

**Wojciech Tomasz Dobrosielski\***

*Uniwersytet Kazimierza Wielkiego w Bydgoszczy*

<https://orcid.org/0000-0001-7756-4259>

## **ZASTOSOWANIE ROZMYTEJ KOLONII MRÓWEK W ROZWIĄZYWANIU PROBLEMÓW KOMIWOJAZERA**

### **Streszczenie**

Celem artykułu jest omówienie zastosowania skierowanych liczb rozmytych OFN (Ordered Fuzzy Numbers) w rozwiązywaniu problemu komiwojażera TSP (Travel Selsman Problem) przy użyciu kolonii mrówek ACO (Ant Colony Optimization). Możliwość użycia liczb OFN w algorytmie mrówkowym, to nowe podejście do problemu optymalizacji danego problemu. Autorzy rozwiązania przedstawiają metodę antOFN do rozwiązania tego problemu optymalizacji. W kolejnych krokach omawiamy sam problem TSP i możliwe rozwiązania. Dalej autorzy przedstawiają rozwiązania problemu przy użyciu ACO. Centralnym punktem artykułu jest omówienie rozwiązania w oparciu o skierowane liczby rozmyte OFN, gdzie wyjaśniamy interpretację feromonu oraz opisujemy wykonywane operacje. W końcowej części, autorzy skupiają na eksperymencie, w którym prezentują wyniki dla rozwiązań z grupy ACO jako i dedykowanych dla problemu komiwojażera. Rozwiązanie antOFN na potrzeby tego badania, zostało zaimplementowane i porównane resztą algorytmów. Końcowe wyniki eksperymentu pokazują, że metody oparte na Ant Colony Optimization są lepsze, jeżeli chodzi o osiągnięcie rozwiązań. Proponowana antOFN także należy do klasy optymalnych rozwiązań.

**Słowa kluczowe:** *ant colony optimization, TSP, feromon, swarm intelligence, przeszukiwanie grafu, skierowane liczby rozmyte OFN*

---

\* **Wojciech Tomasz Dobrosielski** – magister inżynier, Katedra Systemów Inteligentnych Wydziału Informatyki, e-mail: [wdobrosielski@ukw.edu.pl](mailto:wdobrosielski@ukw.edu.pl), [wdobrosielski@gmail.com](mailto:wdobrosielski@gmail.com)

## APPLICATION OF FUZZY ANT COLONY IN SOLVING TRAVELING SALESMAN PROBLEMS

### Summary

The aim of the article is to discuss the use of directed fuzzy numbers OFN (Ordered Fuzzy Numbers) in solving the Traveling Salesman Problem (TSP) using Ant Colony Optimization (ACO). The possibility of using OFN numbers in the ant algorithm is a new approach to the optimization problem of a given problem. The authors of the solution present the antOFN method to solve this optimization problem. In the following steps, we discuss the TSP itself and possible solutions. Then, the authors present solutions to the problem using ACO. The central point of the article is a discussion of the solution based on directed fuzzy numbers OFN, where we explain the interpretation of pheromone and describe the operations performed. In the final part, the authors focus on the experiment in which they present results for solutions from the ACO group and those dedicated to the Traveling Salesman Problem. The antOFN solution for this study has been implemented and compared to the rest of the algorithms. The final results of the experiment show that methods based on Ant Colony Optimization are better when it comes to achieving solutions. The proposed antOFN also belongs to the class of optimal solutions.

**Keywords:** *ant colony optimization, TSP, pheromone, swarm intelligence, graph search, ordered fuzzy numbers OFN*

~ . ~

### Wstęp

W niniejszym rozwiązaniu skoncentrujemy się na trzech powiązanych tematach: problemie komiwojażera (TSP) i systemach mrówkowych ACO oraz skierowanych liczbach rozmytych OFN. Systemy ACO zostały zainspirowane poprzez studiowanie kolonii mrówek i termitów, gdzie zauważono wyjątkową organizację możliwą do osiągnięcia tylko przez kooperację całej grupy. Ta umiejętność efektywnego znalezienia najkrótszej trasy do źródła pokarmu zainteresowała naukowców, stając się interesującym zagadnieniem do dalszego rozwinięcia i zastosowania do problemów optymalizacyjnych. Jednym z takich problemów, jaki można zaadresować przy pomocy algorytmów mrówkowych, jest TSP. W interesujący sposób oba te problemy odniosły korzyści z ich wzajemnej interakcji. Dla systemów mrówkowych, TSP stanowiło doskonałe pole do testowania nowych algorytmów, natomiast TSP zyskało na efektywnym odnajdywaniu najkrótszej trasy. W kontekście TSP, mówimy o podróży, który ma na celu pokonać trasę z punktu A do punktu B, odwiedzając po drodze serię innych, wybranych przez

siebie miast, zaś po ich odwiedzeniu powraca do punktu wyjścia. Głównym celem TSP jest optymalizacja podróży, która może być reprezentowana przez różne kryteria, takie jak: czas, odległość czy koszty. Zarówno systemy mrówkowe, jak i TSP, mają wiele punktów wspólnych, takich jak poszukiwanie najkrótszej trasy. Może to stanowić interesujące pole do porównania skuteczności algorytmów mrówkowych w rozwiązywaniu TSP względem innych stosowanych metod.

Matematyczne problemy związane z TSP zostały również poruszone w 1800 roku przez dwóch naukowców: szkockiego matematyka Ser Williama Hamiltona oraz brytyjskiego matematyka Thomasa Kirkmana. Rozważania na temat wczesnych prac Hamiltona i Kirkmana, oraz Eulera możemy znaleźć w pracy pt: „*Graph Theory 1736-1936*”<sup>1</sup>. W tej pracy przedstawiono ciekawy problem miasteczka Königsberg, w którym jak to żartobliwie napisano: mieszkańcy jako rozrywkę próbują odnaleźć trasę dookoła miasta w taki sposób, aby przekroczyć każdy z siedmiu mostów tylko raz.

Aby lepiej zrozumieć omawianą kwestię, całkowitą liczbę wszystkich potencjalnych rozwiązań powyżej przedstawionego przykładu, który odnosi się do problemu TSP, można opisać jako cykl Hamiltona<sup>2</sup>, który jest reprezentowany za pomocą wzoru:

$$\frac{1}{2} \cdot (n - 1) \cdot ! \rightarrow n > 2 \quad (1)$$

Z wykorzystaniem tego równania można określić złożoność problemu dla różnej liczby miast. Jak widać w poniższej tabeli, liczba wszystkich potencjalnych rozwiązań rośnie wraz z liczbą miast, ale nie jest proporcjonalna do złożoności problemu. To sprawia, że problem TSP jest niezwykle trudny do rozwiązania.

Tabela 1. Wielkość problemu w zależności od ilości miast

Miast	Wielkość problemu
3	1
5	12
7	360
10	181440
15	43589145600

<sup>1</sup> N. Biggs, E K. Lloyd, R. J Wilson. *Graph Theory, 1736-1936*, Oxford 1986.

<sup>2</sup> W. T. Dobrosielski. *The golden ratio of area method based on fuzzy number area as a defuzzifier*, in: Krassimir T. Atanassov, Vassia Atanassova, Janusz Kacprzyk, Andrzej Kauszko, Maciej Krawczak, Jan W. Owsinski, Sotir S. Sotirov, Evdokia Sotirova, Eulalia Szmidt, Sławomir Zadrozny, editors, *Uncertainty and Imprecision in Decision Making and Decision Support: New Advances, Challenges, and Perspectives*, Cham 2022, pages 92-108.

20	60822550204416000
25	310224200866620000000000
30	4420880996869850000000000000
35	14761639951980200000000000000000
40	10198941040598700000000000000000000000000000000
45	132913578739422000000000000000000000000000000000000
50	304140932017134000 0000

Dla naszego przykładu czterech miast całkowita ilość możliwych rozwiązań jest równa  $\frac{1}{2} * 3!$ , czyli 3 rzeczywiste rozwiązania dla 6 dostępnych możliwych tras. Dużą zaletą jest to, iż wzór na cykl Hamiltona wyklucza powtórzenie się tego samego rozwiązania.

1. Metryczny problem TSP<sup>3</sup> polega na szukaniu najkrótszej ścieżki łączącej miasta, gdzie odległości między nimi spełniają warunek nierówności trójkątnej  $d(u, v) \leq d(u, w) + d(w, v)$ . Często odległości są mierzone w metryce euklidesowej, a algorytmy rozwiązujące ten problem muszą uwzględnić specyfikę tych odległości.
2. Euklidesowe TSP (ETSP)<sup>4</sup> odnosi się do wariantu klasycznego problemu komiwojagera, w którym miasta są położone na płaszczyźnie (często w przestrzeni dwuwymiarowej) i odległości między nimi są wyznaczone zgodnie z metryką euklidesową, czyli jako odległość w linii prostej między dwoma punktami. W przypadku euklidesowego TSP, odległości między miastami są mierzone na podstawie współrzędnych geograficznych lub położenia miast na mapie. Algorytmy rozwiązujące ten problem muszą uwzględnić te konkretne geometrie miast i dążyć do znalezienia najkrótszej ścieżki łączącej wszystkie miasta, pamiętając o odległościach euklidesowych między nimi. Algorytmy takie mogą być wykorzystywane do optymalizacji tras w rzeczywistych zastosowaniach, takich jak logistyka czy zarządzanie trasami dla pojazdów.
3. **Asymetryczne TSP (ATSP)**<sup>5</sup> – to wariant klasycznego problemu TSP, w którym odległości między miastami są niesymetryczne, czyli mogą różnić się w zależności od kierunku podróży. Innymi słowy, odległość z miasta A do miasta B może być różna od odległości z miasta B do miasta A. Rozwiązując

<sup>3</sup> L. Engebretsen, M. Karpinski, *Tsp with bounded metrics*, "Journal of Computer and System Sciences" 2006, 72(4), 509-546.

<sup>4</sup> Y. Bartal, L.-A. Gottlieb, *A linear time approximation scheme for euclidean tsp*, in: "2013 IEEE 54th Annual Symposium on Foundations of Computer Science" 2013, pages 698-706.

