ZASTOSOWANIE ALGORYTMÓW SZTUCZNEJ INTELIGENCJI DO ROZWIĄZANIA PROBLEMU KOMIWOJAŻERA

APPLICATION OF ARTIFICIAL INTELLIGENCE ALGORITHMS TO SOLVE THE TRAVELING SALESMAN PROBLEM

Krzysztof Ficoń
k.ficon@amw.gdynia.pl

Grzegorz Krasnodębski
g.krasnodebski@amw.gdynia.pl

Akademia Marynarki Wojennej
Wydział Dowodzenia i Operacji Morskich

STRESZCZENIE

W pracy omówiono heurystyczne metody rozwiązania problemu komiwojażera za pomocą algorytmów sztucznej inteligencji. Oprócz niemal klasycznych algorytmów opartych na sztucznych sieciach neuronowych i algorytmach genetycznych (ewolucyjnych) zostały przeanalizowane nowoczesne algorytmy korzystające z tzw. inteligencji roju (stada). W tej grupie zostały przeanalizowane algorytmy kolonii pszczół i stada ptaków. Szerzej zostały przedyskutowane algorytmy mrówkowe, bardzo ściśle związane z suboptymalizacją tras komunikacyjnych.

SUMMARY

The paper discusses the heuristic methods of solving the traveling salesman problem using artificial intelligence algorithms. In addition to almost classic algorithms based on artificial neural networks and genetic (evolutionary) algorithms, modern algorithms using the so-called swarm intelligence (herd). In this group, the algorithms for colonies of bees and flocks of birds have been analyzed. The formic algorithms, very closely related to the suboptimization of communication routes, have been discussed in more detail.

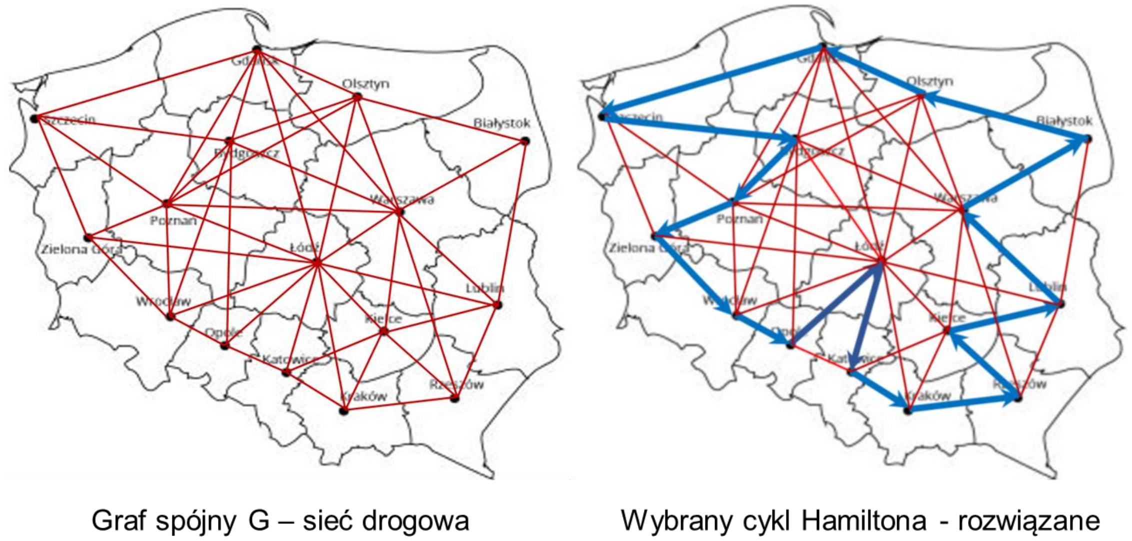
Słowa kluczowe: sztuczna inteligencja, komiwojażer, algorytm, heurystyka
Keywords: artificial intelligence, traveling salesman, algorithm, heuristics

WSTĘP

Do jednych z najtrudniejszych zadań optymalnego zarządzania procesami logistycznymi należy sterowanie systemami transportowymi obejmującymi w ogólności zarządzanie transportem zewnętrznym (międzynarodowym, krajowym) i wewnętrznym (lokalnym, firmowym), a także intermodalnym, multimodalnym czy kombinowanym. Złożoność współczesnych systemów transportowych jest ogromna, o czym świadczą różne jego podziały, a przykładem jest chociażby transport wewnętrzny, który dzieli się na: transport zaopatrzeniowy, dystrybucyjny i technologiczny. Z uwagi na wysoce konkurencyjne kryteria funkcjonowania gospodarki rynkowej optymalizacja stosunkowo kosztownych systemów transportowych jest nakazem *sine qua non*. Według różnych źródeł w ogólnym bilansie kosztów logistyki koszty funkcjonowania transportu oceniane są na poziomie 35-45% ogółu tej kwoty. W tej kwocie decydujący jest udział kosztów operacyjnych związanych *stricte* z przemieszczaniem surowców, towarów i ładunków między poszczególnymi punktami sieci komunikacyjnej – podmiotami gospodarczymi.

Obsługa topologiczna sieci transportowej przez środki transportowe jest więc elementem krytycznym całej strategii transportowej firmy, korporacji, armatora czy przewoźnika. Dlatego nadrzędnym wymogiem w dobie bardzo intensywnego ruchu towarowego jest planowanie optymalnych tras minimalizujących niebagatelne koszty transportu, wyrażone w walorach finansowych, czasowych, a także ekologicznych (Ficoń, 2015). W celu rozwiązania klasycznego zadania transportowego obejmującego przydział środków transportowych do obsługi skończonej liczby kontrahentów wykorzystuje się znane metody programowania liniowego całkowitoliczbowego, a zwłaszcza specjalnie dedykowany tzw. algorytm węgierski, bazujący na metodzie simpleks.

Najbardziej znanym i najtrudniejszym historycznie zadaniem z zakresu optymalizacji transportu jest słynny problem komiwojażera TSP (*Traveling Salesman Problem*) wywodzący się z nurtu praktycznych potrzeb logistycznych (transportowych), jednak z uwagi na duże trudności jego rozwiązania, lokowany powszechnie w sferze optymalizacji dyskretnej, jako skomplikowane zadanie kombinatoryczne (Cook, 2012). Ogólny problem komiwojażera TSP formułujemy następująco: dane jest N miast, a każde dwa z nich połączone są drogą o pewnej długości. W jednym z miast znajduje się komiwojażer (np. współczesny kurier), który chce odwiedzić w celach biznesowych wszystkie miasta w taki sposób, aby w każdym mieście znaleźć się dokładnie jeden raz, a na koniec wędrówki powrócić do miejsca startowego. Naszym celem jest znalezienie najkrótszej możliwej trasy dla komiwojażera, gwarantującej, np. minimalne koszty lub minimalny czas podróży. Tak wiec możemy poszukiwać trasy najkrótszej albo najszybszej, albo najtańszej, czyli optymalnej ze względu na przyjęte kryterium (Rysunek 1).



Rys.1. Graficzne zobrazowanie zadania komiwojażera TSP
Źródło: Opracowanie własne.

Pozornie prosty problem wyboru optymalnej (najkrótszej) trasy w sensie ponoszonych kosztów (kilometrowych, finansowych, czasowych) rozwiązywany dość sprawnie przy małej liczbie punktów topologicznych komplikuje się poważnie przy wzroście liczby odwiedzanych miejscowości (kontrahentów) (Robinson, 1949). Faktycznie problem TSP należy do złożonych, ciągle nie rozwiązanych zadań kombinatorycznych z zakresu optymalizacji dyskretnej rozpatrywanej dziś w kategoriach deterministycznych i stochastycznych, jako zadania symetryczne i niesymetryczne. Złożoność ta wynika z silnej wykładniczej zależności czasu generowania rozwiązania optymalnego od liczby punktów topologicznych leżących na docelowej trasie komiwojażera. W efekcie mamy do czynienia z tzw. problemem NP-zupełnym dla którego nie istnieje efektywny, wielomianowy algorytm uzyskania rozwiązania optymalnego. Jest to jeden z najtrudniejszych i zarazem najbardziej znanych problemów optymalizacyjnych programowania dyskretnego (Dantzig, Fulkerson, Johnson, 1954).

Wobec dużej zawodności i braku efektywnych metod rozwiązania zadania komiwojażera za pomocą znanych teorii i dostępnych narzędzi badawczych trwają intensywne poszukiwania innych metod gwarantujących uzyskanie zakładanych efektów teoretycznych i praktycznych. Obiecujące wyniki uzyskano dzięki stosowaniu miękkich metod z obszaru sztucznej inteligencji (*Soft Computing*), zaliczanych głównie do kategorii algorytmów wieloagentowych. Szczególne nadzieje budzą tzw. algorytmy mrówkowe oparte na mechanizmach naturalnej wędrówki w poszukiwaniu pożywienia. Przyroda po mistrzowsku optymalizuje trasy, jakie muszą pokonać mrówki w codziennej pracy transportowej. Adaptacyjne wykorzystanie mechanizmu algorytmów mrówkowych w problemie komiwojażera wsparte narzędziową technologią komputerową rokuje duże nadzieje na skuteczne (dopuszczalne) rozwiązywanie zwłaszcza zadań praktycznych z zakresu optymalizacji tras transportowych.

