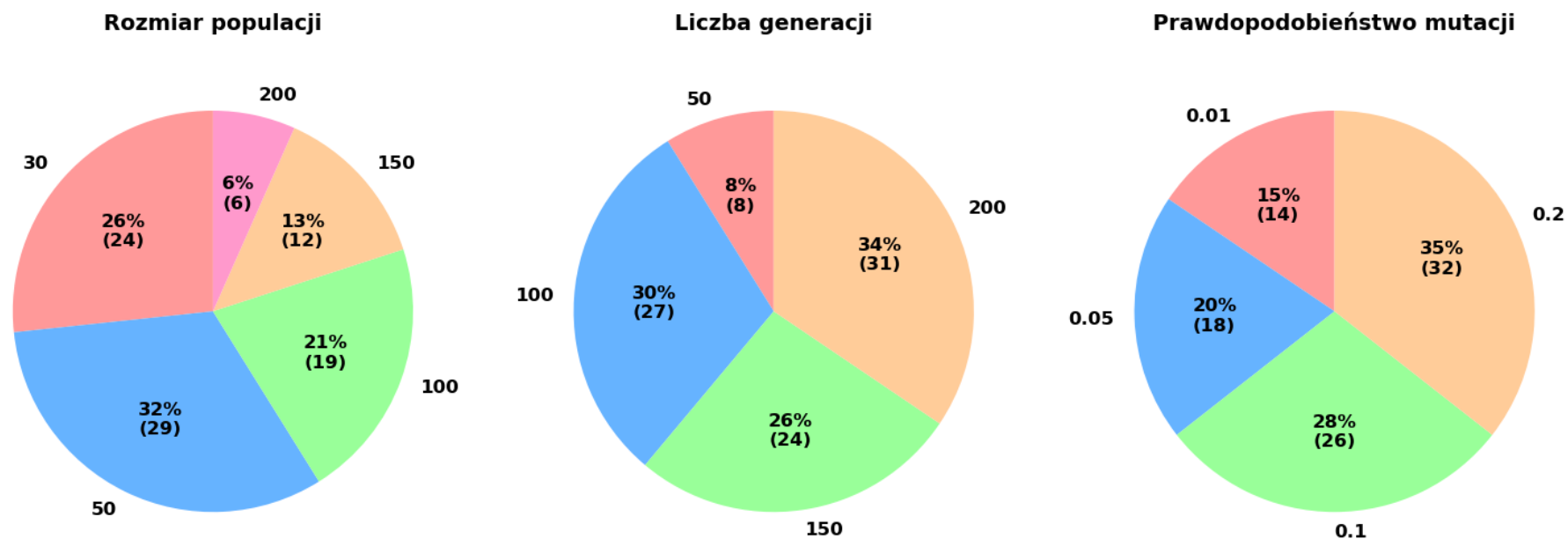
Rysunek 42. Udział procentowy prawdopodobieństwa mutacji w analizowanej puli rozwiązań
Źródło: Opracowanie własne

Zgodnie z wykresem przedstawionym na rys. 42, najwięcej rozwiązań pochodziło z serii eksperymentu z parametrem prawdopodobieństwa mutacji równym 0,2 (35%). W podobnej liczbie rozwiązań (28%) parametr prawdopodobieństwa mutacji przyjmował wartość 0,1. Mniejszym udziałem charakteryzowały się rozwiązania z prawdopodobieństwem mutacji 0,01 i 0,05 odpowiednio 15% i 20%.

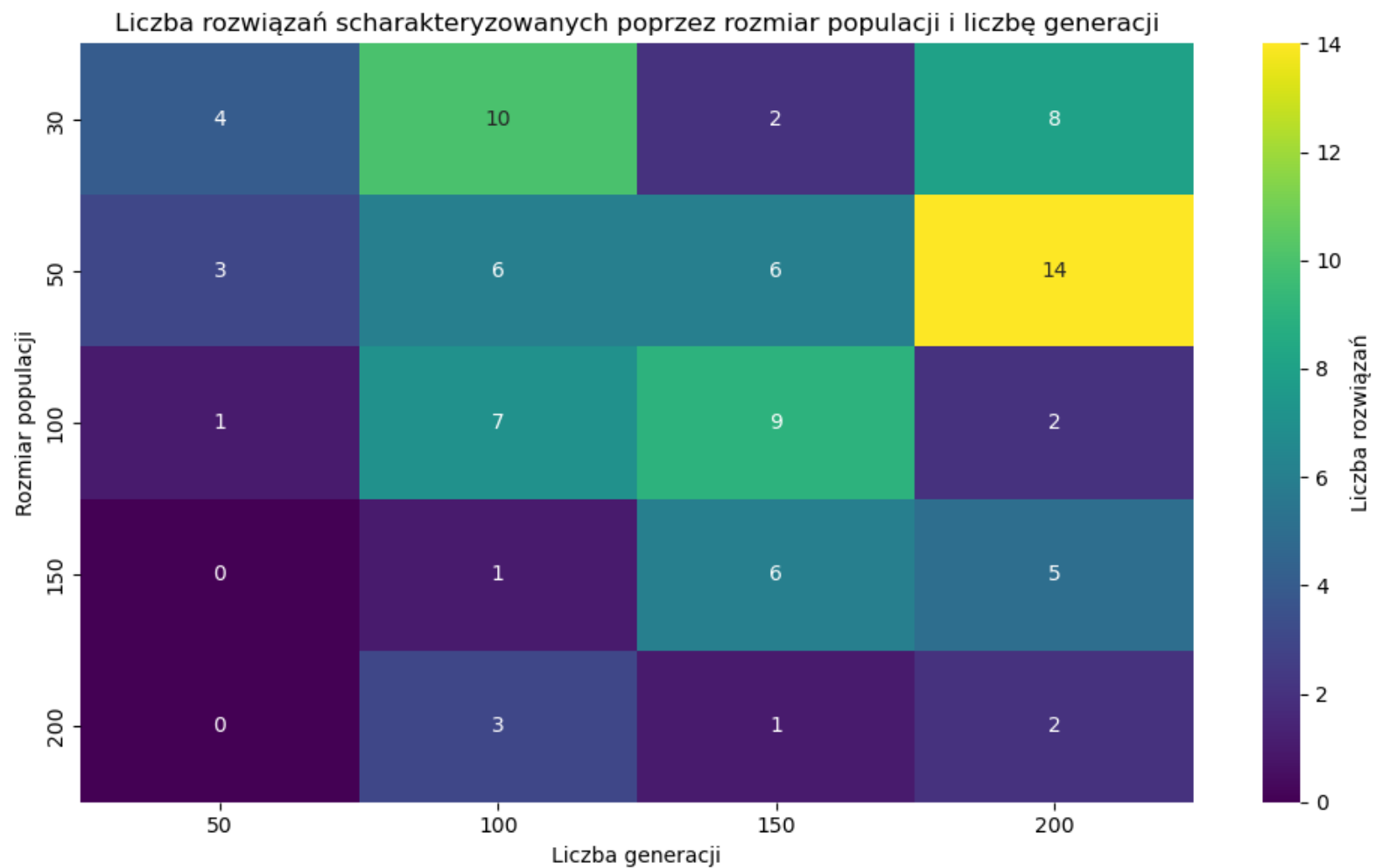
Scharakteryzowano również otrzymane rozwiązania za pomocą zestawów parametrów. Charakterystykę rozwiązań wykonaną za pomocą par parametrów: liczba osobników w populacji i liczba generacji, liczba osobników w populacji i prawdopodobieństwo mutacji oraz liczba generacji i prawdopodobieństwo mutacji zaprezentowano na rys. 44-46.

Dane na rys. 44 wskazują, że najczęściej występowały rozwiązania powstałe z par parametrów charakteryzujących się liczbą osobników równą 30 i liczbą generacji równą 200. Kolejnymi pod względem częstości występowania były pary parametrów: rozmiar populacji 30 i liczba generacji 100, rozmiar populacji 100 i liczba generacji 150 oraz rozmiar populacji 30 i liczba generacji 200.

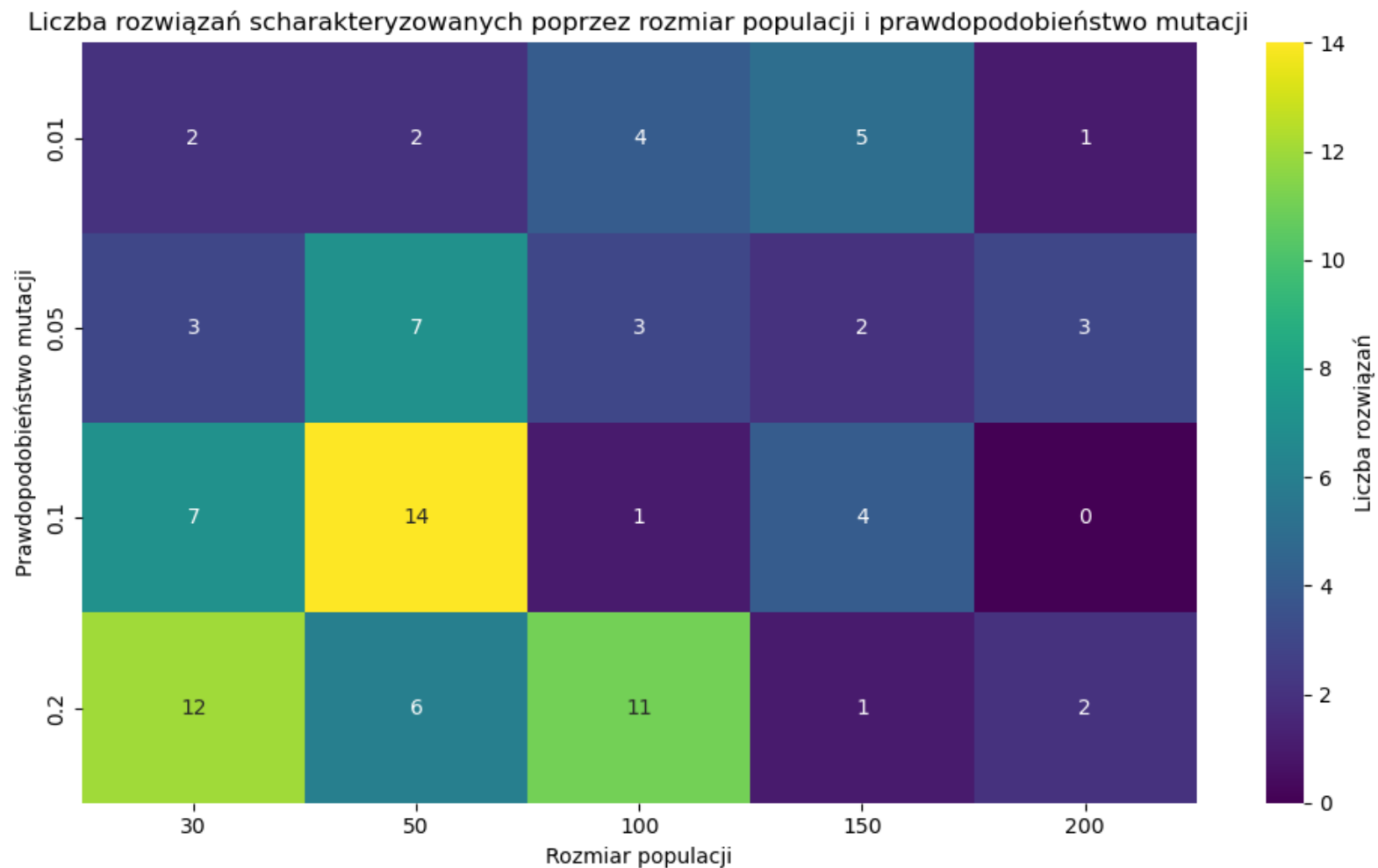
Jak można zauważyć na rys. 45, więcej rozwiązań występowało przy niższych rozmiarach populacji i prawdopodobieństwach mutacji 0,1 i 0,2. Najczęściej występowały rozwiązania powstałe z par parametrów charakteryzujących się liczbą osobników równą 50 i prawdopodobieństwem mutacji 0,1. Kolejnymi pod względem częstości występowania były pary parametrów: rozmiar populacji 30 i prawdopodobieństwo mutacji 0,2 oraz rozmiar populacji 100 i prawdopodobieństwo mutacji 0,2.



