

POLITECHNIKA BIAŁOSTOCKA

WYDZIAŁ INFORMATYKI

PRACA DYPLOMOWA INŻYNIERSKA

TEMAT: TEMAT PRACY INŻYNIERSKIEJ /  
MAGISTERSKIEJ.

WYKONAWCA: GAL ANONIM

.....  
podpis

PROMOTOR: DR INŻ. DOKTOR INŻYNIER

.....  
podpis

BIAŁYSTOK 2021 r.



**Karta dyplomowa**

POLITECHNIKA BIAŁOSTOCKA  Wydział.....  .....	Studia..... stacjonarne/niestacjonarne	Nr albumu studenta.....
	..... studia I stopnia/studia II stopnia	Rok akademicki.....
		Kierunek studiów..... ..... Specjalność..... .....
<p>.....</p> <p><b>Imiona i nazwisko studenta</b></p> <p><b>TEMAT PRACY DYPLOMOWEJ:</b></p> <p>.....</p> <p>.....</p> <p>.....</p> <p><b>Zakres pracy:</b></p> <p>1. ....</p> <p>2. ....</p> <p>3. ....</p> <p>4. ....</p> <p><b>Słowa kluczowe (max 5):</b> .....</p> <p><b>TO JEST SKAN</b></p> <p>.....</p> <p><i>Imiona i nazwisko, stopień/ tytuł promotora - podpis</i></p>		
<p>.....</p> <p><i>Data wydania tematu pracy dyplomowej      Regulaminowy termin złożenia pracy dyplomowej      Data złożenia pracy dyplomowej</i></p> <p><i>- podpis promotora      - potwierdzenie dziekanatu</i></p>		
<p>.....</p> <p><i>Ocena promotora      Podpis promotora</i></p>		
<p>.....</p> <p><i>Imiona i nazwisko, stopień/ tytuł recenzenta      Ocena recenzenta      Podpis recenzenta</i></p>		



Subject of diploma thesis

Temat po angielsku.

## **Summary**

Streszczenie pracy po angielsku.



**Gal Anonim**

Imiona i nazwisko studenta

**12345**

Nr albumu

**informatyka, stacjonarne**

Kierunek i forma studiów

**dr inż. Doktor Inżynier**

Promotor pracy dyplomowej

**OŚWIADCZENIE**

Przedkładając w roku akademickim 2019/2020 Promotorowi **dr inż. Doktor Inżynier** pracę dyplomową pt.: **Temat pracy**, dalej zwaną pracą dyplomową, **oświadczam, że:**

- 1) praca dyplomowa stanowi wynik samodzielnej pracy twórczej;
- 2) wykorzystując w pracy dyplomowej materiały źródłowe, w tym w szczególności: monografie, artykuły naukowe, zestawienia zawierające wyniki badań (opublikowane, jak i nieopublikowane), materiały ze stron internetowych, w przypisach wskazywałem/am ich autora, tytuł, miejsce i rok publikacji oraz stronę, z której pochodzą powoływane fragmenty, ponadto w pracy dyplomowej zamieściłem/am bibliografię;
- 3) praca dyplomowa nie zawiera żadnych danych, informacji i materiałów, których publikacja nie jest prawnie dozwolona;
- 4) praca dyplomowa dotychczas nie stanowiła podstawy nadania tytułu zawodowego, stopnia naukowego, tytułu naukowego oraz uzyskania innych kwalifikacji;
- 5) treść pracy dyplomowej przekazanej do dziekanatu Wydziału Informatyki jest jednakowa w wersji drukowanej oraz w formie elektronicznej;
- 6) jestem świadomy/a, że naruszenie praw autorskich podlega odpowiedzialności na podstawie przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. z 2019 r. poz. 1231, późn. zm.), jednocześnie na podstawie przepisów ustawy z dnia 20 lipca 2018 roku Prawo o szkolnictwie wyższym i nauce (Dz. U. poz. 1668, z późn. zm.) stanowi przesłankę wszczęcia postępowania dyscyplinarnego oraz stwierdzenia nieważności postępowania w sprawie nadania tytułu zawodowego;
- 7) udzielam Politechnice Białostockiej nieodpłatnej, nieograniczonej terytorialnie i czasowo licencji wyłącznej na umieszczenie i przechowywanie elektronicznej wersji pracy dyplomowej w zbiorach systemu Archiwum Prac Dyplomowych Politechniki Białostockiej oraz jej zwielokrotniania i udostępniania w formie elektronicznej w zakresie koniecznym do weryfikacji autorstwa tej pracy i ochrony przed przywłaszczeniem jej autorstwa.

.....

czytelny podpis studenta





# Spis treści

<b>Streszczenie</b>	<b>5</b>
<b>Wstęp</b>	<b>11</b>
<b>1 Ogólny problem</b>	<b>13</b>
1.1 Algorytm genetyczny [Paweł] . . . . .	15
1.2 Algorytm mrówkowy . . . . .	19
1.3 Algorytm zachłanne . . . . .	22
<b>2 Algorytm genetyczny</b>	<b>25</b>
2.1 Chromosom . . . . .	25
2.2 Selekcja . . . . .	25
<b>3 Algorytm mrówkowy</b>	<b>27</b>
3.1 Wprowadzenie do algorytmu . . . . .	27
3.2 Opis działania algorytmu . . . . .	28
3.3 Pozostałe zastosowania algorytmu mrówkowego . . . . .	29
<b>4 Rozdział Przemokowy</b>	<b>33</b>
<b>5 Badania</b>	<b>35</b>
5.1 Wyniki algorytmu genetycznego . . . . .	35
5.2 Wyniki algorytmu mrówkowego . . . . .	35
5.3 Wyniki algorytmu mrówkowego . . . . .	35
<b>6 Porównania wyników</b>	<b>37</b>
<b>Podsumowanie</b>	<b>39</b>
<b>Bibliografia</b>	<b>41</b>
<b>Spis tabel</b>	<b>43</b>
<b>Spis rysunków</b>	<b>45</b>

<b>Spis listingów</b>	<b>47</b>
<b>Spis algorytmów</b>	<b>49</b>

## **Wstęp**

wstęp, wejście do problemu, zakres smieci, cel pracy i zakres, jak jest zorganizowana



# 1. Ogólny problem

Problem komiwojażera (ang. travelling salesman problem - TSP) należy do rodziny problemów NP-trudnych. Znalezienie najlepszego rozwiązania dla tego problemu jest trudne i fascynuje naukowców od wielu lat. Niektórzy poddają pod wątpliwość znalezienie efektywnego rozwiązania czyli takiego którego czas działania jest maksymalnie wielomianowy. Aktualnie istnieje wiele rozwiązań tego problemu, a proponowane podejścia są bardzo interesujące. Niektóre z nich bazują na lokalnych przeszukiwaniach grafu, a inne opierają się na rozwoju.