1. KLASYFIKACJA METOD ROZWIĄZANIA PROBLEMU TSP

Stosowane dotychczas metody rozwiązania NP-zupełnego problemu komiwojażera TSP można podzielić na trzy zasadnicze klasy obejmujące (Nilsson, 2003):

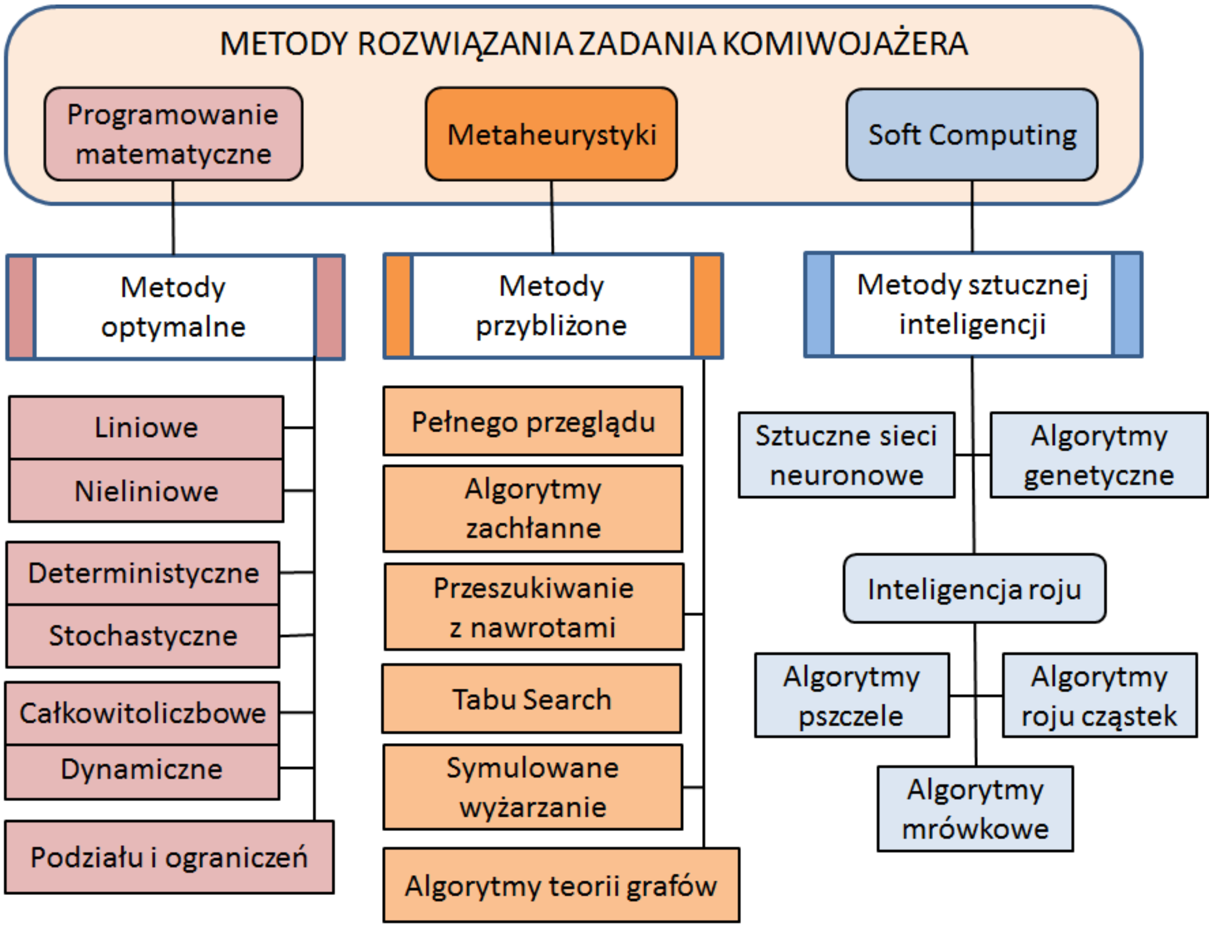
- metody dokładne (optymalizacyjne) dające rozwiązanie optymalne;
- metody przybliżone (heurystyczne) dające rozwiązanie dopuszczalne, ale niekoniecznie optymalne;
- metody sztucznej inteligencji stosunkowo sprawnie generujące rozwiązanie dopuszczalne, przybliżone.

Historycznie do rozwiązania problemu komiwojażera TSP jako pierwsze zostały wykorzystane klasyczne metody bazujące na teorii grafów, takie jak: metoda dróg elementarnych, metoda minimalnego szkieletu (Kulikowski, 1986), czy metoda funkcji kary N. Christofidesa (1976). W dalszej kolejności, już w nurcie badań operacyjnych podjęto prace nad wykorzystaniem licznych metod optymalizacyjnych, takich jak: programowanie liniowe i całkowitoliczbowe, programowanie dynamiczne, programowanie stochastyczne czy metody podziału i ograniczeń. Ze względu na NP-zupełną złożoność obliczeniową problemu TSP, pomimo angażowania coraz bardziej wydajnych komputerów klasyczne metody optymalizacyjne mogły być efektywnie wykorzystane tylko do niewielkich zadań, dla stosunkowo małej liczby miast (Reinelt,1994).

Metody przybliżone (heurystyczne) nie gwarantują rozwiązania optymalnego, natomiast charakteryzują się stosunkowo dużą szybkością, dzięki czemu są z powodzeniem stosowane w rozwiązywaniu praktycznych zadań TSP. Nowoczesne metody potrafią znaleźć rozwiązanie dla ogromnych problemów z wielką liczbą miast w rozsądnym czasie, odbiegające zaledwie 2-3% od optymalnego rozwiązania. Heurystyka w ogólności jest działem dosyć obszernym, który możemy podzielić na heurystykę konstruktywną, gdzie rozwiązanie jest budowane sukcesywnie z otrzymanych fragmentów, które nie są modyfikowane przez algorytm oraz heurystykę ulepszającą, w której znajdują się metody ulepszające już istniejące rozwiązanie (Applegate, Bixby, Chvátal, Cook, 2007).

Jeszcze inne kryterium dzieli algorytmy heurystyczne na zachłanne i tzw. metaheurystyki, które zawierają programowe mechanizmy ucieczki z lokalnych ekstermów, reprezentowane m.in. przez takie algorytmy jak: algorytmy wyszukiwania z nawrotami, metodę *Tabu Search* (Glover, 1990), (Hall, 2012) czy metodę symulowanego wyżarzania

(Metropolis, 1953). Efektywne rozwiązywanie bardziej złożonych zadań TSP umożliwiły dopiero metody przybliżone bazujące na różnych metaheurystykach i mechanizmach zdroworozsądkowych (Applegate, Bixby, Chvátal, Cook, 2007). Zaawansowane metody przybliżone ewoluują dziś w stronę rozwiązań hybrydowych łączących rozmaite heurystyki z algorytmami optymalizacyjnymi, a także z metodami obliczeniowymi sztucznej inteligencji (Rysunek 2).



Rys. 2. Typologia metod wykorzystywanych w problemie komiwojażera TSP
Źródło: Opracowanie własne.