<sup>5</sup> R. Roberti, P. Toth, *Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison*, "EURO Journal on Transportation and Logistics" 2012, 1(1-2), 113-133.

asymetryczny problem TSP, celem jest znalezienie najkrótszej ścieżki łączącej wszystkie miasta, przestrzegając zasady, że odległości między nimi są niesymetryczne. Algorytmy ATSP muszą uwzględniać te różnice w odległościach i znaleźć optymalną trasę biorąc pod uwagę tę asymetrię. Asymetryczny problem TSP może być bardziej złożony do rozwiązania niż klasyczny TSP, ze względu na dodatkowe zmienne wynikające z asymetryczności odległości między miastami. Istnieją specjalne algorytmy i metody, które zostały opracowane specjalnie do rozwiązywania ATSP, aby znaleźć optymalne trasy w przypadku, gdy odległości między miastami są niesymetryczne.

4. **Symetryczne TSP (STSP)** – Symetryczny problem komiwojażera (STSP) to klasyczna wersja problemu TSP, w której odległości między miastami są symetryczne, czyli odległość z miasta A do miasta B jest taka sama jak odległość z miasta B do miasta A. W przypadku STSP, przyjmuje się, że koszty podróży między dwoma miastami pozostają stałe niezależnie od kierunku podróży. Rozwiązując symetryczny problem TSP, celem jest znalezienie najkrótszej ścieżki łączącej wszystkie miasta dokładnie raz i wracającej do miasta początkowego, uwzględniając symetryczność odległości między nimi. Algorytmy rozwiązujące STSP dążą do znalezienia optymalnej trasy, która minimalizuje całkowity koszt podróży. Symetryczny problem komiwojażera jest dobrze zbadany, a istnieje wiele algorytmów i metod rozwiązujących go efektywnie, takich jak algorytm zbioru otwartego czy algorytmy przybliżone, które pomagają znaleźć optymalne trasy w przypadku, gdy odległości między miastami są symetryczne.

## Algorytmy do rozwiązań problemu komiwojażera

Tradycyjnymi sposobami, algorytmami atakującymi NP - trudny problem TSP są:

1. **Algorytmy znajdujące dokładne rozwiązanie** – znajdujące dokładne rozwiązanie dla małych problemów TSP mogą pracować szybko, ale sprawdzenie wszystkich możliwych permutacji staje się niepraktyczne ze względu na fakt, że liczba permutacji wynosi  $n!$ . Techniki dynamicznego programowania pozwalają rozwiązać problem w bardziej efektywny sposób niż sprawdzanie wszystkich permutacji, chociaż jest to nadal rozwiązanie wykładnicze.
2. **Metoda podziałów i ograniczeń** (z ang. *branch-and-bound*, *B&B*) - Metoda *branch-and-bound*<sup>6</sup> polega na rozwiązaniu złożonego problemu poprzez podział na mniejsze, bardziej zarządzalne części. Poprzez kontynuacyjne dzielenie problemu na mniejsze jednostki oraz ustalanie ograniczeń, algorytm

<sup>6</sup> Narendra and Fukunaga. *A branch and bound algorithm for feature subset selection*, "IEEE Transactions on computers" 1977, 100(9), 917-922.

dąży do znalezienia optymalnego rozwiązania. Jest to skuteczna metoda, szczególnie przy problemach, które są zbyt skomplikowane do rozwiązania bezpośrednio. Poprzez analizę częściowych rozwiązań i potencjalnych optymalnych rozwiązań, algorytm może znaleźć najlepsze możliwe rozwiązanie dla danego problemu.

3. **Algorytmy nie w pełni optymalne**, algorytmy heurystyczne, inaczej nazywane algorytmami przybliżonymi, skupiają się na znajdowaniu rozwiązań, które, mimo że nie są optymalne, są szybkie i prawdopodobnie dobre. Oznacza to, że algorytmy heurystyczne nie gwarantują optymalnego rozwiązania, ale zazwyczaj dostarczają rozwiązanie wystarczająco bliskie optymalnemu, aby spełniało wymagania praktyczne. Poprzez skuteczne wykorzystanie różnych metod i strategii, algorytmy heurystyczne szybko znajdują dobre rozwiązania z dużym prawdopodobieństwem ich poprawności. Pomimo że nie są w pełni optymalne, są skutecznym narzędziem do rozwiązywania złożonych problemów.
4. **Algorytm najbliższego sąsiada** (z ang. Nearest neighbour algorithm, NN)<sup>7</sup> był jednym z pierwszych algorytmów stosowanych do rozwiązywania problemu komiwojażera (TSP). Polegał on na wybieraniu najbliższego miasta, które nie zostało jeszcze odwiedzone, jako kolejnego punktu trasy. Mimo że algorytm był stosunkowo szybki, jego rozwiązania nie zawsze były optymalne. Działanie to było wybierane jako prosta metoda rozwiązania problemu komiwojażera, jednak z czasem zauważono, że może dawać jedynie przybliżone rozwiązania, a nie zawsze najlepsze.
5. **Algorytm zachłanny** (z ang. *Greedy algorithm*)<sup>8</sup> – to rozwiązanie metaheurystycznych problemów, które polega na podejmowaniu lokalnych optymalnych wyborów na każdym etapie działania. Główną cechą algorytmu jest dążenie do znalezienia lokalnego optimum, które w teorii może prowadzić do osiągnięcia optimum globalnego. Algorytm zachłanny podejmuje decyzje krok po kroku, zawsze wybierając najbardziej obiecujące rozwiązanie w danym momencie, co ma prowadzić do optymalnego rozwiązania problemu jako całości.

Problem komiwojażera (TSP) stanowi istotne wyzwanie w dziedzinie heurystycznych rozwiązań optymalizacyjnych. Dla tego typu problemów stosuje się różnorodne metody, takie jak algorytmy genetyczne, symulowane wyżarzanie, poszukiwanie tabu czy algorytm mrówkowy (ACO).

<sup>7</sup> K. Arora, S. Agarwal, R. Tanwar, *Solving tsp using genetic algorithm and nearest neighbour algorithm and their comparison*, "International Journal of Scientific & Engineering Research" 2016, 7(1), 1014-1018.

<sup>8</sup> G. Gutin, A. Yeo, *The greedy algorithm for the symmetric tsp*, "Algorithmic Operations Research" 2007, 2(1), 33-36.

## Ogólna charakterystyka systemów mrówkowych

Systemy mrówkowe odnoszą się do koncepcji, w której mrówki jako społeczność o wysokim stopniu zorganizowania prezentują zdolności przekraczające możliwości pojedynczego osobnika. Te systemy informatyczne realizują algorytmy równoległe, działając na zasadach analogicznych do procesów w świecie rzeczywistym. Inspirowane przez zoologię i fizykę molekularną, systemy mrówkowe są wykorzystywane do tworzenia skutecznych algorytmów. Naukowcy doszukują się inspiracji w naturze, która ewoluowała przez miliony lat, aby opracować skuteczne strategie rozwiązywania problemów.

Mrówka w algorytmie ACO (Ant Colony Optimization) działa na zasadzie imitowania zachowania rzeczywistych mrówek w poszukiwaniu najlepszej ścieżki do jedzenia. Na początku mrówka jest umieszczana w losowym miejscu grafu i rozpoczyna swoje poszukiwanie. Idąc ścieżką, mrówka zostawia feromony, czyli substancje chemiczne, które innym mrówkom pomagają znaleźć trasę. Im dłuższa ścieżka, tym więcej feromonów jest zostawiane. Mrówka korzysta z kombinacji feromonów i heurystyki (informacji lokalnej) w celu podjęcia decyzji, w którą stronę iść. W miarę eksploracji grafu mrówka zaczyna preferować ścieżki z większą ilością feromonów, co prowadzi do znalezienia optymalnej trasy.

Ślad feromonowy<sup>9</sup> w algorytmie ACO jest zatem reprezentacją informacji o jakości danego rozwiązania (np. długości trasy) na danych fragmentach grafu. Im więcej mrówek przemieszcza się po danej trasie, tym więcej feromonów zostaje na niej pozostawionych, co prowadzi do wzmocnienia tej ścieżki dla przyszłych mrówek. Dzięki temu system może ewoluować w stronę coraz lepszych rozwiązań, wykorzystując zasoby heurystyczne i informację o jakości tras ze śladów feromonowych. Ta kombinacja pozwala na efektywne znalezienie optymalnego rozwiązania w problemach optymalizacyjnych.

W roku 1992 Marco Dorigo stworzył pojęcie algorytmów mrówkowych, będących wieloagentowym systemem inspirowanym zachowaniami rzeczywistych kolonii mrówek, wykorzystując zagadnienie stigmergy (pośrednia komunikacja). Algorytmy te czerpią swoje korzenie z obserwacji naturalnego środowiska, w którym żyją mrówki. Zagadnienie stigmergy pełni kluczową rolę w algorytmach mrówkowych, umożliwiając działanie poprzez implementację zmiennych STY, które zawierają informacje służące do komunikacji pośredniej między sztucznymi mrówkami. Przykładowo, w algorytmie ACO zmienne STY są szczegółowo zdefiniowane i wykorzystywane przez mrówki do modyfikowania sposobu konstruowania rozwiązania dla danego problemu optymalizacyjnego.

<sup>9</sup> R. Jovanovic, M. Tuba, *Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem*, "Computer Science and Information Systems" 2013, 10(1), 133-149.

Ponadto, algorytmy mrówkowe pozwalają na efektywne przeszukiwanie przestrzeni rozwiązań, wykorzystując informacje zawarte w śladach feromonowych pozostawionych przez mrówki. Dzięki temu możliwe jest znalezienie optymalnego rozwiązania problemu, poprzez ewolucję systemu na podstawie zbieranych i wykorzystywanych danych. W ten sposób algorytmy mrówkowe stanowią potężne narzędzie do rozwiązywania różnorodnych problemów optymalizacyjnych.

