

WPŁYW PARAMETRÓW ALGORYTMU MRÓWKOWEGO NA EFEKTYWNOŚĆ POŁĄCZEŃ W SIECIACH

Beata Iżuk¹, Maciej Piechowiak²

¹ Uniwersytet Kazimierza Wielkiego, Dział Informatyzacji, Chodkiewicza 30, 85-064 Bydgoszcz

² Uniwersytet Kazimierza Wielkiego, Instytut Informatyki, Chodkiewicza 30, 85-064 Bydgoszcz

Streszczenie: W artykule zdefiniowano algorytm mrówkowy ACO, jego miejsce wśród dziedzin sztucznej inteligencji oraz przedstawiono jego przykładowy przebieg. Scharakteryzowano także parametry algorytmu mrówkowego i przeprowadzono testy zachowania mrówek w przykładowych grafach. W szczególności omówiono wpływ każdego parametru na sposób wyznaczania najkrótszych ścieżek w grafach o różnych rozmiarach. Badania symulacyjne przedstawiono dla czterech zestawów parametrów ACO.

Słowa kluczowe: algorytm mrówkowy, badania symulacyjne, najkrótsze ścieżki w grafie, optymalizacja sieci

The Impact of Ant Colony Optimization Parameters on The Connections Efficiency in Networks

Abstract: The article defines Ant Colony Optimization algorithm (ACO), its place among the fields of artificial intelligence and an example of its implementation was presented. The crucial parameters of the algorithm were also characterized and tests of ant behavior in sample graphs were carried out. In particular, the influence of each parameter on the method determining the shortest paths in graphs of various sizes was discussed. Simulation tests were presented for four sets of ACO parameters.

Key words: Ant Colony Optimization, computer simulations, shortest path in graph, network optimization

1. Wprowadzenie

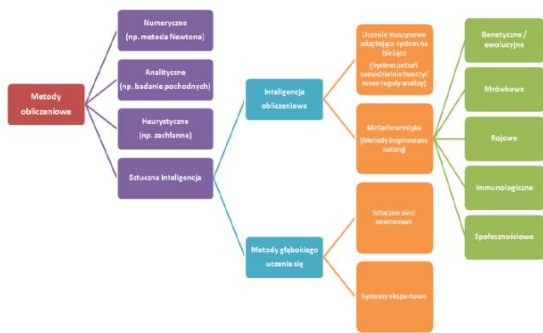
Dziedzinę zagadnień algorytmicznych zdominowały problemy NP-trudne, których dokładnego rozwiązania nie można otrzymać w czasie wielomianowym [1]. Wybierając kompromis pomiędzy czasem potrzebnym do rozwiązania problemu, a efektywnością uzyskanego wyniku, badaczom pozostają algorytmy, które w czasie wielomianowym zwracają wyniki nieoptymalne, lecz wciąż zadowalające.

Artykuł przedstawia wynik badania mającego na celu modelowanie zachowania kolonii mrówek podczas przeszukiwania grafów o różnej ilości węzłów.

2. Miejsce algorytmów mrówkowych wśród dziedzin sztucznej inteligencji

Ciężko znaleźć jasny i wyraźny, w dodatku utrwalony w świecie nauki podział metod obliczeniowych. Nadal odcina się grubą kreską inteligencję obliczeniową i sztuczną inteligencję, co nie jest jednoznacznie uzasadnione [2]. W dodatku pojęcie "inteligentne zachowanie" zawężane jest do ludzi, pomijając zwierzęta. Sztuczna inteligencja postrzegana jako działanie racjonalne obejmuje również zachowania zwierząt np. inteligencja stadna u wilków, ptaków, jak i owadów społecznych, które są uważane za ewolucyjną "kolebkę" społeczeństwa. Postępowanie zgodne z instynktem prowadzi do