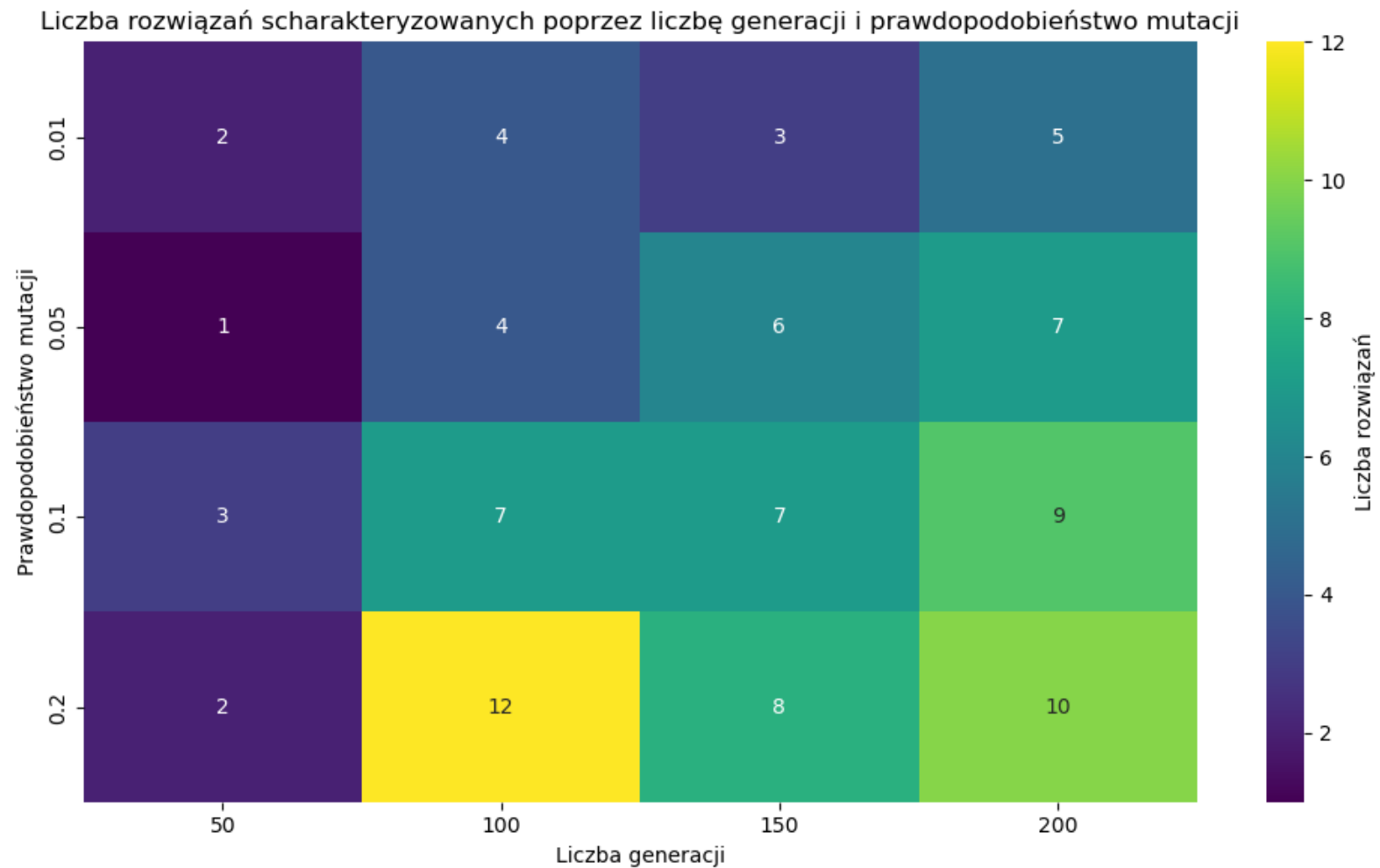
Rysunek 43. Zbiorcze zestawienie udziałów procentowych poszczególnych parametrów w analizowanej puli rozwiązań
 Źródło: Opracowanie własne



Rysunek 44. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i liczby generacji w analizowanej puli rozwiązań
 Źródło: Opracowanie własne



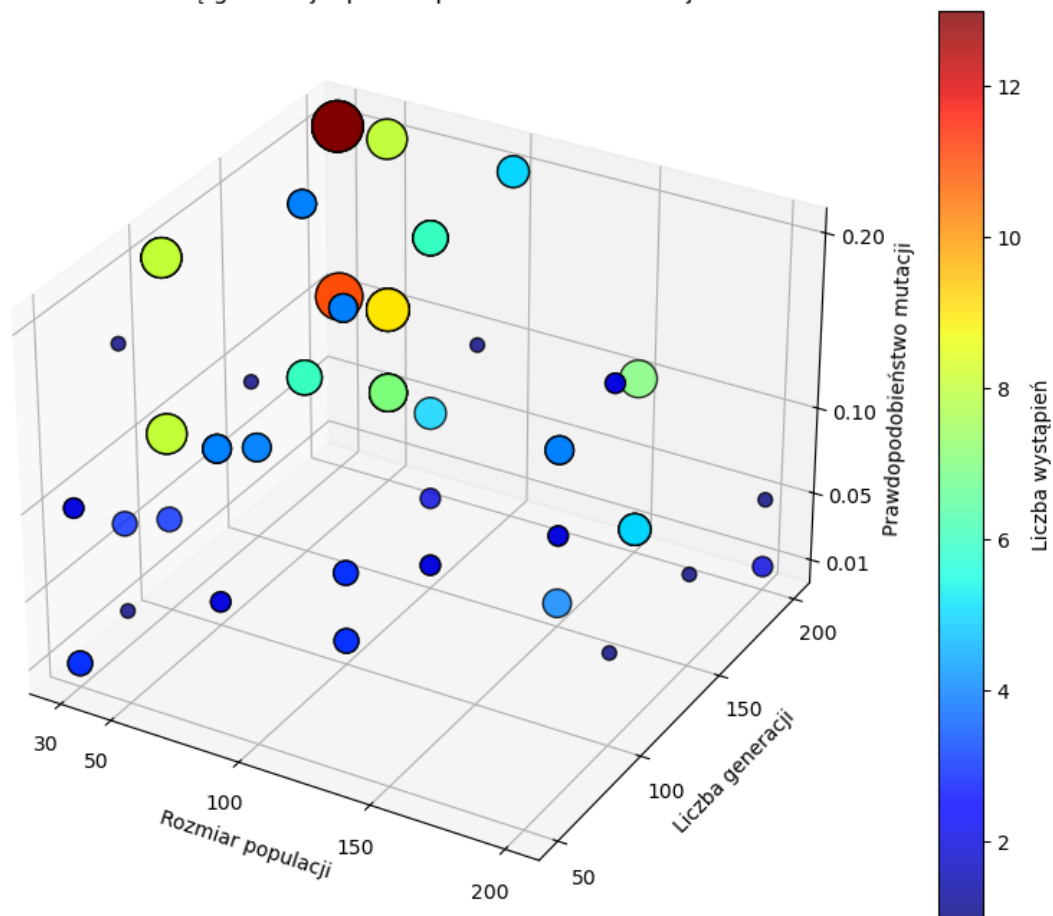
Rysunek 45. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań
 Źródło: Opracowanie własne



Rysunek 46. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań
 Źródło: Opracowanie własne

Jak wynika z rys. 46, więcej rozwiązań występowało przy większych liczbach generacji i prawdopodobieństwach mutacji 0,1 i 0,2. Najczęściej występowały rozwiązania powstałe z par parametrów charakteryzujących się liczbą generacji równą 100 i prawdopodobieństwem mutacji 0,2. Kolejnymi pod względem częstości występowania były pary parametrów: liczba generacji 200 i prawdopodobieństwo mutacji 0,2 oraz liczba generacji 200 i prawdopodobieństwo mutacji 0,1. Rys. 47 przedstawia liczbę rozwiązań w zależności od zestawienia trzech parametrów (rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji) w analizowanej puli rozwiązań.

Liczba rozwiązań scharakteryzowanych poprzez rozmiar populacji, liczbę generacji i prawdopodobieństwo mutacji



Rysunek 47. Wykres 3D przedstawiający liczbę rozwiązań dla różnych wartości rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań

Źródło: Opracowanie własne

Rys. 47 wskazuje, że najwięcej rezultatów otrzymano w serii eksperymentów, w których rozmiar populacji była niski (30 i 50), liczba generacji wynosiła 100 lub więcej, a prawdopodobieństwo mutacji było równe 0,1 lub 0,2.

4.2.3 Ocena wpływu parametru odpowiedzialnego za prawdopodobieństwo wystąpienia mutacji na wartość funkcji celu

Z powodu zaobserwowania podobnej liczebności wyników otrzymanych w opisanym w niniejszej pracy eksperymencie z parametrem odpowiedzialnym za prawdopodobieństwo mutacji równym 0,1 i 0,2 zdecydowano o przeprowadzeniu dodatkowego badania wpływu wartości tego parametru na wartość funkcji celu.

W związku z dużym odstępem pomiędzy wartościami parametru 0,1 i 0,2 postanowiono przeanalizować wartości prawdopodobieństwa mutacji: 0,11; 0,13; 0,15; 0,17 i 0,19.

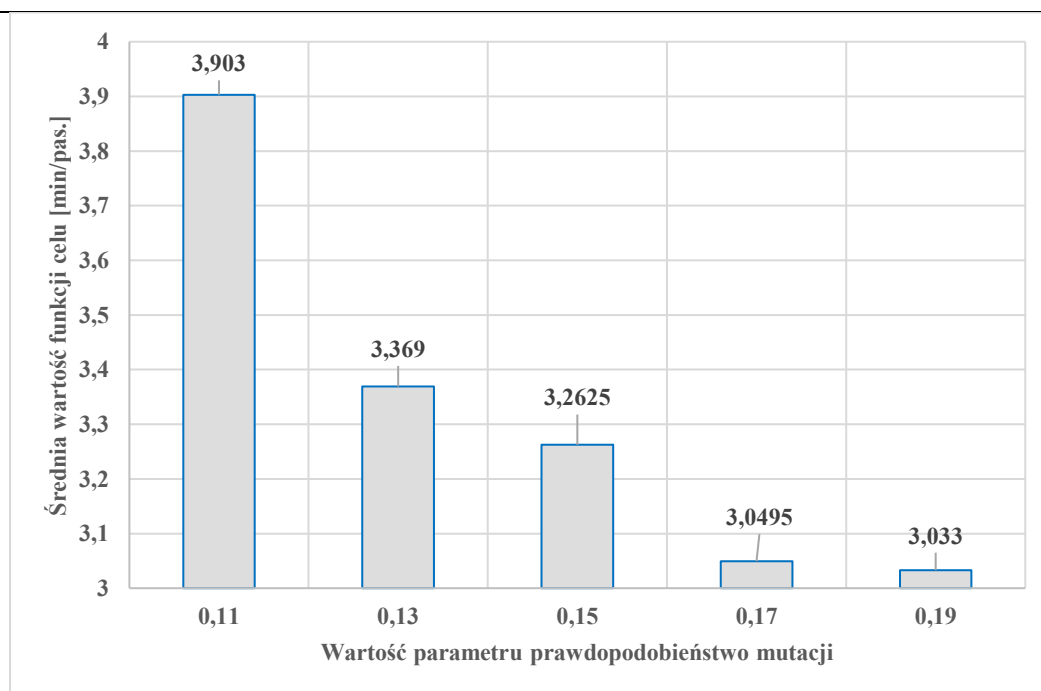
Dla każdej z powyższych wartości prawdopodobieństwa mutacji oraz liczebności populacji równej 50 i liczby generacji równej 200 uruchomiono algorytm genetyczny 200 razy z losowo generowanym popytem. Obliczono średnią arytmetyczną dwustu otrzymanych wartości funkcji celu dla każdego z pięciu zestawów parametrów. Rezultaty poddano analizie statystycznej – wszystkie wyniki mieściły się w 95% przedziale ufności.