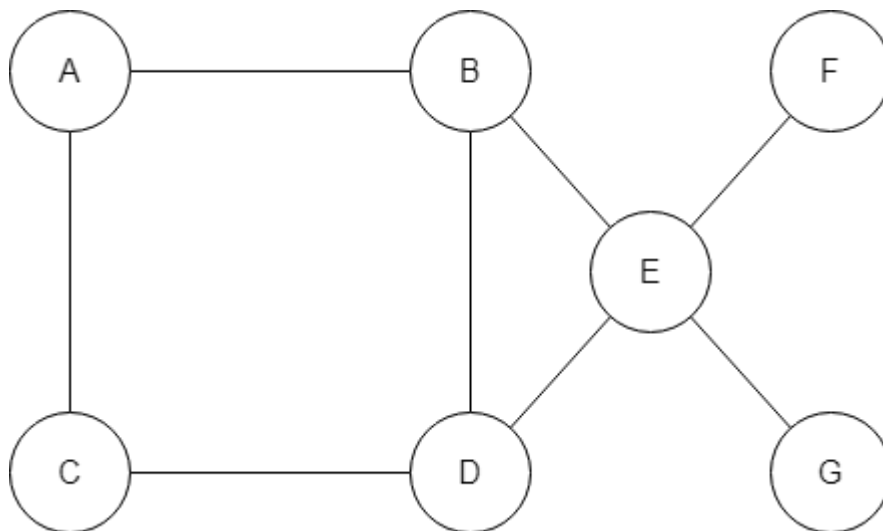
Podobnym problemem do TSP jest problem konika szachowego. Problem ten jest problemem NP-zupełnym. Już w XVIII wieku badania nad tym problemem rozpoczął Euler. Rozwiązanie tego problemu polega na znalezieniu ścieżki jaką ma przebyć konik szachowy, tak aby odwiedzić każde pole na szachownicy tylko i wyłącznie raz. Skoczek porusza się po planszy zgodnie z określonym ruchem, a plansza szachowa może mieć różny rozmiar. Konik porusza się aż do momentu odwiedzenia wszystkich pól lub do momentu w którym nie ma możliwości odwiedzenia kolejnego pola.

Optymalizacja tras od zawsze jest obecna w historii ludzkości. Nawet takie trywialne problemy jak podróż pomiędzy 3 miejscowościami może zostać sklasyfikowany jako problem komiwojażera. Chociaż dokładne wskazanie na źródło problemu TSP nie jest znane, to już w 1832 roku w przewodniku dla podróżujących po Niemczech i Szwajcarii została zawarta informacja o optymalizacji trasy przejazdu. Nie ma tam zawartych żadnych teorii matematycznych w związku z czym nie można uznać tego dzieła za początek rozważań nad problemem komiwojażera.

W XIX wieku William Hamilton stworzył fundamenty pod definicję TSP. W rozwiązaniu problemu komiwojażera należy znaleźć cykl w grafie. W skład takiego cyklu musi zostać zawarty każdy z wierzchołków. Każdy z wierzchołków może znajdować się w rozwiązaniu dokładnie tylko raz. Cykl który spełnia wymieniony warunek jest cyklem Hamiltona. Jeśli w grafie można wyróżnić cykl z opisanymi powyżej warunkami, to graf jest grafem Hamiltonowskim.

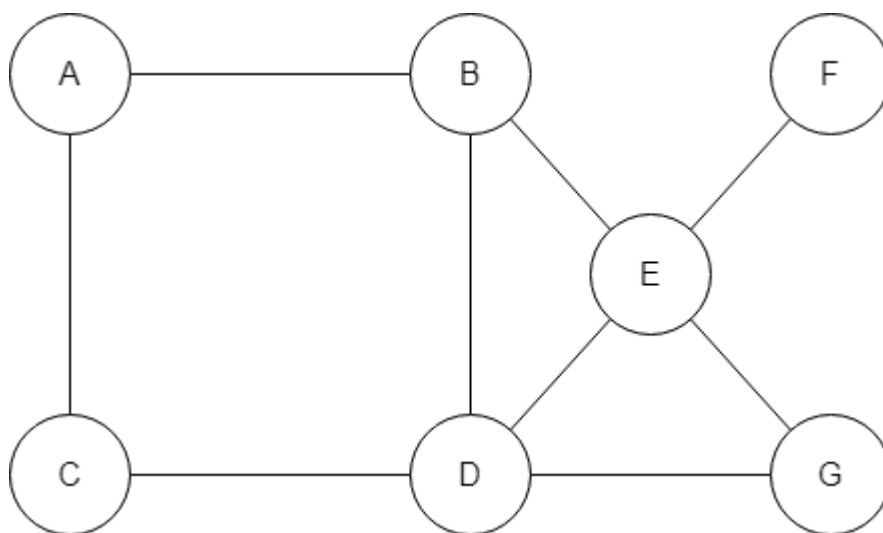
Na 1.1 został przedstawiony graf bez cyklu Hamiltona. W grafie tym nie można znaleźć takiego połączenia które zawiera wszystkie wierzchołki przechodząc przez każdy

z nich dokładnie raz. Istnieje możliwość przejścia przez wszystkie wierzchołki jedynie po powtórnym odwiedzeniu przynajmniej jednego wierzchołka.



Rysunek 1.1: Graf bez cyklu Hamiltona.

Graf z 1.2 posiada połączenie krawędzi dzięki któremu można przejść po wszystkich wierzchołkach dokładnie raz. Takie przejście jest właśnie cyklem Hamiltona w związku z czym graf jest Hamiltonowski. Wyruszając przykładowo z punktu F możemy przejść kolejno do E - G - D - B - A - C. W ten sposób odwiedzimy wszystkie wierzchołki tylko raz.



Rysunek 1.2: Graf z cyklem Hamiltona.

W latach 30 XX wieku Merrill Meeks Flood rozpoczął rozważania nad optymalizacją przejazdu autobusów szkolnych. Działalność tą możemy uznać za początek pracy nad problemem TSP. Wraz z upływem czasu zainteresowanie problemem optymalizacyjnym narastało a co za tym idzie powstawały nowe pomysły na algorytmy. Jednak żaden z

pomysłów nie jest w stanie zaproponować dokładnego rozwiązania które jest w stanie przedstawić rezultat w czasie wyrażonym za pomocą wielomianu.

Jednym z proponowanych rozwiązań jest algorytm Helda Karpa który jest oparty na programowaniu dynamicznym. Złożoność pamięciowa tego algorytmu wynosi  $O(n \text{ razy } 2 \text{ do } n)$ , a czasowa  $O(n \text{ do } 2 \text{ razy } 2 \text{ do } n)$ . W algorytmie tym na każdym kroku wyznaczamy punkt który powinien być przedostatni na trasie. Aby wyznaczyć poprzednika należy skorzystać ze wzoru w którym poszukiwana jest najmniejsza wartość pomiędzy punktami.

Innym przykładem algorytmu który można wykorzystać do rozwiązania problemu komiwojażera jest algorytm najbliższego sąsiada. Rozwiązanie to wykorzystuje strategię zachłanną. W algorytmie szukamy aktualnie najlepszego ruchu. W tym celu przeszukiwani są jedynie sąsiedzi którzy są najbliżej aktualnego punktu. Złożoność takiego algorytmu jest szacowana na  $O(n \text{ do } 2)$ .

Oprócz standardowych przeszukiwań zbiorów na przestrzeni lat pojawiły się propozycje które wprowadzają elementy losowości. Przykładem takiego rozwiązania mogą być algorytm genetyczny oraz algorytm mrówkowy. Powstały również rozwiązania wykorzystujące bla bla bla a jako przykład mogą posłużyć bla bla bla

## 1.1 Algorytm genetyczny [Paweł]

Kolejnym podejściem do rozwiązania problemu komiwojażera jest algorytm genetyczny(z ang. Genetic Algorithm - GA), czyli algorytm, który bazuje na ewolucji. Jest on oparty na zjawiskach zachodzących przyrodzie jak dziedziczenie cech oraz dobór naturalny[4]. Dąży on do tego, aby początkowe pokolenia w raz z kolejnymi iteracjami ewoluowały w coraz to lepsze rozwiązania. Najważniejszą cechą jaką odwzorowują algorytmy genetyczne z przyrody, jest przetrwanie najlepszych osobników. W przyrodzie często najsłabsze osobniki w stadach nie biorą udziału w reprodukcji i giną. Podobnie w algorytmie genetycznym populacja poddawana jest operatorom genetycznym: selekcja najlepszych osobników, krzyżowanie oraz mutacja. Zastosowanie tych trzech operatorów prowadzi do powstawania w każdym kolejnym pokoleniu lepiej przystosowanych osobników, czyli lepszych rozwiązań problemu.