osiągnięcia zamierzonego celu np. współpraca pomiędzy watahami wilków podczas polowania to nic innego jak przetwarzanie dostępnych informacji w oparciu o doświadczenia stada przekazywane wszystkim osobnikom (uczenie się młodszych osobników od bardziej doświadczonych) [3]. Starając się maksymalnie dostosować definicję "sztucznej inteligencji" do maszyn, można powiedzieć, że jest to "sprawność maszyny liczącej zdolnej odwzorować i imitować inteligentne zachowania możliwie dużej ilości inteligentnych osobników, współpracować z nimi, uczyć się oraz szukać zależności pomiędzy danymi" [4]. Odpowiedzią na potencjał natury rozumiany jako jej łatwość adaptacji i efektywność wprowadzanych przez nią zmian, stały się modele inspirowane biologią zgrupowane w pojęciu metaheurystyki. Jak sama nazwa wskazuje są to algorytmy wykraczające poza heurystyczne schematy, pozwalające rozwiązać dowolny problem, oferując własny uniwersalny schemat rozwiązania [5].



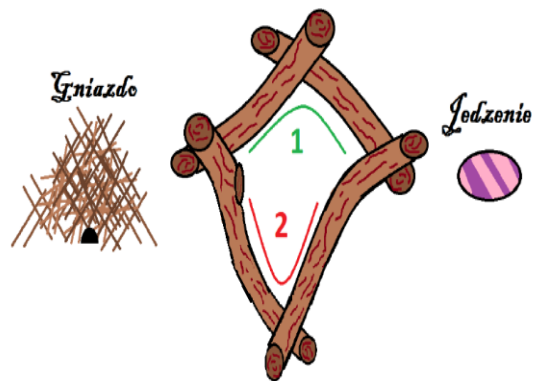
Rysunek 1. Podział dziedzin nauki zaliczanych do sztucznej inteligencji [7, 8, 9, 10].

3. Algorytm mrówkowy i jego pierwowzór

Algorytmy mrówkowe (ang. *Ant Colony Optimization*) czerpią swoją inspirację z entomologii, konkretnie myrmekologii zajmującej się badaniem zachowania mrówek [6]. Do świata maszyn przeniósł je Marco Dorigo, który za agentów przyjął sztuczne mrówki. Kluczowe dla przetrwania mrówek po wyruszeniu z mrowiska jest odnalezienie źródła pokarmu i przetransportowanie go z powrotem do mrowiska. Mrówka powinna stracić przy tym jak najmniej energii i wykonać cały proces w jak najkrótszym czasie. W wersji elektronicznej mrowisko przyjmujemy za punkt startu, a punkt stopu to źródło pokarmu. Poprzez ślad feromonu pozostawionego przez każdą mrówkę odbywa się wartościowanie krawędzi tworzących rozwiązanie problemu. Za przestrzeń rozwiązań przyjęto graf. Im miejsce w przestrzeni rozwiązań jest atrakcyjniejsze, tym wyższe stężenie feromonu będzie je cechować. Agent nie funkcjonuje w oderwaniu od czasu, gdyż feromon stopniowo odparowuje ze ścieżki i nieodnawiany w końcu zanika. W wyborze krawędzi nowy agent kieruje się

intensywnością zapachu pozostawionego feromonu oraz długością ścieżki [11].

Ilość pozostawionego przez mrówkę feromonu jest odwrotnie proporcjonalna do długości krawędzi, mierząc go w określonej jednostce czasu. Im trasa pokonana przez mrówkę jest dłuższa tym poziom jej feromonu spada i musi być częściej odświeżany, czego dowiódł eksperyment Jean-Louis'a Deneubourga zatytułowany "double bridge experiment". Polegał on na stworzeniu mrówkom dwóch ścieżek od mrowiska do pokarmu, z których jedna była dwukrotnie dłuższa od drugiej. W początkowej fazie eksperymentu owady były zainteresowane obiema drogami, jednak z czasem zainteresowanie to zaczęło maleć, aż w końcu ruch na dłuższej ścieżce zamarł. Innymi słowy krótszą krawędź pokonało więcej mrówek niż dłuższą, co pociągnęło za sobą częstsze odświeżanie feromonu na krótszej krawędzi i całkowite odparowanie go na dłuższej trasie. [12]