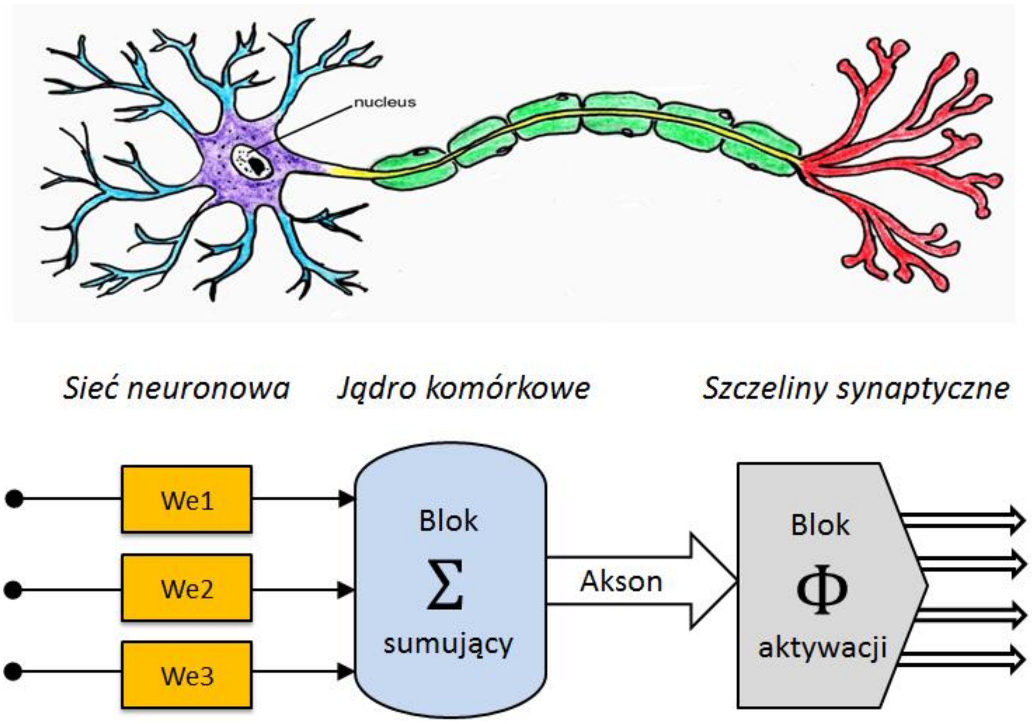
Burzliwy rozwój technologii komputerowej powoduje szeroki wzrost zainteresowania metodami obliczeniowymi sztucznej inteligencji (*Artificial Intelligence*), zwłaszcza grupą tzw. metod inteligencji obliczeniowej (*Computational Intelligence*), do których tradycyjnie zaliczamy: sztuczne sieci neuronowe (*Neural Networks*), algorytmy genetyczne (*Genetic Algorithms*), czy teorię zbiorów rozmytych (*Fuzzy Sets*) (Ficon, 2013). W ostatnich latach dynamicznie rozwija się, głównie na potrzeby zadań optymalizacji kombinatorycznej nowy dział metod optymalizacyjnych bazujących na inteligencji roju (SI – *Swarm Intelligence*), których zasady działania zostały zaczerpnięte z obserwacji świata zwierząt. Do tej klasy algorytmów należą m.in. algorytmy mrówkowe (ACO – *Ant Colony Optimization*), algorytmy pszczele (ABC – *Artificial Bee Colony*) i szeroka klasa algorytmów roju cząstek (PSO – *Particle Swarm Optimization*) (Kennedy, Eberhart 1995).

Dynamicznie rozwijane ostatnio metody inteligencji roju (*Swarm Intelligence*) wykorzystują naturalny efekt synergii wynikający z inteligentnej współpracy określonej populacji osobników, gwarantujący m.in. jej przetrwanie w naturalnych warunkach konkurencji ewolucyjnej. Ogół tych metod często klasyfikuje się do grupy "miękkich" metod obliczeniowych pod wspólną nazwą *Soft Computing*. W grupie *Soft Computing* szczególną użytecznością w stosunku do wymagań problemu TSP odznacza się kategoria tzw. algorytmów (metod) mrówkowych (ACO) pozwalająca na bardzo elastyczne dochodzenie do rozwiązań suboptymalnych kosztem stosunkowo niedużych nakładów czasowych i niewielkich mocy obliczeniowych.

2. SZTUCZNE SIECI NEURONOWE

Historia sztucznych sieci neuronowych (NN) daje początek zaawansowanym narzędziom inteligencji obliczeniowej, które po dzień dzisiejszy budzą liczne kontrowersje. Koncepcja NN pochodzi od słynnego perceptronu F. Rosenblatta i Ch. Wightmana. Zaproponowane przez nich w latach 70. XX w. cybernetyczne symulatory komórki nerwowej – neuronu okazały się bardzo atrakcyjne poznawczo i były intensywnie rozwijane przez wielu badaczy w różnych placówkach naukowych. Sukcesywnie rozwijane od pojedynczej komórki poprzez struktury jednowarstwowe, aż do modeli wielowarstwowych oferowały nieznaną dotychczas zdolność programowego uczenia i samodoskonalenia. Centralną pozycję w teorii sztucznych sieci neuronowych zajmuje sformalizowany model neuronu zobrazowany za pomocą elementarnej jednostki logiczno-arytmetycznej przeznaczonej do przetwarzania zewnętrznych sygnałów źródłowych na sygnały wyjściowe sterujące, np. pracą innych neuronów (Ficoń, 2013).

Sztuczna sieć neuronowa jest odpowiednio skonfigurowanym układem sztucznych (matematyczno-logicznych) modeli neuronów połączonych ze sobą licznymi więzami, poprzez specjalny system złączy (przełączników) zwanych synapsami, które pełnią kluczową rolę w przetwarzaniu informacji. Synapsy są najbardziej aktywnymi elementami sztucznych sieci neuronowych, gdyż poprzez współczynniki wagowe bezpośrednio wpływają na wielkość sygnału wyjściowego, przetwarzanego przez neurony. Sztuczny neuron przejmuje wszystkie nadchodzące sygnały wejściowe i rozpoczyna ich interpretację, zgodnie z przyjętą funkcją transformacji sygnału. Wynikowy sygnał sumaryczny jest przesyłany za pomocą właściwego aksonu przez szczelinę synaptyczną do innych neuronów, zgodnie z aktualną konfiguracją sieci połączeń. Neurony są połączone w sieć za pomocą połączeń wagowych modyfikowanych w trakcie tzw. procesu uczenia. Topologia połączeń oraz ich parametry stanowią program działania sieci, zaś sygnały pojawiające się na wyjściach, w odpowiedzi na określone sygnały wejściowe, są rozwiązaniami stawianych zadań (rysunek 3).



Rys. 3. Model neuronu w strukturze sztucznej sieci neuronowej
Źródło: Opracowanie własne.

Problematyka badawcza sztucznych sieci neuronowych wywodzi się z rodziny nauk dedukcyjnych, których podstawy teoretyczne opierają się na założeniu, że jeśli dana sieć neuronowa poświęci dostatecznie dużo czasu na rozwiązywanie zadań treningowych, to jej odpowiedzi będą coraz bardziej zbliżone do rozwiązania rzeczywistego problemu. Po latach eksperymentów, pozbawionych niestety solidnych podstaw teoretycznych okazało się, że sieci neuronowe rzeczywiście rozwiązują niektóre nawet skomplikowane zadania w sposób zadawalający, ale ich szersze możliwości zgodnie z zasadą NP-zupełności są bardzo ograniczone (Mandziuk, 2000).