Algorytmy inspirowane zruchowaniami mrówek zyskały popularność dzięki swojej skuteczności w rozwiązywaniu problemów optymalizacji i kontroli zaburzeń. Jednym z kluczowych pojęć wykorzystywanych w tych algorytmach jest sztuczny agent, który reprezentuje działania i interakcje zwykłej mrówki. Sztuczne mrówki wraz z innymi agentami współpracują, aby znaleźć optymalne rozwiązania problemów obliczeniowych. Istnieje wiele różnych czynników w otoczeniu kolonii mrówek, które mogą być wykorzystane do stworzenia różnorodnych algorytmów i modeli takich jak:

1. **Teoria poszukiwania** (z ang. *Foraging Theory*)<sup>10</sup> – znana również jako zachowanie wyszukiwania pokarmu, jest dziedziną ekologii zajmującą się analizą zachowań zwierząt w kontekście interakcji z otoczeniem i zdobywania pokarmu. Jednym z kluczowych przykładów wykorzystania tej teorii jest pozostawianie feromonów przez mrówki, co zwiększa prawdopodobieństwo, że inne osobniki pójdą tą samą ścieżką. W kontekście algorytmów metaheurystycznych, teoria foragowania była inspiracją do stworzenia efektywnych rozwiązań optymalizacyjnych. Na przykład, inżynierowie Wagner, Lindenbaum i Bruckstein opracowali dwa algorytmy służące do przeszukiwania grafów, znane jako Edge Ant Walk (1996) oraz Verte Ant Walk (1998). W tych algorytmach, sztuczne mrówki podróżują po krawędziach grafu, pozostawiając feromony na odwiedzonych wierzchołkach. Następnie inne sztuczne mrówki wykorzystują te feromony do wyboru swojej ścieżki. Choć zasada działania tych algorytmów przypomina działanie ACO, różnią się one głównie implementacją. W kolejnych latach, algorytmy te zostały rozszerzone, umożliwiając ich użycie do rozwiązywania problemów w dynamicznie zmieniających się grafach.
2. **Podział pracy** (z ang. *Division of labor*)<sup>11</sup> – Podział pracy, zwany również specjalizacją, polega na przydziale określonych zadań i odpowiedzialności w celu zwiększenia efektywności pracy. W kontekście świata mrówek, można zaobserwować, że niektóre osobniki specjalizują się w konkretnych zadaniach. Jednak mrówki wykazują także elastyczność, ponieważ w zależności od sytu-

<sup>10</sup> J. FA Traniello, *Foraging strategies of ants*, "Annual review of entomology" 1989, 34(1), 191-210.

<sup>11</sup> F. Ravary, E. Lecoutey, G. Kaminski, N. Châline, P. Jaisson, *Individual experience alone can generate lasting division of labor in ants*, "Current Biology" 2007, 17(15), 1308-1312.



acji np. mrówka żołnierz może przejąć inne zadanie, co ilustruje możliwość wymienności ról. Różnorodność specjalizacji i elastyczność zachowań mrówek zachęca do wykorzystania podejścia wieloagentowego w optymalizacji i kontroli, umożliwiając dostosowywanie się do zmieniającego się otoczenia. Badaniami nad tym podejściem zajął się Robinson w 1992 roku, wprowadzając model wartości progowej. Według tego modelu, mrówki o niskim priorytecie zajmują się zadaniami na niższych poziomach, podczas gdy mrówki o wyższym priorytecie mogą pomóc w rozwiązaniu bardziej złożonych problemów, gdy zajdzie taka potrzeba. Analogicznie do rzeczywistości, mrówki o różnym priorytecie mogą współpracować w celu skutecznego rozwiązania trudniejszych zadań.

3. **Sortowanie strumieniowe** (z ang. *Brood sorting*)<sup>12</sup> – zwane również sortowaniem rodzinnym, polega na organizowaniu składników mrówki, takich jak larwy, jaja, kokony i jedzenie, w sposób scentralizowany, gdzie mniejsze elementy są trzymane w środku, a większe na obrzeżach. To zachowanie można zaobserwować u wielu gatunków mrówek, takich jak *Lep-tothorax unifaciat*, *Laskus Niger* czy *Pheidole pallidula*. Pierwszy model tego zachowania został zaproponowany przez Deneubourga w 1991 roku, w którym mrówka decyduje, gdzie umieścić lub przenieść jakąś rzecz, bazując na otaczających ją elementach. Na przykład, gdy mrówka niesie małe jajo, prawdopodobnie zostanie ono pozostawione w miejscu, gdzie znajduje się wiele innych małych jajeczek. Natomiast, jeśli znajdzie larwę otoczoną małymi jajkami, istnieje duże prawdopodobieństwo, że przeniesie larwę w inne miejsce. Model ten został zastosowany do problemu grupowania przez Lumera, Faieta (1994) oraz Kuntza, Snysersa, Layzella (1999). Algorytm polega na reprezentacji pewnej ilości punktów w  $n$ -wymiarowej przestrzeni i metryce  $d$ , która mierzy odległości między nimi. Sztuczni agenci w losowy sposób wybierają swoje drogi na płaszczyźnie, przenosząc i pozostawiając zadane rzeczy zgodnie z regułami zaproponowanymi przez Deneubourga. Rezultaty działania tych algorytmów są obiecujące i dorównują klasycznym metodom grupowania.

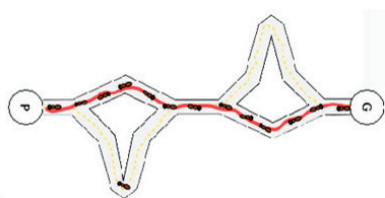
Optymalizacja kolonii mrówek, znana także jako Ant Colony Optimization (ACO), to algorytm opracowany przez Marco Dorigo, Gianni Di Caro i Luca Marius Gambardella w 1999 roku do rozwiązywania problemów dyskretnej optymalizacji<sup>13</sup>. Inspiracją dla ACO była teoria foragingu kolonii mrówek oraz obserwacje zachowań tych insektów w środowisku.

<sup>12</sup> L. Gaubert, P. Redou, F. Harrouet, J. Tisseau, *A first mathematical model of brood sorting by ants: Functional self-organization without swarm-intelligence*, "Ecological complexity" 2007, 4(4), 234-241.

<sup>13</sup> M. Dorigo, M. Birattari, T. Stutzle, *Ant colony optimization*, "IEEE computational intelligence magazine" 2006, 1(4), 28-39.

Algorytm ACO jest wykorzystywany zarówno do problemów optymalizacyjnych statycznych (np. TSP) jak i dynamicznych, gdzie informacje się zmieniają, takich jak problem wyznaczania trasy w zmieniającym się ruchu. Kluczowym elementem algorytmu jest ślad feromonowy, który umożliwia komunikację i współpracę między sztucznymi agentami przemieszczającymi się po grafie w poszukiwaniu optymalnych rozwiązań.

ACO unika konwergencji do lokalnych optymalnych rozwiązań poprzez parowanie feromonu jak pokazano na rysunku 1, co sprawia, że ścieżki o wysokiej jakości stają się preferowane przez kolejne mrówki. Algorytm ten przynosi obiecujące rezultaty w porównaniu z innymi technikami optymalizacyjnymi, takimi jak algorytmy genetyczne czy symulowane wyżarzanie. Jego efektywność została potwierdzona w rozwiązaniu problemu TSP, zbliżając się do optymalnego rozwiązania.



Rysunek 1. Efekt zanikania śladu feromonowego [źródło własne]

Powyższy pseudokod przedstawia ogólny schemat algorytmu ACO dla rozwiązania problemu TSP. Algorytm ten polega na iteracyjnym przeszukiwaniu grafu przez mrówki, które zgodnie z zasadami opartymi na feromonach i heurystyce wybierają kolejne trasy. Aktualizacja feromonów następuje na podstawie jakości tras, co pozwala na stopniowe poprawianie ścieżek i znalezienie optymalnego rozwiązania. Poniżej możemy zobaczyć pseudo kod algorytmu ACO:

#### Kod 1. Pseudo kod algorytmu ACO [źródło własne]

Inicjalizacja:

- Inicjalizuj graf  $G$  z wagami  $d(i, j)$  dla każdego wierzchołka  $i$  oraz  $j$
- Inicjalizuj feromony  $T(i, j)$  dla każdego sąsiedztwa  $(i, j)$  na grafie
- Ustal parametry: liczba mrówek, współczynnik parowania feromonu, współczynnik wpływu feromonu ( $\alpha$ ), współczynnik wpływu heurystyki ( $\beta$ )

Iteracje:

Dla każdej iteracji:

Dla każdej mrówki:

Inicjalizuj trasę mrówki

Powtarzaj, dopóki nie odwiedzi wszystkich wierzchołków:

Wybierz następne przejście na podstawie prawdopodobieństwa zależnego od feromonu i heurystyki

Zaktualizuj trasę mrówki i feromon na przejściu

Zaktualizuj feromony na wszystkich przejściach

Do najważniejszych wzorów które opisują zagadnienie systemów mrówkowych należą:

**Wybór ścieżki** – mrówka przebędzie drogę z punktu  $i$  do punktu  $j$  z prawdopodobieństwem równym:

$$p_{ij} = \frac{(\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)}{\sum (\tau_{i,j}^\alpha)(\eta_{i,j}^\beta)} \quad (2)$$

gdzie

$\tau_{i,j}$  - ilość feromonu na ścieżce  $i,j$ ,       $\eta_{i,j}$  - określa atrakcyjność ścieżki  $i,j$ ,  
 $\alpha$  - parametr do kontroli wpływu  $\tau_{i,j}$ ,       $\beta$  - parametr do kontroli wpływu  $\eta_{i,j}$ .

**Uaktualnienie feromonu** – to zagadnienie reprezentowane jest wzorem

$$\tau_{i,j} = p\tau_{i,j} + \Delta\tau_{i,j} \quad (3)$$

gdzie

$\tau_{i,j}$  - ilość feromonu na ścieżce  $i,j$ ,       $\Delta\tau_{i,j}$  - oznacza ilość feromonu pozostawionego, określane wzorem  
 $p$  – skala parowania feromonu,

$$\Delta\tau_{i,j} = \begin{cases} 1/L_k & \text{Jeśli } k \text{ mrówek przemierza ścieżkę } i,j \\ 0 & \text{Inny przypadek} \end{cases} \quad (4)$$

gdzie  $L_k$  jest kosztem podróży  $k$ -ej mrówki.