Algorytmy genetyczne są od dawna stosowane informatyce do rozwiązywania problemów komiwojażera oraz innych NP trudnych zagadnień. Pionierem algorytmów genetycznych był John Henry Holland[7], który w latach 70 napisał książkę o algorytmach

ewolucyjnych "Adaptation in Natural and Artificial Systems". Mają zastosowanie w takich dziedzinach jak np: optymalizacje funkcji, minimalizacja kosztów, przemysł lotniczy, projektowanie sieci przemysłowych itp. [7]

### 1.1.1 Opis działania algorytmu

Przed przejściem do omawiania algorytmu, należy wyjaśnić podstawowe pojęcia, które występują w algorytmie genetycznym:

- **OSOBNIK** pojedyncze rozwiązanie problemu, zakodowane w postaci chromosomu.
- **POPULACJA** zbiór osobników o stałej liczbie  $N$  w przekroju trwania całego algorytmu.
- **GEN** przechowuje informację o dowolnej cenie osobnika. W zależności od sposobu kodowania może to być bit, dowolna cyfra, znak itp.
- **CHROMOSOM** składa się z uporządkowanego ciągu genów, przechowuje wszystkie cechy osobnika
- **GENOTYP** w przyrodzie może składać się z kilku chromosomów i określa skład osobnika.  
W algorytmach genetycznych przyjmuje się, że jest to pojedynczy chromosom [7].
- **FUNKCJA PRZYSTOSOWANIA**

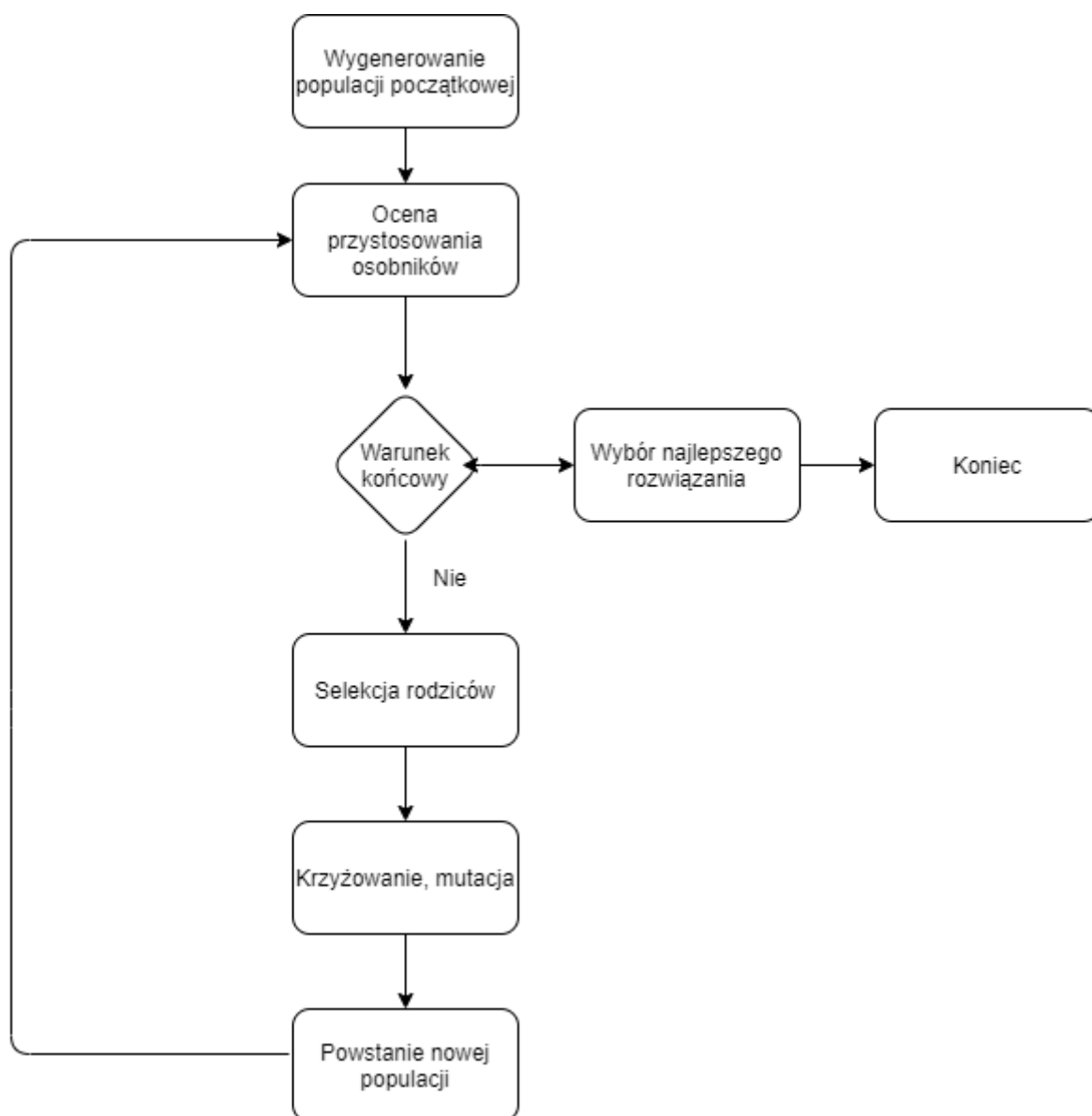
Schemat blokowy klasycznego algorytmu genetycznego został pokazany na rysunku 1.3

Pierwszym krokiem jest wylosowanie populacji początkowej algorytmu. Wielkość populacji podczas trwania całego algorytmu jest stała  $N$ . Ważne jest, aby wszystkie osobniki były jak najbardziej zróżnicowane i wygenerowane losowo. Każdy z nich następnie musi zostać zakodowany do postaci chromosomów, które będą przechowywać w sobie informację o odwiedzanych punktach w postaci genów.

W populacji każdy osobnik musi zostać poddany ocenie funkcji przystosowania. Jej wynik determinuje jak dobre jest dane rozwiązanie. W klasycznym algorytmie dąży się do maksymalizacji tej funkcji. Określenie jak dana funkcja przystosowania będzie wyglądać, jest to jedną z najważniejszych części algorytmu genetycznego. Jeśli zostanie źle zdefiniowana, znalezione osobnik może nie spełniać wymagań rozwiązania problemu.

Po ocenie osobników zostaje sprawdzony warunek końcowy algorytmu. W zależności od problemu zostaje zdefiniowany inny warunek. W klasycznych podejściach są dwa