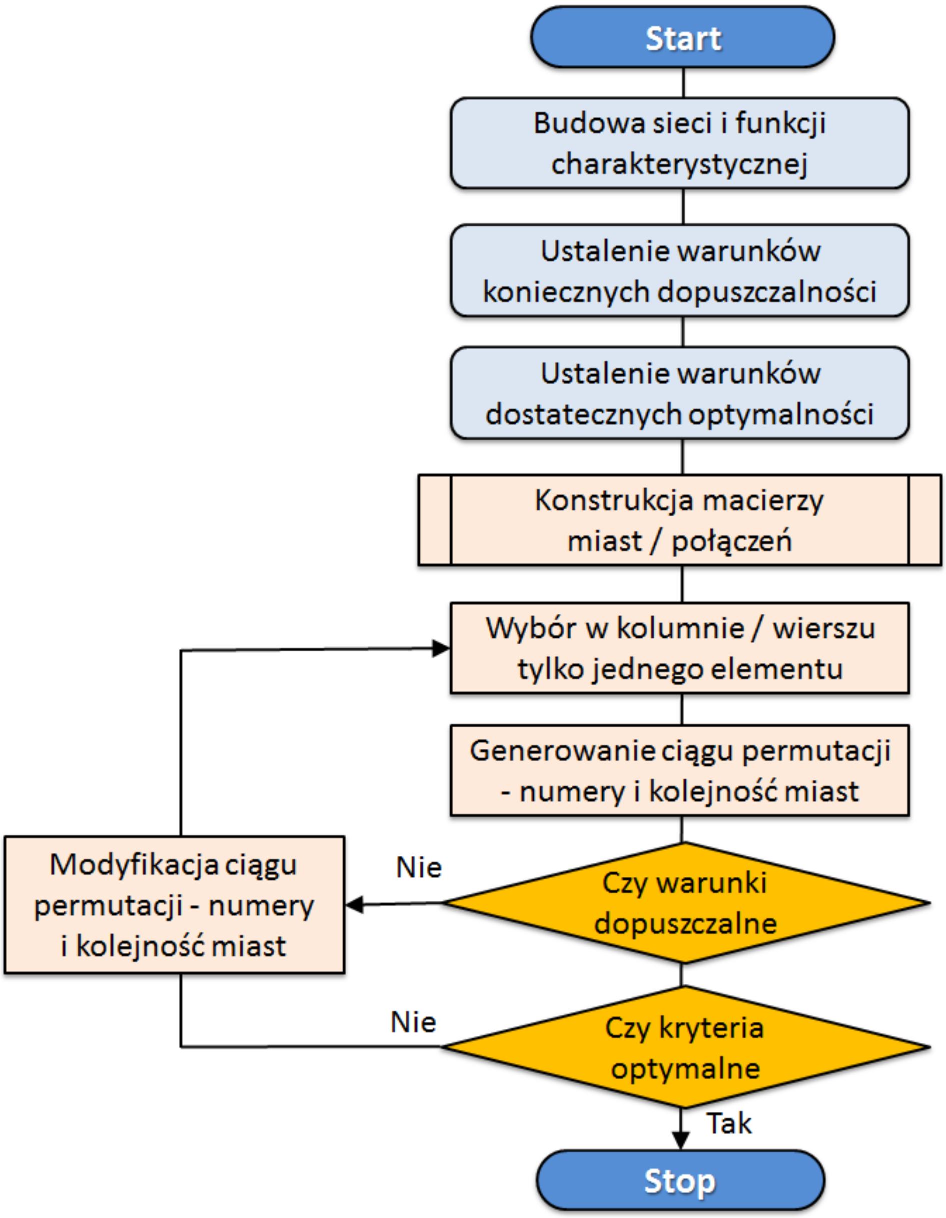
Potwierdzeniem tej tezy jest pozornie prosty problem komiwojażera polegający na wyznaczeniu optymalnej trasy jego podróży. Z wielką energią wyzwanie to podjął znakomity matematyk i informatyk J. Hopfield – ojciec i entuzjasta tzw. sieci Hopfielda. Pomysł ten opierał się na następujących założeniach. Po pierwsze, należy zbudować sieć neuronową i opisać ją takim równaniem, aby można było na podstawie tego równania ustalić wagi synaptyczne tej sieci. Po drugie, stosownie do trasy komiwojażera należy zaproponować odpowiednią funkcję charakterystyczną, która będzie definiować warunki konieczne i wystarczające konstruowania optymalnej, a przynajmniej dopuszczalnej trasy (Rysunek 4). Dla potrzeb problemu TSP sieć Hopfielda została zapisana w postaci macierzy kwadratowej $[M]_{N \times N}$, gdzie N – liczba rozpatrywanych miast na trasie komiwojażera. Elementy macierzy symbolizują elementarne neurony, których łączna liczba dla sieci obejmującej N-miast wynosi N^2. W każdym wierszu tej macierzy i w każdej kolumnie może znajdować się tylko jeden aktywny element przyjmujący wartość 1. Dla każdego aktywnego neuronu wartość jego wyjścia przyjmujemy równą 1, wówczas gdy jego pozycja jest dostatecznie bliska jedności np. 0,96. W przeciwnym przypadku wartość wyjścia neuronu wynosi zero. Numer kolumny w macierzy neuronów określa pozycje miasta (numer) na trasie komiwojażera, kodowanego przez tą kolumnę. Pozycja, w której neuron przyjmuje wartość 1 określa jednoznacznie numer miasta i kolejność w jakiej ma ono być odwiedzane (Tabela 1.).

Tabela 1. Przykładowa macierz 6-miast w modelu sztucznych sieci neuronowych

Kol. / Miasta	I	II	III	IV	V	VI
A	1	0	0	0	0	0
B	0	0	0	0	1	0
C	0	0	0	0	0	1
D	0	0	1	0	0	0
E	0	0	0	1	0	0
F	0	1	0	0	0	0

Źródło: Opracowanie własne.

Z tabeli 1 wynika, że zgodnie z aktywnością poszczególnych neuronów (1-aktywny, 0-nieaktywny) komiwojażer odwiedzi miasta w następującej kolejności A-F-D-E-B-C.



Rys. 4. Idea wykorzystania sztucznych sieci neuronowych do rozwiązania zadania komiwojażera
Źródło: Opracowanie własne.

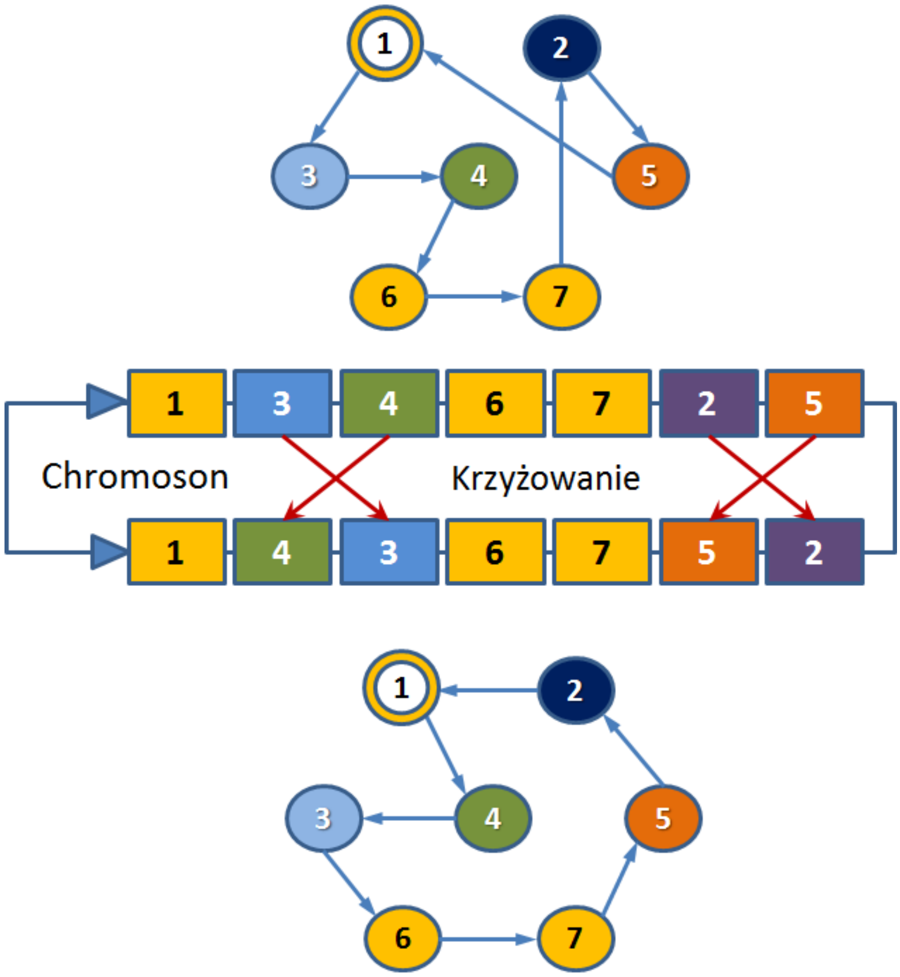
Funkcja charakterystyczna opisująca aktywną sieć neuronową przedstawia postać rozwiązania w formie ciągu permutacji (numerów) miast na trasie komiwojażera. W kolejnych cyklach jest ona stopniowo poprawiana pod kątem coraz lepszych rozwiązań, czyli ścieżek o krótszej trasie. Jeśli poprawnie zdefiniujemy równanie charakterystyczne sieci to podejście takie jest uniwersalne i umożliwia konstruowanie sieci neuronowych rokujących nadzieje na rozwiązanie zadania komiwojażera przynajmniej w sposób zadawalający. Niestety, jak wykazał A.K. Dewdney (1997) teoretyczna i praktyczna użyteczność sztucznych sieci neuronowych w rozwiązaniu optymalizacyjnego problemu TSP okazała się bardzo ograniczona i zupełnie nieprzydatna do badań bardziej złożonych zadań obejmujących większą liczbę miast niż kilkadziesiąt.

3. ALGORYTMY GENETYCZNE

Algorytm genetyczny (GA – *Genetic Algoritms*) to rodzaj algorytmu heurystycznego ukierunkowanego na przeszukiwanie w sposób mniej lub bardziej losowy przestrzeni rozwiązań alternatywnych w celu znalezienia rozwiązań najlepszych – niekoniecznie optymalnych. Sposób działania algorytmów genetycznych nieprzypadkowo przypomina zjawisko ewolucji biologicznej, ponieważ ich twórca J.H. Holland (1975) właśnie z biologii czerpał inspiracje do swoich prac nad algorytmami genetycznymi. Ze względu na fakt, że koncepcja algorytmów genetycznych ma swoje korzenie w biologii i genetyce, a rozwijana jest za pomocą technologii komputerowych jej terminologia jest kompilacją słownictwa używanego w obu tych dziedzinach (Ficoń, 2013).