Średnią wartość funkcji celu (średni czas oczekiwania na przesiadkę) dla każdego zestawu parametrów przedstawiono w tabeli 23 oraz na rys. 48.

Tabela 23. Średnia wartość funkcji celu dla poszczególnych zestawów parametrów

Parametry			Średnia wartość funkcji celu
Rozmiar populacji	Liczba generacji	Prawdopodobieństwo mutacji	
50	200	0,11	3,903
50	200	0,13	3,369
50	200	0,15	3,263
50	200	0,17	3,05
50	200	0,19	3,033

Źródło: Opracowanie własne



Rysunek 48. Średnia wartość funkcji celu w zależności od parametru prawdopodobieństwo mutacji
Źródło: Opracowanie własne

Wyniki, prezentowane w tabeli 23 oraz na rys. 48, wskazują, że wraz ze wzrostem wartości parametru prawdopodobieństwa mutacji średnia wartość funkcji celu (średni czas oczekiwania pasażera na przesiadkę) malała. Dla wartości parametru odpowiadającego za prawdopodobieństwo mutacji równego 0,11 średni czas oczekiwania pasażera na przesiadkę wynosił 3,903 minuty. Niższe czasy oczekiwania otrzymano dla parametrów o wartościach 0,17 i 0,19, które były do siebie zbliżone i wynosiły odpowiednio 3,0495 i 3,033 minuty.

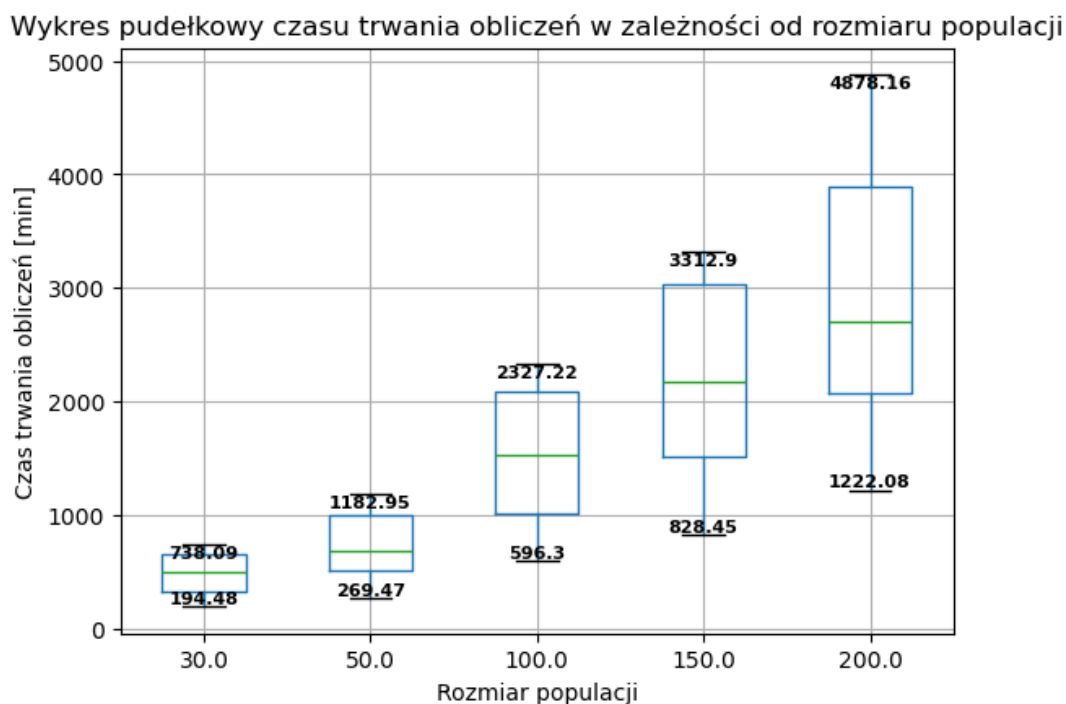
Podsumowując, w dodatkowej analizie wpływu wartości prawdopodobieństwa mutacji na wartość funkcji celu, podobnie jak w eksperymencie opisanym w poprzednim rozdziale, zaobserwowano otrzymywanie niższych wartości funkcji wraz ze zwiększaniem wartości tego parametru.

4.3 Wpływ parametrów algorytmu genetycznego na czas wykonania obliczeń

Podczas uruchomienia algorytmu genetycznego z 80 różnymi zestawami parametrów w kroku pierwszej głównego eksperymentu opisanego w niniejszej pracy zapisywano również czas trwania obliczeń dla każdej jego serii. Czasy trwania obliczeń przeanalizowano w zależności od rozmiaru populacji, liczby generacji, prawdopodobieństwa mutacji oraz par tych parametrów.

4.3.1 Wpływ rozmiaru populacji na czas trwania obliczeń

W celu lepszego zobrazowania zależności czasu trwania obliczeń od rozmiaru populacji dane przedstawiono na wykresie pudełkowym (rys. 49), gdzie zaznaczone są minimalny i maksymalny czas trwania obliczeń, średni czas trwania obliczeń (zielona linia) oraz zakresy pomiędzy pierwszym i trzecim kwantylem (niebieski prostokąt). W tabeli 24 zebrano minimalny, średni i maksymalny czas trwania obliczeń dla badanych rozmiarów populacji.



Rysunek 49. Zależność czasu trwania obliczeń od rozmiaru populacji

Źródło: Opracowanie własne

Tabela 24. Czasy trwania obliczeń dla różnych rozmiarów populacji

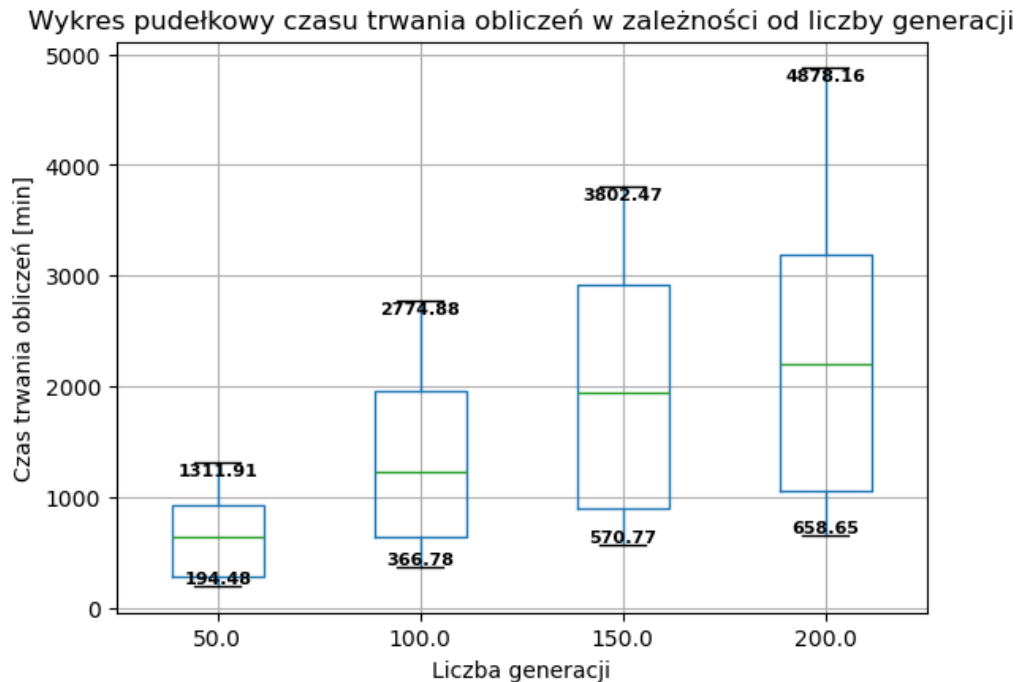
	Rozmiar populacji				
	30	50	100	150	200
Minimalny czas trwania obliczeń [min]	194,48	269,47	596,3	828,45	1222,08
Średni czas trwania obliczeń [min]	474,58	735,58	1513,12	2175,71	2891,4
Maksymalny czas trwania obliczeń [min]	738,09	1182,95	2327,22	3312,9	4878,16

Źródło: Opracowanie własne

Zgodne z danymi zaprezentowanymi w tabeli 24 oraz na rys. 49, najkrócej obliczenia były wykonywane dla rozmiaru populacji równego 30. Średni czas obliczeń dla tej wartości parametru trwał 474,58 minuty. Największy średni czas wynosił 2891,4 minuty dla rozmiaru populacji równego 200. Można zauważyć, że im większy rozmiar populacji tym dłużej trwają obliczenia, a także zwiększa się różnica pomiędzy minimalnym a maksymalnym czasem trwania obliczeń.

4.3.2 Wpływ liczby generacji na czas trwania obliczeń

W celu lepszego zobrazowania zależności czasu trwania obliczeń od liczby generacji dane przedstawiono na wykresie pudełkowym (rys. 50), gdzie zaznaczone są minimalny i maksymalny czas trwania obliczeń, średni czas trwania obliczeń (zielona linia) oraz zakresy pomiędzy pierwszym i trzecim kwantylem (niebieski prostokąt). W tabeli 25 zebrano minimalny, średni i maksymalny czas trwania obliczeń dla badanych liczb generacji.



Rysunek 50. Zależność czasu trwania obliczeń od liczby generacji
Źródło: Opracowanie własne

Tabela 25. Czasy trwania obliczeń dla różnych liczb generacji

	Liczba generacji			
	50	100	150	200
Minimalny czas trwania obliczeń [min]	194,48	366,78	570,77	658,65
Średni czas trwania obliczeń [min]	662,19	1337,32	1922,24	2310,65
Maksymalny czas trwania obliczeń [min]	1311,91	2774,88	3802,47	4878,16

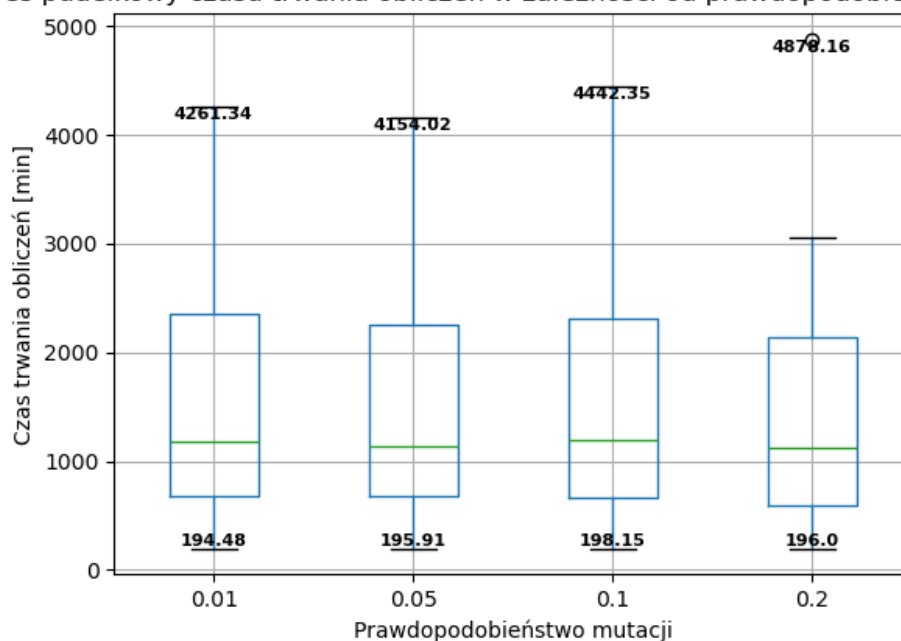
Źródło: Opracowanie własne

Zgodne z danymi zaprezentowanymi w tabeli 25 oraz na rys. 50, najkrócej obliczenia były wykonywane dla liczby generacji wynoszącej 50. Średni czas obliczeń dla tej wartości parametru trwał 662,19 minuty. Wraz ze wzrostem liczby generacji czas trwania obliczeń rósł oraz zwiększała się różnica pomiędzy minimalnym a maksymalnym czasem trwania obliczeń. Największy średni czas wyniósł 2310,65 minuty dla liczby generacji równej 200.