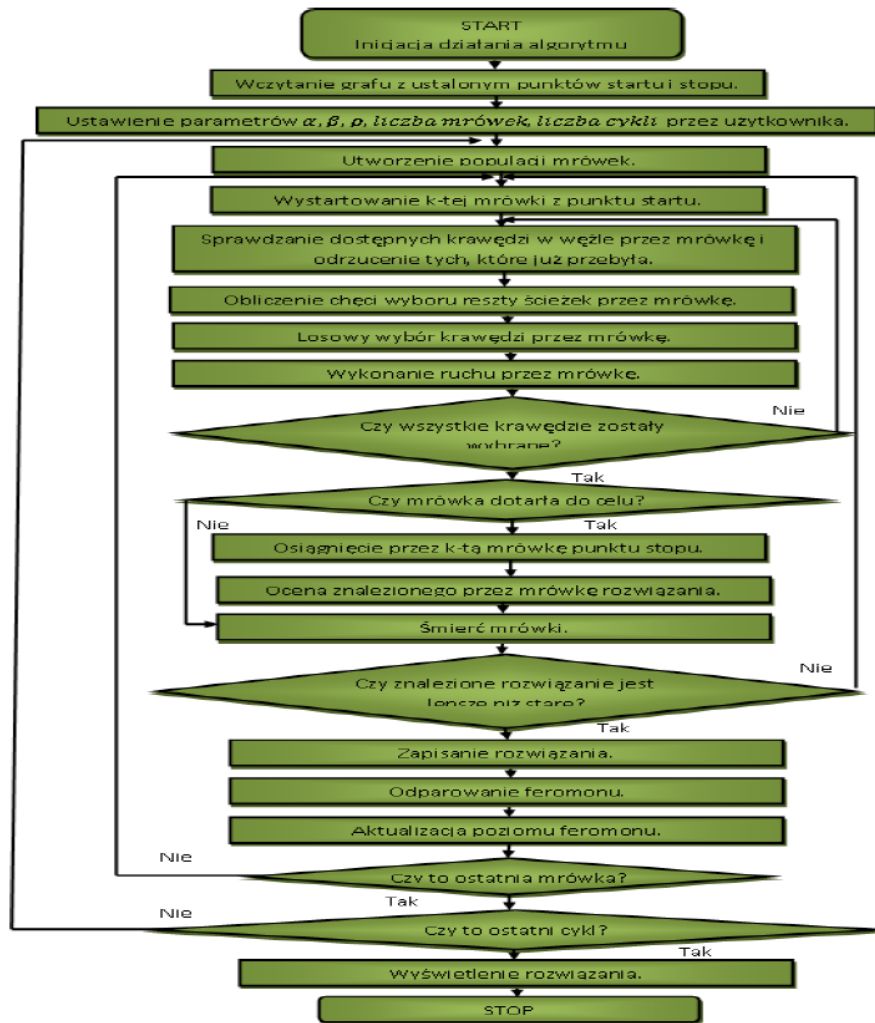


Rysunek 2. Eksperyment podwójnego mostu [12].

4. Algorytm mrówkowy i jego parametry

Istnieje wiele odmian algorytmów mrówkowych m.in.: AS (*Ant Systems*), EAS (*Elite Ant System*), ACS (*Ant Colony System*), MMAS (*Max-Min Ant System*), AS_{rank} (*Rank Based Ant System*). Każdy z nich posiada stałe elementy w postaci agentów (mrówek), krawędzi grafu, na których mrówki pozostawiają feromon, moment odświeżenia feromonu na ścieżce oraz moment jego

parowania, węzeł grafu, który stanowi miejsce podejmowania decyzji przez agenta [12, 13, 14, 16]. Za ogólny schemat działania ACO możemy przyjąć poniższy schemat (Rys. 3):



Rysunek 3. Schemat działania ACO [12, 15, 16, 17].

Parametry, które wyróżniają się w działaniu algorytmu mrówkowego można podzielić na dwie grupy – te którymi może zarządzać użytkownik algorytmu (modelować je) oraz te, które zależą od samego algorytmu (programisty) [12, 15]:

- τ ("tau") odzwierciedla proces pozostawiania śladu feromonu przez każdą mrówkę jaka przejdzie przez krawędź xy . Od tego jaką ilość feromonu mrówka pozostawią zależy będzie jak bardzo mrówka będzie skłonna wybierać krawędź xy (jest to zmienna zależna od mrówki).
- α ("alfa") to zmienna wpływająca na "chęć" mrówki, którą może zarządzać użytkownik. Należy do przedziału od zera (z zerem włącznie) do plus nieskończoności. Dzięki niej określamy jak bardzo sztuczna mrówka jest podatna na działanie feromonu (jak bardzo się go słucha przy wyborze drogi).
- η ("eta") jest tworem wprowadzonym przez danego twórcę algorytmu, odzwierciedla ona heurystykę na podstawie której już na początku

mrówka dostaje informacje na temat ścieżki. Może być to np. odwrotność długości krawędzi. Jest zmienną zależną od grafu. Natomiast użytkownik może sterować tym parametrem przez parametr β .

