

# Zastosowanie algorytmu mrówkowego do wyznaczania przydziału pojazdów do zadań w miejskim transporcie zbiorowym<sup>1,2</sup>

**MARIUSZ IZDEBSKI**

dr hab. inż., prof. PW, Politechnika Warszawska, Wydział Transportu, 00-662 Warszawa, ul. Koszykowa 75, mizdeb@wt.pw.edu.pl

**MARIANNA JACYNA**

prof. dr hab. inż., Politechnika Warszawska, Wydział Transportu, 00-662 Warszawa, ul. Koszykowa 75, maja@wt.pw.edu.pl

**PIOTR KISIELEWSKI**

dr hab. inż., prof. PK, Politechnika Krakowska, Wydział Mechaniczny, 31-155 Kraków, ul. Warszawska 24, pkisielewski@pk.edu.pl

**Streszczenie:** W artykule opisano zastosowanie algorytmu mrówkowego w wyznaczaniu przydziału pojazdów do zadań w transporcie zbiorowym. Analizowany problem przydziału jest złożonym zagadnieniem optymalizacyjnym, klasyfikującym go do problemów NP-trudnych. W obszarze dotyczącym zagadnień miejskiego transportu zbiorowego jest podstawowym problemem, który należy rozwiązać w procesie konstruowania rozkładów jazdy oraz planów pracy pojazdów i kierowców. Celem niniejszej publikacji było opracowanie nowego narzędzia optymalizacyjnego adekwatnego do analizowanego zagadnienia przydziału pojazdów do zadań w komunikacji miejskiej. Przedstawiony algorytm mrówkowy jest nowym podejściem zastosowanym do rozwiązywania zagadnień przydziału w transporcie zbiorowym i stanowi podstawę do dalszych badań nad tematyką opracowywania nowych metod optymalizacyjnych w badanym problemie. Opracowany algorytm minimalizuje liczbę pojazdów przy jednoczesnej minimalizacji czasu pracy oraz przebytej drogi przez wykorzystane pojazdy. Opracowano model matematyczny zagadnienia przydziału pojazdów do zadań w transporcie publicznym, tj. zdefiniowano zmienne decyzyjne, ograniczenia oraz funkcje kryterium. Ograniczenia przydziału wynikają z czasu realizacji kursów w danym dniu roboczym, ograniczeń prawnych w zakresie czasu pracy i jazdy kierowcy, a także dostępnej liczby pojazdów. Problem został przedstawiony w aspekcie wielokryterialnym, gdzie decydujące znaczenie w ocenie efektywnego przydziału mają czas i dystans pokonany przez wszystkie pojazdy realizujące zlecone zadania. W artykule przedstawiono ogólną koncepcję algorytmu mrówkowego, która jest w trakcie procesu weryfikacji na danych teoretycznych i rzeczywistych bazach danych przedsiębiorstw komunikacji miejskiej.

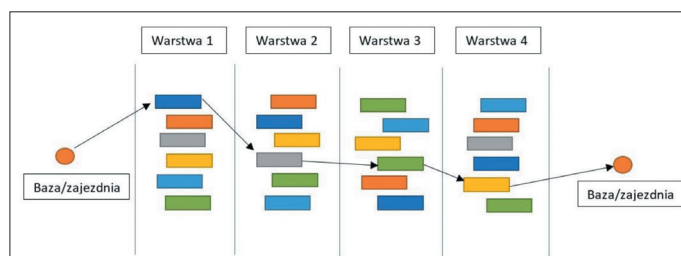
**Słowa kluczowe:** transport publiczny, przydział pojazdów do zadań, algorytm mrówkowy.

## Wprowadzenie

Problem przydziału pojazdów do zadań w transporcie zbiorowym jest zagadnieniem znanym i szeroko omawianym w literaturze przedmiotu [1-5]. Problem nawiązuje do ogólnego zagadnienia przydziału opisywanego w teorii grafów czy badaniach operacyjnych [6-7] oraz jest osadzony w ogólnej problematyce przydziału pojazdów do zadań, analizowaną w przedsiębiorstwach transportowych [9-20].

W obszarze dotyczącym zagadnień transportu zbiorowego prezentowane zagadnienie jest podstawowym problemem, który należy rozwiązać w procesie konstruowania

rozkładów jazdy i planów pracy kierowców [21-32]. Problem przydziału pojazdów do zadań polega na efektywnym przypisaniu pojazdów wraz z kierowcami do zleconych zadań przewozowych. Zadania przewozowe w transporcie publicznym zdefiniowane są jako kursy rozkładu jazdy, które można podzielić bardzo szybkim algorytmem na warstwy/poziomy [1,21], co schematycznie przedstawiono na rysunku 1. Efektywność przydziału wynika z opracowanej funkcji kryterium, która może określać minimalny koszt realizacji wszystkich kursów lub czas ich realizacji. Ograniczenia przydziału wynikają z czasu realizacji kursów w danym dniu roboczym, ograniczeń czasu pracy i jazdy kierowcy, a także dostępnej liczby pojazdów. Szukając rozwiązania problemu, dąży się do minimalizacji liczby pojazdów i kierowców użytych do wykonania wszystkich kursów w danym dniu roboczym.



