

Bogusław Filipowicz*, Joanna Kwiecień*

Algorytmy stadne w optymalizacji problemów przydziału przy kwadratowym wskaźniku jakości (QAP)

1. Wprowadzenie

W ciągu ostatnich kilkunastu lat nastąpił intensywny rozwój algorytmów stadnych, których zasady działania zostały zaczerpnięte z obserwacji natury. Naśladują one zachowania istniejące w świecie owadów, zwierząt, ptaków i dostarczają bardzo wydajnych metod. Ich główną zaletą jest to, że bazują na zbiorze rozwiązań danego problemu, łatwo dostosowując się do ograniczeń, niezależnie od liczby zmiennych oraz rozmiaru przestrzeni rozwiązań.

Do klasy algorytmów stadnych należą algorytmy mrówkowe oparte na zachowaniu kolonii mrówek (ACO – *ant colony optimization*), algorytmy pszczele bazujące na zachowaniu roju pszczoł (BA – *bee algorithm*) i algorytmy optymalizacji rojem cząstek (PSO, *particle swarm optimization*) oparte na obserwacji zachowania całej populacji (stada ptaków, ławicy ryb), przy możliwości wymiany informacji między osobnikami. Algorytmy ACO, BA i PSO należą do klasy dynamicznie rozwijających się technik optymalizacji. Ich zastosowania w wielu dziedzinach wskazują na olbrzymi potencjał, ze względu na efektywność w poszukiwaniu globalnego rozwiązania.

Algorytmy te można wykorzystać do rozwiązania problemów przydziału przy kwadratowym wskaźniku jakości (QAP – *quadratic assignment problem*), które należą dla klasy zagadnień NP-trudnych.

2. Matematyczny model problemu QAP

Zagadnienie QAP określone jest poprzez dwie macierze $n \times n$: $A = [a_{i,j}]$, $B = [b_{k,l}]$. Niech $\pi(i) \in N$, $i = 1, \dots, n$ określa numer obiektu przydzielonego do pozycji i , zaś zbiór numerów obiektów oznaczony jest jako $N = \{1, \dots, n\}$. Macierz A określa odległości pomiędzy pozycjami rozmieszczenia obiektów, a macierz B opisuje powiązania pomiędzy tymi

* AGH Akademia Górniczo-Hutnicza, Wydział Elektrotechniki, Automatyki, Informatyki i Elektroniki, Katedra Automatyki

obiektami. Celem jest znalezienie takiej permutacji $\pi = (\pi(1), \dots, \pi(n))$ elementów zbioru obiektów, która minimalizuje funkcję celu $z(\pi)$ określającą globalny koszt realizacji systemu [1, 3, 4]:

$$z(\pi) = \sum_{i=1}^n \sum_{j=1}^n a_{\pi(i)\pi(j)} b_{ij} \quad (1)$$

3. Wybrane algorytmy stadne i ich zastosowanie do QAP

W latach 90. ubiegłego wieku M. Dorigo, opracował algorytm wzorowany na zachowaniu kolonii mrówek do kombinatorycznych problemów optymalizacyjnych. Do zapewnienia przetrwania kolonii mrówek konieczna jest komunikacja, odbywająca się poprzez wydzielanie feromonów, których detekcja determinuje określone reakcje. Mrówki podążające za nimi, wybierają drogę na podstawie intensywności pozostawionego feromonu. Im więcej feromonu na ścieżce, tym istnieje większe prawdopodobieństwo obrania danej trasy przez mrówki. Mrówki dostosowują się do zmian zachodzących w środowisku, modyfikując trasę w przypadku pojawienia się przeszkody, przerywającej ślad feromonowy, a następnie poruszając się w sposób losowy, wybierają drogę, przy czym prawdopodobieństwo owego wyboru jest jednakowe dla mrówek przed przeszkodą i powracających do gniazda. Mrówki wybierające krótszą drogę przyczyniają się do szybszego odbudowania śladu feromonowego [10, 11].

W 1995 roku Kennedy i Eberhart opracowali algorytm optymalizacji rojem cząstek (PSO). Algorytm ten bazuje na zachowaniu całej populacji (np. stado ptaków, rój owadów), w której osobniki komunikują się między sobą i dzielą się informacjami. Cząstki przemieszczają się do nowych położeń, poszukując optimum. Cały rój podąża za przywódcą (najlepszym rozwiązaniem), przyspieszając i zmieniając kierunek, gdy zostanie znalezione lepsze rozwiązanie [2, 6, 11].

Algorytmy pszczele (BA) są kolejną metodą należącą do klasy algorytmów stadnych, których rozwój datuje się na okres 2004–2007. Organizacja roju pszczoł chcących zdobyć optymalną ilość kwiatostanów polega na rozsyłaniu we wszystkie strony pszczoł-zwiadowców, które przeszukują w odległości kilkunastu kilometrów od ula obszary zasobne w nektar. Początkowe szukanie nektaru odbywa się w sposób losowy. Po powrocie do ula pszczoła-zwiadowca powiadamia pozostałe pszczoły o najlepszym swoim odkryciu. W trakcie wykonywania tańca pszczelego następuje wymiana informacji (nt. jakości, kierunku i odległości pożywienia od punktu bazowego) między zwiadowcami a pozostałymi pszczołami niezatrudnionymi, które z kolei wybierają najlepsze miejsca i rozpoczynają zbiory nektaru. Im obfitsze źródło pokarmu tym więcej pszczoł dowiaduje się o tym miejscu. Pszczoły powracające z wyprawy z pyłkiem przekazują pozostałym osobnikom podejmującym na tej podstawie decyzję, za śladem, której pszczoły-zwiadowcy podążać. Pszczoła zbierająca nektar może również powiadomić pozostałe o miejscu występowania nektaru [4, 9].