4.3.3 Wpływ prawdopodobieństwa mutacji na czas trwania obliczeń

W celu lepszego zobrazowania zależności czasu trwania obliczeń od prawdopodobieństwa mutacji dane przedstawiono na wykresie pudełkowym (rys. 51), gdzie zaznaczone są minimalny i maksymalny czas trwania obliczeń, średni czas trwania obliczeń (zielona linia), zakresy pomiędzy pierwszym i trzecim kwantylem (niebieski prostokąt) oraz wartość odstająca zaznaczona punktem. W tabeli 26. zebrano minimalny, średni i maksymalny czas trwania obliczeń dla badanych wartości prawdopodobieństwa mutacji.

Wykres pudełkowy czasu trwania obliczeń w zależności od prawdopodobieństwa mutacji



Rysunek 51. Zależność czasu trwania obliczeń od prawdopodobieństwa mutacji

Źródło: Opracowanie własne

Tabela 26. Czasy trwania obliczeń dla różnych prawdopodobieństw mutacji

	Prawdopodobieństwo mutacji			
	0,01	0,05	0,1	0,2
Minimalny czas trwania obliczeń [min]	194,48	195,91	198,15	196
Średni czas trwania obliczeń [min]	1586	1576,46	1593,61	1476,32
Maksymalny czas trwania obliczeń [min]	4261,34	4154,02	4442,35	4878,16

Źródło: Opracowanie własne

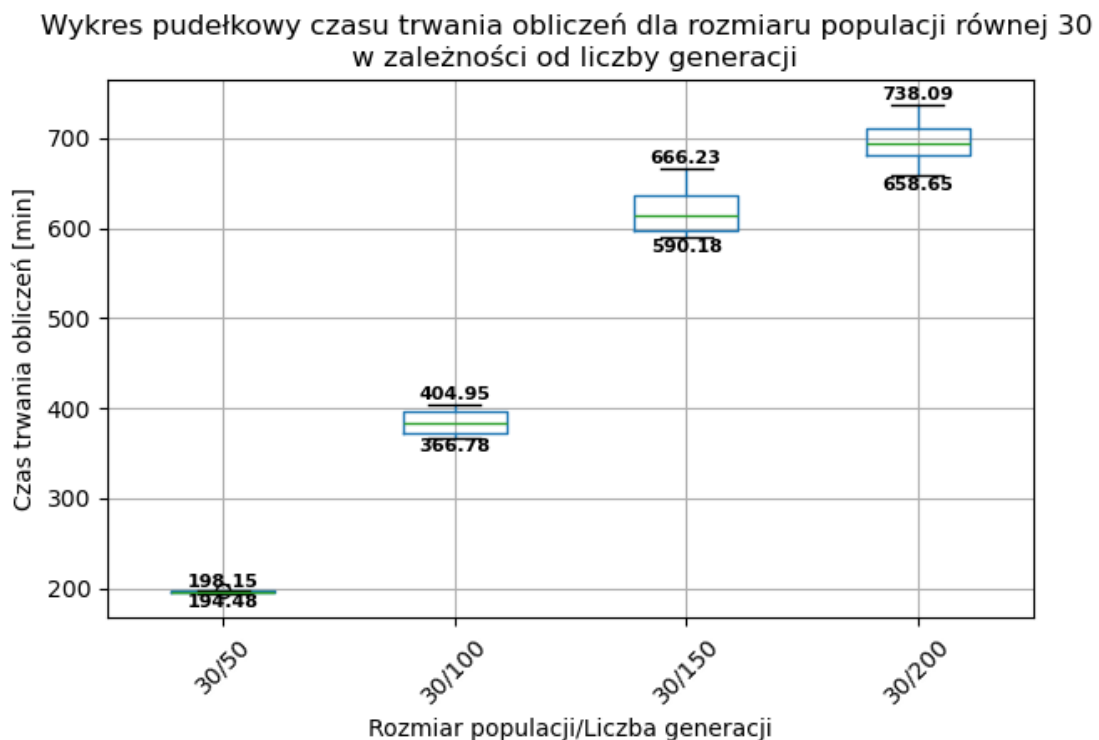
Zgodne z danymi zaprezentowanymi w tabeli 26 oraz na rys. 51, minimalne, średnie i maksymalne czasy trwania obliczeń dla różnych prawdopodobieństw mutacji nie różniły się znacząco od siebie. Dla wszystkich wartości prawdopodobieństwa mutacji można zauważyć szeroką rozpiętość zakresu pomiędzy minimalnym a maksymalnym czasem trwania obliczeń.

4.3.4 Wpływ rozmiaru populacji wraz z liczbą generacji na czas trwania obliczeń

W związku z zauważeniem zależności pomiędzy czasem trwania obliczeń a rozmiarem populacji oraz pomiędzy czasem trwania obliczeń a liczbą generacji przeprowadzono analizę wpływu zestawień tych parametrów na czas trwania obliczeń.

Podobnie jak w poprzednim rozdziale w celu lepszego zobrazowania zależności czasu trwania obliczeń od rozmiaru populacji i liczby generacji dane przedstawiono na wykresach pudełkowych, gdzie zaznaczone są minimalny i maksymalny czas trwania obliczeń, średni czas trwania obliczeń (zielona linia), zakresy pomiędzy pierwszym i trzecim kwantylem (niebieski prostokąt) oraz wartości odstające zaznaczone punktem.

Na rys. 52 i w tabeli 27 zaprezentowano czas trwania obliczeń dla rozmiaru populacji równego 30 w zależności od liczby generacji w serii eksperymentu.



Rysunek 52. Zależność czasu trwania obliczeń dla populacji równej 30 od liczby generacji

Źródło: Opracowanie własne

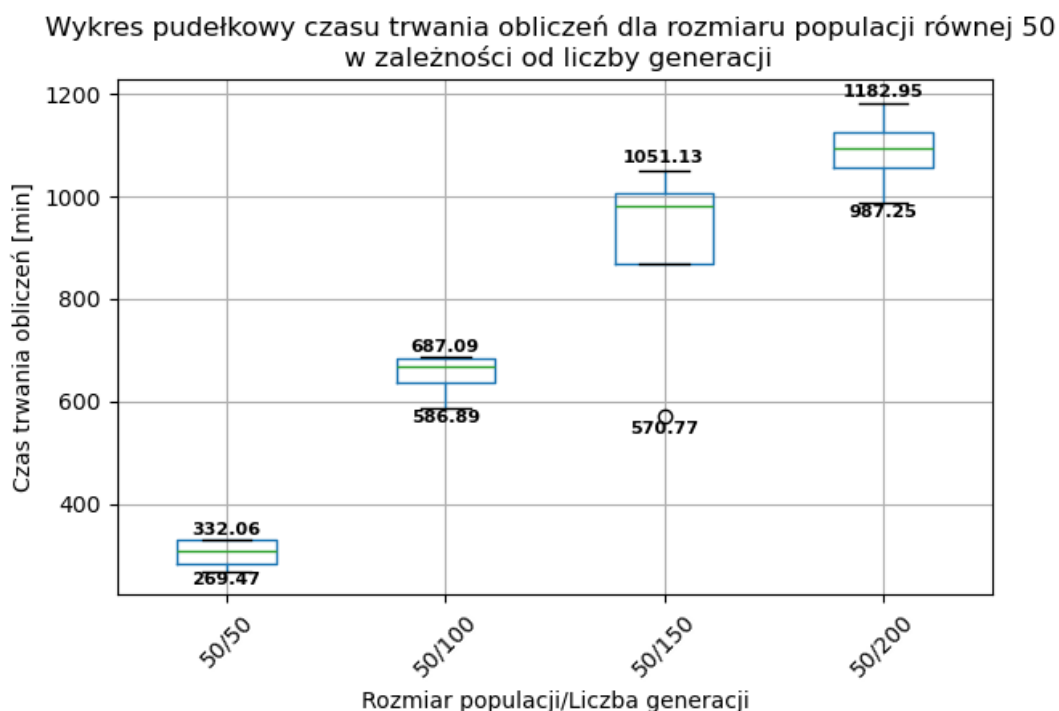
Tabela 27. Czasy trwania obliczeń dla populacji równej 30 i różnych liczb generacji

	Liczba generacji			
	50	100	150	200
Minimalny czas trwania obliczeń [min]	194,48	366,78	590,18	658,65
Średni czas trwania obliczeń [min]	196,14	384,98	620,87	696,65
Maksymalny czas trwania obliczeń [min]	198,15	404,95	666,23	738,09

Źródło: Opracowanie własne

Zgodnie z danymi zaprezentowanymi na rys. 52 oraz w tabeli 27, przy stałym rozmiarze populacji równym 30 najkrócej obliczenia były wykonywane dla liczby generacji wynoszącej 50. Różnica pomiędzy minimalnym i maksymalnym czasem trwania obliczeń wynosiła mniej niż 3 minuty. Wraz ze wzrostem liczby generacji minimalne, średnie i maksymalne czasy obliczeń rosły.

Na rys. 53 i w tabeli 28 zaprezentowano czas trwania obliczeń dla rozmiaru populacji równego 50 w zależności od liczby generacji w serii eksperymentu.



Rysunek 53. Zależność czasu trwania obliczeń dla populacji równej 50 od liczby generacji

Źródło: Opracowanie własne

Tabela 28. Czasy trwania obliczeń dla populacji równej 50 i różnych liczb generacji

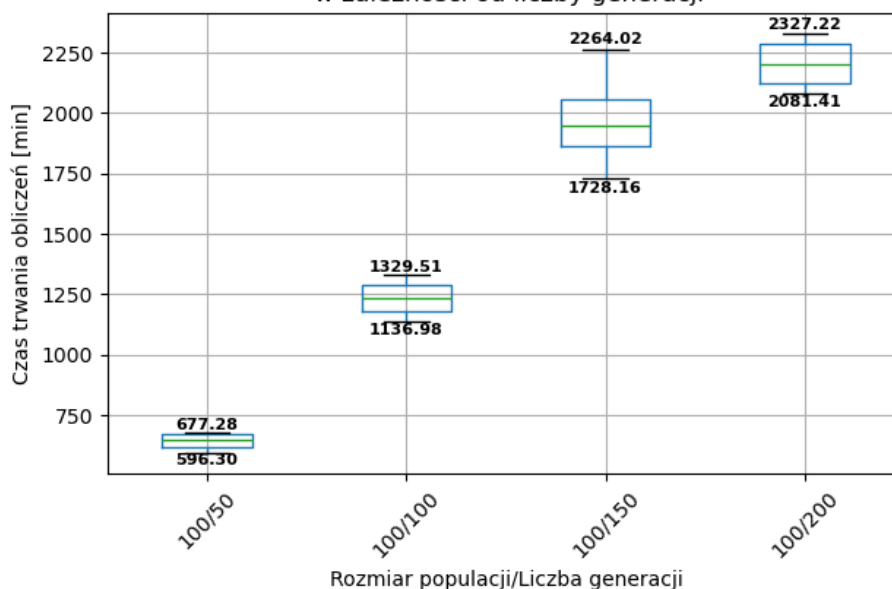
	Liczba generacji			
	50	100	150	200
Minimalny czas trwania obliczeń [min]	269,47	586,89	570,77	987,25
Średni czas trwania obliczeń [min]	304,97	652,04	895,75	1089,56
Maksymalny czas trwania obliczeń [min]	332,06	687,09	1051,13	1182,95

Źródło: Opracowanie własne

Zgodnie z danymi zaprezentowanymi na rys. 53 oraz w tabeli 28, przy stałym rozmiarze populacji równym 50 najkrócej obliczenia były wykonywane dla liczby generacji wynoszącej 50. Dla liczby generacji 150 zaobserwowano szeroki zakres pomiędzy minimalną a maksymalną wartością czasu trwania obliczeń, minimalny czas trwania obliczeń wyniósł 570,77 minuty. Wartość ta była niższa o 16 minut od minimalnego czasu obliczeń dla rozmiaru populacji równego 50 i 100 generacji.