## Logika rozmyta, skierowane liczby rozmyte OFN

Teoria przedstawiona w publikacji autorstwa Zadeha<sup>14</sup> ewoluowała w narzędzie matematyczne, umożliwiające rozwiązywanie różnych dylematów. Wykorzystanie pojedynczego zbioru rozmytego, reprezentowanego jako stopień przynależności elementu  $x$  do zbioru  $A$ , pozwala na opisywanie tego zbioru jako tzw. zbioru rozmytego.

$$A = \{(x, \mu_A(x)), x \in X\} \quad (5)$$

<sup>14</sup> L.A. Zadeh, *Fuzzy sets*, "Information and Control" 1965, 8(3), 338-353.

W kontekście badań nad logiką rozmytą, istotnym postępowaniem było zdefiniowanie nowej kategorii tzw. klas przedziałowych liczb rozmytych typu L-R. To podejście zainicjowane przez Dubois i Prade stanowiło istotny krok naprzód w rozwoju tej dziedziny<sup>15</sup>.

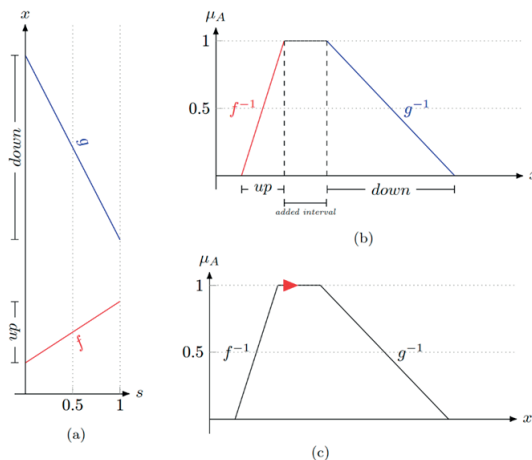
$$\mu_X(x) = \begin{cases} L(\frac{m-x}{m-a}) & \text{dla } x < m \\ 1 & \text{dla } m \leq x < \bar{m} \\ R(\frac{x-\bar{m}}{b-\bar{m}}) & \text{dla } x > \bar{m} \end{cases} \quad (6)$$

Staje się jasne, że wprowadzenie teorii logiki rozmytej stanowiło przełom w postrzeganiu klasycznej logiki, która ograniczała się do przyjmowania wartości 0 lub 1. Podobnym momentem przełomowym jest rozwój teorii skierowanych liczb rozmytych OFN<sup>16</sup>. Jednym z kluczowych czynników, które pociągnęły za sobą zmianę klasycznej teorii zbiorów rozmytych, był brak spójności w wykonywaniu operacji arytmetycznych na liczbach rozmytych, co stoi w kontraście do działań na liczbach rzeczywistych. Przekształcenie klasycznego zbioru rozmytego w uporządkowaną parę funkcji rozwiązało problem arytmetyki liczb rozmytych. W związku z tym, liczby OFN definiujemy w następujący sposób:

**Definicja 1.** Skierowaną liczbę rozmytą nazywamy uporządkowaną parę funkcji

$$A = (f_A, g_A) \quad (7)$$

Interpretacja skierowanej liczby rozmytej można rozpatrywać w oparciu o poniższą ilustrację.



**Rysunek 5.** Interpretacja skierowanej liczby rozmytej [źródło własne]

<sup>15</sup> D. Dubois, H. Prade, *Operations on fuzzy numbers*, "International Journal of systems science" 1978, 9(6), 613-626.

<sup>16</sup> W. Kosinski, P. Prokopowicz, D. Slézak, *Ordered fuzzy numbers*, "Bulletin of the Polish Academy of Sciences" 2003, 51(3), 327-338.

Na której mamy typowe odwzorowanie dla terminologii OFN (a) oraz przejście do wypukłych liczb rozmytych jak pokazano na rysunku 5 (b,c). W przypadku gdy porządek zaczyna się od funkcji  $f$  a kończy się na  $g$ , to mówimy o pozytywnym skierowaniu.

Działania arytmetyczne dla skierowanych liczb rozmytych wyglądają następująco. Dla  $A = (f_A, g_A), B = (f_B, g_B), C = (f_C, g_C)$  mamy wówczas:

$$f_C(x) = f_A(x) * f_B(x) \wedge g_C(x) = g_A(x) * g_B(x) \quad (8)$$

gdzie symbol  $*$  zastępuje odpowiednio  $+$ ,  $-$ ,  $\cdot$ ,  $\div$ , działanie dzielenia  $C = A \div B$  zdefiniowane jest przy założeniu, że liczba  $B$  nie zawiera zera (tj.  $|f_B| > 0$  oraz  $|g_B| > 0$ ).

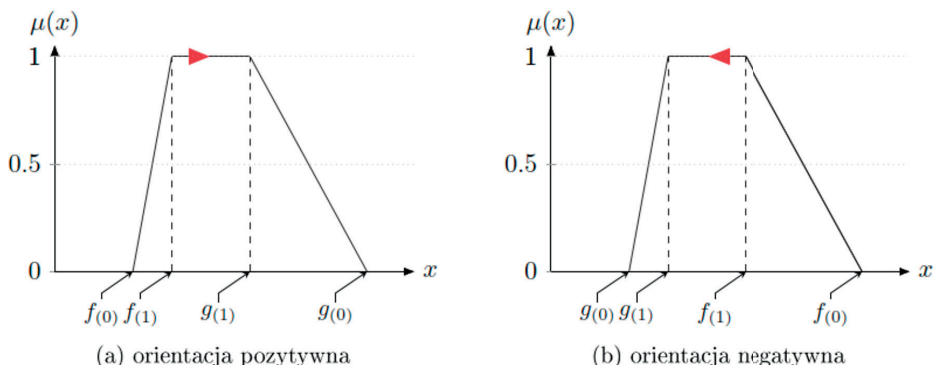
Tak określone działania są analogiczne do działań na liczbach rzeczywistych, w szczególności dla dowolnej skierowanej liczby rozmytej mamy:

$$A - A = A + (-1) \cdot A = 0 \quad (9)$$

oraz

$$A \cdot A^{-1} = A \cdot \frac{1}{A} = 1 \quad (10)$$

Wykorzystując powyższe cechy skierowanych liczb rozmytych możemy z łatwością obejść ograniczenia klasycznych wypukłych zbiorów rozmytych. Tym samym operacje algebraiczne są zgodne z liczbami rzeczywistymi, wyróżniają się rozdzielnością mnożenia względem dodawania, przemiennością dodawania, istnieniem elementów neutralnych względem dodawania i mnożenia, i tym samym spełniają własności algebry liczb rzeczywistych.



Rysunek 6. Skierowana liczba rozmyta w dwóch wariantach orientacji wraz z punktami charakterystycznymi [źródło własne]

Inne istotne aspekty przedstawione czytelnikowi w tym artykule dotyczą dyskusji na temat samego procesu defuzyfikacji. Ostra wartość wynikająca z defuzyfikacji zbioru rozmytego, będąca wnioskiem, pozwala na kontrolę nad określonym procesem<sup>17</sup>. Ta operacja jest przeprowadzana dzięki komponentowi defuzyfikacji<sup>18</sup>, która wykorzystuje specyficzną metodę<sup>19</sup>. W związku z tym, mechanizm defuzyfikacji można określić jako:

$$W = \{f: X \rightarrow [0,1]\} \rightarrow X \quad (11)$$

gdzie: operator wyostrzenia, funkcja przynależności oraz zbiór uniwersum, na którym określone są funkcje przynależności.

W treści tego opracowania metoda wyostrzania jest zamiennie nazywana jako funkcjonal wyostrzania. W literaturze znajdujemy wiele sposobów wyznaczania ostrej wartości w procesie wyostrzania<sup>20</sup>. Różnorodność form uzyskiwania ostrej wartości skupia się w kilku głównych kategoriach tj. klasie metod maksimum /maxima methods/ Random Choice Of Maxima (RCOM), First Of Maxima (FOM), Last Of Maxima (LOM), Middle Of Maxima (MOM) – wybór ostrej wartości dokonuje się z jądra zbioru rozmytego; klasie metod rozkładu /distribution methods/ Centre Of Gravity (COG), Mean Of Maxima (MOM), Basic Defuzzification Distribution (BADD), Generalized Level Set Defuzzification (GLSD), Indexed Centre of Gravity (ICOG), Fuzzy Mean (FM) – przekształceniu funkcji przynależności do probability distribution oraz powierzchni /area methods/ Centre of Area (COA) – podział pola pod krzywą na części oraz innych jak GR (Golden Ratio)<sup>21</sup>, GROA (Golden Ratio of Area)<sup>22</sup>.

<sup>17</sup> T. J. Ross et al. *Fuzzy logic with engineering applications*, volume 2. Wiley Online Library, 2004.

<sup>18</sup> W. Kosinski. *On defuzzification of ordered fuzzy numbers*, in: Leszek Rutkowski, Ryszard Tadeusiewicz, and Lotfi A. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, "Lecture Notes in Computer Science", vol. 3070, pages 326-331.

<sup>19</sup> W. T. Dobrosielski, J. M. Czerniak, J. Szczepański, H. Zarzycki, *Two new defuzzification methods useful for different fuzzy arithmetics*, in: *International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets*, Springer 2016, pages 83-101.

<sup>20</sup> R. R. Yager, D. Filev, *On the issue of defuzzification and selection based on a fuzzy set*, "Fuzzy sets and Systems" 1993, 55(3), 255-271.

<sup>21</sup> W. T. Dobrosielski, J. Szczepański, H. Zarzycki, *A proposal for a method of defuzzification based on the golden ratio—gr*, in: K. T. Atanassov, O. Castillo, J. Kacprzyk, M. Krawczak, P. Melin, S. Sotirov, E. Sotirova, E. Szmidt, G. De Tré, S. Zadrożny, editors, *Novel Developments in Uncertainty Representation and Processing*, Cham 2016, pages 75-84.

<sup>22</sup> W. T. Dobrosielski, *The golden ratio of area method based on fuzzy number area as a defuzzifier*, in: K. T. Atanassov, V. Atanassova, J. Kacprzyk, A. Kauszko, M. Krawczak, J. W. Owsinski, S. S. Sotirov, E. Sotirova, E. Szmidt, S. Zadrożny, editors, *Uncertainty and Imprecision in Decision Making and Decision Support: New Advances, Challenges, and Perspectives*, Cham 2022, pages 92-108.