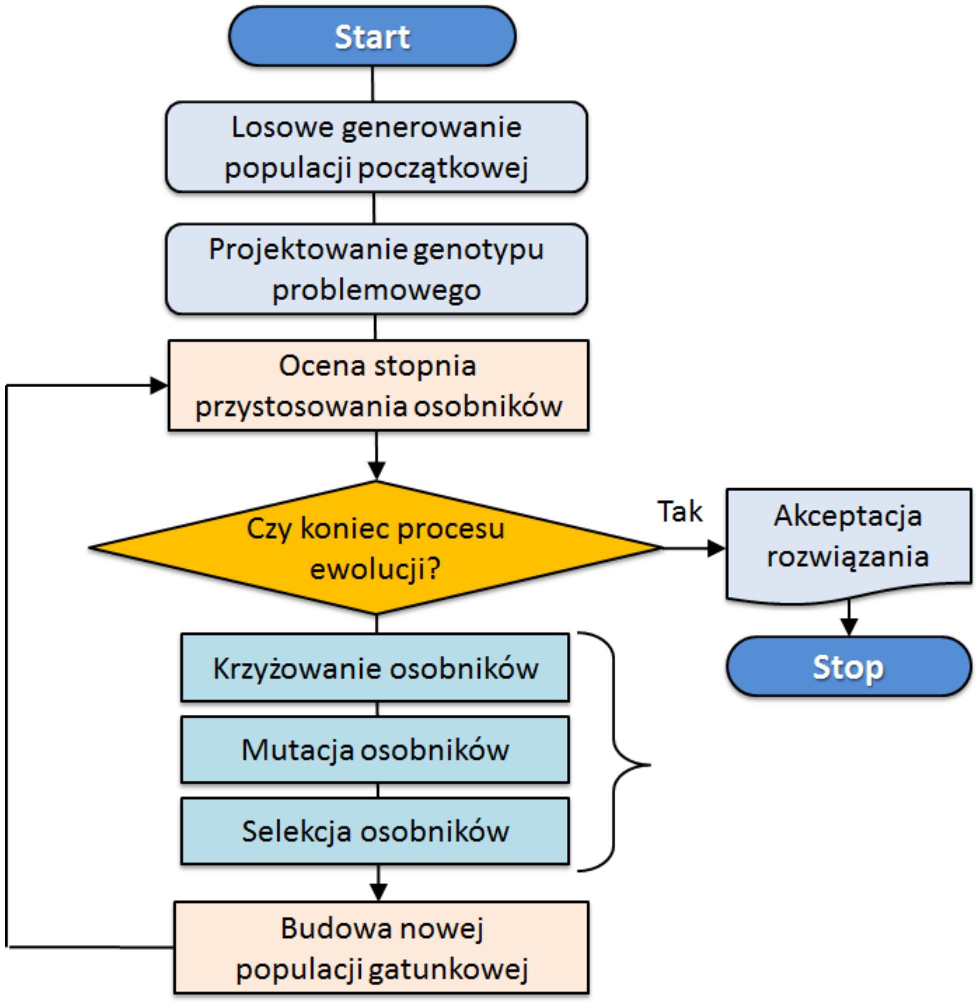
Kluczowymi pojęciami algorytmów genetycznych są klasyczne terminy genetyki, którym odpowiadają następujące określenia: gen – bit, chromosom – ciąg bitów, osobnik – punkt w przestrzeni rozwiązań, populacja – zbiór rozpatrywanych osobników, krzyżowanie – wymiana bitów czy mutacja – losowa negacja wartości bitów, selekcja – eliminacja osobników zgodnie z funkcją przystosowania (Ramani, 2011). Algorytmy genetyczne działając na genach, chromosomach i osobnikach pracują faktycznie na symbolicznym, informacyjnym opisie osobnika, a nie na samych osobnikach, co pozwala ograniczyć niezbędną informację do absolutnego minimum podnosząc sprawność ich działania. Algorytmy genetyczne, podobnie jak ewolucja pracują na całych populacjach podlegających w tym przypadku sztucznym procesom selekcji, krzyżowania i mutacji, których efektem jest gatunkowy rozwój tej populacji. Zgodnie z teorią K. Darwina (1809-1882) natura posługuje się prostym kryterium zdolności do przetrwania, natomiast algorytm czy program komputerowy będzie operował sztucznym, ilościowym kryterium oceny jakościowej każdego osobnika.

Idea działania algorytmów genetycznych zaproponowana przez J.H. Hollanda (1975) jest nadzwyczaj prosta i jednoznaczna. Formalnie zaczynamy od losowej populacji N ciągów kodowych, dokonujemy ich replikacji, preferując najlepsze osobniki, losowo kojarzymy je wymieniając fragmenty ciągów i na dodatek od czasu do czasu mutujemy (negujemy) jakiś bit (Rysunek 5).



Rys. 5. Mechanizm działania operacji krzyżowania w problemie komiwojażera
Źródło: Opracowanie własne.

Proces działania algorytmu genetycznego inicjuje projektowanie (kodowanie) struktury genotypu – zespołu chromosomów składających się na obraz osobnika. Efektem działania tego algorytmu jest fenotyp reprezentujący rozwiązanie końcowe problemu. Standardowy (kanoniczny) algorytm genetyczny (SGA – *Simple Genetic Algorithm*) obejmuje następujące etapy: 1 – losowanie populacji, 2 – ocena osobników, 3 – selekcja osobników, 4 – krzyżowanie osobników, 5 – mutacja osobników, 6 – badanie sztucznego warunku stop, który praktycznie w przyrodzie nie istnieje. Rekurencyjne powtarzanie algorytmu w pętli ewolucyjnej powinno skutkować generowaniem coraz krótszych wariantów dopuszczalnych tras, co gwarantuje ogólna filozofia działania algorytmów genetycznych. Po dostatecznie długim czasie działania algorytmu genetycznego powinniśmy otrzymać osobnika, który będzie zawierał wszystkie miasta, a łączna długość całej trasy będzie najkrótsza ze wszystkich dotychczas rozpatrywanych (Rysunek 6).



Rys. 6. Schemat standardowego algorytmu genetycznego SGA
Źródło: Opracowanie własne.

Po zdefiniowaniu chromosomu gatunkowego punktem startowym algorytmu SGA jest pierwsza populacja, która zawiera osobnika (marszrutę) dla którego trasa została wygenerowana losowo ze zbioru możliwych permutacji tras komiwojażera. Na tak zdefiniowanej populacji aktywowane są operatory genetyczne – krzyżowania, mutacji i selekcji. Krzyżowanie polega na wymianie elementów trasy (miast) pomiędzy dwoma osobnikami. Mutacja oznacza zamianę w pojedynczym osobniku, np. dwóch miast między sobą, czyli zmianę kolejności ich odwiedzania. Operator mutacji dopuszcza też przestawienie między sobą kilku miast na trasie komiwojażera. Kolejnym etapem jest selekcja, czyli wybór do następnej populacji osobników najlepiej przystosowanych tzn. takich które w tym przypadku reprezentują najkrótsze warianty trasy komiwojażera (Michalewicz, 2006).

Warto także wspomnieć o algorytmie ewolucji różnicowej. Algorytm ten został przedstawiony przez Pricea i Storma w 1995 roku jako algorytm optymalizacji z niewielką liczbą parametrów. Populacja składa się z określonej liczby wektorów zawierających wariancje rozwiązania problemu. Cechą charakterystyczną tego algorytmu jest metoda krzyżowania, która wykorzystuje trzy wektory wylosowane z populacji rodziców. Wybór najlepszego wektora do populacji potomnej następuje między wektorami z populacji rodziców a wektorami z populacji powstałej przy wykorzystaniu metod krzyżowania i mutacji. W wielu problemach optymalizacji algorytm okazał się lepszy w porównaniu do innych algorytmów ewolucyjnych (Huibin, Mingguang, 2009). Z kolei Bubal i Lee (2016) zaproponowali planowanie połączeń transportowych i komunikacyjnych w taki sposób, aby były one optymalnie dopasowane do zapotrzebowania osób w dużych aglomeracjach miejskich za pomocą algorytmu ewolucji różnicowej. Opracowany algorytm uwzględnia nie tylko zadowolenie pasażerów, ale również koszty ponoszone przez zarządców autobusów, tramwajów oraz pociągów, tak aby połączenia były opłacalne.

Reasumując, metody ewolucyjne polegają na przeszukiwaniu alternatywnych rozwiązań naśladując procesy naturalnej ewolucji, takie jak dziedziczenie, mutacja, krzyżowanie, selekcja (Michalewicz, 2006). Prowadzą do otrzymywania coraz lepiej przystosowanych osobników, co w tym wypadku oznacza znalezienie coraz krótszej trasy (fenotypu). Duża prostota i wysoka sprawność algorytmów genetycznych została wykorzystana do próby rozwiązania niektórych problemów NP-zupełnych i flagowego okrętu, jakim jest problem komiwojażera. Populację gatunkową będziemy odnosić do topograficznej trasy komiwojażera. Poszczególne osobniki tej klasy będą zawierać informację o kolejności miast odwiedzanych w danym wariancie marszruty. Takich osobników przy nietrywialnym problemie komiwojażera obejmujących większą liczbę miast będzie dostatecznie dużo, co podnosi jej wrażliwość na działanie mechanizmu ewolucji.