Rys. 1. Schemat podziału kursów na warstwy z przykładową trasą pojazdu

Problem przydziału pojazdów do zadań jest problemem NP-trudnym, dlatego też w celu jego wyznaczenia wskazane jest zastosowanie zaawansowanych algorytmów optymalizacyjnych. Analizowany problem przydziału rozwiązywany jest różnymi metodami i algorytmami, np. za pomocą programowania liniowego [33,34], algorytmów zachłannych [35], Tabu Search [36,37], algorytmów genetycznych [38] i ewolucyjnych [21], wyżarzania [39] czy roju cząstek [40]. W większości przypadków rozpatrywany jest jednokryterialne, rzadziej jako wielokryterialne [21] zadanie optymalizacyjne.

Celem artykułu jest opracowanie nowego narzędzia optymalizacyjnego dostosowanego do wielokryterialnego zagadnienia przydziału pojazdów do zadań w transporcie zbiorowym. Zastosowanie algorytmu mrówkowego jest unikalnym podejściem w wyznaczaniu przydziału i wzbogaca ono istniejące metody i narzędzia optymalizacyjne

<sup>1</sup> ©Transport Miejski i Regionalny, 2020. Procentowy udział wkładu autorów w publikację: M. Izdebski 34%, M. Jacyna 33%, P. Kisielewski 33%.

<sup>2</sup> Badania przeprowadzone w artykule są realizowane w ramach projektu NCBR/POIR.01.01.01-00-0970/17-00 „System informatyczny komputerowego wspomaganie planowania komunikacji miejskiej”

stosowane przy tego typu problemach. Problem przydziału został przedstawiony w ujęciu wielokryterialnym, w którym uwzględniono kryterium czasu i przebytego dystansu w realizacji wszystkich zleconych kursów. Na potrzebę opracowania algorytmu mrówkowego zdefiniowano model matematyczny przydziału pojazdów do zadań, tj. określono dane wejściowe, zmienne decyzyjne, ograniczenia oraz funkcje kryterium. Opracowany model matematyczny oraz algorytm mrówkowy mogą stanowić punkt odniesienia do kolejnych badań nad analizowaną tematyką.

### Model matematyczny przydziału pojazdów do zadań

W celu zaimplementowania algorytmu mrówkowego opracowano model matematyczny problemu przydziału pojazdów do zadań (kursów). Model matematyczny odnosi się do przydziałów wykonanych w danym dniu roboczym. W celu zdefiniowania modelu określono następujące dane wejściowe:

- zbiór kursów do zrealizowania w danym dniu roboczym,  $K = \{1, \dots, k, \dots\}$ ,
- zbiór pojazdów,  $P = \{1, \dots, p, \dots\}$ ,
- zbiór typów pojazdów,  $R = \{1, \dots, r, \dots\}$ ,
- wektor określający liczbę pojazdów określonego typu gotowych do przydziału w danym dniu roboczym,  $N = [n(r): n(r) \in \mathbb{N}, r \in R]$ ,
- wektor określający czas trwania danego kursu,  $T1 = [t1(k): t1(k) \in \mathbb{R}^+, k \in K]$ ,
- macierz określającą czas jazdy pojazdu pomiędzy dwoma kursami,  $T2 = [t2(p, r, k, k'): t2(p, r, k, k') \in \mathbb{R}^+, p \in P, r \in R, k \in K, k' \in K]$ ,
- macierz określającą dystans pomiędzy dwoma kursami,  $D = [d(k, k'): d(k, k') \in \mathbb{R}^+, k \in K, k' \in K]$ ,
- $tdop$  – dopuszczalny czas realizacji zadań,
- $tpracy$  – dopuszczalny czas pracy kierowcy/kierowców.

Zmienne modelu matematycznego przyjmują postać wektorów:

- Zmienna określająca typ pojazdu przypisany do danego kursu –  $Z = [z(r, k): z(r, k) \in \{0, 1\}, r \in R, k \in K]$ ,
- Podstawowa niezależna zmienna decyzyjna określająca przypisanie pojazdu określonego typu do kursu –  $X = [x(p, r, k): x(p, r, k) \in \{0, 1\}, p \in P, r \in R, k \in K]$ ,
- Zmienna określająca czas pracy pojazdu w danym dniu roboczym –  $T = [t(p, r): t(p, r) \in \mathbb{R}, p \in P, r \in R]$ ,
- Zmienna określająca liczbę pojazdów danego typu przypisanych do kursów w danym dniu roboczym –  $Y = [y(r): y(r) \in \mathbb{N}, r \in R]$ ,
- Zmienna określająca połączenie między kursami realizowane danym pojazdem –  $L = [l(p, r, k', k): l(p, r, k', k) \in \{0, 1\}, p \in P, r \in R, k \in K, k' \in K]$ ,

Ograniczenia przyjmują następującą formę:

- Ograniczenie wynikające z czasu realizacji wszystkich zadań: suma czasów realizacji wszystkich kursów oraz dojazdów do tych kursów nie może przekraczać dopuszczalnego czasu realizacji zadań:

$$\sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \alpha(r, k) \cdot x(p, r, k) \cdot t1(k) + \sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \sum_{k' \in K} l(p, r, k, k') \cdot t2(p, r, k, k') \leq tdop \quad (1)$$