Na rys. 54 i w tabeli 29 zaprezentowano czas trwania obliczeń dla rozmiaru populacji równego 100 w zależności od liczby generacji w serii eksperymentu.

Wykres pudełkowy czasu trwania obliczeń dla rozmiaru populacji równej 100 w zależności od liczby generacji



Rysunek 54. Zależność czasu trwania obliczeń dla populacji równej 100 od liczby generacji

Źródło: Opracowanie własne

Tabela 29. Czasy trwania obliczeń dla populacji równej 100 i różnych liczb generacji

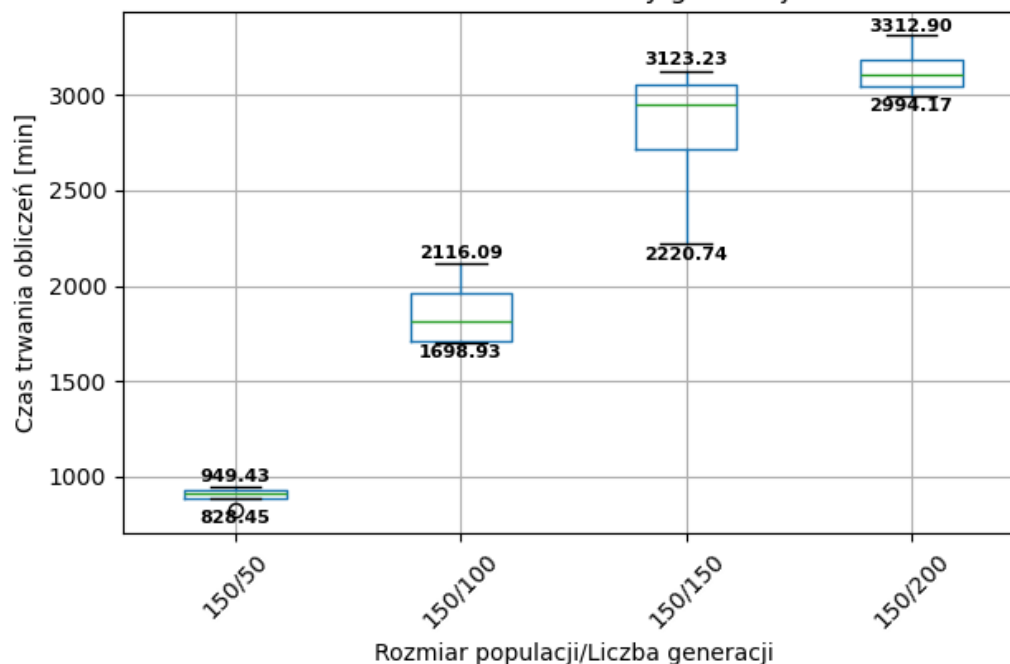
	Liczba generacji			
	50	100	150	200
Minimalny czas trwania obliczeń [min]	596,3	1136,98	1728,16	2081,41
Średni czas trwania obliczeń [min]	641,18	1234,65	1972,21	2204,47
Maksymalny czas trwania obliczeń [min]	677,28	1329,51	2264,03	2327,22

Źródło: Opracowanie własne

Zgodnie z danymi zaprezentowanymi na rys. 54 oraz w tabeli 29, przy stałym rozmiarze populacji równym 100 najkrócej obliczenia były wykonywane dla liczby generacji wynoszącej 50. Wraz ze wzrostem liczby generacji minimalne, średnie i maksymalne czasy trwania obliczeń rosły. Dla liczby generacji 150 zaobserwowano największą różnicę pomiędzy minimalnym a maksymalnym czasem trwania obliczeń. Zaobserwowano również podobne maksymalne czasy trwania obliczeń dla 150 i 200 generacji.

Na rys. 55 i w tabeli 30 zaprezentowano czas trwania obliczeń dla rozmiaru populacji równego 150 w zależności od liczby generacji w serii eksperymentu.

Wykres pudełkowy czasu trwania obliczeń dla rozmiaru populacji równej 150 w zależności od liczby generacji



Rysunek 55. Zależność czasu trwania obliczeń dla populacji równej 150 od liczby generacji

Źródło: Opracowanie własne

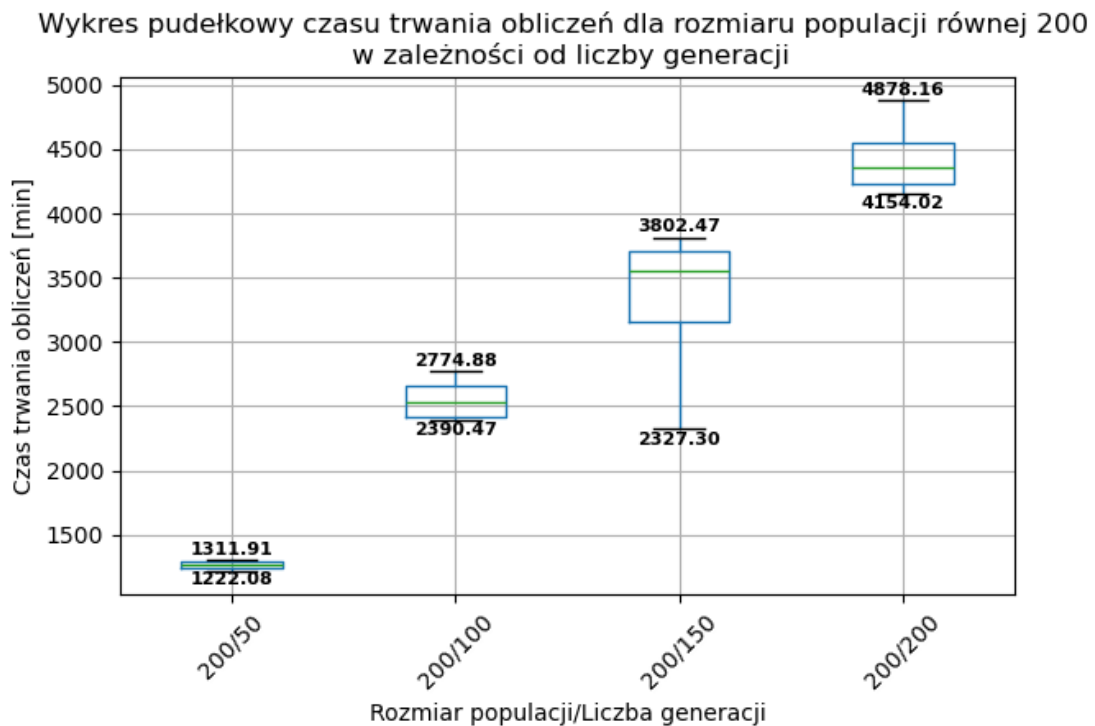
Tabela 30. Czasy trwania obliczeń dla populacji równej 150 i różnych liczb generacji

	Liczba generacji			
	50	100	150	200
Minimalny czas trwania obliczeń [min]	828,45	1698,93	2220,74	2994,17
Średni czas trwania obliczeń [min]	902,38	1859,27	2812,3	3128,57
Maksymalny czas trwania obliczeń [min]	949,43	2116,09	3123,23	3312,90

Źródło: Opracowanie własne

Zgodnie z danymi zaprezentowanymi na rys. 55 oraz w tabeli 30, przy stałym rozmiarze populacji równym 150 najkrócej obliczenia były wykonywane dla liczby generacji wynoszącej 50. Wraz ze wzrostem liczby generacji minimalne, średnie i maksymalne czasy obliczeń rosły. Dla liczby generacji 150 zaobserwowano największą różnicę pomiędzy minimalnym a maksymalnym czasem trwania obliczeń.

Na rys. 56 i w tabeli 31 zaprezentowano czas trwania obliczeń dla rozmiaru populacji równego 200 w zależności od liczby generacji w serii eksperymentu.



Rysunek 56. Zależność czasu trwania obliczeń dla populacji równej 200 od liczby generacji

Źródło: Opracowanie własne

Tabela 31. Czasy trwania obliczeń dla populacji równej 200 i różnych liczb generacji

	Liczba generacji			
	30	100	150	200
Minimalny czas trwania obliczeń [min]	1222,08	2390,47	2327,3	4154,02
Średni czas trwania obliczeń [min]	1266,29	2555,66	3309,72	4433,97
Maksymalny czas trwania obliczeń [min]	1311,91	2774,88	3802,47	4878,16

Źródło: Opracowanie własne

Zgodnie z zaprezentowanymi na rys. 56 oraz w tabeli 31, przy stałym rozmiarze populacji równym 200 najkrócej obliczenia były wykonywane dla liczby generacji wynoszącej 50. Wraz ze wzrostem liczby generacji minimalne, średnie i maksymalne czasy obliczeń rosły. Zaobserwowano również, że minimalny czas trwania obliczeń dla liczby generacji równej 150 był mniejszy od minimalnego czasu trwania obliczeń dla liczby generacji równej 100. Dla liczby generacji 150 zaobserwowano największą różnicę pomiędzy minimalnym a maksymalnym czasem trwania obliczeń.

4.3.5 Podsumowanie analizy wpływu parametrów algorytmu na czas obliczeń

Podsumowując analizę czasu trwania obliczeń w zależności od rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji, należy zauważyć wzrost czasu trwania obliczeń wraz z zwiększeniem rozmiaru populacji i liczby generacji. Czasy trwania obliczeń przy różnych parametrach prawdopodobieństwa mutacji były podobne, więc w praktycznych rozwiązaniach, implementujących proponowane podejście do synchronizacji rozkładów jazdy, należy stosować prawdopodobieństwo mutacji zapewniającą najniższy łączny czas oczekiwania pasażerów na przystankach.

Podsumowanie

Przeprowadzone badania eksperymentalne pozwoliły zrealizować główny cel pracy, jakim było opracowanie i przetestowanie metody synchronizacji rozkładów jazdy w systemie publicznego transportu zbiorowego z wykorzystaniem algorytmu genetycznego. W wyniku przeprowadzonych eksperymentów wykazano, że zastosowana metoda umożliwia skuteczne skrócenie czasu oczekiwania pasażerów na przesiadki, a jednocześnie pozwala na wskazanie optymalnych zakresów parametrów algorytmu sprzyjających równowadze pomiędzy jakością rozwiązań a czasem obliczeń. Osiągnięte rezultaty potwierdzają zasadność wykorzystania algorytmów genetycznych do rozwiązywania problemu synchronizacji rozkładów jazdy, a także stanowią podstawę do dalszych badań nad ich udoskonalaniem i adaptacją w warunkach rzeczywistych systemów transportu publicznego.

Wnioski z pracy

Badania eksperymentalne przeprowadzone w niniejszej pracy udowodniły, że wygenerowanie rozkładu jazdy, dzięki któremu można skrócić czas oczekiwania pasażerów na przesiadkę za pomocą algorytmu genetycznego, jest możliwe i praktycznie osiągalne. Aby zwiększyć szansę znalezienia rozwiązania skracającego czas oczekiwania pasażerów na przesiadkę konieczne było jednak wiele uruchomień algorytmu, co wiązało się z długim czasem trwania obliczeń.

W związku z tym kluczowa jest taka parametryzacja algorytmu, aby zminimalizować czas trwania obliczeń. Na czas trwania obliczeń w niniejszej pracy wpływ miały rozmiar populacji i liczba generacja, nie zaobserwowano natomiast wpływu prawdopodobieństwa mutacji. Dla niskich wartości liczby populacji czas trwania obliczeń może trwać do kilku godzin. Dla dużych populacji składających się z 200 osobników czas ten wyrażany jest w dniach.