4. ALGORYTMY INTELIGENCJI ZBIOROWEJ

Wobec zawodności metod optymalizacyjnych, a także niewielkiej skuteczności klasycznych metod sztucznej inteligencji, zaliczanych do *SoftComputing* w procesie rozwiązania zadania komiwojażera wielkie nadzieje rodzi burzliwie rozwijające się na przełomie XX/XXI wieku metaheurystyki oparte na naturalnej inteligencji zbiorowej – roju cząstek (*Swarm Intelligence*) wywodzącej się ze świata zwierząt. Rój cząstek stanowi grupę osobników najczęściej tego samego gatunku zwierząt – owadów, ptaków, ryb żyjących w określonym środowisku i związanych ze sobą pewną formą organizacji (inteligencji) społecznej. Łączenie się pojedynczych osobników w stada jest podyktowane koniecznością przetrwania i rozrodu populacji oraz zdobywaniem pożywienia.

Ogólna idea algorytmu optymalizacji roju cząstek PSO (*Particle Swarm Optimization*) wywodzi się z naśladowania zachowania się populacji żywych istot – ptaków, mrówek, pszczół, świetlików, nietoperzy, karaluchów, kryli, kukułek, nietoperzy itp. w przypadku życiowej konieczności przetrwania i rozwoju. Z reguły pojedynczy osobnik każdej populacji ma bardzo ograniczone anatomiczne możliwości samodzielnego podejmowania efektywnych decyzji, a siłą każdej zbiorowości jest skuteczny sposób komunikowania się i wzajemnej wymiany informacji. Całość populacji pomimo braku centralnego systemu sterowania wykazuje cechy dysponowania pewną inteligencją i skutecznego reagowania na zmiany w otoczeniu oraz podejmowania dopuszczalnych decyzji (Lisowski, 2017).

Model matematyczny traktuje rój jako złożoną populacje (zbiór), a każdego jej osobnika jako pojedyncza cząstkę (element zbioru). W trakcie kolejnych kroków procedury algorytmicznej na osi zdyskretyzowanego czasu poszczególne cząstki przemieszczają się celowo w czasoprzestrzeni do nowych położeń symulując adaptacje populacji do zmieniającego się środowiska, czyli dążenie do pewnej suboptymalizacji. Algorytmy roju cząstek PSO do poszukiwania ekstremalnej wartości funkcji celu wykorzystują populację losowo przemieszczających się cząstek, które potrafią zapamiętać i przekazać informację o swoim położeniu w przestrzeni poszukiwań do całej populacji. Ruchy kolejnych cząstek są wstępnie programowane przez pozyskane informacje, co pozwala na znajdowanie najlepszych wartości funkcji celu i bardziej skuteczne działanie całej populacji. Na podstawie bezustannie modyfikowanej informacji spływającej od poszczególnych osobników cząstki mogą sukcesywnie zmieniać swoje położenie, dążąc do punktu o najlepszej wartości funkcji celu (Lisowski, 2017).

Aktualnie najbardziej zadawalające wyniki dają algorytmy bazujące na zbiorowej inteligencji pszczół (ABC – *Artificial Bee Colony*), ptaków (PSO – *Particle Swarm Optimization*) i mrówek (ACO – *Ant Colony Optimization*) (Clerc, 2004). Algorytmy te, chociaż znacząco się różnią, posiadają wspólną cechę, jaką jest działanie poszczególnych agentów (osobników) bez wcześniej określonego planu oraz bez nadrzędnej jednostki zarządzającej. Ponadto, typowym dla nich jest korzystanie ze zbiorowej wiedzy gromadzonej podczas ewaluacji algorytmu. Każdy agent działa w sposób autonomiczny, ale uwzględnia zbiorczą wiedzę całej populacji zgromadzoną na dany moment. W przypadku ABC mówić możemy o inspiracji wynikającej z zachowania się pszczół, PSO – ptaków, natomiast ACO – mrówek.

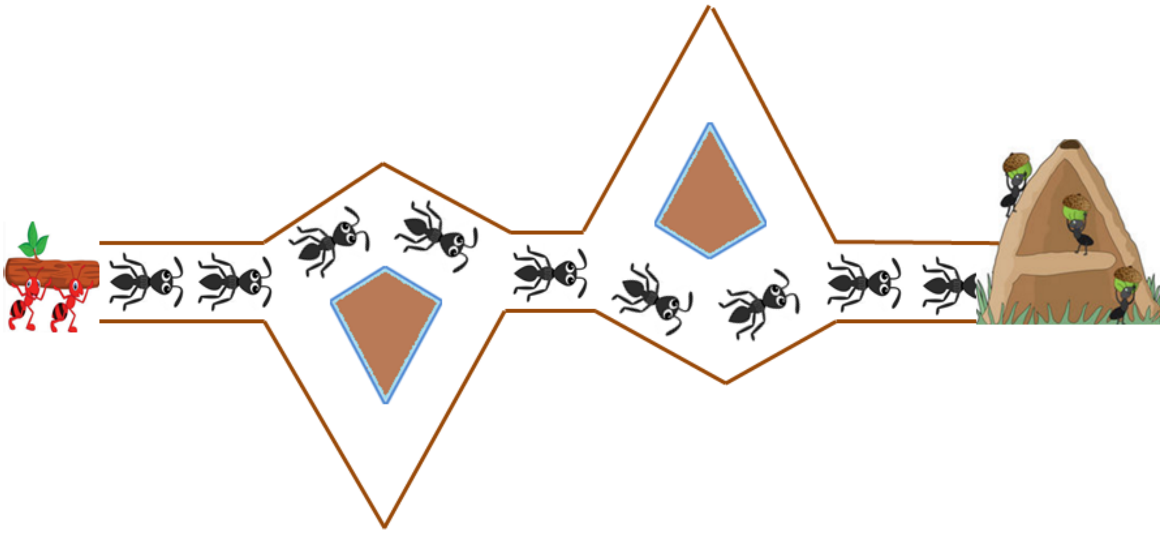
Cieszące się coraz większym zainteresowaniem metody inteligencji roju stanowią ogromny potencjał badawczy i aplikacyjny do rozwiązywania zagadnień optymalizacji kombinatorycznej, w tym problemu komiwojażera TSP (Clerc, 2004).

5. ALGORYTMY MRÓWKOWE

Inspiracją dla twórców algorytmów mrówkowych było zachowanie mrówek argentyńskich, które M. Dorigo (1995) zaliczył do tzw. systemów wieloagentowych (MAS – *Multi Agent Systems*). Okazało się, że jeśli od gniazda mrówek do miejsca składowania żywności prowadzą dwie drogi – jedna krótsza, a druga dłuższa, to po pewnym czasie większość mrówek będzie korzystać z drogi krótszej (Dorigo, Maniezzo, Colorni, 1996). Algorytm mrówkowy (ACO) jest przeznaczony do heurystycznego wyszukiwania najlepszych (optymalnych) ścieżek w grafach. Algorytm ACO stanowi podstawę dla istotnej liczby różnych rozszerzeń takich jak: *Ranked-Based ant System* czy *Recursive ant Colony Optimization*.

Pomysł pochodzi ze świata mrówek, które potrafią znaleźć najkrótszą trasę między mrowiskiem a pożywieniem. Początkowo mrówki wędrując w stronę pożywienia obierają drogę losowo, ale wracając do mrowiska w drodze powrotnej pozostawiają na swojej trasie tzw. ślad feromonowy. Mrówki orientują się w poszukiwaniu pokarmu przy pomocy wydzielanej substancji chemicznej – feromonu, która posiada określoną trwałość i ograniczoną w czasie aktywność. Podążające określonym szlakiem kolejne mrówki dokonują wyboru kierunku drogi na podstawie intensywności pozostawionego feromonu. Substancja ta pełni rolę wspólnej bazy danych dla całej kolonii determinującej topografię najbardziej uczęszczanych szlaków (trajektorii).

Feromony stopniowo parują, co powoduje utratę cennej informacji. Na krótszej trasie intensywnie uczęszczanej przez mrówki parowanie jest stosunkowo wolniejsze niż na innych mniej uczęszczanych trasach. W efekcie kolejne mrówki wybierają tę krótszą trasę chętniej niż inne trasy. Kiedy zostanie znaleziona najkorzystniejsza dla mrówek trasa, kolejne mrówki wybierają ją chętniej wzmacniając ślad feromonowy. Jest to znane z teorii sterowania zjawisko pozytywnego sprzężenia zwrotnego. Korzystna o dużej intensywności feromonu ścieżka staje się coraz bardziej atrakcyjna dla danej kolonii mrówek. Mniej korzystne trasy tracą na znaczeniu, gdyż ulatniający się ślad feromonowy powoduje coraz mniejszą aktywność mrówek. Odparowywanie feromonu z kolei sprawia, że mrówki nie poprzestają na znalezieniu lokalnego maksimum, co umożliwia im dostosowanie się do zmian środowiska (Rysunek 7).



Rys. 7. Zobrazowanie wpływu śladu feromonowego na wybór trasy przez mrówki
Źródło: Opracowanie własne.