- ograniczenie wynikające z czasu pracy kierowców/kierowcy:

$$\forall p \in P, \forall r \in R$$

$$\sum_{k \in K} \alpha(r, k) \cdot x(p, r, k) \cdot t1(k) + \sum_{k \in K} \sum_{k' \in K} l(p, r, k, k') \cdot t2(p, r, k, k') \leq tpracy \quad (2)$$

- całkowity czas pracy pojazdów (kierowców) musi być równy bądź większy od całkowitego czasu realizacji zadań:

$$\sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \alpha(r, k) \cdot x(p, r, k) \cdot t1(k) + \sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \sum_{k' \in K} l(p, r, k, k') \cdot t2(p, r, k, k') \leq \sum_{r \in R} y(r) \cdot t(p, r) \quad (3)$$

- ograniczenie liczby pojazdów danego typu wykorzystanych do realizacji zadań:

$$\forall r \in R \quad y(r) \leq n(r) \quad (4)$$

Funkcje kryteriów minimalizujących czas realizacji zadań (5) i odcinki dojazdowe do poszczególnych zadań (6) przedstawiono odpowiednio:

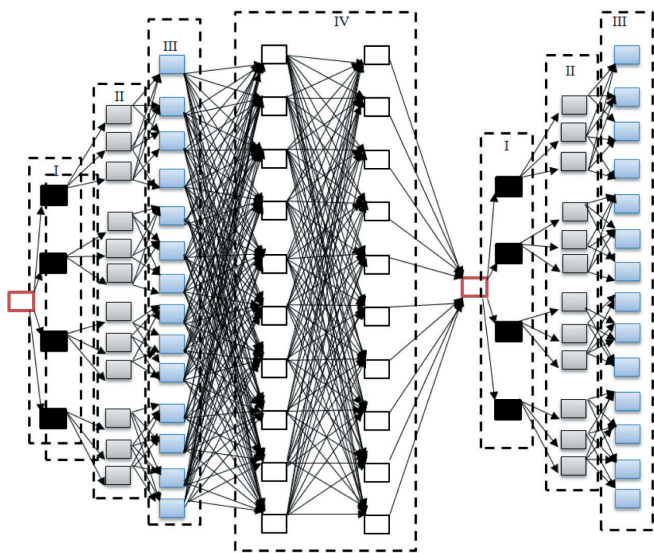
$$F(Z, X, L) =$$

$$\sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \alpha(r, k) \cdot x(p, r, k) \cdot t1(k) + \sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \sum_{k' \in K} \alpha(r, k) \cdot l(p, r, k, k') \cdot t2(p, r, k, k') \longrightarrow \min \quad (5)$$

$$F(L) = \sum_{p \in P} \sum_{r \in R} \sum_{k \in K} \sum_{k' \in K} l(p, r, k, k') \cdot d(k, k') \longrightarrow \min \quad (6)$$

### Algorytm mrówkowy przydziału pojazdów do zadań

W celu sprecyzowania działania algorytmu mrówkowego należy zdefiniować pojęcie „trasy mrówki”. Trasa pojedynczej mrówki składa się ze wszystkich kursów, jakie są realizowane na wszystkich każdych liniach w dany dzień roboczy. Punktem startowym trasy mrówki jest zajezdnia. Trasa pojedynczej mrówki składa się z czterech warstw, które występują kolejno po sobie jak na rysunku 2. W czasie konstruowania trasy w I warstwie ustawiana jest zmienna decyzyjna dotycząca użytego typu pojazdu realizującego zadania przewozowe. Liczba elementów punktowych tej warstwy uzależniona jest od liczby dostępnych typów pojazdów. W II warstwie dobierana jest kolejna zmienna decyzyjna modelu matematycznego, tj. liczba pojazdów określonego typu, który został wybrany w warstwie I. W warstwie III ustalana jest zmienna decyzyjna dotycząca czasu zaangażowania danego typu środka transportu. Czas zaangażowania uzależniony jest od liczby kierowców (służb) w brygadzie, tj. pojedyncza obsada lub podwójna brygada. Brygada rozumiana jest jako dzienne zadanie pojazdu. Liczba elementów punktowych tej warstwy ograniczona jest dopuszczalnym czasem pracy kierowcy i określa godziny jego pracy, np. dziewięć elementów w warstwie oznacza pierwszy element – 1h pracy kierowcy, drugi element 2h pracy, dziewiąty element – pierwsza godzina pracy



Rys. 2. Warstwy trasy mrówek

drugiego kierowcy. W warstwie IV wyznaczane są zmienne decyzyjne określające połączenia między poszczególnymi kursami. Mrówka przechodzi przez poszczególne elementy warstwy IV do momentu zakończenia realizacji wszystkich kursów.

Przykładową trasę mrówki można przedstawić graficznie jak na rysunku 3. Kursy są realizowane w odpowiedniej kolejności: 1,4,3,7,5,6. Moc zainstalowana do zrealizowania tych kursów to 3 pojazdy typu pierwszego, każdy pojazd obsadzony pojedynczym kierowcą, czas pracy każdego kierowcy 5 h.