Przedstawione omówienie działań arytmetycznych, metod wyostrzania dla skierowanych liczb rozmytych przybliża czytelnika w rozwiązywaniu optymalizacji problemu TSP przy pomocy kolonii mrówek OFN.

## Systemy mrówkowe a problem komiwojażera

Badania nad rozwiązaniem problemu TSP przyniosły wiele różnych wyników, które znalazły się w obszernej literaturze. Optymalizacja kolonią mrówek w przypadku TSP okazała się szczególnie użyteczna z uwagi na:

1. TSP jako problem NP-trudny ma szerokie zastosowanie,
2. W dosyć łatwy sposób można zaimplementować algorytmy ACO do tego problemu,
3. Kod jest łatwy do zrozumienia, dzięki TSP samo serce ACO nie jest zasłaniane dodatkowymi technicznymi sprawami, które mogłyby skomplikować sam kod i zmniejszyć przejrzystość ACO.

Początkowo wyniki testów AS na bazie TSP nie były zbyt obiecujące, jednak po upływie pewnego czasu sama filozofia wykorzystania mrówek zaczęła się rozwijać i powstawały kolejne rozszerzenia AS, mam tu na myśli m.in. EAS, MMAS, czy ASrank. Wszystkie te rozwinięcia pierwszego bazowego AS, spowodowały bardzo duże usprawnienie systemu i wzrost jego wydajności. Na chwilę obecną algorytmy ACO uznawane są za jedno z najbardziej skutecznych, efektywnych algorytmów zajmujących się problemami optymalizacyjnymi, które rozwiązują problem TSP, oraz wiele innych trudnych problemów (patrz 3.4). Jeśli chodzi o różnice pomiędzy bazowym AS a rozszerzeniami to dotyczą one sposobu uaktualniania śladu feromonowego, jak i również pewne dodatkowe rzeczy związane z zarządzaniem tymi śladami. Oprócz algorytmów, rozszerzeń AS, które nie wniosły dużych zmian, w późniejszym okresie zaproponowano również algorytmy, które wprowadziły większe zmiany. Do tych algorytmów możemy zaliczyć systemy kolonii mrówek (ACS), Ant-Q, ANTS (z ang. Approximate Nondeterministic Tree Search) wykorzystujące techniki dolnej granicy w programowaniu, czy środowisko hyper-cube dla ACO. Poniżej można zobaczyć okresy, w jakich poszczególne systemy były prezentowane.

Tabela 2. Algorytmy ACO w chronologicznej kolejności ich pojawienia się

Algorytmy ACO	Używany do TSP	Stworzony
System Mrówkowy AS	TAK	Dorigo 1992
Elitarny System Mrówkowy	TAK	Dorigo 1992
Ant-Q	TAK	Gambardella i Dorigo 1995-96

System Kolonii Mrówek ACS	TAK	Dorigo i Gambardella 1997
Max-Min AS	TAK	Stützle i Hoos 1996, 2000
System oceny AS <sub>rank</sub>	TAK	Bullnheimer, Hartl i Strauss 1997-99
ANTS	NIE	Maniezo 1999
Hyoeer Cube AS	NIE	Blum Roli i Dorigo 2001, Blum i Dorigo 2004

Szukając zależności pomiędzy tymi systemami można zauważyć, że ACS jest oparte na systemie Ant-Q, wczesnym algorytmie zaproponowany przez Gambardella i Dorigo w 1995 roku. Jedną rzeczą, jaką różni ACS od Ant-Q jest definicja parametru  $\tau_0$ , który dla Ant-Q ma postać  $\tau_0 = \gamma \max_{j \in N_i^k} \{\tau_j\}$ . Tego typu zmiana została zainspirowana przez *Q-learning*, który jest dobrze znanym algorytmem wzmocnionego uczenia. Podobieństwo można również zauważyć w algorytmach ACS i MMAS. Obydwa algorytmy używają limitów dla feromonu, gdzie dla MMAS jest on wyraźnie określony, a w ACS jest on parametrem domyślny. Obowiązkowo należy wspomnieć o tym, iż system ACS, był pierwszym algorytmem, który zastosował tzw. „listę kandydatów”. Lista była stosowana, aby ograniczyć liczbę możliwych wyborów, które są rozważane na początku tworzenia kolejnego kroku. Uogólniając, jest to lista zawierająca najlepiej ocenione decyzje na bazie określonych heurystycznych kryteriów. Patrząc z punktu widzenia TSP „lista kandydatów” zawiera dla każdego miasta  $i$  te miasta  $j$ , która są w bliskiej odległości. Lista oprócz informacji o tym, jakie miasta  $j$  znajdują się w bliskiej odległości, posiada mechanizm oznaczania odwiedzonych miasta, tak, aby mrówka wiedziała, do których miast z pośród wszystkich może się udać.

Stwierdzono, iż wprowadzenie listy kandydatów spowodowało zwiększenie wydajności samego algorytmu. Jeszcze jedną rzeczą, o której należy wspomnieć jest zastosowanie algorytmów lokalnego przeszukiwania, które w znaczący sposób przyspieszyły jego działanie i pozwoliły na osiągnięcie lepszych rezultatów w rozwiązaniach TSP. Do stosowanych algorytmów zaliczane są: 2-opt, 2,5-opt jak i najskuteczniejszy 3-opt. W książce pt. „*Ant Colony Optimization*” autorstwa Dorigo i Stützle można znaleźć wiele badań dotyczących tego zagadnienia, pokazujących skuteczność wszystkich trzech sposobów przeszukiwania w oparciu o dwa wybrane problemy z bazy TSPLIB.

Tak jak można zauważyć TSP był pierwszym problemem kombinatorycznej optymalizacji, który posłużył do sprawdzania skuteczności funkcjonowania algorytmów ACO<sup>23</sup>. Pierwszym testowanym algorytmem był AS, który uzyskał dobre rezultaty, ale tylko dla problemów TSP o małej wielkości. Przy problemach

<sup>23</sup> M. Dorigo, L. M. Gambardella, *Ant colony system: a cooperative learning approach to the traveling salesman problem*, “IEEE Transactions on evolutionary computation” 1997, 1(1), 53-66.



o większej ilości miast wyniki były dosyć słabe. Głównym celem zastosowania AS dla TSP było potwierdzenie przez Dorigo koncepcji zastosowania tego typu algorytmów do rozwiązywania problemów optymalizacyjnych. Na chwilę obecną dostępna jest dosyć duża gama algorytmów ACO. Można powiedzieć, iż te algorytmy wniosły duży wkład w rozwój rozwiązywania trudnych problemów i osiągnęły bardzo dobre rezultaty niejednokrotnie znajdując najlepsze rozwiązania danego problemu.

## Rozwiązanie systemu mrówkowego OFN w problemie komiwożera

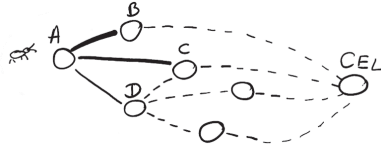
Zupełnie innym podejściem do rozwiązania problemu optymalizacji problemu komiwożera będzie użycie kolonii mrówek bazującej na skierowanych liczbach rozmytych. W przeciwieństwie do klasycznych rozwiązań logiki rozmytej<sup>24</sup>, skierowana liczba rozmyta pozwala płynnie przeprowadzać działania arytmetyczne w logice rozmytej oraz zawierają informację o trendzie zmiennej lingwistycznej. W wielu przypadkach własność ta jest tożsama z terminem tendencji, trendu danego zjawiska. Właściwym przykładem omawiającym skierowanie dla naszego przypadku będzie ocena ilości feromonu pozostawionego przez mrówkę na krawędzi danej drogi. Odpowiednio możemy powiedzieć, że ilość feromonu „około 30 jednostek” ma tendencję wzrostową lub malejącą. W interpretacji rozwiązania będzie chodziło nam o atrakcyjność danej ścieżki dla poszczególnej mrówki, a w ogólnym ujęciu wyznaczało optymalne rozwiązanie.

Przedstawiając powyższe zagadnienie optymalizacji kolonią mrówek OFN, zaznaczamy, że działanie tradycyjnego algorytmu ACO nie zostaje zmienione i poszczególne etapy są zachowane. Jak wiadomo w pierwszym kroku, mamy do czynienia z inicjacją parametrów algorytmu. Co oznacza, że przypisujemy skierowane liczby do krawędzi łączących poszczególne węzły grafu. W niektórych przypadkach mogą to być liczby OFN typu singleton. Przechodząc do kolejnego kroku, jesteśmy w sytuacji, w której jest generowana kolonia mrówek. Spełniając warunki początkowe algorytmu, gdzie mamy wstępnie zainicjowane ścieżki oraz wygenerowaną kolonię mrówek przechodzimy do typowego zachowania mrówki. Omówienie tego punktu przedstawimy na kluczowych zachowaniach mrówki.

Jak wiadomo mrówka kierując się śladem feromonowym wybiera drogę, na której jest jego więcej. Powyższą sytuację przedstawia rysunek 7, na którym gru-

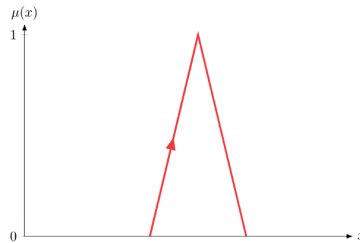
<sup>24</sup> O. Castillo, H. Neyoy, J. Soria, P. Melin, F. Valdez, *A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot*, "Applied soft computing" 2015, 28, 150-159; E. Amiri, H. Keshavarz, M. Alizadeh, M. Zamani, T. Khodadadi, *Energy efficient routing in wireless sensor networks based on fuzzy ant colony optimization*, "International Journal of Distributed Sensor Networks" 2014, 10(7),768936.

bość linii to odpowiednio ilość feromonu na danej ścieżce. W przypadku drogi między węzłem AB śladu jest najwięcej w stosunku do pozostałych możliwych przejść.