Dlatego też istotny jest dobór odpowiednich parametrów, który pozwoli na skrócenie czasu trwania obliczeń przy zachowaniu jakości dostarczanych rozwiązań. Wyniki serii przeprowadzonego eksperymentu wskazują, że zestawy przesunięć czasowych z najniższą wartością funkcji celu, jaką był średni czas oczekiwania pasażera na przesiadkę, można uzyskać przy niskiej liczbie osobników w populacji jaką jest 30 lub 50 osobników oraz liczbie generacji wynoszącej 100 lub 150. Najwięcej wyników z najkrótszym czasem oczekiwania na przesiadkę uzyskano przy najwyższym z badanych prawdopodobieństw wystąpienia mutacji (0,2). Fakt, że wyniki najlepsze pod kątem wartości funkcji celu

osiągnano wraz ze wzrostem prawdopodobieństwa mutacji potwierdzają również dodatkowe badania opisane w rozdziale 4.2.3.

Wyniki przeprowadzonego eksperymentu sugerują, że do rozwiązania problemu synchronizacji rozkładów jazdy w publicznym transporcie zbiorowym właściwym jest użycie algorytmu genetycznego z następującymi parametrami:

- liczebność populacji pomiędzy 30 a 50,
- liczba generacji pomiędzy 100 a 150,
- prawdopodobieństwo mutacji pomiędzy 0,1 a 0,2.

Chociaż mniejsza liczebność populacji skutkuje mniejszą jej różnorodnością to w przeprowadzonym eksperymencie pozwalała na znajdowanie najniższych wartości funkcji celu. Pod względem czasu trwania obliczeń to dłuższe niż to konieczne działanie algorytmu powoduje zwiększone zużycie zasobów oraz nie gwarantuje uzyskania wyraźnie lepszego rozwiązania. Stosunkowo wysokie prawdopodobieństwo wystąpienia mutacji w rozwiązaniach pozwala na efektywne przeszukiwanie dużej przestrzeni rozwiązań oraz na nieutykanie w ekstremach lokalnych.

Kierunki dalszych badań

Opisany w pracy eksperyment nie wyczerpuje całkowicie problematyki synchronizacji rozkładów jazdy w systemach publicznego transportu zbiorowego.

Z pewnością konieczne jest sprawdzenie efektywności algorytmu genetycznego do synchronizacji rozkładów jazdy większej liczby linii. W średniej wielkości miastach liczba linii obsługujących system to kilkadziesiąt, a w największych ich liczba może sięgnąć do kilkuset.

Naturalnym kierunkiem badań wydaje się również synchronizowanie rozkładów jazdy linii, które mają różne interwały czasowe odjazdów.

Do rozważenia wydaje się również sprawdzenie dalszego zwiększania wartości parametru odpowiadającego za prawdopodobieństwo mutacji na dostarczane rozwiązania.

Zgodnie z opisem w rozdziale 1.4.6 parametrów algorytmów genetycznych jest więcej niż analizowano w niniejszej pracy. Właściwym wydaje się zbadanie wpływu na otrzymywane rozwiązania m.in.:

- metod selekcji najlepszych rozwiązań do kolejnych generacji,
- liczby prób wystąpienia mutacji,
- różnych prawdopodobieństw krzyżowania,
- różnych metod krzyżowania.

Podczas synchronizacji rozkładów jazdy w niniejszej pracy nie brano pod uwagę koordynacji rozkładów na wspólnych ciągach komunikacyjnych. Jest to ważny element dobrze zorganizowanej siatki publicznego transportu zbiorowego pozwalający na równomierną obsługę pasażerów. Tak skoordynowany rozkład jazdy jest również łatwiejszy do zapamiętania przez pasażera co podnosi atrakcyjność systemu.

Bibliografia

Abdolmaleki, M., Masoud, N. i Yin, Y., 2020. Transit timetable synchronization for transfer time minimization. *Transportation Research Part B: Methodological*, Tom 131, pp. 143-159.

Aibinu, A. i inni, 2016. A novel Clustering based Genetic Algorithm for route optimization. *Engineering Science and Technology*.

Aksu, D. i Yılmaz, S., 2014. Transit coordination with heterogeneous headways. *Transportation Planning and Technology*, 37(5), pp. 450-465.

Ansari, N. i Hou, E., 1997. *Computational Intelligence for Optimization*. Newark: Kluwer Academic Publishers.

Asadujjaman, A., Rahman, H. F., Chakraborty, R. K. i Ryan, M. J., 2022. Multi-operator immune genetic algorithm for project scheduling with discounted cash flows. *Expert Systems with Applications*, Issue 195.

Ataeian, S., Solimanpur, M., Amiripour, S. i Shankar, R., 2021. Synchronized timetables for bus rapid transit networks in small and large cities. *Scientia Iranica*, 28(1), pp. 477-491.

Bielli, M., Caramia, M. i Carotenuto, P., 2002. Genetic algorithms in bus network optimization. *Transportation Research Part C: Emerging Technologies*, 10(1).

Bruno, G., Improta, G. i Sgalambro, A., 2009. Models for the schedule optimization problem at a public transit terminal.. *OR Spectrum*, Issue 31, pp. 465-481.

Cao, N., Tang, T. i Gao, C., 2020. Multiperiod Transfer Synchronization for Cross-Platform Transfer in an Urban Rail Transit System. *Symmetry*.

Cao, Z., Ceder, A. (., Li, D. i Zhang, S., 2019. Optimal synchronization and coordination of actual passenger-rail timetables. *Journal of Intelligent Transportation Systems*, 23(3), pp. 231-249.

Cats, O., West, J. i Eliasson, J., 2016. A dynamic stochastic model for evaluating congestion and crowding effects in transit systems. *Transportation Research Part B: Methodological*, Issue 89, pp. 43-57.

Ceder, A., 2007. *Public Transit Planning and Operation (1st ed.)*.. Londyn: CRC Press.

Ceder, A., 2016. *Public Transit Planning and Operation: Modeling, Practice and Behavior*, Second Edition (2nd ed.).

Ceder, A., Golany, B. i Tal, O., 2001. Creating bus timetables with maximal synchronization. *Transportation Research Part A: Policy and Practice*, 35(10).

Cevallos, F. i Zhao, F., 2006. A Genetic Algorithm for Bus Schedule Synchronization. *Applications of Advanced Technology in Transportation*, pp. 737-742.

Cevallos, F. i Zhao, F., 2006. Minimizing Transfer Times in Public Transit Network with Genetic Algorithm. *Transportation Research Record*, pp. 74-79.

Chakroborty, P., Kalyanmoy, D. i Subrahmanyam, P., 1995. Optimal Scheduling of Urban Transit Systems Using Genetic Algorithms. *Journal of Transportation Engineering*, 121(6), pp. 544-553.

Chen, Y. i inni, 2019. Timetable synchronization of last trains for urban rail networks with maximum accessibility. *Transportation Research Part C: Emerging Technologies*, Tom 99, pp. 110-129.

Chowdhury, M. S. i Chien, S., 2011. Joint Optimization of Bus Size, Headway, and Slack Time for Efficient Timed Transfer. *Transportation Research Record*, 2218(1), pp. 48-58.

Chung, H. S., Tam, . E. P., Hui, S. i Lo, W. L., 2001. An Optimized Fuzzy Logic Controller for Active Power Factor Corrector Using Genetic Algorithm. W: L. Chambers, red. *The Practical Handbook of Genetic Algorithms Applications*. wydanie drugie red. Boston: CHAPMAN & HALL/CRC.

Corne, D. i Lones, M., 2018. *Evolutionary Algorithms*. W: R. Martí, P. Panos i M. Resende, redaktorzy *Handbook of Heuristics*. Cham: Springer.

Daduna, J. i Voß, S., 1995. *Practical Experiences in Schedule Synchronization*. Berlin, Heidelberg, Springer, pp. 39-55.

Dane Miejskiego Zakładu Komunikacji w Zielonej Górze (2018).

Dane Miejskiego Zakładu Komunikacji w Zielonej Górze (2023).

De Jong, K. A., 1975. *Analysis of the behavior of a class of genetic adaptive systems*, Hampton: University of Michigan.

Dou, X. i Guo, X., 2017. Schedule Coordination Method for Last Train Transfer Problem. *Transportation Research Record*, 2648(1), pp. 86-95.

Dou, X., Meng, Q. i Guo, X., 2015. Bus schedule coordination for the last train service in an intermodal bus-and-train transport network. *Transportation Research Part C: Emerging Technologies*, Tom 60, pp. 360-376.

Eiben, A. i Smith, J., 2015. *Introduction to Evolutionary*. II red. Berlin, Heidelberg: Springer.

Elsayed, S. M., Sarker, R. A. i Essam, D. L., 2011. Multi-operator based evolutionary algorithms for solving constrained optimization problems. *Computers & Operations Research*, 38(12), pp. 1877-1896.

Eranki, A., 2004. *A Model to Create Bus Timetables to Attain Maximum Synchronization Considering Waiting Times at Transfer Stops*. brak miejsca:USF Tampa Graduate Theses and Dissertations.

Fogel, G. B., 2012. Evolutionary Programming. W: G. Rozenberg, T. Bäck i J. N. Kok, redaktorzy *Handbook of Natural Computing*. Berlin: Springer, pp. 700-708.

Fonseca, J. P., van der Hurk, E., Roberti, R. i Larsen, A., 2018. A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. *Transportation Research Part B: Methodologica*, Tom 109, pp. 128-149.

Gkiotsalitis, K. i Maslekar, N., 2018. Towards transfer synchronization of regularity-based bus operations with sequential hill-climbing. *Public Transport*, Tom 10, pp. 335-361.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization & Machine Learning*. Boston: Addison-Wesley Publishing Company, Inc..

Grundler, D., 2001. Multilevel Fuzzy Process Control Optimized by Genetic Algorithm. W: L. Chambers, red. *The Practical Handbook of Genetic Algorithms Applications*. Boston: CHAPMAN & HALL/CRC.

Gschwender, A., Jara-Díaz, S. i Bravo, C., 2016. Feeder-trunk or direct lines? Economies of density, transfer costs and transit structure in an urban context. *Transportation Research Part A: Policy and Practice*, Tom 88, pp. 209-22.

Guihaire, V. i Hao, J., 2008. *Transit Network Re-timetabling and Vehicle Scheduling*. Berlin, Heidelberg, Springer, pp. 134-144.

Guo, X. i inni, 2017. Multiperiod-based timetable optimization for metro transit networks. *Transportation Research Part B: Methodological*, Tom 96, pp. 46-67.

Guo, X. i inni, 2016. Timetable coordination of first trains in urban railway network: A case study of Beijing. *Applied Mathematical Modelling*, 40(17-18), pp. 8048-8066.

Guo, X. i inni, 2020. Scheduling synchronization in urban rail transit networks: Trade-offs between transfer passenger and last train operation. *Transportation Research Part A: Policy and Practice*, Tom 138, pp. 463-490.

GUS, 2025. *Bank Danych Lokalnych*, brak miejsca: brak nazwiska

Hadas, Y. i Ceder, A. (., 2010. Optimal coordination of public-transit vehicles using operational tactics examined by simulation. *Transportation Research Part C: Emerging Technologies*, 18(6), pp. 879-895.

Hancock, P. J. B., 1994. *An empirical comparison of selection methods in evolutionary algorithms*. Berlin, Heidelberg, Springer, pp. 80-94.

Haodong, Y. i inni, 2018. Optimizing last trains timetable in the urban rail network: social welfare and synchronization. *Transportmetrica B: Transport Dynamics*.

Hassold, S. i Ceder, A., 2012. Multiobjective Approach to Creating Bus Timetables with Multiple Vehicle Types. *Transportation Research Record*, pp. 56-62.

Hassold, S. i Ceder, A., 2014. Public transport vehicle scheduling featuring multiple vehicle types. *Transportation Research Part B: Methodological*, Issue 67, pp. 129-143.

Holland, J., 1975. *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.