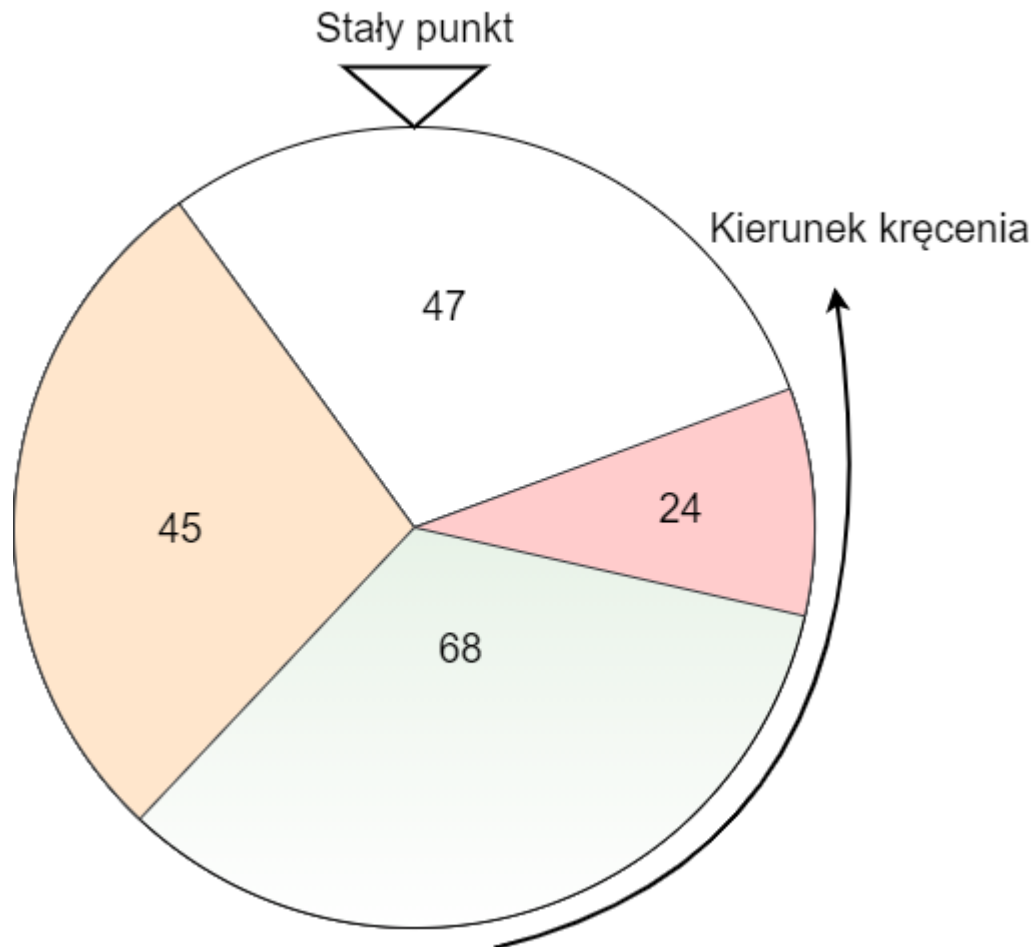


Rysunek 1.3: Schemat algorytmu genetycznego

rodzaje warunków końcowych. Pierwszym popularnym warunkiem końcowym jest stała ilość iteracji algorytmu, czyli po wykonaniu określonej ilości razy ewolucji, wybierany jest najlepszy osobnik z populacji. Drugim warunkiem zazwyczaj jest przetwarzanie algorytmu dopóki nie zostanie znaleziony dostatecznie dobry osobnik. Należy również założyć, że jeśli w kolejnych pokoleniach nie zachodzi poprawa najlepszego rozwiązania, należy przerwać. Wybór w jaki sposób będzie wyglądać warunek końcowy zależy od wielu czynników. Jeśli ważny jest krótki czas, należy założyć pierwszy wariant. Jeśli natomiast algorytm może szukać rozwiązania nawet kilka godzin, można przyjąć drugi wariant.

Kolejnym krokiem algorytmu jest wyselekcjonowanie rodziców do reprodukcji. Polega ona na tym, że osobniki lepsze(mają większą wartość oceny przystosowania)mają

większe szansę na pozostanie rodzicem i przekazanie swoich cech. [3]. Najpopularniejszymi metodami wyboru rodziców jest metoda ruletki oraz turniejowa.



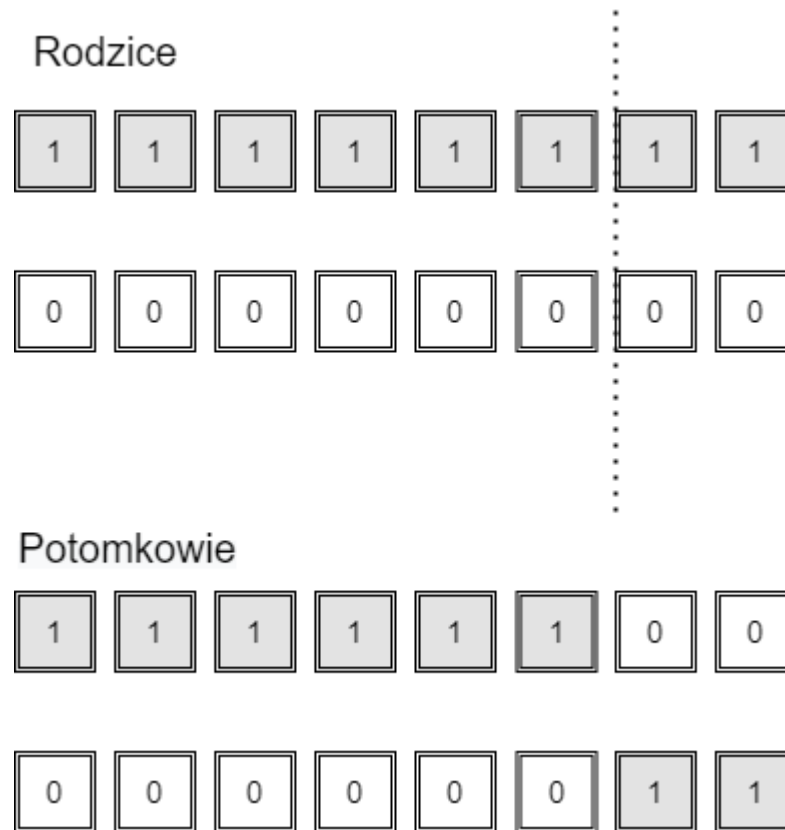
Rysunek 1.4: Metoda ruletki

Na rysunku 1.4 została zilustrowana pierwsza metoda ruletki. Każdy z osobników dostaje wirtualny wycinek koła fortuny. Jego wielkość zależy od wartości funkcji prawdopodobieństwa. Przy każdym wyborze rodzica następuje zakręceniem koła i do reprodukcji zostaje wybrany osobnik na który będzie wskazywał stały punkt.

W metodzie turniejowej zostaje wybranych  $r$  osobników z populacji  $N$ . Z pośród nich zostaje wybrany zwycięzca(największa wartość funkcji przystosowania), który trafia do puli rodzicielskiej.Im większa jest ilość osobników  $r$  tym mniejsze szanse, że słabsze osobniki zostaną wybrane.

Wybrani rodzice zostają poddani operatorom genetycznym: krzyżowanie(ang. crossover) oraz mutacji(ang. mutation). Krzyżowanie polega na połączeniu części chromosomu jednego rodzica z częścią drugiego. Wynikiem takiego połączenia jest nowy osobnik. Proces krzyżowania w zależności może przebiegać w różny ale zawsze określony sposób.

Wszystko zależy od metody zakodowania chromosomu oraz od tego czy kolejność genów i ich unikalność ma znaczenie. W klasycznym podejściu polega na rozcięciu w dowolnym miejscu genotypu u dwóch osobników oraz skrzyżowaniu ich ze sobą w tym punkcie rys. 1.5. Następnie u nowego osobnika może z prawdopodobieństwem  $pm$  wystąpić



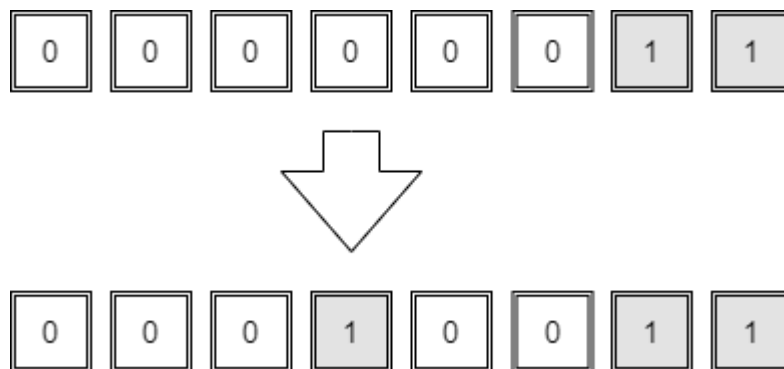
Rysunek 1.5: Klasyczne krzyżowanie

mutacja. Jest to zmiana dowolnego pojedynczego rys ?? lub ciągu genów na inny. Wartość  $pm$  w klasycznych algorytmach jest stosunkowo niskie. Mutacja ma na celu delikatne zróżnicowanie osobników w celu przeszukania nowej przestrzeni rozwiązań. Natomiast gdyby zachodziła często, mogłaby powodować niszczenie dobrych rozwiązań.