Baza	1 typ	3 poj.	5h	Kurs 1	Kurs 4	Kurs 3	Kurs 7	Kurs 5	Kurs 6	Baza
------	-------	--------	----	--------	--------	--------	--------	--------	--------	------

Rys. 3. Trasa mrówki

Mrówka wraca do bazy z trasy w przypadku, gdy czas realizacji kursów dobiega końca lub wszystkie kursy zostały już zrealizowane. Liczba pojazdów wyjeżdżających z bazy uzależniona jest od czasu trwania kursów wyznaczonych przez mrówkę, np. w przypadku, gdy czas wszystkich kursów przedstawionych na rysunku 3 wynosi 13 h, do realizacji tych kursów należy przydzielić 3 pojazdy. Po powrocie do bazy mrówka rozpoczyna tworzenie nowej trasy z kolejno niezrealizowanych kursów.

Głównym zadaniem mrówki jest wyznaczenie minimalnej trasy składającej się ze wszystkich kursów. Trasa każdej mrówki reprezentowana jest przez ciąg liczb naturalnych określających numery kursów (na potrzeby implementacji algorytmu kurs rozumiany jest jako przejazd autobusu od pętli do pętli). Pod pojęciem minimalnej trasy należy rozumieć trasę charakteryzującą się minimalnym czasem oczekiwania na następny kurs i dystansem do pokonania w przypadku dojazdu pojazdu do innej pętli, niż pętla w której skończył realizowany kurs. Minimalna trasa rozważana jest więc w ujęciu wielokryterialnym, tj. decydujące znaczenie ma czas i dystans dojazdu do następnego kursu. Trasa ta oceniana jest przez znormalizowany (uwzględniający zarówno dystans i czas dojazdu) współczynnik (wzór 9), wyznaczany w każdym kroku przejścia mrówki pomiędzy kursami.

W trakcie realizowanej trasy przez pojedynczą mrówkę wyznaczane są pojazdy i kierowcy. Każda mrówka jest więc reprezentantem określonej liczby pojazdów i kierowców. Liczby te w każdej trasie mogą być różne i są określane na początku tworzenia trasy mrówki.

Na potrzebę opisu elementów składowych algorytmu zdefiniowano zbiór punktów trasy realizowanej przez pojedynczą mrówkę  $WTP$ , zbiór mrówek w mrowisku  $MR$ , tj.:

$$WTP = \{1, \dots, y, \dots, z, \dots, \overline{WTP}\} \quad (7)$$

gdzie  $\overline{WTP}$  – licznosc zbioru  $WTP$ ,  $y \neq z$ .

Punktem startowym trasy każdej mrówki jest zajezdnia autobusowa. Kursy w każdej trasie mrówki wybierane są zgodnie z określonym prawdopodobieństwem [41, 42]:

$$PR^{mr}_{yz}(t) = \begin{cases} \frac{[\tau_{yz}(t)]^\alpha \cdot [\eta_{yz}(t)]^\beta}{\sum_{l \in \Omega^{mr}} [\tau_{yl}(t)]^\alpha \cdot [\eta_{yl}(t)]^\beta}, & z \in \Omega^{mr} \\ 0, & z \notin \Omega^{mr} \end{cases} \quad (8)$$

gdzie:

- $\tau_{yz}(t)$  – natężenie śladu feromonu między  $y$ -tym punktem trasy a  $z$ -tym punktem w  $t$ -iteracji, na początku (w pierwszej iteracji algorytmu), gdy jeszcze nie ma wyznaczonego śladu przez mrówki przyjmujemy wartość 1 na każdej relacji,
- $\eta_{yz}(t)$  – informacja heurystyczna ukazuje aspekt przejścia mrówki do kolejnych punktów trasy np. jak w poniższym wzorze (9):

$$\eta_{yz}(t) = \frac{w_1}{t_2(z) - t_1(y)} + \frac{w_2}{d(y,z)} + \frac{w_3}{t(z)} \quad (9)$$

gdzie:

- $t_1(y)$  – czas zakończenia kursu  $y$ -tego,
- $t_2(z)$  – czas rozpoczęcia kursu  $z$ -tego,
- $d(y,z)$  – dystans pomiędzy pętlami w przypadku kursów zaczynających się w różnych pętlach; wzór dotyczy punktów trasy określających kursy w trasie,
- $t(z)$  – czas trwania kursu  $z$  (realizowanie w pierwszej kolejności kursów krótkich może przyczynić się do minimalizacji liczby służ),
- $\alpha, \beta$  – wpływ feromonów i danych heurystycznych na zachowanie mrówek,
- $w_1, w_2, w_3$  – współczynniki wagowe o wymiarze czasu lub odległości,
- $\Omega^{mr}$  – zbiór nieodwiedzanych (niezrealizowanych) dotychczas  $l$  kursów przez  $mr$ -tą mrówkę.

Każde przejście mrówki wiąże się ze sprawdzeniem warunku na fizyczną możliwość przejścia mrówki do kolejnego kursu ( $v$ - średnia prędkość pojazdu):

$$t_2(z) - t_1(y) \geq \frac{d(y,z)}{v} \quad (10)$$

W przypadku spełnienia warunku informacja heurystyczna przyjmuje wartość ze wzoru (9), w przeciwnym przypadku wartość 0. Powyższy warunek jest realizowany w algorytmie za pomocą macierzy przejść. Macierz przejścia jest macierzą binarną określającą możliwość przejścia mrówki do kolejnego kursu: wartość elementu macierzy 1, w innym przypadku 0.