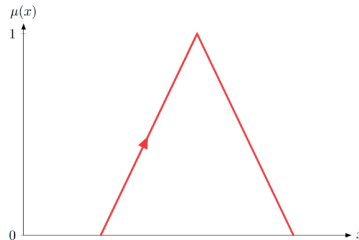


Rysunek 7. Wybór trasy przez mrówkę [źródło własne]

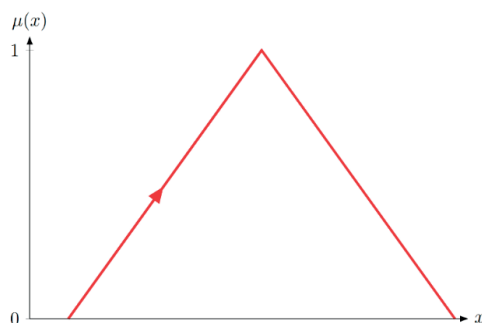
Wyjaśniając znaczenie ilość śladu feromonowego na tematykę skierowanych liczb rozmytych, mamy odpowiednio trzy przypadki liczby rozmytej OFN. Dla węzła AB jest to liczba przedstawiona na rysunkach 8, 9, 10. Poszczególne warianty liczby dotyczą procesu budowania śladu feromonowego. Słusznie możemy zauważyć, że w tym rozwiązaniu mamy do czynienia z poszerzaniem nośnika liczby rozmytej. Przyjęcie takiego sposobu budowy informacji o feromonie, w znaczący sposób upraszcza działanie algorytmu. Inaczej mówiąc, mrówka przechodzą ścieżką powiększa liczbę OFN.



Rysunek 8. Liczba  $\tau_{AB}$  w chwili inicjacji algorytmu antOFN [źródło własne]

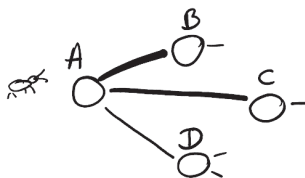


Rysunek 9. Liczba  $\tau_{AC}$  w kolejnej chwili działania algorytmu antOFN [źródło własne]



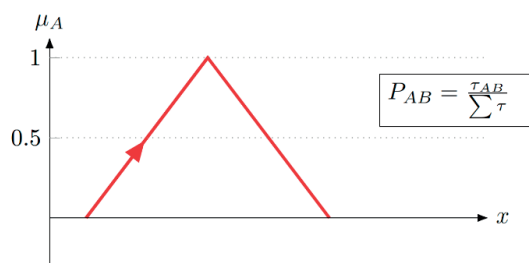
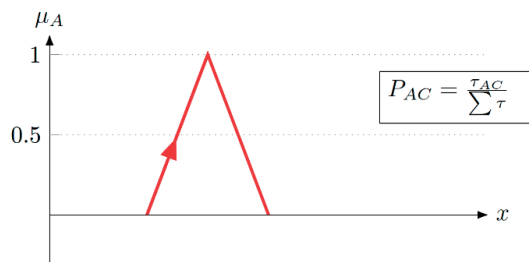
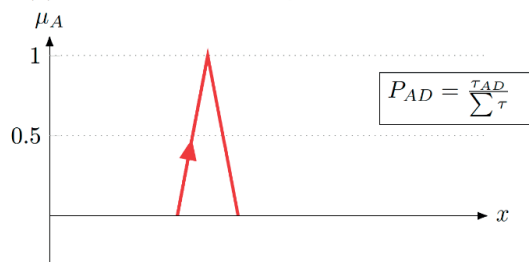
Rysunek 10. Liczba  $\tau_{AD}$  w kolejnej chwili działania algorytmu antOFN [źródło własne]

Znając budowę ścieżki w interpretacji skierowanych liczb rozmytych, przejdziemy kolejno do omówienia scenariusza wyboru trasy przez mrówkę. W działaniu algorytmu ACO mrówka stoi przed wyborem atrakcyjnej ścieżki, która pozwoli na optymalne rozwiązanie danego problemu. Jedną z wielu możliwych sytuacji została zainscenizowana na rysunku 11, gdzie węzeł AB jest reprezentowany przez liczbę OFN, feromon węzła AC jest opisany jako  $\tau_{AC}$ , z kolei połączenie AD to liczba  $\tau_{AD}$ .



Rysunek 11. Mrówka przed wyborem trasy [źródło własne]

W trakcie kolejnych iteracji algorytmu antOFN dla przedstawionego przykładu wyboru trasy przez mrówkę, jak podaje ilustracja 11. Nasze liczby OFN przedstawiają się jak ujęto to na rysunku 12.

(a) Skierowana liczba rozmyta  $\tau_{AB}$  dla ścieżki AB(b) Skierowana liczba rozmyta  $\tau_{AC}$  dla ścieżki AC(c) Skierowana liczba rozmyta  $\tau_{AD}$  dla ścieżki AD

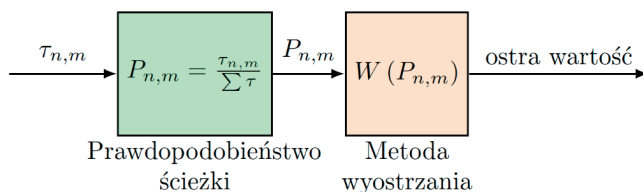
Rysunek 12. Interpretacja ilości feromonu dla poszczególnych możliwości wyboru ścieżki przez mrówkę [źródło własne]

Rzeczą istotną jest to, iż dla poszczególnych liczb OFN produkujemy kolejną liczbę skierowaną związaną z prawdopodobieństwem wyboru danej ścieżki. W tym przykładzie jest to stosunek ilości feromonu do sumy feromonu wszystkich możliwych dróg. Ogólna formuła przedstawia się tak jak to pokazano we wzorze 5,

$$P_{m,n} = \frac{\tau_{m,n}}{\sum \tau} \quad (12)$$

gdzie  $P_{m,n}$  to prawdopodobieństwo danej ścieżki, ilość feromonu danej ścieżki oraz  $\tau$  to suma feromonu wszystkich ścieżek jakie są dostępne z danego węzła dla mrówki.

Obliczone prawdopodobieństwo dla każdej możliwej ścieżki, wyrażone jako skierowana liczba rozmyta OFN jest właściwą informacją dla mrówki, podejmującej decyzję o wyborze drogi do kolejnego węzła. Z uwagi na fakt, że jest to liczba skierowana OFN, proponowany algorytm wymaga przeprowadzenia dodatkowego działania jakim jest przekształcenie do wartości ostrej. Termin jest bezpośrednio związany z sterownikiem rozmytym, gdzie z wynikowej funkcji przynależności, otrzymujemy ostrą wartość. W wolnej interpretacji jest to wartość, która bezpośrednio steruje urządzeniem wykonawczym. Opisany proces jest w literaturze nazywany procesem wyostrzania. Dla celów rozwiązania antOFN, wyostrzanie obliczamy z prawdopodobieństwa danej ścieżki  $P_{n,m}$  wraz z metodą wyostrzania jednoznacznie wyznacza atrakcyjność ścieżki. Opisaną sytuację odzwierciedla rysunek 13 z zaznaczonym blokiem wyostrzania, która została przedstawiona w rozdziale o skierowanych liczbach rozmytych.



Rysunek 13. Proces obliczenia wartości ostrej dla dowolnej ścieżki [źródło własne]

W dalszym etapie, mrówka podejmuje decyzję o wyborze ścieżki z zbioru dostępnych dróg. Zbiór potencjalnych rozwiązań to możliwe wybory jaki mogą zostać podjęte. Formalnie decyzja o wyborze drogi zapiszemy jako formuła 6, zbiór wyostrzeń z poszczególnych prawdopodobieństw danej ścieżki z którego wybieramy największe prawdopodobieństwo za pomocą operatora  $\max$ . Dla innych rozwiązań można zastosować dowolne konstrukcję wyboru elementu z zbioru.

$$\max(W(P_{AB}), W(P_{AC}), W(P_{AD})) \quad (13)$$

Przedstawiona zasada działania algorytmu antOFN została zaimplementowana jako oprogramowanie, wspomagające przeprowadzenie eksperymentu do porównania się z innymi algorytmami.



Rysunek 14. Oprogramowania antOFN realizujące proces optymalizacji problemu TSP w oparciu o skierowaną kolonię mrówek [źródło własne]

## Badanie efektywności działania algorytmów mrówkowych w porównaniu do algorytmów klasycznych w rozwiązaniu TSP

W tym paragrafie zostanie dokonane porównanie efektywności oprogramowania ACOTSP, CONCORE oraz antOFN, jako dwóch programów mogących poradzić sobie z problemem NP – trudnym, jakim jest TSP. Będzie to również konfrontacja algorytmów klasycznych z całkowicie nowym podejściem, jakim jest naśladowanie kolonii mrówek do rozwiązania problemów optymalizacyjnych za pomocą skierowanych liczb rozmytych OFN.

Dzięki ustalonym parametrom możemy przejść do porównania algorytmów klasycznych z algorytmami mrówkowymi. Ostatnią rzeczą, jaką nam pozostaje ustalić jest wybór problemów z biblioteki TSPLIB oraz ustalenie ilości tych problemów. Ustalam, iż 10 wybranych problemów pozwoli w znaczący sposób zbadać efektywności oby dwóch typów algorytmów. Będą one badane według poniższych zasad:

1. Jeśli chodzi o ACOTSP, to dla każdego problemu program będzie uruchamiany trzykrotnie dla  $r = 10$ , natomiast CONCORDE trzykrotnie bez dodatkowych parametrów,
2. Skuteczność danego algorytmu będzie oceniana następująco:
  - przez podanie wyniku otrzymanej długości trasy,
  - procentowo w postaci: optimum spełnione w  $x\%$ , tak jak to zostało przedstawione w tabeli numer 8,
3. Dla każdego z 10 problemów, przedstawiony zostanie wykres ilustrujący efektywność poszczególnych algorytmów,
4. Każdy taki wykres będzie posiadał krótkie podsumowanie omawiające otrzymane wyniki,
5. Po przedstawieniu wszystkich wyników, sporządzony zostanie ogólny wykres efektywności wszystkich algorytmów. Dla każdego algorytmu zsumowana zostanie wartość „optimum spełnione w  $x\%$ ”, przez co otrzymamy całkowitą ilość punktów na 1000 możliwych. Ten sposób prezentacji danych pomoże w łatwy sposób ocenić, jaka jest hierarchia wszystkich testowanych algorytmów na przekroju 10 zbadanych próbek.