Następnie nowa populacja jest poddawana ocenie przystosowania i jeśli wystąpił warunek końcowy, wybierane jest najlepsze rozwiązanie.

## 1.2 Algorytm mrówkowy

Obserwacje nad zachowaniami w przyrodzie wielokrotnie miały wpływ na rozwój nowych rozwiązań. Tak jak w przypadku algorytmu genetycznego, tak i w przypadku algorytmu mrówkowego pomysł został zaczerpnięty z przyrody. Dokładne działanie algorytmu



Rysunek 1.6: Mutacja genotypu

mrówkowego wzoruje się na zachowaniu kolonii mrówek. Dzięki pracy zespołowej, owady te są w stanie wypracować optymalną ścieżkę między siedliskiem a znalezionym pokarmem.

Dla niejednego gatunku problematyczne mogłoby być odtworzenie przebytej ścieżki. Na początku należałoby zadać sobie pytanie w jaki sposób te niewielkich rozmiarów owady są w stanie znacząco ułatwić sobie przetrwanie? Istotną rolę odgrywa tutaj wspomniana już praca zespołowa. To dzięki współpracy mrówki są w stanie optymalizować trasę. Innym ważnym czynnikiem determinującym poprawę ścieżki jest zapach jaki zostawiają mrówki.

Zapach nie jest niczym innym jak feromonem wytwarzanym przez mrówki. Dzięki pozostawionemu zapachowi mrówki wiedziały w jaki sposób poruszali się ich poprzednicy w związku z czym odtworzenie trasy nie stanowiło już poważnego problemu. Przy kolejnych iteracjach kolonia próbuje optymalizować aktualną trasę. W tym celu również wykorzystuje zapach pozostawiony w poprzednich przejściach. Ścieżka jest losowo zmieniana w celu optymalizacji. Jeśli modyfikacja przyniosła oczekiwany efekt, to trasa zostaje zmieniona.

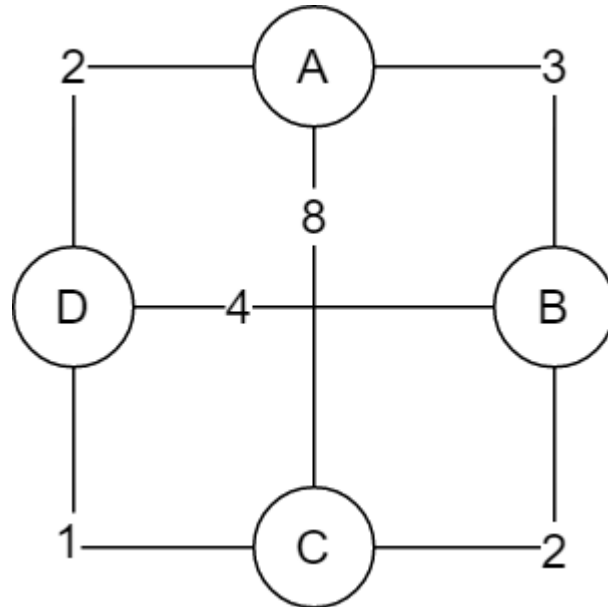
Feromony są istotnym czynnikiem w całym procesie. To dzięki nim trasa jest ulepszana. Zapach posiada jedną z charakterystyk która na początku może wydawać się problematyczna. Wraz z upływem czasu siła zapachu słabnie aż do całkowitego zaniknięcia. Właściwość ta jest zaletą, a nie wadą. To dzięki pracy zespołowej zapach na najlepszej trasie jest podtrzymywany, a na słabszych zanika. Dzięki tej selekcji dłuższe trasy nie są brane pod uwagę w wyniku czego zostaje trasa najkorzystniejsza.

### 1.2.1 Opis działania algorytmu

Do wyznaczenia optymalnej trasy potrzebne są długości jakie należy przebyć do przemieszczania się między punktami. W 1.1 przedstawione są przykładowe odległości.

Tabela 1.1: Wartości kosztów

	A	B	C	D
A	0	3	8	2
B	3	0	2	4
C	8	2	0	1
D	2	4	1	0



Rysunek 1.7: Graf z wagami

Równie ważne jest wyznaczenie początkowych współczynników feromonów. Na początku nadajmy wszystkim krawędziom w grafie wartości równe 1, a współczynnik parowania niech wynosi 0.5.

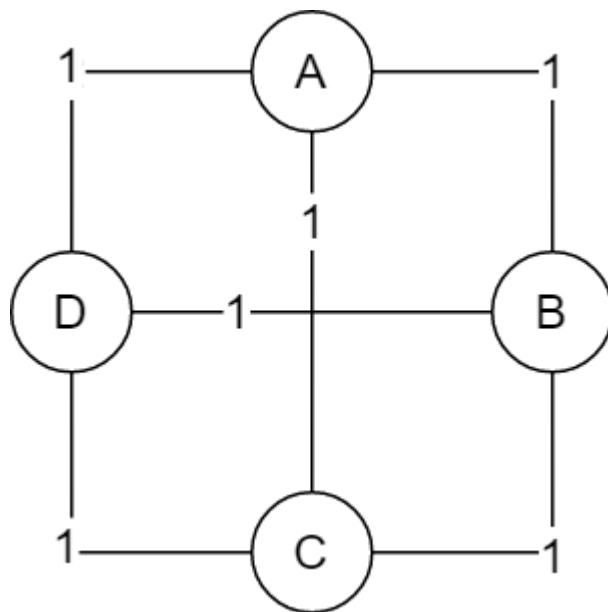
Tabela 1.2: Wartości feromonów

	A	B	C	D
A	0	1	1	1
B	1	0	1	1
C	1	1	0	1
D	1	1	1	0

Posiadając początkowe dane wyznaczmy w sposób losowy trasy dla dwóch agentów: L1 i L2. Agent L1 poruszał się trasą w której odwiedził wierzchołki w następującej kolejności: A, B, C, D, A.

Agent L2 wyznaczył następującą trasę: A, C, B, D, A.

Mrówki odpowiednio pokonały dystans 8 i 16 punktów. Dzięki tej informacji można zaktualizować wartości feromonów na poszczególnych krawędziach. Do obliczeń

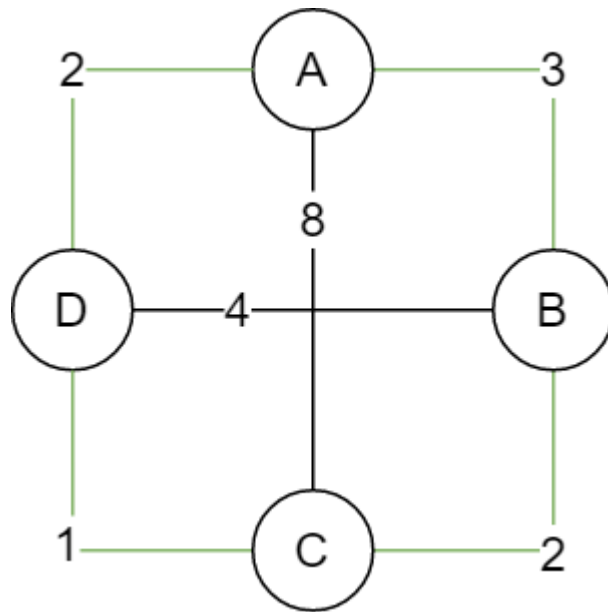


Rysunek 1.8: Graf z początkowymi wartościami feromonów

wykorzystany jest wzór: TUTAJ WZÓR DODAC. Krawędzie A-B i C-D zostały odwiedzone jedynie przez agenta L1 w wyniku czego wartość feromonu zostaje zmieniona na  $10/16$ . Następnie krawędzie A-C i B-D zostają zaktualizowane na  $9/16$ . Krawędzie A-D i B-C są odwiedzone dwukrotnie a wartość feromonów wynosi  $11/16$ .