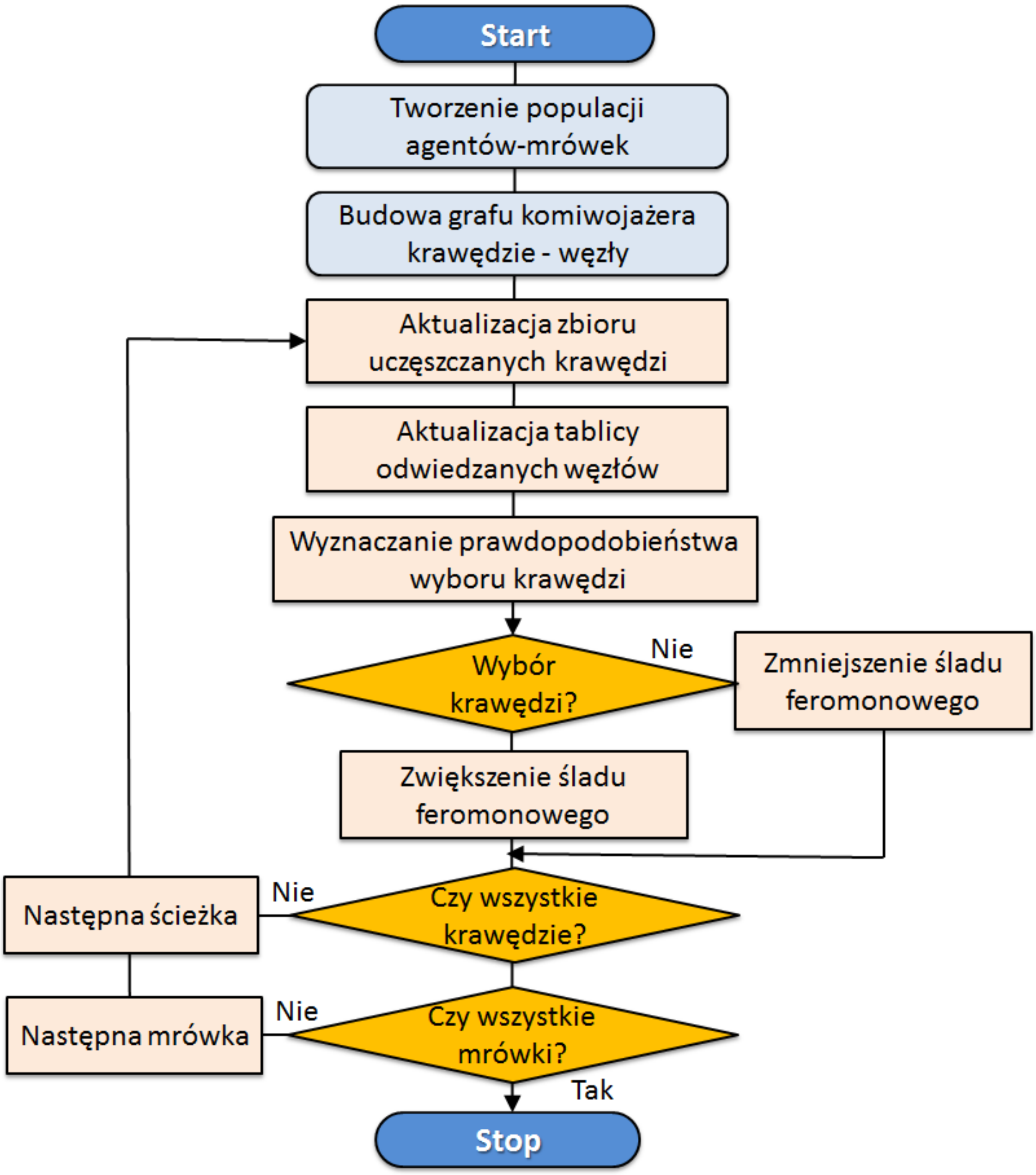
Zachowaniem mrówek kierują trzy proste zasady. Pierwsza – każda mrówka znaczy ślad swojej wędrówki feromonem, druga – owad wybiera ścieżkę z prawdopodobieństwem proporcjonalnym do ilości feromonu na niej, trzecia – feromon z czasem stopniowo odparowuje, co zmniejsza trakcyjność ścieżki. Inteligencja mrówek polega na interaktywnych relacjach między kolonią mrówek, które dzielą się zdobytym doświadczeniem na etapie poszukiwania i wykorzystania najkrótszej trasy. W miarę upływu czasu mrówki wspólnie wypracowują pewien zbiór najkrótszych ścieżek wiodących do określonych celów. Strategia wędrówek mrówek (agentów) oparta jest więc na inteligencji zbiorowej (stadnej). Działanie poszczególnych agentów w takim systemie oparte jest na działaniu pojedynczych mrówek poszukujących niezbędnych dla ich wegetacji zasobów. Każdy agent takiego systemu (mrówka) realizuje identyczną strategię poszukiwania najbardziej korzystnej drogi prowadzącej do celu (Dorigo, Stutzle, 2004).

Z przedstawionej koncepcji intuicyjnie rodzi się pomysł wykorzystania algorytmu mrówkowego ACO albo strategii wieloagentowej do rozwiązania problemu komiwojażera. W algorytmicznej implementacji ACO wyróżnić można trzy główne kroki. Pierwszym z nich jest generowanie rozwiązań dla każdej sztucznej mrówki, czyli generowanie indywidualnej ścieżki w zadanym grafie. Następnym krokiem jest uaktualnienie sztucznych śladów feromonowych, czyli symulacja wyparowywania oraz wzmacniania śladów feromonowych na podstawie wygenerowanych poprzednio rozwiązań. Krok ten jest ściśle związany z dziedziną problemową, w której algorytm jest wykorzystywany. W ostatnim etapie wykonywane są zazwyczaj akcje dodatkowe oraz takie, które operują na wiedzy globalnej, np. dodatkowe wzmocnienie danego śladu, który okazał się być najbardziej korzystnym.

Ogólny zarys tej metody obejmuje następujące etapy. Tworzona jest populacja mrówek, której rozmiar jest jednym z parametrów algorytmu. Pojedyncza mrówka generuje swoją ścieżkę niezależnie (losowo) od reszty populacji. Dla każdej mrówki generowane jest losowo miejsce startowe (miasto), z którego rozpoczyna wędrówkę. Formalnie mrówki poruszają się po grafie (sieci) szukając sekwencji wierzchołków grafu tworzącej najkrótszą drogę od wierzchołka startowego do końcowego.

Aby ukierunkować przypadkowe błądzenie i wyeliminować wielokrotne odwiedzanie tych samych wierzchołków, każda wirtualna mrówka została wyposażona w podręczną pamięć, w której przechowuje listę odwiedzanych wierzchołków. Na starcie lista ta jest pusta, natomiast po dojściu do celu zawiera sekwencję odwiedzanych wierzchołków, w kolejności ich odwiedzania. Lista ta jest tworzona na podstawie śladu feromonowego znaczącego trasę przemarszu każdej mrówki. Poszczególne krawędzie grafu są etykietowane śladem feromonowym. Przy wyborze kolejnej marszruty, po pierwsze, wirtualna mrówka nie uwzględnia wierzchołków już odwiedzonych, a po drugie, kieruje się bardziej aktywnym śladem feromonowym pozostawionym przez inne mrówki, które wybrały już tą trasę. Wybór kolejnego odcinka jest więc losowy, ale zgodny z zasadą – im więcej feromonu znajduje się na danej krawędzi, tym większe jest prawdopodobieństwo wyboru danej trasy przez mrówkę.

Algorytm mrówkowy ma charakter iteracyjny. Po zakończeniu bieżącej iteracji każda mrówka czeka na wyznaczenie nowego węzła (miasta) startowego, z którego w następnej iteracji ponownie wyruszy na trasę poszukiwań. W procedurze iteracji bardzo istotny jest fakt gromadzenia feromonu w miarę upływającego czasu na poszczególnych krawędziach grafu. Jednocześnie w miarę upływu czasu, czyli liczbie wykonywanych iteracji feromon na ścieżkach mniej uczęszczanych bardziej paruje niż na szlakach intensywnie odwiedzanych. Aktualizacja poziomu (śladu) feromonu uwzględnia zarówno procesy składania odpowiedniej porcji przez mrówki poruszające się po danej marszrucie, jak też jego odparowanie w wyniku mniej intensywnego ruchu na danym odcinku. Strategiczny ślad feromonowy jest dynamicznie budowany przez całą populację mrówek podczas kolejnych iteracji algorytmu (Rysunek 8).



Rys. 8. Sieć działań algorytmu mrówkowego ACO w problemie TSP
Źródło: Opracowanie własne.