Ibarra-Rojas, O. J., López-Irarragorri, F. i Rios-Solis, Y. A., 2015. Multiperiod Bus Timetabling. *Transportation Science*, 50(3), pp. 805-822.

Ibarra-Rojas, O. J. i Rios-Solis, Y. A., 2012. Synchronization of bus timetabling. *Transportation Research Part B: Methodological*, 46(5).

Institute for Transportation & Development Policy, 2024. *The BRT Standard*, Nowy Jork: Institute for Transportation & Development Policy.

Jänsch, P. i Reinhold, R., 1990. Hochgeschwindigkeitsverkehr international. *Schienenschnellverkehr*, Issue 4.

Jansen, L. N., Pedersen, M. B. i Nielsen, O. A., 2002. *Minimizing Passenger Transfer Times in Public Transport Timetables*. brak miejsca, brak nazwiska, pp. 229-239.

Kang, L. i inni, 2015. A case study on the coordination of last trains for the Beijing subway network. *Transportation Research Part B: Methodological*, Tom 72, pp. 112-127.

Kang, L. i inni, 2015. A practical model for last train rescheduling with train delay in urban railway transit networks. *Omega*, Tom 50, pp. 29-42.

Kang, L. i inni, 2019. Last train timetabling optimization and bus bridging service management in urban railway transit networks. *Omega*, pp. 31-44.

Kapica, D., 2021. *Ocena wskaźnikowa systemu transportu zbiorowego w Zielonej Górze*. Kraków: Politechnika Krakowska, Praca magisterska pisana pod kierunkiem prof. dr. hab. W. Starowicza.

Kapica, D., 2021. Ocena wskaźnikowa systemu transportu zbiorowego w Zielonej Górze. *Transport miejski i regionalny*, Issue 7-8, pp. 35-37.

Kapica, D., Yulia, M. i Naumov, V., 2025. Synchronization in public transportation: A review of challenges and techniques. *Future Transportation*, 5(1), pp. 1-25.

Ke, Y. i inni, 2020. Improving Synchronization in an Air and High-Speed Rail Integration Service via Adjusting a Rail Timetable: A Real-World Case Study in China. *Journal of Advanced Transportation*.

Klemmt, W. i Stemme, W., 1988. *Schedule Synchronization for Public Transit Networks*. Berlin, Heidelberg, Springer, pp. 327-335.

Kron, L. i inni, 2009. The New Dutch Timetable: The OR Revolution. *Interfaces*, 39(1), pp. 6-17.

Kumar, R., Memoria, M. i Chandel, A., 2020. *Performance Analysis of proposed Mutation Operator of Genetic Algorithm under Scheduling Problem*. Londyn, brak nazwiska, pp. 193-197.

Lee, K. i inni, 2022. Path-oriented synchronized transit scheduling using time-dependent data. *Transportation Research Part C: Emerging Technologies*, Issue 136.

Liebchen, C., 2008. The First Optimized Railway Timetable in Practice. *Transportation Science*, 42(4), pp. 420-435.

Lipowski, A. i Lipowska, D., 2012. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6), pp. 2193-2196.

Liujiang, K. i inni, 2015. A practical model for last train rescheduling with train delay in urban railway transit networks. *Omega*, Issue 50, pp. 29-42.

Liu, T. i Ceder, A., 2017. User and Operator Perspectives in Public Transport Timetable Synchronization Design. *Transportation Research Record*, 2667(1), pp. 154-163.

Liu, T., Ceder, A. i Chowdhury, S., 2017. Integrated public transport timetable synchronization with vehicle scheduling. *Transportmetrica A: Transport Science*, 13(10), pp. 932-954.

Liu, Z., Shen, J., Wang, H. i Yang, W., 2007. Regional Bus Timetabling Model with Synchronization. *Journal of Transportation Systems Engineering and Information Technology*, 7(2), pp. 109-112.

Lu, J. i inni, 2024. The impact of congestion and dedicated lanes on on-demand multimodal transit systems. *Travel Behaviour and Society*.

Madej, Pruciak i Madej, 2017. *Publiczny Transport Miejski. Zasady tworzenia rozkładów jazdy*. II red. Warszawa: Akademia Transportu i Przesiębiorczości.

Mohammed, M. i Jimi, O., 2023. Origin-destination inference in public transportation systems: A comprehensive review. *International Journal of Transportation Science and Technology*, pp. 315-328.

Nachtigall, K. i Voget, S., 1996. A genetic algorithm approach to periodic railway synchronization. *Computers & Operations Research*, 23(5), pp. 453-463.

Naeini, H. M., Shafahi, Y. i SafariTaherkhani, M., 2022. Optimizing and synchronizing timetable in an urban subway network with stop-skip strategy. *Journal of Rail Transport Planning & Management*, Tom 22.

Naumov, V., 2017. Synchronisation of Timetables for Public Bus Lines Using Genetic Algorithms and Computer Simulations. W: *Reliability and Statistics in Transportation and Communication*. Cham: Springer, pp. 44-53.

Naumov, V., 2019. Modeling demand for passenger transfers in the bounds of public transport network. W: E. G. Nathanail i I. D. Karakikes, redaktorzy *Data Analytics: Paving the Way to Sustainable Urban Mobility*. Cham: Springer, pp. 156-163.

Naumov, V., 2020. Genetic-based algorithm of the public transport lines synchronization in a transfer node. *Transportation Research Procedia*, Tom 47, pp. 315-322.

Naumov, V., 2021. [Online] Available at: <https://github.com/naumovvs/publictransportnet> [Data uzyskania dostępu: 27 11 2023].

Naumov, V. i Samchuk, G., 2017. Class Library for Simulations of Passenger Transfer Nodes as Elements of the Public Transport System. *Procedia Engineering*, Issue 187, pp. 77-81.

Nesheli, M. M. i Ceder, A., 2014. Optimal combinations of selected tactics for public-transport transfer synchronization. *Transportation Research Part C: Emerging Technologies*, Issue 48.

Ngamchai, S. i Lovell, D. J., 2003. Optimal Time Transfer in Bus Transit Route Network Design Using a Genetic Algorithm. *Journal of Transportation Engineering*, 129(5), pp. 510-521.

Niu, H. i Zhou, X., 2013. Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transportation Research Part C: Emerging Technologies*, Tom 36, pp. 212-230.

Parbo, J., Nielsen, O. A. i Prato, C. G., 2014. User perspectives in public transport timetable optimisation. *Transportation Research Part C: Emerging Technologies*, Tom 48, pp. 269-284.

Poorjafari, V., Yue, W. L. i Holyoak, N., 2014. Application of Simulated Annealing in Transit Schedule Synchronization. *International Journal of Modeling and Optimization*, 4(6), pp. 476-481.

PTV, 2025. *PTV Visum 2025 Manual*. Karlsruhe: PTV Planung Transport Verkehr GmbH.

Rozporządzenie (WE) nr 1370/2007 Parlamentu Europejskiego i Rady z dnia 23 października 2007 r. dotyczące usług publicznych w zakresie kolejowego i drogowego transportu pasażerskiego (2007).

Rydzikowski, W. i Wojewódzka-Król, K., 2010. *Transport*. Warszawa: Wydawnictwo Naukowe PWN.

Schröder, M. i Solchenbach, I., 2006. *Optimization of Transfer Quality in Regional Public Transit*, brak miejsca: Fraunhofer-Institut für Techno- und Wirtschaftsmathematik.

Shafahi, Y. i Khani, A., 2010. A practical model for transfer optimization in a transit network: Model formulations and solutions. *Transportation Research Part A: Policy and Practice*, 44(6), pp. 377-389.

Shang, H.-Y., Huang, H.-J. i Wu, W.-X., 2019. Bus timetabling considering passenger satisfaction: An empirical study in Beijing. *Computers & Industrial Engineering*, Issue 135, pp. 1155-1166.

Shang, H. i Liu, Y., 2019. *An Extended Vehicle Scheduling Model Considering Interests of both Passengers and Bus Operators*. Guangzhou, brak nazwiska, pp. 1995-2000.

Shang, H., Liu, Y., Huang, H. i Guo, R., 2019. Vehicle Scheduling Optimization considering the Passenger Waiting Cost. *Journal of Advanced Transportation*, 2019(1).

Shang, P. i inni, 2018. Timetable Synchronization and Optimization Considering Time-Dependent Passenger Demand in an Urban Subway Network. *Transportation Research Record*, pp. 243-254.

Sharapov, R., 2007. *Genetic Algorithms: Basic Ideas, Variants and*. W: G. Obinata i A. Dutta, redaktorzy *Vision Systems: Segmentation and Pattern Recognition*. Rijeka: I-Tech Education and Publishing, pp. 407-422.

Shen, Y. i Wang, S., 2015. *Maximal Synchronization of Feeder Buses to Metro Using Particle Swarm Optimization*. Berlin, Heidelberg, Springer.

Shrivastava, P. i Dhingra, S., 2002. Development of Coordinated Schedules using Genetic Algorithms. *Journal of Transportation Engineering*, 128(1), pp. 89-96.

Shrivastava, P., Dhingra, S. i Gundaliya, P., 2002. Application Of Genetic Algorithm For Scheduling And Schedule Coordination Problems. *Journal of Advanced Transportation*, Issue 36, pp. 26-41.

Shrivastava, P. i O'Mahony, M., 2006. A Model for Development of Optimized Feeder Routes and Coordinated Schedules—A Genetic Algorithms Approach. *Transport Policy*, Tom 13, pp. 413-425.

Shukla, A., Pandey, H. M. i Mehrotra, D., 2015. *Comparative review of selection techniques in genetic algorithm*. Greater Noida, IEEE.

Sivanandam, S. i Deepa, S., 2007. *Introduction to Genetic Algorithms*. Heidelberg: Springer Berlin.

Takamatsu, M. i Taguchi, A., 2020. Bus Timetable Design to Ensure Smooth Transfers in Areas with Low-Frequency Public Transportation Services. *Transportation Science*, 54(5), pp. 1238-1250.

Teodorović, D. i Lučić, P., 2005. Schedule synchronization in public transit using the fuzzy ant system. *Transportation Planning and Technology*, 28(1), pp. 47-76.

Tzieropoulos, P., Daniel, E. i Buri, J.-D., 2010. *Regular-interval timetables: Theoretical foundations and policy implications*. brak miejsca, brak nazwiska

Ustawa z dnia 16 grudnia 2010 r. o publicznym transporcie zbiorowym (2010).

Ustawa z dnia 16.04.2004 o czasie pracy kierowców (2004).

Vansteenwegen, P. i inni, 2022. A survey on demand-responsive public bus systems. *Transportation Research Part C: Emerging Technologies*, Issue 137.

Vansteenwegen, P. i Van Oudheusden, D., 2007. Decreasing the passenger waiting time for an intercity rail network. *TRANSPORTATION RESEARCH PART B-METHODOLOGICAL*, 41(4), pp. 478-492.

Vuchic, V., 2005. *Urban Transit: Operations, Planning, and Economics*. Hoboken: Wiley.

Wang, H. i Shen, J., 2007. Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Applied Mathematics and Computation*, 190(2), pp. 1237-1249.

Wang, Y., Li, D. i Cao, Z., 2020. Integrated timetable synchronization optimization with capacity constraint under time-dependent demand for a rail transit network. *Computers & Industrial Engineering*, Tom 142.

Wang, Y. i inni, 2019. Scheduling overnight trains for improving both last and first train services in an urban subway network. *Advances in Mechanical Engineering*, 11(5).

Wang, Y., Zhou, Y. i Yan, X., 2022. Autonomous bus timetable synchronization for maximizing smooth transfers with passenger assignment. *Expert Systems With Applications*.

Wardman, M., 2004. Public transport values of time. *Transport Policy*, 11(4), pp. 363-377.

Wen-Yang, L., Wen-Yuan, L. i Tzung-Pei, H., 2003. Adapting Crossover and Mutation Rates in Genetic Algorithms. *Journal of Information Science and Engineering*.