Aplikacja CONCORDE została opracowana do rozwiązywania problemów typu Symetryczne TSP i innych problemów optymalizacji sieciowej. Jest ona wspierana przez Biuro Badań Marynarki Wojennej, Międzynarodową Fundację Badań Naukowych oraz Szkołę Systemów Informatycznych z Zastosowaniem w Przemysle na Georgia Tech. Program obsługuje różne formaty plików, w tym \*.tsp z biblioteki TSPLIB. W aplikacji znajdują się różne algorytmy tworzące krawędzie używane do znalezienia optymalnych rozwiązań, takie jak:

1. Delaunay Triangulation,
2. Minimum Spanning Tree,
3. Różne odmiany Nearest Neighbours

Program zawiera również kilka algorytmów heurystycznych obliczających problem TSP. Do tych algorytmów zaliczamy:

1. Algorytm zachłanny (ang. Greedy algorithm, GR)
2. Algorytmy Boruvka (BOR) oraz Qucik Boruvka (QBOR),
3. Algorytm najbliższego sąsiada (ang. Nearest neighbour algorithm, NN),
4. Algorytm Lin-Keringhana (LK)

## Dane eksperymentalne

Poniżej w tabeli zostało przedstawione 10 problemów, które zostaną podane testom wraz z oczekiwaną wartością optimum, która będzie docelowa dla testowanych algorytmów.

Tabela 3. Lista rozpatrywanych problemów wraz z podanym optimum [źródło własne]

Lp	Oznaczenie problemu	Optimum	Opis	Twórca
1	Eil51	426	Problem dla 51 miast	Christofides / Eilon
2	D198	15780	Reprezentuje problem wywierconych dziur w materiale (z ang. <i>Dribling Problem</i> ). Wielkość problemu 198 dziur	Reinelt
3	Gil262	2378	Problem dla 262 miast	Gillet/Johnson
4	Lin318	42029	Problem dla 318 miast	Lin / Kernighan
5	Pcb442	50778	Reprezentuje problem wywierconych dziur w materiale (z ang. <i>Dribling Problem</i> ). Wielkość problemu 442 dziury	Groetschel/Juenger/Reinelt
6	Rat783	8806	Problem połączonej sieci elektrycznej dla 783 punktów	Pulleyblank
7	Pcb1173	56892	Reprezentuje problem wywierconych dziur w materiale (z ang. <i>Dribling Problem</i> ). Wielkość problemu 1173 dziur	Juenger/Reinelt
8	D1291	50801	Reprezentuje problem wywierconych dziur w materiale (z ang. <i>Dribling Problem</i> ). Wielkość problemu 1291 dziur	Reinelt
9	Nrw1379	56638	Problem 1379 miejscowości w Nadreni Północna-Westfalia	Bachem/Wottawa
10	Pr2392	378032	Problem 2392 miast	Padberg/Rinaldi

## Rezultaty eksperymentu

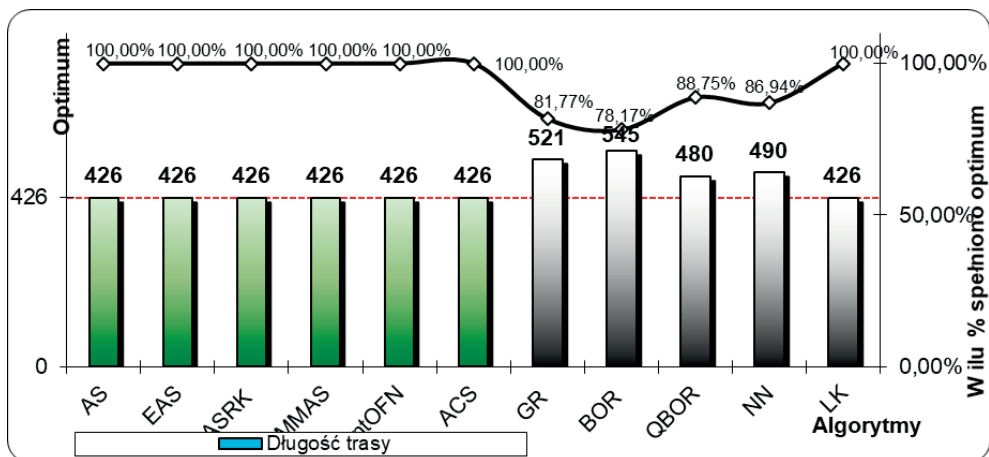
Zostało przeprowadzonych szereg testów zgodnie z ustalonymi powyżej zasadami oraz używając wybranych dziesięciu problemów. Wyniki poszczególnych testów zostały przedstawione poniżej wraz z wykresem oraz krótką notatką do każdego z nich.

Tabela 4. Wyniki dla poszczególnych algorytmów w rozwiązaniu problemu TSP [źródło własne]

Zbiór danych		ACOTSP				CONCORDE		
*.tsp file	optimum	AS	ASRK	antOFN	ACS	GR	QBOR	LK
<b>eil51</b>	<b>426</b>	426	426	426	426	521	480	426
%		100,00%	100,00%	100,00%	100,00%	81,77%	88,75%	100,00%
<b>d198</b>	<b>15780</b>	15781	15780	15780	15780	18399	18140	15828
%		99,99%	100,00%	100,00%	100,00%	85,77%	86,99%	99,70%
<b>gil262</b>	<b>2378</b>	2380	2378	2378	2378	2846	2818	2380
%		99,92%	100,00%	100,00%	100,00%	83,56%	84,39%	99,92%

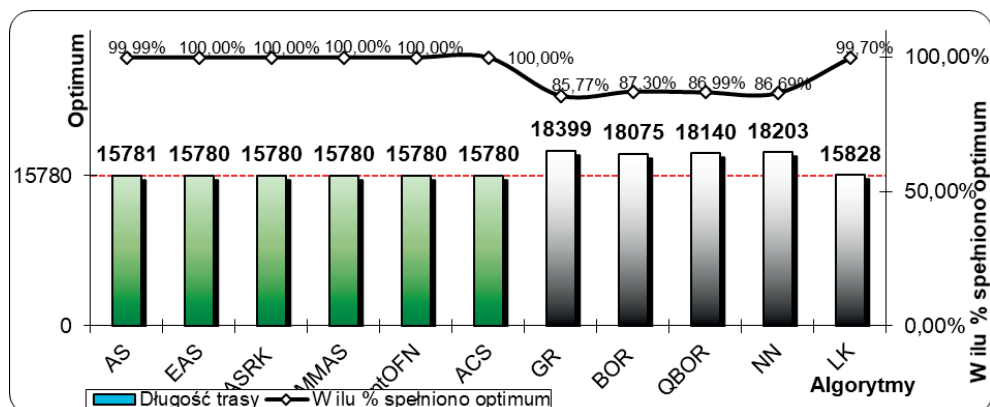


<b>lin318</b>	<b>42029</b>	42091	42029	42029	42029	49744	54090	42272
%		99,85%	100,00%	100,00%	100,00%	84,49%	77,70%	99,43%
<b>pcb442</b>	<b>50778</b>	50964	50883	50778	50778	61891	58695	51071
%		99,64%	99,79%	100,00%	100,00%	82,04%	86,51%	99,43%
<b>rat783</b>	<b>8806</b>	8833	8812	8808	8806	10294	10402	8831
%		99,69%	99,93%	99,98%	100,00%	85,54%	84,66%	99,72%
<b>pcb1173</b>	<b>56892</b>	57612	56950	57040	56897	65829	66493	57063
%		98,75%	99,90%	99,74%	99,99%	86,42%	85,56%	99,70%
<b>d1291</b>	<b>50801</b>	51020	50824	50870	50820	59293	57228	52729
%		99,57%	99,95%	99,86%	99,96%	85,68%	88,77%	96,34%
<b>nrw1379</b>	<b>56638</b>	57281	56859	56917	56770	66371	66110	56756
%		98,88%	99,61%	99,51%	99,77%	85,34%	85,67%	99,79%
<b>pr2392</b>	<b>378032</b>	386541	382089	381077	379602	444853	448641	383277
%		97,80%	98,94%	99,20%	99,59%	84,98%	84,26%	98,63%



Wykres 1. Wykres efektywności algorytmów dla problemu eil51 [źródło własne]

**eil51** Powyżej przedstawiono eksperyment dla 51 miast, dla którego optimum jest równe 426. Zgodnie z obliczeniami 7 algorytmów na 10 otrzymało wynik optymalny. Co warto zauważyć tylko jeden algorytm klasyczny otrzymał wynik najlepszy. Pozostałe heurystyki wykazały się słabą skutecznością przy stosunkowo niewielkim problemie. Wszystkie algorytmy mrówkowe doskonale poradziły sobie z powyższym problemem.



Wykres 2. Wykres efektywności algorytmów dla problemu d198 [źródło własne]

**d198** Kolejny przykład testowy, tym razem dla 198 miast, powtórnie pokazał przewagę algorytmów mrówkowych nad klasycznymi. W tym przypadku wszystkie 6 metod mrówkowych znalazło optimum, przy czym AS uzyskał wynik gorszy o jedną setną procenta od optimum. Najlepszy algorytm klasyczny, czyli Lin-Kernighan spełnił optimum w 99,70%, co nie jest złym wynikiem. Pozostałe algorytmy spełniły optimum w zaledwie 88%.