Po zakończeniu budowania trasy przez pojedynczą mrówkę należy sprawdzić warunek na poprawność wygenerowanej trasy. Czas zaangażowania kierowców w realizację kursów musi być większy bądź równy sumie czasów trwania kursów wraz z przerwami międzykursowymi i dojazdami w każdej relacji wyjazd z bazy – powrót do bazy. W przypadku niespełnienia warunku etap tworzenia trasy przez mrówkę należy powtórzyć. Warunek ten jest sprawdzany dla każdej trasy od wyjścia mrówki z bazy do jej powrotu.

Po zakończeniu budowania tras przez wszystkie mrówki następuje aktualizacja śladu feromonu. W procesie zastosowano aktualizację cykliczną jako najbardziej wydajną spośród stosowanych aktualizacji gęstościowej czy ilościowej. Na początku przyjmuje się, iż ślad na połączeniach między kursami jest jednakowo silny np. 1. W kolejnych iteracjach ślad feromonu obliczany jest według wzoru  $\tau_{yz}(t+1)$ , wartość feromonu jest odkładana w każdej iteracji na danym odcinku trasy  $yz$ , ślad jest odkładany na odcinkach przez które przechodzi mrówka:

$$\tau_{yz}(t+1) = (1 - \rho)\tau_{yz}(t) + \sum_{mr=1}^{MR} \Delta\tau_{yz}^{mr}(t) \quad (11)$$

gdzie:

- $mr$  – kolejna mrówka w mrowisku  $mr \in MR$ ,
- $\rho$  – współczynnik ulatniania się feromonów ( $0 < \rho \leq 1$ ),
- $\tau_{yz}(t)$  – wzmocnienie feromonu, dla pierwszej iteracji przyjmuje wartość na każdym połączeniu równą  $\tau_0$  .np. 1.

Pierwszy składnik wzoru 11 określa szybkość ulatniania feromonu  $(1 - \rho)\tau_{yz}(t)$ , natomiast drugi określa wzmocnienie feromonu i przyjmuje wartość:

$$\Delta\tau_{yz}^{mr}(t) = \begin{cases} \left( \frac{1}{T^{mr}(t)} + \frac{CK}{CZ} \right) & \text{gdy trasa } (y,z) \text{ była użyta} \\ & \text{przez mrówkę } mr \\ 0 & \text{w przeciwnym wypadku } 0 \end{cases} \quad (12)$$

gdzie:

- $T^{mr}(t)$  – całkowity czas i dystans dla danej trasy przebytej w  $t$ -ej iteracji przez  $mr$ -tą mrówkę, jeżeli czas i dystans jest minimalny iloraz  $\frac{1}{T^{mr}(t)}$  rośnie, więc ta trasa jest atrakcyjna dla kolejnych mrówek,
- CK – całkowity czas trwania wszystkich kursów w trasie mrówki,
- CZ – całkowity czas zaangażowania wszystkich pojazdów wygenerowanych w trasie mrówki.

Iloraz w przypadku osiągnięcia wartości zbliżonej do jedności, określi minimalną liczbę pojazdów (brygad) po-

trzebnych do realizacji kursów. W przypadku, gdy iloraz będzie dążył do zera, trasa taka jest trasą mało atrakcyjną dla kolejnych mrówek.

Jeśli odcinek tras  $(y, z)$  był realizowany przez  $mr$ -tą mrówkę to  $\Delta\tau_{yz}^{mr}(t)$  wynosi  $\left( \frac{1}{T^{mr}(t)} + \frac{CK}{CZ} \right)$ , w innym przypadku 0.

Każda trasa pojedynczej mrówki składa się z wszystkich kursów przeznaczonych do realizacji w danym dniu roboczym. Przyjęto założenie, że pojedynczy kurs charakteryzuje się takimi parametrami, jak godzina rozpoczęcia kursu, zakończenia oraz czas jazdy w danym kursie.

Kroki algorytmu można przedstawić jako:

- Krok 1. Zdefiniowanie liczby mrówek w mrowisku oraz liczby iteracji. Algorytm mrówkowy jest algorytmem iteracyjnym, w każdej iteracji wszystkie mrówki tworzą własne trasy w oparciu o feromon zgromadzony na nich.
- Krok 2. Losowy wybór typu pojazdu.
- Krok 3. Losowy wybór liczby pojazdów.
- Krok 4. Losowy wybór czasu zaangażowania pojazdu.
- Krok 5. Mrówka opuszcza zajezdnię. Wyznaczane jest prawdopodobieństwo przejścia mrówki do pierwszego punktu trasy (kursu). Należy rozważyć dwa przypadki. Jeżeli mrówka opuszcza zajezdnię wzór (9) zostanie zmodyfikowany do postaci:

$$\eta_{yz}(t) = \frac{1}{d(y,z)} \quad (13)$$

W drugim przypadku, w celu uniknięcia stałego wyboru punktów najbliższych znajdujących się bazy należy w sposób losowy wybrać pierwszy punkt trasy. W przypadku przejścia mrówki pomiędzy kursami ma zastosowanie wzór (9). Natężenie feromonu na poszczególnych połączeniach w każdej warstwie przyjmuje się ustaloną wartość np. 1.