- β ("beta") to zmienna należąca do przedziału od 1 (z jedynką włącznie do plus nieskończoności). Dzięki niej użytkownik wpływa na parametr η .
- ρ ("rho") to zmienna dobierana zazwyczaj przez użytkownika algorytmu. Odpowiada za współczynnik parowania feromonu. Zawiera się w przedziale od zera do jeden. Przyjmuje się, że 1 oznacza całkowite odparowanie feromonu a 0 przeciwnie - feromon nie odparowuje.

Jest to związane ze wzorem wyboru trasy przez mrówkę znajdującą się na węzle grafu oraz współczynnika parowania feromonu.

$$p_{xy} = \frac{(\tau_{xy}^\alpha)(\eta_{xy}^\beta)}{\sum (\tau_{xy}^\alpha)(\eta_{xy}^\beta)}$$

τ – ilość feromonu na ścieżce xy (jego intensywność)
 η – atrakcyjność ścieżki xy ($1/d_{xy}$ gdzie d_{xy} oznacza dystans)
 α – parametr do kontroli wpływu "tau"
 β – parametr do kontroli wpływu "ety"

Rysunek 4. Wzór przedstawiający wybór trasy przez mrówkę [12].

5. Strojenie parametrów ACO

W poprzednim rozdziale przedstawiono parametry, które użytkownik może dowolnie ustawiać: α , β , ρ , liczba iteracji algorytmu oraz liczba mrówek.

Z czego pierwsze trzy parametry mają kluczowy wpływ dla powodzenia przeszukiwania przestrzeni rozwiązań. Zarówno α jak i β należą do zbioru dodatnich liczb rzeczywistych nie posiadających górnej granicy, natomiast dla ρ górną granicą jest 1.

Zadaniem mrówki jest uzyskanie jak najkrótszej drogi do celu. Podczas eksperymentu zaobserwowano, że najkorzystniejsza droga znaleziona przez mrówki mieści się w przedziale „małych” wartości parametrów α oraz β , stąd pominięto zakres „dużych” wielkości wspomnianych parametrów. Zawężono tym samym zakres przedziałów w następujący sposób: od 1,0 do 5,0 dla β i od 0,1 do 5,0 dla α , oraz dla ρ od 0,1 do 0,9.

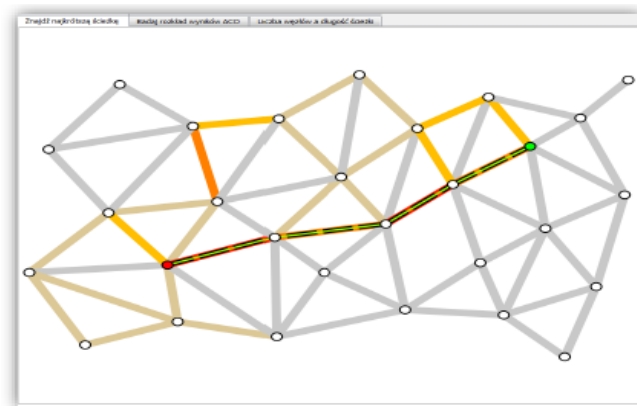
Następnie wytypowano cztery zestawy kombinacji powyższych parametrów:

TESTOWY ZESTAW PARAMETRÓW DLA ACO				
Parametry	Zestaw 1	Zestaw 2	Zestaw 3	Zestaw 4
Alfa	0,1	1,6	5,0	1,8
Beta	1,6	1,0	5,0	1,6
Ro	0,9	0,5	0,1	0,4
Iteracje	10	10	10	10
Ilość mrówek	½ węzłów	½ węzłów	½ węzłów	½ węzłów

Rysunek 5. Wartości testowanych parametrów ACO

Mrówki oddziałują na siebie wzajemnie za sprawą feromonu komunikując reszcie stopień atrakcyjności danej krawędzi. Intensywność zapachu feromonu pozostawionego przez agentów odzwierciedla wzrostowa gradacja koloru żółtego przez pomarańczowy, aż do czerwonego (najintensywniejszy).