## Podsumowanie

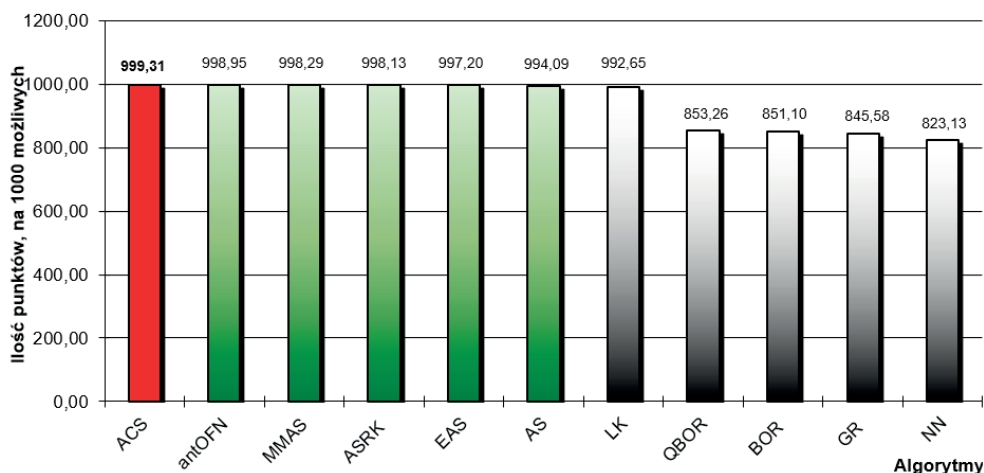
Po przeprowadzeniu serii eksperymentów zgodnie z wcześniej ustaloną metodologią, można jednoznacznie stwierdzić przewagę algorytmów mrówkowych nad tradycyjnymi metodami w większości testowanych przypadków. Z dziesięciu benchmarków tylko algorytm Lin-Kernighan (LK) osiągnął lepsze wyniki w jednym przypadku, co mogło być spowodowane specyfiką problemu nrw1379. W pozostałych przypadkach algorytm LK był minimalnie skuteczniejszy od starszych algorytmów mrówkowych, takich jak AS i EAS. Klasyczne metody optymalizacji zajęły niższe pozycje, osiągając wyniki od 79% do 89%, co nie jest najlepszym rezultatem. Zauważalny jest trend pogarszania się wyników wraz ze wzrostem rozmiaru problemu, potwierdzony szczególnie przez testy dla problemu pcb1173.

Podsumowując, wykres prezentujący wyniki wszystkich testów jasno wskazuje, że algorytmy mrówkowe osiągnęły najlepsze rezultaty, zwłaszcza antOFN, który osiągnął 998,95 punktu. Algorytmy mrówkowe reprezentują nową generację skutecznych algorytmów optymalizacji, które sprawdziły się w wielu dziedzinach, nie tylko w problemie TSP. Metoda antOFN, łącząca ACO i OFN, wykazała się szczególną efektywnością, uzyskując najlepsze wyniki dla problemu pr2392. Eksperymenty potwierdziły skuteczność tej metody w rozwiązywaniu problemów komiwojażera, a obecnie trwają badania nad jej zastosowaniem w innych klasach problemów.

Tabela 5. Zestawienie porównawcze algorytmów mrówkowych i klasycznych [źródło własne]

Zbiór testowy		Algorytmy mrówkowe				Algorytmy klasyczne		
*.tsp file	optimum	AS	ASRK	AntOFN	ACS	GR	QBOR	LK
<b>eil51</b>	<b>426</b>	426	426	426	426	521	480	426
%		100,00%	100,00%	100,00%	100,00%	81,77%	88,75%	100,00%
<b>d198</b>	<b>15780</b>	15781	15780	15780	15780	18399	18140	15828
%		99,99%	100,00%	100,00%	100,00%	85,77%	86,99%	99,70%
<b>gil262</b>	<b>2378</b>	2380	2378	2378	2378	2846	2818	2380
%		99,92%	100,00%	100,00%	100,00%	83,56%	84,39%	99,92%
<b>lin318</b>	<b>42029</b>	42091	42029	42029	42029	49744	54090	42272
%		99,85%	100,00%	100,00%	100,00%	84,49%	77,70%	99,43%
<b>pcb442</b>	<b>50778</b>	50964	50883	50778	50778	61891	58695	51071
%		99,64%	99,79%	100,00%	100,00%	82,04%	86,51%	99,43%
<b>rat783</b>	<b>8806</b>	8833	8812	8808	8806	10294	10402	8831
%		99,69%	99,93%	99,98%	100,00%	85,54%	84,66%	99,72%
<b>pcb1173</b>	<b>56892</b>	57612	56950	57040	56897	65829	66493	57063
%		98,75%	99,90%	99,74%	99,99%	86,42%	85,56%	99,70%
<b>d1291</b>	<b>50801</b>	51020	50824	50870	50820	59293	57228	52729
%		99,57%	99,95%	99,86%	99,96%	85,68%	88,77%	96,34%
<b>nrw1379</b>	<b>56638</b>	57281	56859	56917	56770	66371	66110	56756
%		98,88%	99,61%	99,51%	99,77%	85,34%	85,67%	99,79%
<b>pr2392</b>	<b>378032</b>	386541	382089	378998	379602	444853	448641	383277
%		97,80%	98,94%	99,75%	99,59%	84,98%	84,26%	98,63%

Oprogramowanie		Algorytmy mrówkowe						Algorytmy klasyczne				
Plik *.tsp	optimum	ACS	antOFN	MMAS	Asrank	EAS	AS	Lin Kernighan	Quick Borůvka	Borůvka	Greedy	Nearest Neighbor
		ACS	antOFN	MMAS	ASRK	EAS	AS	LK	QBOR	BOR	GR	NN
liczba pkt na 1000		999,31	998,95	998,29	998,13	997,20	994,09	992,65	853,26	851,10	845,58	823,13
średnia %		99,93%	99,89%	99,83%	99,81%	99,72%	99,41%	99,26%	85,33%	85,11%	84,56%	82,31%



Wykres 3. Podsumowanie efektywności algorytmów na próbce 10 problemów [źródło własne]

## References

### Bibliografia

- Amiri E., Keshavarz H., Alizadeh M., Zamani M., Khodadadi T., *Energy efficient routing in wireless sensor networks based on fuzzy ant colony optimization*, "International Journal of Distributed Sensor Networks" 2014, 10(7), 768936.
- Arora K., Agarwal S., Tanwar R., *Solving tsp using genetic algorithm and nearest neighbour algorithm and their comparison*, "International Journal of Scientific & Engineering Research" 2016, 7(1), 1014–1018.
- Bartal Y., Gottlieb L.-A., *A linear time approximation scheme for euclidean tsp*, in: "2013 IEEE 54th Annual Symposium on Foundations of Computer Science" 2013, pages 698-706.
- Biggs N., Lloyd E. K., Wilson R. J., *Graph Theory, 1736-1936*, Oxford 1986.
- Castillo O., Neyoy H., Soria J., Melin P., Valdez F., *A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot*, "Applied soft computing" 2015, 28, 150-159.
- Diestel R., *Graph Decotnpositions: A Study in Infinite Graph Theory*, Oxford 1990.
- Dobrosielski W. T., *The golden ratio of area method based on fuzzy number area as a defuzzifier*, in: K. T. Atanassov, V. Atanassova, J. Kacprzyk, A. Kauszko, M. Krawczak, J. W. Owsinski, S. S. Sotirov, E. Sotirova, E. Szmids, S. Zadrozny, editors, *Uncertainty and Imprecision in Decision Making and Decision Support: New Advances, Challenges, and Perspectives*, Cham 2022, pages 92-108.
- Dobrosielski W. T., Czerniak J. M., Szczepański J., Zarzycki, H. *Two new defuzzification methods useful for different fuzzy arithmetics*, in: *International Workshop on Intuitionistic Fuzzy Sets and Generalized Nets*, Springer 2016, pages 83-101.
- Dobrosielski W. T., Szczepański J., Zarzycki H., *A proposal for a method of defuzzification based on the golden ratio—gr*, in: K. T. Atanassov, O. Castillo, J. Kacprzyk, M. Krawczak, P. Melin, S. Sotirov, E. Sotirova, E. Szmids, G. De Tré, S. Zadrozny, editors, *Novel Developments in Uncertainty Representation and Processing*, Cham 2016, pages 75-84.
- Dorigo M., Birattari M., Stutzle T., *Ant colony optimization*, "IEEE computational intelligence magazine" 2006, 1(4), 28-39.
- Dorigo M., Gambardella L.M., *Ant colony system: a cooperative learning approach to the traveling salesman problem*, "IEEE Transactions on evolutionary computation" 1997, 1(1), 53-66.
- Dubois D., Prade H., *Operations on fuzzy numbers*, "International Journal of systems science" 1978, 9(6), 613-626.
- Engelbrechtsen L., Karpinski M., *Tsp with bounded metrics*, "Journal of Computer and System Sciences" 2006, 72(4), 509-546.
- Gaubert L., Redou P., Harrouet F., Tisseau J., *A first mathematical model of brood sorting by ants: Functional self-organization without swarm-intelligence*, "Ecological complexity" 2007, 4(4), 234-241.
- Gutin G., Yeo A., *The greedy algorithm for the symmetric tsp*, "Algorithmic Operations Research" 2007, 2(1), 33-36.
- Jovanovic R., Tuba M., *Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem*, "Computer Science and Information Systems" 2013, 10(1), 133-149.
- Kosinski W., *On defuzzification of ordered fuzzy numbers*, in: L. Rutkowski, J. Siekmann, R. Tadeusiewicz, L. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, "Lecture Notes in Computer Science" 2004, volume 3070, pages 326-331.

- Kosinski W., Prokopowicz P., Slézak D., *Ordered fuzzy numbers*, "Bulletin of the Polish Academy of Sciences" 2003, 51(3), 327-338.
- Narendra and Fukunaga, *A branch and bound algorithm for feature subset selection*, "IEEE Transactions on computers" 1977, 100(9), 917-922.
- Ravary F., Lecoutey E., Kaminski G., Châline N., Jaisson P., *Individual experience alone can generate lasting division of labor in ants*, "Current Biology" 2007, 17(15), 1308-1312.
- Roberti R., Toth P., *Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison*, "EURO Journal on Transportation and Logistics" 2012, 1(1-2), 113-133.
- Ross T. J., et al, *Fuzzy logic with engineering applications*, volume 2, Wiley Online Library, 2004.
- Traniello J. FA, *Foraging strategies of ants*, "Annual review of entomology" 1989, 34(1), 191-210.
- Yager R. R., Filev D., *On the issue of defuzzification and selection based on a fuzzy set*, "Fuzzy sets and Systems" 1993, 55(3), 255-271.
- Zadeh L.A., *Fuzzy sets*, "Information and Control" 1965, 8(3), 338-353.