- Krok 6. Sprawdzenie warunku na czas realizacji wszystkich zadań oraz czas pracy kierowcy lub kierowców. W przypadku, kiedy czas trwania trasy mrówki jest mniejszy niż czas realizacji zadań lub czas pracy kierowcy/kierowców, wykonywany jest krok 7, gdy większy kurs wybrany w kroku 5 nie wchodzi do trasy mrówki, mrówka wraca do zajezdni i kroki 1–5 są ponownie powtarzane dla niezrealizowanych kursów. Warstwa III (liczba pojazdów) jest zaktualizowana i pomniejszona o pojazdy już wykorzystane.
- Krok 7. Sprawdzenie warunku na kompletność kursów. Jeżeli wszystkie kursy są zrealizowane mrówka udaje się do zajezdni i kończy budowanie trasy. W przypadku niespełnienia warunku algorytm realizuje krok 5.
- Krok 8. Sprawdzenie warunku na liczbę mrówek realizujących trasy. Jeżeli wszystkie mrówki zrealizowały trasy następuje krok 9 – aktualizacja feromonu. Jeśli nie wszystkie mrówki, kolejna mrówka tworzy trasę, algorytm powtarzany jest od kroku 1 do kroku 7.
- Krok 9. Aktualizacja feromonu do następnych iteracji zgodnie z wzorami (11) i (12).
- Krok 9a. Wyznaczenie całkowitego czasu i dystansu  $T^{mr}(t)$  dla całej trasy, czyli suma odcinków dojazdowych między kursami, odcinków między zajezdnią a pierwszym kursem oraz między ostatnim kursem a zajezdnią.

- Krok 10. Po zaktualizowaniu feromonu algorytm sprawdza warunek na liczbę iteracji (warunek Stopu). Jeżeli warunek jest niespełniony, powtarzane są kroki 1-9a, jeśli spełniony – algorytm kończy działanie. Z populacji wybierana jest mrówka o największym natężeniu feromonu.

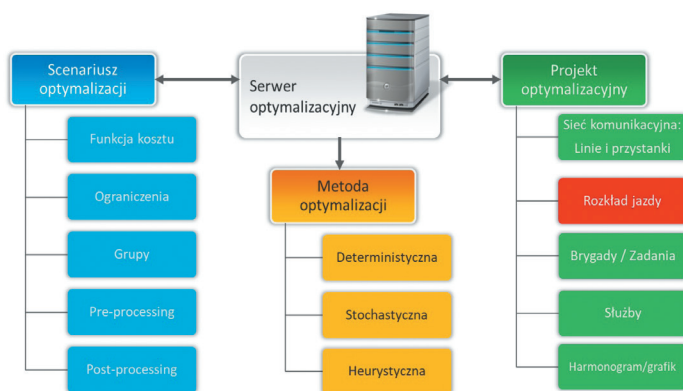
### Praktyczna implementacja algorytmu mrówkowego

Badania symulacyjne zaprezentowanego algorytmu prowadzone są aktualnie w ramach projektu NCBiR POIR.01.01.01-00-0970/17-00 „System informatyczny komputerowego wspomaganie planowania komunikacji miejskiej” w dwóch etapach. W pierwszym etapie badania prowadzone są na środowisku testowym w systemie *Mathematica* z wykorzystaniem sztucznych, teoretycznych danych wejściowych. Badania w tym etapie koncentrują się na walidacji modelu matematycznego, ocenie zbieżności algorytmu oraz doborze parametrów sterujących algorytmu.

W kolejnym etapie zaplanowane są badania symulacyjne na rzeczywistych danych komunikacyjnych różnych miast, w komercyjnym systemie planistycznym *CityLineDesigner*. Badania te pozwolą na ocenę efektywności proponowanego algorytmu mrówkowego w porównaniu z innymi algorytmami metaheurystycznymi, pod kątem szybkości działania i jakości uzyskiwanych rozwiązań w zastosowaniu do planowania zadań w komunikacji miejskiej.

Na rysunku 3 przedstawiono schemat serwera optymalizacji zastosowanego w systemie *cityLineDesigner*. Serwer optymalizacji działa w tle serwera aplikacji systemu, przyjmując zlecenia optymalizacji wybranego procesu np. projektowania rozkładu jazdy czy też brygad tj. zadań pojazdów. Użytkownik systemu ma możliwość opracowania własnego scenariusza optymalizacji, w tym funkcji kosztu i ograniczeń, oraz wyboru metody optymalizacji. Funkcje kosztu w zależności od złożoności problemu mogą być analizowane jako funkcje jedno bądź wielokryterialne. Serwer optymalizacyjny przystosowany jest do rozwiązywania problemów o różnej złożoności obliczeniowej, zarówno w optymalizacji liniowej, jak i nieliniowej. Praca systemu *cityLineDesigner* opiera się na obliczeniach wstępnych (ang. pre-processing) w którym początkowe rozwiązania są poprawiane (ang. post-processing) i ukierunkowane na rozwiązanie optymalne bądź suboptymalne w zależności od skali analizowanego problemu. Scenariusze optymalizacyjne są grupowane w zależności od przyjętych funkcji kryteriów i poddawane dalszej analizie.