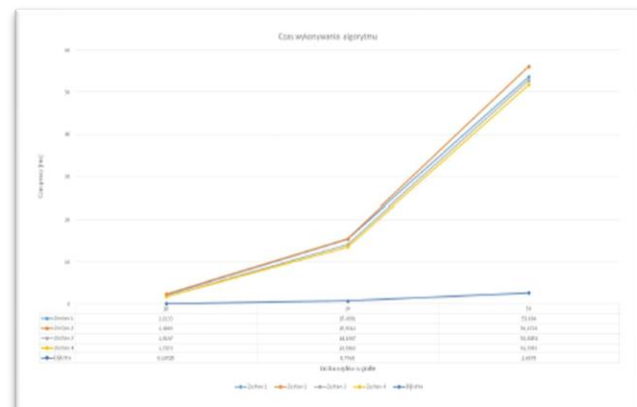
Grafy z kolei tworzyły kolejno grupy węzłów po 10 węzłów (15 krawędzi) – sieci małe, 30 węzłów (50 krawędzi) – sieci średnie i 50 węzłów (90 krawędzi) – sieci duże. Miały za zadanie imitować prawdziwą sieć urządzeń dystrybucyjnych. Punkty (miejsce znajdowania się urządzenia) i połączenia między nimi były generowane losowo, przestrzegając zasadę, iż każdy punkt musi stanowić część przynajmniej jednej krawędzi (dopuszczano grafy otwarte).



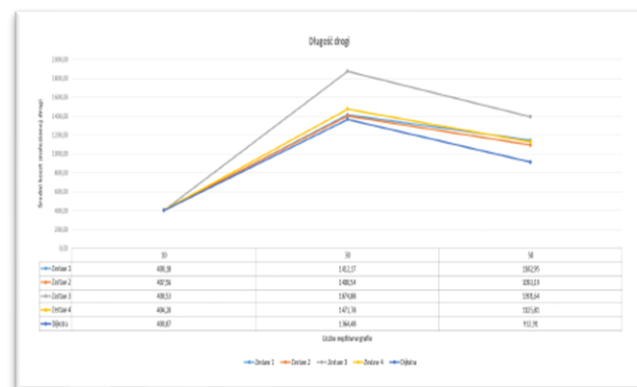
Rysunek 6. Rozkład poziomy feromonu (odcienie czerwieni) oraz najkrótsza ścieżka znaleziona przez ACO (czarna przerywana kreska, start - zielony punkt, stop czerwony punkt)

6. Efektywność algorytmu ACO

Efektywność algorytmu wyznaczały dwie wartości: czas działania algorytmu zakończony sukcesem (czas) oraz długość znalezionej najlepszej rozwiązania (droga).



Rysunek 7. Porównanie wyników parametrów ACO i algorytmu Dijkstry dla kosztów czasu

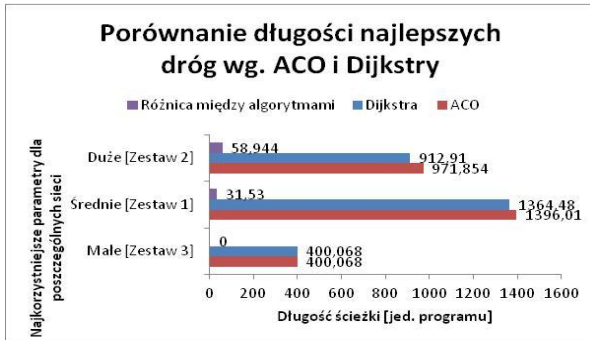


Rysunek 8. Porównanie wyników parametrów ACO i algorytmu Dijkstry dla kosztów drogi

Za wartość referencyjną dla ACO został przyjęty algorytm Dijkstry – powszechnie wykorzystywany w protokołach routingu, m. in. w sieciach komputerowych (protokół trasowania stanu łącza) [18].