Wong, R., Yuen, T., Fung, K. i Leung, Y., 2008. Optimizing Timetable Synchronization for Rail Mass Transit. *Transportation Science*, pp. 57-69.

Wu, J. i inni, 2015. Equity-based timetable synchronization optimization in urban subway network. *Transportation Research Part C: Emerging Technologies*, Tom 51, pp. 1-18.

Wu, W., Liu, R. i Jin, W., 2016. Designing robust schedule coordination scheme for transit networks with safety control margins. *Transportation Research Part B: Methodological*, 93(A), pp. 495-519.

Wu, Y., Tang, J., Yu, Y. i Pan, Z., 2015. A stochastic optimization model for transit network timetable design to mitigate the randomness of traveling time by adding slack time. *Transportation Research Part C: Emerging Technologies*, Tom 52, pp. 15-31.

Wu, Y., Yang, H., Tang, J. i Yu, Y., 2016. Multi-objective re-synchronizing of bus timetable: Model, complexity and solution. *Transportation Research Part C: Emerging Technologies*, Tom 67, pp. 149-168.

Xin, G. i inni, 2016. Timetable coordination of first trains in urban railway network: A case study of Beijing. *Applied Mathematical Modelling*, 40(17-18), pp. 8048-8066.

Xin-She, Y., 2014. *Nature-Inspired Optimization Algorithms*. Londyn, Waltham: Elsevier.

Yap, M. i inni, 2019. Where shall we sync? Clustering passenger flows to identify urban public transport hubs and their key synchronization priorities. *Transportation Research Part C: Emerging Technologies*, Tom 98.

Yin, H. i inni, 2018. Optimizing last trains timetable in the urban rail network: social welfare and synchronization. *Transportmetrica B: Transport Dynamics*, 7(1), pp. 473-497.

Yin, H. i inni, 2019. Optimizing last trains timetable in the urban rail network: social welfare and synchronization. *Transportmetrica B: Transport Dynamics*, 7(1), pp. 473-497.

Yu, B., Yang, Z. i Yao, J., 2010. Genetic Algorithm for Bus Frequency Optimization. *Journal of Transportation Engineering*, 136(6), pp. 576-583.

Zhang, L. i inni, 2022. Evaluation of spatial fairness of urban public transport service: a case of Wuhan City. *The International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, pp. 579-585.

Zhao, F. i Zeng, X., 2008. Optimization of transit route network, vehicle headways and timetables for large-scale transit networks. *European Journal of Operational Research*, 186(2), pp. 841-855.

Zhao, L., 2006. A heuristic method for analyzing driver scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 36(3), pp. 521-531.

Żak, J., 2011. Wybór taboru dla systemu publicznego transportu miejskiego z wykorzystaniem metodyk: wielokryterialnego i grupowego podejmowania decyzji. *Logistyka*, Issue 6.

Spis tabel

Tabela 1. Liczba kursów rozpatrywanych oddzielnie dla każdej linii i wspólnie w ciągu komunikacyjnym	17
Tabela 2. Fragment przykładowego, skoordynowanego rozkładu jazdy na przystanku X ..	18
Tabela 3. Obowiązkowe przerwy dla prowadzących w publicznym transporcie zbiorowym	21
Tabela 4. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą algorytmu genetycznego w dokonanym przeglądzie literatury	37
Tabela 5. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą programowania całkowitoliczbowego i jego wariantów w dokonanym przeglądzie literatury	42
Tabela 6. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą symulowanego wyżarzania w dokonanym przeglądzie literatury	46
Tabela 7. Zestawienie prac nad synchronizacją rozkładów jazdy za pomocą innych metod w dokonanym przeglądzie literatury	47
Tabela 8. Metody oraz liczba prac, w których wykorzystano je do synchronizacji rozkładów jazdy.....	50
Tabela 9. Typy synchronizowanych sieci w dokonanym przeglądzie literatury.....	52
Tabela 10. Liczba prac w zależności od metody, części oraz typu synchronizowanej sieci w dokonanym przeglądzie literatury	53
Tabela 11. Liczba pasażerów korzystających z poszczególnych linii w dzień powszedni w systemie publicznego transportu zbiorowego w Zielonej Górze.....	80
Tabela 12. Liczba pasażerów korzystających z wybranych sześciu linii w dzień powszedni w systemie publicznego transportu zbiorowego w Zielonej Górze.....	81
Tabela 13. Lista przystanków, na których możliwe są przesiadki	83
Tabela 14. Fragment przykładowego rozkładu jazdy zgodnego z chromosomem przedstawionym na rysunku 14.	85
Tabela 15. Zapis binarny i dziesiętny dwóch rozwiązań.....	86
Tabela 16. Fragmenty dwóch rozkładów jazdy.....	86
Tabela 17. Zapis binarny i dziesiętny dwóch rozwiązań po operacji krzyżowania	87
Tabela 18. Fragmenty dwóch rozkładów jazdy po operacji krzyżowania.....	87
Tabela 19. Zapis binarny i dziesiętny rozwiązania przed i po wystąpieniu mutacji	88
Tabela 20. Fragmenty rozkładów jazdy przed i po wystąpieniu mutacji	88
Tabela 21. Parametry algorytmu genetycznego użyte w eksperymencie	89

Tabela 22. Wyniki analizy statystycznej zbioru rezultatów.....	92
Tabela 23. Średnia wartość funkcji celu dla poszczególnych zestawów parametrów	116
Tabela 24. Czasy trwania obliczeń dla różnych rozmiarów populacji.....	118
Tabela 25. Czasy trwania obliczeń dla różnych liczb generacji	119
Tabela 26. Czasy trwania obliczeń dla różnych prawdopodobieństw mutacji	120
Tabela 27. Czasy trwania obliczeń dla populacji równej 30 i różnych liczb generacji	122
Tabela 28. Czasy trwania obliczeń dla populacji równej 50 i różnych liczb generacji	123
Tabela 29. Czasy trwania obliczeń dla populacji równej 100 i różnych liczb generacji ...	124
Tabela 30. Czasy trwania obliczeń dla populacji równej 150 i różnych liczb generacji ...	125
Tabela 31. Czasy trwania obliczeń dla populacji równej 200 i różnych liczb generacji ...	126

Spis rysunków

Rysunek 1. Uproszczony schemat planowania oferty przewozowej.....	15
Rysunek 2. Przykład wykresu ruchu dwóch linii obsługiwanych przez dwa pojazdy	19
Rysunek 3. Przykład wykresu ruchu dwóch linii obsługiwanych przez jeden pojazd	20
Rysunek 4. Sprzężenia zwrotne w planowaniu oferty publicznego transportu zbiorowego	22
Rysunek 5. Uproszczony schemat synchronizacji rozkładu jazdy na jednym przystanku...	24
Rysunek 6. Synchronizacja rozkładów jazdy dwóch linii na jednym przystanku w godzinach szczytu porannego.	24
Rysunek 7. Synchronizacja rozkładów jazdy dwóch linii na jednym przystanku w godzinach szczytu popołudniowego.	24
Rysunek 8. Uproszczony schemat synchronizacji kilku linii autobusowych w sieci publicznego transportu zbiorowego.....	25
Rysunek 9. Fragment schematu sieci pekińskiego metra	27
Rysunek 10. Schemat przedstawiający działanie algorytmu genetycznego	31
Rysunek 11. Metody krzyżowania	32
Rysunek 12. Sposoby mutacji osobników	33
Rysunek 13. Procentowy udział metod prac dotyczących synchronizacji rozkładów jazdy w dokonanym przeglądzie literatury	50
Rysunek 14. Procentowy udział typów synchronizowanych sieci w dokonanym przeglądzie literatury.....	52
Rysunek 15. Atrybuty i metody w pakiecie i klasie <i>Stochastic</i>	66
Rysunek 16. Atrybuty i metody w klasach należących do pakietu <i>transportnet</i> oraz relacje pomiędzy nimi	69
Rysunek 17. Atrybuty i metody w klasach należących do pakietu <i>Genetics</i> oraz relacje pomiędzy nimi	71
Rysunek 18. Relacje i zależności pomiędzy pakietami	73
Rysunek 19. Granice Zielonej Góry oraz ważniejsze szlaki drogowe i kolejowe	76
Rysunek 20. Ważniejsze generatory ruchu w mieście	77
Rysunek 21. Plan sieci zielonogórskiego systemu transportu zbiorowego w 2019 roku.....	79
Rysunek 22. Uproszczony schemat tras sześciu wytypowanych linii.....	82
Rysunek 23. Przykładowy chromosom	85
Rysunek 24. Charakterystyka zestawu przesunięć czasowych pod względem częstości występowania	91

Rysunek 25. Udział procentowy rozmiarów populacji w analizowanej puli rozwiązań	93
Rysunek 26. Udział procentowy liczby generacji w analizowanej puli rozwiązań	93
Rysunek 27. Udział procentowy prawdopodobieństwa mutacji w analizowanej puli rozwiązań	94
Rysunek 28. Zbiorcze zestawienie udziałów procentowych poszczególnych parametrów w analizowanej puli rozwiązań	95
Rysunek 29. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i liczby generacji w analizowanej puli rozwiązań.....	96
Rysunek 30. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań	97
Rysunek 31. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań	98
Rysunek 32. Wykres 3D przedstawiający liczbę rozwiązań dla różnych wartości rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań	99
Rysunek 33. Histogram rozkładu średnich czasów oczekiwania na przesiadkę dla rzeczywistego popytu i rozkładu jazdy	101
Rysunek 34. Histogram rozkładu średnich czasów oczekiwania na przesiadkę dla zestawów przesunięć czasowych zaproponowanych przez algorytm genetyczny.....	101
Rysunek 35. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [10, 2, 9, 1, 1, 4]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy	103
Rysunek 36. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [9, 1, 7, 1, 1, 0]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy	104
Rysunek 37. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [11, 2, 9, 0, 1, 4]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy	105
Rysunek 38. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [8, 1, 4, 6, 0, 1]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy	106
Rysunek 39. Porównanie histogramu rozkładu średnich czasów oczekiwania dla zestawu przesunięć $\Psi = [7, 2, 6, 0, 2, 3]$ z histogramem rozkładu średnich czasów oczekiwania dla rzeczywistego rozkładu jazdy	107
Rysunek 40. Udział procentowy rozmiaru populacji w analizowanej puli rozwiązań	108

Rysunek 41. Udział procentowy liczby generacji w analizowanej puli rozwiązań.....	109
Rysunek 42. Udział procentowy prawdopodobieństwa mutacji w analizowanej puli rozwiązań.....	109
Rysunek 43. Zbiorcze zestawienie udziałów procentowych poszczególnych parametrów w analizowanej puli rozwiązań	111
Rysunek 44. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i liczby generacji w analizowanej puli rozwiązań	112
Rysunek 45. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości rozmiaru populacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań	113
Rysunek 46. Mapa cieplna przedstawiająca liczbę rozwiązań dla różnych wartości liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań.....	114
Rysunek 47. Wykres 3D przedstawiający liczbę rozwiązań dla różnych wartości rozmiaru populacji, liczby generacji i prawdopodobieństwa mutacji w analizowanej puli rozwiązań	115
Rysunek 48. Średnia wartość funkcji celu w zależności od parametru prawdopodobieństwo mutacji	117
Rysunek 49. Zależność czasu trwania obliczeń od rozmiaru populacji	118
Rysunek 50. Zależność czasu trwania obliczeń od liczby generacji	119
Rysunek 51. Zależność czasu trwania obliczeń od prawdopodobieństwa mutacji	120
Rysunek 52. Zależność czasu trwania obliczeń dla populacji równej 30 od liczby generacji	121
Rysunek 53. Zależność czasu trwania obliczeń dla populacji równej 50 od liczby generacji	122
Rysunek 54. Zależność czasu trwania obliczeń dla populacji równej 100 od liczby generacji	123
Rysunek 55. Zależność czasu trwania obliczeń dla populacji równej 150 od liczby generacji	124
Rysunek 56. Zależność czasu trwania obliczeń dla populacji równej 200 od liczby generacji	125