Metody zaimplementowane w systemie dostosowane są do złożoności problemu i mogą generować wyniki dokładne lub przybliżone. Dla prostych problemów klasy NP-zupełnych funkcjonalnymi metodami optymalizacyjnymi są metody deterministyczne, dzięki którym system w bardzo krótkim czasie generuje optymalne rozwiązania. Dla złożonych problemów klasy NP-trudnych wybierane są metody stochastyczne bądź heurystyczne. W grupie metod heurystycznych, zaimplementowane są w systemie m.in. metody metaheurystyczne jak algorytmy ewolucyjne, przewidziana jest również implementacja algorytmów mrówkowych.



Rys. 3. Schemat serwera optymalizacji w systemie *cityLineDesigner*

Baza danych zaimplementowana w systemie *cityLineDesigner* przetwarza zarówno dane wprowadzane do niej bezpośrednio przez użytkownika aplikacji lub pobierane z innych serwerów, jak i dane generowane w procesie optymalizacyjnym. Wyniki działania programu można przedstawić w postaci tabelarycznej oraz graficznej.

### Podsumowanie

Omówione w artykule zastosowanie algorytmu mrówkowego w problematyce przydziału pojazdów do zadań w komunikacji miejskiej jest podejściem nowatorskim. Opracowany algorytm minimalizuje liczbę pojazdów i kierowców, przy jednoczesnej minimalizacji czasu realizacji wszystkich kursów oraz przebytej drogi przez wykorzystane pojazdy. Dalsze prace badawcze nad opracowanym algorytmem mrówkowym mogą obejmować jego modyfikacje pod kątem szybkości działania i jakości uzyskiwanych rozwiązań, np. z uwzględnieniem systemu mrowiskowego. Należy zaznaczyć, iż algorytm mrówkowy jest algorytmem heurystycznym. Zaletą algorytmów heurystycznych jest możliwość szybkiego generowania wyników, co ma decydujące znaczenie w obszarze praktycznego planowania w komunikacji miejskiej. Wadą algorytmów heurystycznych jest to, iż algorytmy te najczęściej generują rozwiązanie suboptymalne, akceptowalne praktycznie, ale odległe mniej lub bardziej od optimum globalnego. Z praktycznego, inżynierskiego punktu widzenia takie rozwiązania są akceptowalne, mając na uwadze złożoność badanej tematyki.

### Literatura

1. Valouxis C., Housos E., *Combined bus and driver scheduling*, "Computers & Operations Research", 2002, no. 29.
2. Wedelin D., *An algorithm for large scale 0-1 integer programming with application to airline crew scheduling*, "Annals of Operational Research", 1995, no 57.
3. Borndörfer R., Löbel A., Weider S., *A Bundle Method for Integrated Multi-Depot Vehicle and Duty Scheduling in Public Transit*. In: Hickman M., Mirchandani P., Voß S. (eds), *Computer-aided Systems in Public Transport. Lecture Notes in Economics and Mathematical Systems*, Springer, Berlin, Heidelberg, 2008, vol. 600.
4. Gintner V., Kliewer N., Suhl L., *A Crew Scheduling Approach for Public Transit Enhanced with Aspects from Vehicle Scheduling*, In: Hickman M., Mirchandani P., Voß S. (eds) *Computer-aided Systems in Public Transport. Lecture Notes in*

- Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 2008, vol. 600.
5. Desrochers, M., Soumis F., *A column generation approach to the urban transit crew scheduling problem*, "Transportation Science", 1989, no. 23.
  6. Wilson J.M., *A genetic algorithm for the generalized assignment problem*, "Journal of the Operational Research Society", 1997, vol. 48.
  7. Pentico D.W., *Assignment problems: A golden anniversary survey*, "European Journal of Operational Research", 2007, vol. 176, issue 2.
  8. Kisiielewski P., *Optymalizacja przydziału zadań transportowych*, Politechnika Krakowska, „Problemy Eksploatacji”, 2007, nr 2.
  9. Jacyna-Gółda I., Izdebski M., Podwieszko A., *Assessment of efficiency of assignment of vehicles to tasks in supply chains: A case study of a municipal company*, "Transport", 2017, vol. 32, issue 3.
  10. Izdebski M., *Modelowanie i analiza problemów decyzyjnych przydziału pojazdów do zadań w zagadnieniach transportowych*, Oficyna Wydawnicza Politechniki Warszawskiej, 2018.
  11. Jacyna M., Izdebski M., Szczepański E. [i in.], *The task assignment of vehicles for a production company*, w: Symmetry-Basel, 2018, vol. 11, no. 10.
  12. Jacyna-Gółda I., *Inżynieria oceny efektywności sieci dostaw*, Wydawnictwo Naukowe PWN, 2019.
  13. Gółda P., Izdebski M., Szczepański E., *The application of ant algorithm in the assignment problem of aircrafts to stops points on the apron*, "Journal of KONES Powertrain and Transport", 2018, vol. 25, no. 1.
  14. Gółda P., *Wspomaganie decyzji wykonywania operacji lotniskowych z zastosowaniem narzędzi symulacyjnych*, Monografia. Wydawnictwo ITWL, 2018.
  15. Izdebski M., Jacyna M., *The organization of the municipal waste collection: the decision model*, Rocznik Ochrona Środowiska/Annual set The Environment Protection, 2018, nr 20.
  16. Szczepański E., Jachimowski R., Izdebski M., Jacyna-Gółda J., *Warehouse location problem in supply chain designing: a simulation analysis*, Archives of Transport, 2019, vol. 50, issue 2.
  17. Jacyna-Gółda I., Lewczuk K., *The method of estimating dependability of supply chain elements on the base of technical and organizational redundancy of process*, in: „Eksploatacja i Niezawodność”, 2017, vol. 3, no. 19.
  18. Izdebski M., Jacyna M., *The algorithm solving the problem of allocation of tasks to resources in the transport company*, w: CLC 2013: Carpathian Logistics Congress – Congress Proceedings / Feliks Jerzy (red.), TANGER Ltd., Ostrava, 2014.
  19. Wasiak M., Jacyna M., Lewczuk K. [i in.], *The method for evaluation of efficiency of the concept of centrally managed distribution in cities*, w: "Transport", 2017, vol. 32, nr 4.
  20. Jacyna M., Wasiak M., *Multicriteria Decision Support in Designing Transport Systems*, w: Tools of Transport Telematics, red. Mikulski J., Springer, 2015.
  21. Kisiielewski P., *Komputerowe wspomaganie planowania komunikacji miejskiej. Monografia*, Oficyna wydawnicza Politechniki Warszawskiej, Warszawa 2019.
  22. Gintner V., Kliwer N., Suhl L., *Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice*, OR Spectrum, 2005, no. 27(4).
  23. Haase K., Desaulniers G., Desrosiers J., *Simultaneous vehicle and crew scheduling in urban mass transit systems*, "Transportation Science", 2001, no. 35.
  24. Huisman D., Freling R., Wagelmans A.P.M., *Multiple-depot integrated vehicle and crew scheduling*, "Transportation Science", 2005, no. 39.
  25. Freling R., Huisman D., Wagelmans A.P.M., *Models and algorithms for integration of vehicle and crew scheduling*, "Journal of Scheduling", 2003, no. 6.
  26. Löbel A., *Solving Large-Scale Multiple-Depot Vehicle Scheduling Problems*, in: Wilson N.H.M. (eds) Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 1999, vol. 471.
  27. Gaffi A., Nonato M., *An Integrated Approach to Ex-Urban Crew and Vehicle Scheduling*, in: Wilson N.H.M. (eds), Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg. 1999, vol. 471.
  28. Freling R., Huisman D., Wagelmans A.P.M., *Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice*, in: Voß S., Daduna J.R. (eds) Computer-Aided Scheduling of Public Transport. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 2001, vol. 505
  29. Daduna J.R., Pinto Paixão J.M., *Vehicle Scheduling for Public Mass Transit — An Overview*, in: Daduna J.R., Branco I., Paixão J.M.P. (eds), Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 1995, vol. 430.
  30. de Groot S.W., Huisman D., *Vehicle and Crew Scheduling: Solving Large Real-World Instances with an Integrated Approach*, in: Hickman M., Mirchandani P., Voß S. (eds), Computer-aided Systems in Public Transport. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 2008, vol. 600.
  31. Mesquita M., Paixão J., *Exact Algorithms for the Multi-Depot Vehicle Scheduling Problem Based on Multicommodity Network Flow Type Formulations*, in: Wilson N.H.M. (eds), Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 1999, vol. 471.
  32. Carraresi P., Nonato M., Girardi L., *Network Models, Lagrangean Relaxation and Subgradients Bundle Approach in Crew Scheduling Problems*, in: Daduna J.R., Branco I., Paixão J.M.P. (eds), Computer-Aided Transit Scheduling. Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, Heidelberg, 1995, vol. 430.
  33. Scott D., *A large linear programming approach to the public transport scheduling and cost model*, in J. Rousseau, editor, *Computer Scheduling of Public Transport 2*, North Holland, Amsterdam 1985.
  34. Friberg C., Haase K., *An exact branch and cut algorithm for the vehicle and crew scheduling problem*, in N. Wilson, editor, *Computer-Aided Transit Scheduling*, Springer, Berlin 1999.
  35. Feo T.A., Resende M.G.C., *Greedy randomized adaptive search heuristic*, Journal of Global Optimization 1995, no. 6.
  36. Garcia B.-L., Potvin J.-Y., Rousseau J.-M., *A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints*, „Computers & Operations Research”, 1994, vol. 21.
  37. Zhang Q., Manier M., *A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times*, „Computers & Operations Research”, 2012, vol. 39, Issue 7.
  38. Li J., Kwan R.S., *A fuzzy genetic algorithm for driver scheduling*, „European Journal of Operational Research”, 2003, vol. 147.
  39. Peng K., Shen Y., Li J., *A multi-objective simulated annealing for bus driver rostering*, in: Bio-Inspired Computing-Theories and Applications, Springer, 2015.
  40. Fu A., Lei S., *Intelligent dispatching of public transit vehicles using particle swarm optimization algorithm*, Comput Eng Appl 44, 2008, no. 15.
  41. Dorigo M., Gambardela L.M., *Ant Colonies for the Travelling Salesman Problem*, BioSystems, 1997, vol. 43.
  42. Dorigo M., Caro D., Gambardela L.M., *Ant Algorithms for Discrete Optimization*, Artificial Life, 1999, vol. 5 (2).