Ostatnią fazą algorytmu jest wyznaczenie prawdopodobieństwa z jakim kolejni agenci będą wybierać kolejny wierzchołek. W kolejnej iteracji agent L3 znajduje się w wierzchołku B. Do wyboru ma krawędzie A, C i D. Według wzoru TUTAJ WZÓR DODAC wyliczane jest prawdopodobieństwo dla wszystkich możliwości. Przejście z krawędzi B do krawędzi A wynosi około 20 PROCENT, do krawędzi D 30 procent, a do krawędzi C około 50 PROCENT.

Optymalna trasa zostanie wykształcona po wykonaniu wielu iteracji. Ilość iteracji nie jest zdefiniowana i dla każdego przypadku może być różna. Sytuacja wygląda identycznie w przypadku wyboru wartości  $p$ . Ważne jest natomiast to, aby w trakcie działania algorytmu nie modyfikować tej wartości. Powinno ona być taka sama na każdym kroku algorytmu.



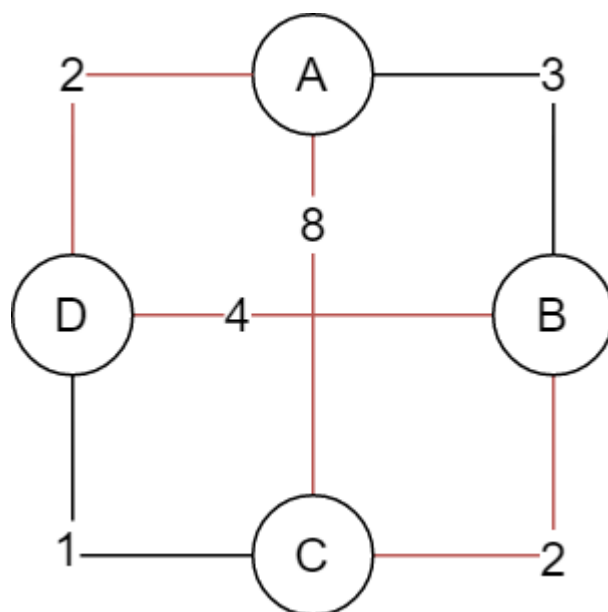
Rysunek 1.9: Trasa przebyta przez agenta L1

### 1.3 Algorytm zachłanne

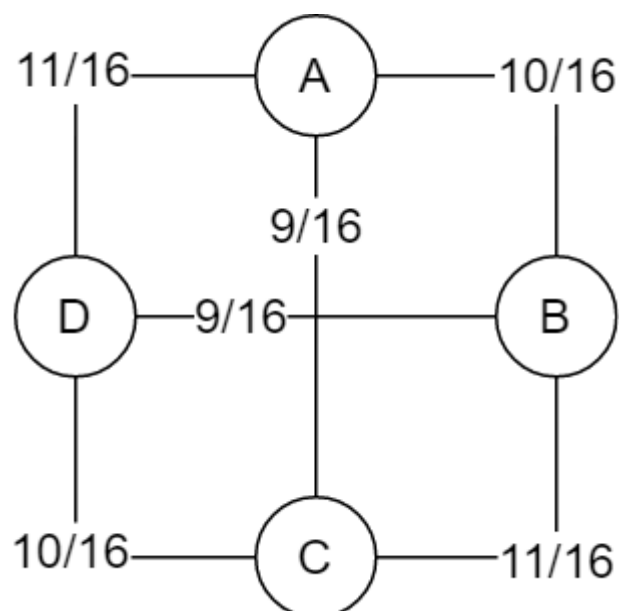
#### 1.3.1 Metoda A\*

#### 1.3.2 Metoda A+

#### 1.3.3 Metoda A-



Rysunek 1.10: Trasa przebyta przez agenta L2



Rysunek 1.11: Graf z wartościami feromonów po modyfikacji



## **2. Algorytm genetyczny**

### **2.1 Chromosom**

Chromosom z definicji jest to ciąg genów reprezentujący dane rozwiązanie. Z kolei gen przenosi informację o cechach. Możliwość osiągnięcia sukcesu jest tylko wtedy, gdy odpowiednio zakoduje się informację i ustali funkcję przystosowania. Do zakodowania badanego problemu użyję metody permutacyjnej, gdzie każdy punkt musi zostać odwiedzony tylko raz. Każdemu punktowi przed wylosowanie tras zostanie przypisany unikalny indeks, będzie on odpowiadał genowi. Następnie dla każdego z  $K$  osobników zostanie zapisany chromosom w postaci ciągu permutacyjnego. Może to być [1234567] jak również [6124573]. Te dwa chromosomy odwiedzają wszystkie punkty w różnej kolejności oraz tylko raz.

### **2.2 Selekcja**

W algorytmie genetycznym przed wykonaniem operatorów należy dokonać selekcji rodziców. Istnieje wiele różnych sposobów, ja przebadam dwie najpopularniejsze. Jest to metoda ruletki oraz turniejowa

#### **2.2.1 Metoda ruletki**



### 3. Algorytm mrówkowy

Słowo rozwój może być zestawiane z wieloma rzeczownikami. Wiele dziedzin ciągle się rozwija, powstają nowe udogodnienia które wpływają na wiele dziedzin życia. Dzięki rozwojowi techniki ludzie są w stanie osiągać cele które jakiś czas temu mogły być tylko marzeniami. Rozwój techniki również potrzebuje inspiracji do tworzenia nowych, lepszych rozwiązań

Szybkość oraz dokładność rozwoju zależy od wielu czynników. Dzięki pracy zespołowej pewne problemy mogą być rozwiązywane szybciej i dokładniej. Wysiłek włożony przez grupę procentuje szybko, a same efekty mogą być również satysfakcjonujące. Istotnym czynnikiem jest wysiłek wkładany przez każdego członka grupy.

Obserwując przyrodę możemy zauważyć w jaki sposób zwierzęta radzą sobie z różnymi problemami. Złożone grupy mogą być spokojniejsze o zdobycie pożywienia czy też o przetrwanie w ciężkich warunkach. Praca zespołowa jest jedną z cech której osobniki w grupie uczą się nie będąc nawet tego do końca świadomym.

#### 3.1 Wprowadzenie do algorytmu

Na przestrzeni czasu wiele gatunków zwierząt żyjących na ziemi przystosowało się do panujących tu warunków. Jedną z takich gatunków są mrówki. Te niewielkich rozmiarów owady posiadają zdolności pomagające im przetrwać wśród najcięższych warunków. Mrówki żyją w stadach w związku z czym wykorzystują pracę zespołową do rozwiązywania problemów jakie codziennie napotykają na swojej drodze.

Aby zapewnić przetrwanie stada mrówki potrzebują zapewnić sobie dostęp do pokarmu. Dziesiątki tysięcy mrówek mają swoje schronienie w mrowiskach. To tam trafia zdobyty przez nich pokarm. Wystarczy aby jedna mrówka znalazła miejsce z pokarmem, to po powrocie do mrowiska inne osobniki są w stanie odtworzyć drogę do pożywienia. Na tym etapie należałoby się zastanowić, w jaki sposób mrówki są w stanie komunikować się między sobą?