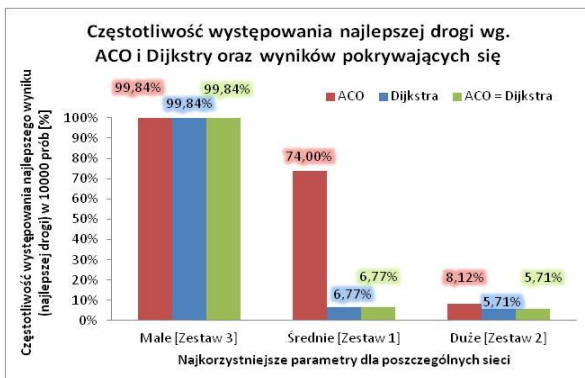
Z powyższych wykresów nie trudno zauważyć, że dla każdego z grafów można wyróżnić jeden zestaw ustawień, w którym wyniki były najkorzystniejsze.

Na Rys. 9. Przedstawiono porównanie długości faktycznej najkrótszej ścieżki (algorytm Dijkstry) oraz najkrótszej ścieżki wybranej przez ACO w poszczególnych grafach dla najkorzystniejszych parametrów.



Rysunek 9. Porównanie najkrótszych dróg wyznaczonych przez ACO i algorytm Dijkstry.

Ciekawie prezentuje się też częstotliwość wystąpienia najkorzystniejszych dróg w 10000 prób.



Rysunek 10. Porównanie ilości wystąpień najkrótszej ścieżki wskazanej przez ACO oraz algorytm Dijkstry dla 10000 prób. Przez wyniki pokrywające się należy rozumieć taką samą liczbę najkrótszych ścieżek wyznaczonych zarówno przez ACO, jak i algorytm Dijkstry.

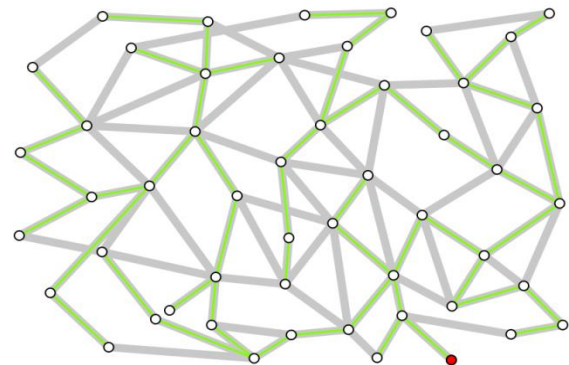
7. Plusy i minusy

Na korzyść algorytmu mrówkowego zdecydowanie przemawia jego elastyczna konfiguracja. Z dużym prawdopodobieństwem można stwierdzić, że ilość mrówek w postaci 1/2 ilości wierzchołków to ilość nadmiarowa i można tę liczbę zmniejszać nawet do 1/3, co z kolei skróci czas wykonania algorytmu mrówkowego. Badania symulacyjne wykazują jednocześnie, że ilość mrówek poniżej 1/4 zdaje się być daleko niewystarczającą.

Z kolei Algorytm Dijkstry zwiększa swój koszt pracy wraz z zwiększającą się liczbą węzłów grafu, jest tym samym bardziej stabilny i przewidywalny, jednak w przypadku badania wielu ścieżek jednocześnie (tworzenie drzewa rozpinającego) koszt uzyskany dla

jednej ścieżki wzrasta znacząco i powoduje obciążenie maszyny tymi samymi obliczeniami wykonywanymi kilkukrotnie w tym samym czasie. Wprowadzając element dynamicznego odczytu i zapisu zgromadzonych wcześniej obliczeń dodamy do Dijkstry "wąskie gardło" algorytmu ACO, co może drastycznie obniżyć jego wydajność.

Z drzewami rozpinającymi dobrze radzi sobie algorytm mrówkowy, ponieważ opiera się na losowej eksploracji przestrzeni grafu. W sensownym zakresie czasu potrafi rozwiązać problem optymalizacyjny na dużym obszarze dzieląc się zdobytymi informacjami z resztą agentów.