Nie trudno zauważyć, że w trakcie działania algorytmu mrówkowego, żadna informacja „nawigacyjna", która jest niezbędna w efektywnym wyborze trasy nie jest gromadzona w mrówkach. Informacja użyteczna dla poszczególnych mrówek (agentów) jest przechowywana i aktualizowana na poszczególnych krawędziach grafu w postaci mniej lub bardziej aktywnego śladu fermonowego. Rola mrówek polega na wyborze kolejnych odcinków trasy na podstawie ilości feromonu złożonego na danym odcinku (krawędzi) oraz na systematycznym znakowaniu wybranego odcinka własnym śladem feromonowym. Najważniejszą informacją w algorytmie mrówkowym będącą jednocześnie rozwiązaniem problemu komiwojażera jest zbiór krawędzi grafu (marszruta) zawierających najwyższą wartość śladu feromonowego. Rozwiązanie optymalne jest superpozycją najbardziej aktywnych śladów feromonowych wszystkich mrówek (agentów) tworzących daną populację.

Problem komiwojażera był jednym z pierwszych zagadnień, w którym wykorzystano rokujące realne nadzieje algorytmy mrówkowe. Algorytmy mrówkowe znajdują zastosowanie przy rozwiązywaniu wielu NP-złożonych zagadnień optymalizacyjnych, takich jak, np.: zadanie transportowe, problemy przydziału, szeregowanie zadań, kolorowanie grafu czy sortowanie sekwencyjne. Obiecujące efekty przynoszą w praktycznych aplikacjach dotyczących np. marszrutacji środków i szlaków transportowych, wykorzystania korporacyjnej floty transportowej, a także w zadaniach routingu w sieciach telekomunikacyjnych.

6. PODSUMOWANIE

Przeprowadzone powyżej rozważania pozwoliły na pozytywne zweryfikowanie postawionej na wstępie tezy orzekającej o dużej użyteczności metod sztucznej inteligencji w szczególności algorytmów mrówkowych podczas rozwiązywania zadania komiwojażera, a ponadto na sformułowanie następujących wniosków końcowych:

1. Pomimo stosowania najbardziej zaawansowanych metod matematycznych i ultraszybkich narzędzi komputerowych problem komiwojażera ze względu na swoją ogromną złożoność obliczeniową wynikającą z matematycznej NP-zupełności pozostaje ciągle nierozwiązany. Przynależność zadania TSP do klasy problemów NP-zupełnych powoduje, że w ogólnym przypadku nie istnieje efektywny algorytm wielomianowy dla zadania komiwojażera, a jedynie dla mniejszych rozmiarów zadania można oczekiwać rozwiązań optymalnych.
2. Wobec braku efektywnych metod algorytmicznych zadanie komiwojażera jest rozwiązywane za pomocą przybliżonych metod heurystycznych, prowadzących do uzyskania rozwiązania dopuszczalnego z określoną dokładnością. W początkowym okresie najbardziej popularne metody oparte były na teorii grafów, obrazujących różne struktury topologiczne. Niestety większość z tych metod napotykając barierę NP-zupełności nie radziła sobie z dużymi rozmiarami zadań kombinatorycznych. Dlatego do arsenału metod zostały włączone inne koncepcje heurystyczne bazujące na metodzie symulowanego wyżarzania, *Tabu Search*, wyszukiwania z nawrotami i inne.
3. Zawodność metod optymalizacyjnych, a także mała efektywność metod heurystycznych zwróciły uwagę badaczy problemu komiwojażera na bardziej miękkie metody oferowane przez burzliwie rozwijającą się sztuczną inteligencję. Pierwotnie stosowane klasyczne

metody sztucznej inteligencji, takie jak: sztuczne sieci neuronowe J. Hopfielda i algorytmy ewolucyjne J. Hollanda okazały się również mało przydatne i stosunkowo złożone przy próbach rozwiązania nietrywialnych problemów TSP, przekraczających próg kilkudziesięciu miast. Istotnym ich mankamentem jest duża trudność transformacji zadania komiwojażera do indywidualnych wymagań i specyficznych konwencji sztucznych sieci neuronowych i algorytmów genetycznych.

4. W grupie metod oferowanych przez sztuczną inteligencję najbardziej efektywne okazały się metody zaliczane do kategorii algorytmów wieloagentowych, w tym głównie algorytmy mrówkowe, zaproponowane przez M. Dorigo. Właśnie algorytmy mrówkowe okazały się najbardziej użyteczne, co wynika m.in. z ich procesualnej natury, ściśle związanej z problematyką znajdowania optymalnych tras między mrowiskiem, a miejscami żerowania. Odwieczna przyroda okazała się mistrzem w omijaniu bariery NP-zupełności, generując rozwiązania suboptymalne w atrakcyjnym czasie, przy niewielkich ograniczeniach formalnych. Wielką zaletą algorytmów mrówkowych jest stosunkowo prosty sposób zapisu problemu komiwojażera według wymagań tej teorii. Interpretacja uzyskanych wyników jest równie prosta i oczywista. Zasadniczą wadą algorytmów mrówkowych jest fakt, że w ogólnym przypadku generują jedno rozwiązanie dopuszczalne z określonym poziomem suboptymalności i dostatecznie dużą dokładnością.

Na zakończenie należy zwrócić uwagę, że problem komiwojażera TSP jest w aspekcie proceduralnym typowym zadaniem decyzyjnym, z jakim niemal codziennie muszą radzić sobie przede wszystkim menedżerowie logistyki i w tym sensie jest to klasyczne zadanie logistyczne. Z drugiej jednak strony, efektywne jego rozwiązanie leży poza sferą logistyki i już dla niewielkich rozmiarów danych wejściowych wymaga wykorzystania zaawansowanych metod kombinatoryki, teorii algorytmów, teorii grafów, a także szczegółowych metod i algorytmów programowania matematycznego, a ostatnio innowacyjnych metod i algorytmów sztucznej inteligencji. Burzliwy rozwój nauki, technologii i życia gospodarczego wymaga posługiwania się coraz bardziej zaawansowanymi metodami badawczymi, których szczególnym przypadkiem jest NP-zupełny problem komiwojażera, czekający ciągle na formalne, naukowe rozwiązanie.

LITERATURA

Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J. (2007), The Traveling Salesman Problem. A Computational Study. Lanchester Prize, Princeton University Press.

Bubal, A.T., Lee, L.S. (2016), Differential Evolution for Urban Transit Routing Problem, Journal of Computer and Communications, 4/2016.

Christofides, N. (1976), Worst-case analysis of a new heuristic for the travelling salesman problem, Report 388. Graduate School of Industrial Administration, CMU.

Clerc, M., (2004). Discrete Particle Swarm Optimization Illustrated by the Travelling Salesman Problem. Springer Berlin.

Cook, W.J. (2012), In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation. Princeton University Press.

Dantzig, G., Fulkerson, R., Johnson, S. (1954), Solution of a Large-Scale Traveling – Salesman Problem. RAND Corp. Santa Monica.

Davendra, D. (2010), Traveling Salesman Problem, Theory and Applications. InTech.

Dorigo, M., Maniezzo, V., Colorni, A. (1996), The Ant System: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26 (1).

Dorigo, M., Stützle, T., Ant Colony Optimization. MIT Press. 2004.

Ficoń, K. (2013), Sztuczna inteligencja. Nie tylko dla humanistów. BEL Studio Warszawa.

Ficoń, K. (2015), Dualizm logistyczno-kombinatoryczny zadania komiwojażera. ZN WAT SLW 42/2015.

Glover, F. (1990), Tabu Search. Part 1,2, ORSA Journal on Computing.

Hall, S.N. (2012), A Group Theoretic Tabu Search Approach to the Traveling Salesman Problem. Biblioscholar.

Holland, J.H. (1975), Adaptation in Natural and Artificial Systems. University of Michigan Press.

Huibin, J., Mingguang, L. (2009). An Improved Differential Evolution Algorithm for Optimization. IITA International Conference on Control, Automation and Systems Engineering.

Kennedy, J., Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of the 1995 IEEE International Conference on Neural Networks, IEEE Press.

Kulikowski, J.J. (1986). Zarys teorii grafów. Zastosowania w technice. PWN Warszawa.

Lisowski, J. (2017), Metody optymalizacyjne. Wyd. AM Gdynia.

Mańdziuk, J. (2000), Sieci neuronowe typu Hopfielda. Teoria i przykłady zastosowań. AOW EXIT Warszawa.

Metropolis, N. et al. (1953). Equation of State Calculations by Fast Computing Machines. The Journal of Chemical Physics 21 (6): 108.

Michalewicz, Z. (2006) i in., Adaptive Business Intelligence. Springer-Verlag Berlin.

Nilsson, C. (2003), Heuristics for the Traveling Salesman Problem, Department of Computer Science, Linkoping University.

Ramani, G. (2011), Travelling Salesman Problem (TSP) optimization through Genetic Algorithm. Improvised solution to VLSI Detailed Routing and National Tour Paperback – September 8/2011.

Reinelt, G. (1994), The Traveling Salesman. Computational Solutions for TSP Applications.
Robinson, J. B. (1949), On the Hamiltonian Game. A Traveling – Salesman Problem. RAND Corp. Santa Monica.