Jednym z opisywanych przez nas rozwiązań do wyznaczania zoptymalizowanej trasy jest algorytm mrówkowy, inaczej nazywany ACO - Ant Colony Optimization. Pomysł ten został zaczerpnięty z natury. Jak sama nazwa wskazuje działanie algorytmu jest związane z

mrówkami, a dokładnie z kolonią mrówek. Pomysł na algorytm został zaproponowany na początku lat 90 XX wieku przez włoskiego badacza - Marco Dorigo.

Tak jak wspomniałem wcześniej, algorytm opiera się na pracy mrówek. Chodzi tutaj dokładnie o trasę jako mrówki pokonują od swojego siedliska do miejsca w którym znajduje się pożywienie. Ważne jest znaczenie tutaj pracy zespołowej. Trasę kształtuje cała kolonia mrówek, a nie pojedyncze przypadki. Mrówki z każdą kolejną podróżą wykształcają coraz to bardziej optymalną trasę.

### 3.2 Opis działania algorytmu

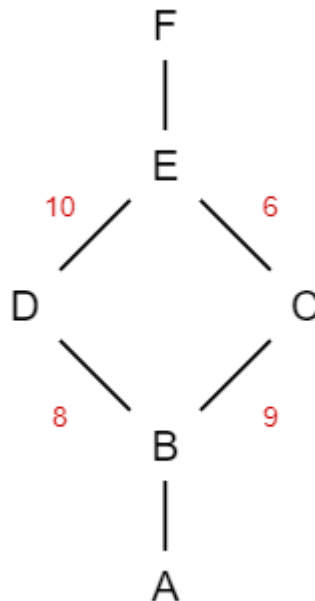
Mrówka w celu znalezienia pokarmu w sposób losowy wyrusza z mrowiska. Losowo przeszukując teren szuka pokarmu. Gdy już go znajdzie wraca do siedliska i informuje o tym fakcie pozostałe mrówki. Aby dostarczyć więcej pokarmu mrówki wyruszają do miejsca spoczynku pożywienia. Chcąc uniknąć sytuacji w której zdobycz może zostać zabrana przez inne owady, mrówki muszą jak najbardziej zoptymalizować trasę jaką mają do pokonania.

Mimo posiadania informacji o znalezionym pokarmie, każda mrówka sama musi zlokalizować źródło. Czy mrówki poruszają się tą samą trasą przy każdym wyjściu z mrowiska? Aby trafić do miejsca w którym znajduje się pokarm, wspomniane owady wykorzystują ślady pozostawione przez osobników które już natrafiły na pożywienie. W ten sposób mrówka która wyrusza w sposób losowy, natrafia na ślad poprzednika który jest wskazówką do znalezienia poszukiwanego pokarmu. Wspomniany ślad nazywa się feromonem. To dzięki tej właściwości mrówki są w stanie lokalizować trasy prowadzące do pokarmu.

Na rysunku ?? został przedstawiony graf z wierzchołkami A-B-C-D-E-F. Nad krawędziami kolorem czerwonym zostały oznaczone wagi. Na początku założmy, że mamy do dyspozycji 80 agentów. Przez kilka pierwszych iteracji mrówki poruszały się losowo i powstał następujący podział:

Tak jak można to zauważyć na grafice ??, po pierwszych iteracjach, przez obie ścieżki przechodzi taka sama ilość agentów. Dzieje się tak ponieważ mrówki rozpoczynają pracę w sposób losowy. W dalszych krokach następują modyfikacje i agenci dążą do wyznaczenia najoptymalniejszej ścieżki.

Ilość agentów odwiedzających ścieżki zmienia się. Bardziej optymalna trasa zyskuje widoczną przewagę. W kolejnych iteracjach mrówki wykorzystują siłę feromonów. Bardziej



Rysunek 3.1: Początkowy graf

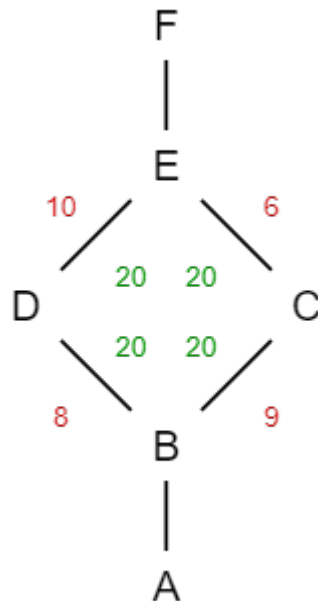
optymalne ścieżki są częściej odwiedzane w związku z czym zapach na tych krawędziach jest silniejszy oraz podtrzymywany. Na mniej optymalnych trasach zapach zanika i przestają one być atrakcyjne dla agentów. Opisana sytuacja prowadzi do wyznaczenia trasy która jest najatrakcyjniejsza do przebycia dla mrówek.

Feromony posiadają cechę która może się wydawać, że negatywnie wpływa na wyznaczanie ścieżki. Chodzi tutaj o ulatnianie się zostawionego zapachu. Na pierwszy rzut oka może się to zjawisko wydawać niepożądanym, ale w rzeczywistości ma duży wpływ na optymalizację. Jeśli feromony nie straciłyby na swojej sile, to bardzo prawdopodobne, że pierwotna ścieżka mogłaby zostać uznana za najbardziej optymalną.

W jaki sposób wyznaczona zostaje najbardziej optymalna ścieżka? Zapach feromonów jest podtrzymywany przez wędrujące mrówki. Z czasem owady te same zbaczają z drogi w celu poszukiwania alternatywnej trasy. Jeśli wybrana trasa jest optymalniejsza od pozostałych to ślad jest podtrzymywany, a na innych zanika. Dzięki temu w sposób iteracyjny można wyróżnić trasę najoptymalniejszą, a słabsze z czasem zostają odrzucone ponieważ przestają być odwiedzane.

### 3.3 Pozostałe zastosowania algorytmu mrówkowego

Algorytm mrówkowy znajduje swoje zastosowanie w rozwiązywaniu innych problemów. Problem plecakowy jest jednym z takich przykładów. Pojawia się on najczęściej przy

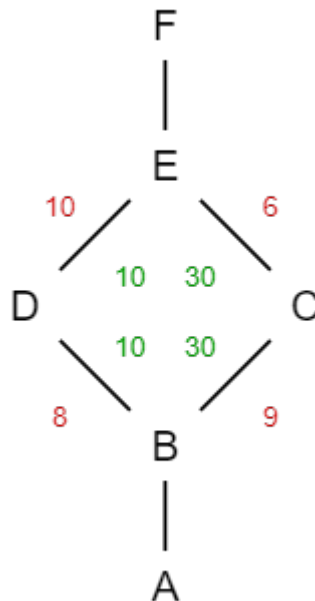


Rysunek 3.2: Rozkład ścieżek na grafie po pierwszych iteracjach

optymalnym zarządzaniu zasobami. Mamy dany zbiór jakichś przedmiotów z czego każdy z nich posiada określony ciężar i wartość. Do plecaka musimy załadować przedmioty o jak największej wartości. Naszym ograniczeniem jest jednak łączny ciężar przedmiotów które możemy udźwignąć.

Algorytm mrówkowy wygląda bardzo podobnie w tym przypadku. Na początku mamy pewną generację agentów - mrówek. Każdy agent iteracyjnie poszukuje jak najlepszego rozwiązania. Po każdej iteracji można wyróżnić trzy typy rozwiązań: rozwiązanie pośrednie, rozwiązanie częściowe lub stan. Agenci w celu znalezienia rozwiązania wykorzystuje swoje naturalne umiejętności czyli zostawia na wszystkich obiektach w plecaku feromony. Dzięki lotności feromonów mrówki są w stanie identyfikować bardziej zadowalające przedmioty.