Rysunek. 12 Drzewo rozpinające zbudowane (ACO)

8. Podsumowanie

Algorytm Dijkstry osiągał zdecydowanie lepsze wyniki niż algorytm ACO biorąc pod uwagę czas wykonania, jednak nie jest to już tak oczywiste dla kosztów znalezionej drogi. Wyniki uzyskane przez ACO jako najlepsze niewiele odbiegają od najlepszego rozwiązania.

Kluczowe dla ACO są parametry, których dobór dla każdego rozpatrywanego problemu jest niezwykle trudny. W trakcie prowadzonych badań można dojść do wniosku, że dla małych grafów parametr β ma mniejsze wartości w stosunku do α , z kolei im problem jest reprezentowany za pomocą złożonego grafu o większej liczbie węzłów tym parametr α zyskuje na znaczeniu. Parametr ρ odpowiada za konsekwencje wyborów mrówek względem kolonii, ustawiając go na wysokie parowanie skróceniu ulega czas dostępny na odnalezienie feromonu przez mrówki, odwrotnie – jest on wydłużany.

Dla małych grafów, które posiadają zdecydowanie zróżnicowane krawędzie sprawdza się dominujący parametr β (od 1,6 do 5,0) i wysoki stopień parowania - ρ od 0,8 do 0,9. Natomiast duże oraz średnie grafy osiągają lepsze wyniki dla dominującego parametru α (od 1,6 do 1,8) i parametru β od 1,00 do 1,6. Z kolei w dużych grafach powinien występować niski stopień parowania ρ (od 0,5 do 0,4).

Reasumując, algorytmy mrówkowe mają większe szanse zastosowania w obrębie robotyki, niż sieci

komputerowych. Wiąże się to przede wszystkim z faktem, że czas rozpropagowania informacji o sieci przez wielu agentów jest zbyt długi, przy czym nie stanowi to problemu przy eksploracji nowych przestrzeni. Celem w sieciach jest przede wszystkim jak najniższy czas dostarczenia wiadomości, stąd lepszym rozwiązaniem wydają się algorytm Dijkstry. W drugim przypadku celem samym w sobie jest zbadanie nowej przestrzeni, którą można później

dostosować. ACO w połączeniu z bardziej zaawansowaną formą sztucznej inteligencji i inteligencji obliczeniowej np. głębokie sieci neuronowe ma szansę dobrze sprawdzić się w przypadku awarii sieci. Dzięki optymalizacji kolonii mrówek można w rozsądnym czasie zbadać spektrum wielu parametrów i wzajemne zależności między nimi.

Literatura

- [1] Szawdyński, P. (2005-2019). *problem NP-trudny - Znaleźnienie rozwiązania problemu nie jest możliwe ze złożonością obliczeniową wielomianową. (pojęcie)*. Pobrano z lokalizacji Serwis programistyczny C/C++: <http://cpp0x.pl/dokumentacja/problem-NP-Trudny/1219>
- [2] Duch, W. (2006, 03 14). *Dokąd zmierza inteligencja obliczeniowa?* Pobrano z lokalizacji Katedra Informatyki Stosowanej, Uniwersytet Mikołaja Kopernika: <https://fizyka.umk.pl/publications/kmk/03-CI-przyszlosc.pdf>
- [3] Dul, F. (2014). *POLITECHNIKA WARSZAWSKA WYDZIAŁ MECHANICZNY ENERGETYKI I LOTNICTWA*. Pobrano z lokalizacji WPROWADZENIE DO SZTUCZNEJ INTELIGENCJI NS 586: <https://docplayer.pl/15447888-Wprowadzenie-do-sztucznej-inteligencji.html>
- [4] Horzyk, A. (2013/2014). *Akademia Górniczo-Hutnicza Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej*. Pobrano z lokalizacji METODY ROZWIĄZYWANIA PROBLEMÓW ORAZ MODELOWANIE WIEDZY I INTELIGENCJI W INFORMATYCE: <http://home.agh.edu.pl/~horzyk/lectures/wdi/WDI-WiedzaInteligencja.pdf>
- [5] Wikipedia. (2020, 1 2). *Metaheuristic*. Pobrano z lokalizacji Wikipedia, the free encyclopedia: <https://en.wikipedia.org/wiki/Metaheuristic>
- [6] Sołtys, K. (2008, 05). *Mrówki, czyli piękno metaheurystyk*. Pobrano z lokalizacji [deltami.edu.pl: http://www.deltami.edu.pl/temat/matematyka/zastosowania/2010/12/16/Mrowki_czyli_piekno_metaheuryst/](http://www.deltami.edu.pl/temat/matematyka/zastosowania/2010/12/16/Mrowki_czyli_piekno_metaheuryst/)
- [7] Wikipedia. (2018, 6 23). *Obliczenia inteligentne*. Pobrano z lokalizacji Wikipedia, the free encyclopedia: https://pl.wikipedia.org/wiki/Obliczenia_inteligentne
- [8] Globema. (2017, 06 19). *KONFERENCJA 2017: SZTUCZNA INTELIGENCJA W PRAKTYCZNYCH ZASTOSOWANIACH*. Pobrano z lokalizacji Globema: <https://www.globema.pl/konferencja-globemy-2017-sztuczna-inteligencja-w-praktycznych-zastosowaniach/>
- [9] Oracle. (brak daty). *Sztuczna inteligencja (SI) od Oracle — czym jest sztuczna inteligencja?* Pobrano z lokalizacji Czym jest sztuczna inteligencja?: <https://www.oracle.com/pl/artificial-intelligence/what-is-artificial-intelligence.html>
- [10] Globema. (2017, 06 19). *KONFERENCJA 2017: SZTUCZNA INTELIGENCJA W PRAKTYCZNYCH ZASTOSOWANIACH*. Pobrano z lokalizacji Globema: <https://www.globema.pl/konferencja-globemy-2017-sztuczna-inteligencja-w-praktycznych-zastosowaniach/>
- [11] Flasiński, M. (2011). *Wstęp do sztucznej inteligencji*. Warszawa: Wydawnictwo Naukowe PWN.
- [12] Dorigo, M., Birattari, M. i Stützle, T. (2006, 09). *IEEE COMPUTATIONAL INTELLIGENCE MAGAZIN*. Pobrano z lokalizacji Ant Colony Optimization – Artificial Ants as a Computational Intelligence Technique: <https://courses.cs.ut.ee/all/MTAT.03.238/2011K/uploads/Main/04129846.pdf>
- [13] Bullnheimer, B., F.Hartl, R. i Strauß, C. (1997, 06). *A New Rank Based Version of the Ant System - A Computational Study*. Pobrano z lokalizacji Wirtchaftsuniversitat Wien: <https://epub.wu.ac.at/616/1/document.pdf>
- [14] Karmakar, R. B., Mitra, R. B., Dey, A. B., Chakraborty, V. B. i Nayak, A. B. (2016). *Solving TSP Using Improved Elitist - Ant System Based on Improved Pheromone - Strategy and Dynamic Candidate List*. Pobrano z lokalizacji MAYFEB Journal of Electrical and Computer Engineering Vol 2 - Pages 8-15: <http://webcache.googleusercontent.com/search?q=cache:Gg0YaTyQ0awJ:www.mayfeb.com/OJS/index.php/CON/article/download/543/221+&cd=6&hl=pl&ct=clnk&gl=pl&client=firefox-b-d>
- [15] Boryczka, U. (2006). *Algorytmy optymalizacji mrowiskowej*. Katowice: Wydawnictwo Uniwersytetu Śląskiego.
- [16] http://people.idsia.ch/%7Eluca/ij_23-alife99.pdf
- [17] Jakubowska, A. i Piechocka, K. (2015). *W POSZUKIWANIU OPTYMALNEJ TRASY – WYBRANE ALGORYTMY W ZASTOSOWANIU DO PROBLEMU KOMIWOJAŻERA*. Pobrano z lokalizacji JOURNAL OF TRANSLOGISTICS: <http://translogistics.pl/files/jtl/2015/R1.pdf>
- [18] Graziani, R. i Johnson, R. (2008). *Akademia sieci Cisco, CCNA Exploration, Semestr 2, Protokoły i koncepcje routingu*. Warszawa: Wydawnictwo Naukowe PWN.