Krzysztof Schiff w jednym z artykułów przedstawił rozwiązanie problemu plecakowego przy użyciu trzech metod optymalizacji algorytmu mrówkowego. Wszystkie te metody można opisać za pomocą wzorów. Zgodnie z przyjętą konwencją przez autora artykułu metody mają odpowiednie nazwy: AKA1, AKA2 oraz AKA3. Atrakcyjność jest obliczana według wzorów:  $AKA1 = z / w / V$ , gdzie: -  $z$  jest zyskiem wybranego obiektu; -  $w$  jest wagą wybranego obiektu; -  $V$  jest aktualną ładownością plecaka  $AKA2 = z / w^2$ , gdzie: -  $z$  jest zyskiem wybranego obiektu; -  $w$  jest wagą wybranego obiektu;  $AKA3 = z / w / C$ , gdzie: -  $z$  jest zyskiem wybranego obiektu; -  $w$  jest wagą wybranego obiektu; -  $C$  jest całkowitą wagą plecaka



Rysunek 3.3: Rozkład agentów w grafie po optymalizacji

Poniżej przedstawione zostały wyniki badań:

#### WYNIKI BADAN

Problem komiwojażera i problem plecakowy są problemami kombinatorycznymi i ich rozwiązanie polega na poszukiwaniu optymalnej ścieżki na grafie pełnym. Inną wariacją problemu jest poszukiwanie rozwiązania problemu od razu zadanego na grafie. Jako przykład może posłużyć problem kolorowania grafu. Dla wszystkich wierzchołków w grafie należy dobrać takie kolory, aby żadne dwa sąsiednie wierzchołki nie miały tego samego koloru.

Mrówki nie działają bezpośrednio na grafie początkowym ponieważ graf ten nie musi być grafem pełnym. Należy stworzyć dla mrówek alternatywę podobną do oryginału z zachowaniem takiego samego zbioru wierzchołków ale z pełnymi krawędziami. Następnie należy dobrać numeryczne wartości odpowiadające konkretnym kolorom. Jeśli mrówka odwiedzi dany wierzchołek, to zostaje on pokolorowany na najniższy kolor który nie został dotychczas użyty do kolorowania któregoś z sąsiadów.

Tak jak w przypadku poprzednich algorytmów wykorzystywane są zapachy pozostawiane przez mrówki. Ilość użytych unikalnych kolorów byłaby odwrotnie proporcjonalna do ilości feromonów. W efekcie czego heurystyka byłaby odwrotnie proporcjonalna do wykorzystanych kolorów po kolejnych iteracjach. Wynikiem tych operacji będzie rozwiązanie w którym w grafie oryginalnym każdy wierzchołek będzie odwiedzany tylko raz.

Ostatni z przykładów wykorzystania algorytmu mrówkowego jest harmonogram produkcji. W porównaniu do poprzednich metod w tym algorytmie zachodzi pewna modyfikacja. Głównym problemem w harmonogramie produkcji jest znalezienie takiej kolejności przetwarzanych zadań, aby jak najszybciej je przetworzyć. Aby lepiej zobrazować tą sytuację należy sobie wyobrazić fabrykę w której znajdują się linie produkcyjne. Na linii są przetwarzane zadania w odpowiedniej kolejności oraz każde z zadań może zostać wykonane w różnym czasie.

W tej metodzie, podobnie jak w metodzie do rozwiązania problemu plecakowego, należy stworzyć graf pełny z wierzchołkami odpowiadającymi konkretnym zadaniom. Następnie mrówki przechodzą przez wszystkie wierzchołki i zostawiają feromony. Czynnikiem decydującym o wyborze wierzchołków nadal jest związana z feromonami. Do rozwiązania tego problemu nie jest brana pod uwagę ilość feromonów na krawędzi pomiędzy wierzchołkiem a jego sąsiadami. Wykorzystywana jest natomiast suma feromonów na wszystkich krawędziach do odwiedzanych wierzchołków z wierzchołkami już odwiedzionymi.



## 4. Rozdział Przemokowy

Prosta tabela 4.1.

Tabela 4.1: Długi podpis tabeli 1, który pojawi się nad nią. Jak chcecie podpis pod tabelą, umieście caption przed samym `end{table}` - ale to niezgodne z wytycznymi.

Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4
Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4
Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4
Kolumna 1	Kolumna 2	Kolumna 3	Kolumna 4

Przykładowa tabela 4.2, nieco bardziej skomplikowana.

Tabela 4.2: Długi podpis tabeli 2, który pojawi się nad nią

Kolumna wyróżniona	Kolumna pierwsza	Kolumna druga	Kolumna kolejna długa nazwa	Przeniesienie słowa	Kolumna kolejna	Kolumna kolejna	Kolumna kolejna	Kolumna kolejna	Kolumna kolejna	Kolumna kolejna
Wiersz jakiś tam	1	2	3	4	5	6	7	8	9	10
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	<b>92,38</b>	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	<b>92,38</b>	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	<b>41,92</b>	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82
Wiersz ze statystykami	11,56	92,38	827,21	41,92	29,71	28,77	29,61	55,02	72,33	95,82



## **5. Badania**

### **5.1 Wyniki algorytmu genetycznego**

### **5.2 Wyniki algorytmu mrówkowego**

### **5.3 Wyniki algorytmu mrówkowego**



## **6. Porównania wyników**



## **Podsumowanie**

Tutaj będzie podsumowanie.





## Bibliografia

- [1] J. Autor. Nazwa strony internetowej. `http://www.dlugi.adres.url.zlamie.sie.gdzies.w.srodku.com`, stan z 01.01.2010 r.
- [2] U. Autor and W. Kolejny. Tytuł publikacji. *Nazwa czasopisma*, 12(2):132–145, May 2012.
- [3] Ewa Figielska. Algorytmy ewolucyjne i ich zastosowania. *Zeszyty Naukowe Warszawskiej Wyższej Szkoły Informatyki*, 1(1):81–92, 2006.
- [4] Anna Gryko-Nikitin. Niektóre osobliwości algorytmów genetycznych na przykładzie zagadnień logistycznych. *Ekonomia i Zarządzanie*, 1(2):129–138, 2010.
- [5] B. Inny. Tytuł publikacji. In *Tytuł książki*, pages 5–32, Feb 2011.
- [6] Z. Test and K. Następny. *Tytuł książki*. Wydawca, Adres, 1995.
- [7] Radosław Winiczenko. Algorytmy genetyczne i ich zastosowania. *Postępy Techniki Przetwórstwa Spożywczego*, 1(2):107–110, 2008.



## Spis tabel

Tablica 1.1	Krótki podpis tabeli 1 – do spisu trešci . . . . .	20
Tablica 1.2	Krótki podpis tabeli 1 – do spisu trešci . . . . .	21
Tablica 4.1	Krótki podpis tabeli 1 – do spisu trešci . . . . .	33
Tablica 4.2	Krótki podpis tabeli 2 – do spisu trešci . . . . .	33



## Spis rysunków

Rysunek 1.1	Graf bez cyklu Hamiltona. . . . .	14
Rysunek 1.2	Graf z cyklem Hamiltona. . . . .	14
Rysunek 1.3	Schemat algorytmu genetycznego . . . . .	17
Rysunek 1.4	Metoda ruletki . . . . .	18
Rysunek 1.5	Klasyczne krzyżowanie . . . . .	19
Rysunek 1.6	Graf z wagami . . . . .	21
Rysunek 1.7	Graf z początkowymi wartościami feromonów . . . . .	22
Rysunek 1.8	Trasa przebyta przez agenta L1 . . . . .	23
Rysunek 1.9	Trasa przebyta przez agenta L2 . . . . .	23
Rysunek 1.10	Graf z wartościami feromonów po modyfikacji . . . . .	24
Rysunek 3.1	Początkowy graf . . . . .	29
Rysunek 3.2	Rozkład ścieżek na grafie po pierwszych iteracjach . . . . .	30
Rysunek 3.3	Rozkład agentów w grafie po optymalizacji . . . . .	31



## **Spis listingów**





## **Spis algorytmów**