3.1. Zastosowanie algorytmów mrówkowych do problemów QAP

System mrówkowy AS (*ant system*) jest algorytmem, który można stosować do problemów QAP. W trakcie konstruowania rozwiązania mrówka przypisuje obiekt i do lokalizacji j z prawdopodobieństwem (2). W celu obliczenia informacji heurystycznej wykorzystywane są dwa wektory a oraz b . Element i -ty wektora a oznacza sumę odległości lokalizacji i od wszystkich pozostałych, natomiast w przypadku wektora b przedstawia sumę przepływów z danego obiektu i do pozostałych obiektów. Należy wyznaczyć macierz $E = b \cdot a^T$, w której każdy element $e_{ij} = b_i \cdot a_j$, co zapewni większe prawdopodobieństwo przypisania małych wartości d_{ij} , reprezentujących odległości między lokacjami, do obiektów z największymi przepływami b_i . Przypisane obiekty i lokalizacje są blokowane do czasu ukończenia rozwiązania. Mając kompletne rozwiązanie uruchamiane jest przeszukiwanie lokalne, w wyniku którego otrzymujemy permutację liczb, z której można odczytać przypisanie obiektów do lokalizacji [10].

W każdym kroku mrówce k zostaje przydzielony następny wolny obiekt i do wolnej pozycji j z prawdopodobieństwem:

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha \cdot [\eta_{il}]^\beta}, \quad j \in N_i^k \quad (2)$$

gdzie τ_{ij} jest śladem feromonowym w iteracji t , α – to parametr kontrolujący wagę feromonu, β – parametr kontrolujący wagę wartości heurystycznych, N_i^k jest sąsiedztwem wężła i (tylko wolne pozycje).

Uaktualnienie śladu feromonowego jest realizowane według zależności:

$$\tau_{ij}(k+1) = \rho \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k \quad (3)$$

przy czym dla k -tej mrówki:

$$\Delta \tau_{ij}^k = \begin{cases} Q/J^k, & \text{obiekt } i \text{ przypisany do lokacji } j \\ 0 & \end{cases} \quad (4)$$

gdzie: $\rho \in (0, 1)$ jest współczynnikiem wyparowywania feromonu, zaś J^k jest funkcją celu, Q – wartość feromonu pozostawionego przez mrówkę.

Kolejnym algorytmem mrówkowym, który można zastosować do problemu QAP, jest system mrówkowy ze strategią max-min (MMAS, ang. *max-min ant system*), w którym wprowadza się maksymalny τ_{\max} i minimalny τ_{\min} poziom feromonu. Tylko jedna mrówka

pozostawia ślad feromonowy, ta która stanowi globalnie najlepsze rozwiązanie lub najlepsze rozwiązanie w danej iteracji. Początkowo wszystkie ślady feromonowe są inicjalizowane wartością τ_{\max} . Mrówka k wybiera losowo nieprzypisane jeszcze i -te zadanie i umieszcza w wolnej pozycji j z prawdopodobieństwem [10]:

$$p_{ij}^k(t) = \frac{\tau_{ij}(t)}{\sum_{l \in N_i^k} \tau_{il}(t)}, \quad j \in N_i^k \quad (5)$$

Po skonstruowaniu rozwiązania przez wszystkie mrówki, ślad feromonowy jest uaktualniony według zależności:

$$\tau_{ij}(k+1) = \rho \tau_{ij}(t) + \Delta \tau_{ij}^{best} \quad (6)$$

przy czym:

$$\Delta \tau_{ij}^{best} = \begin{cases} 1/J^{best}, & \text{obiekt } i \text{ przypisany do lokacji } j \\ 0 & \end{cases} \quad (7)$$

gdzie J^{best} jest funkcją celu.

3.2. Zastosowanie algorytmu optymalizacji rojem cząstek do rozwiązania QAP

W algorytmie PSO każdej cząstce przypisujemy określone położenie i prędkość. Cząstki znają swoich sąsiadów oraz wartość funkcji ewaluacyjnej dla swoich położen [2, 5, 6, 11]. Położenie oraz prędkość i -tej cząstki w d -wymiarowej przestrzeni można przedstawić odpowiednio w postaci wektorów $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$ oraz $v_i = [v_{i1}, v_{i2}, \dots, v_{id}]$. Każda z cząstek zna swoją własną najlepszą pozycję $p_i = [p_{i1}, p_{i2}, \dots, p_{id}]$ odpowiadającą najlepszej uzyskanej dotychczas wartości funkcji celu oraz najlepszą pozycję cząstki-przywódcy w całym roju określoną jako $p_d = [p_{d1}, p_{d2}, \dots, p_{dd}]$.

Podstawowy algorytm optymalizacji rojem cząstek można przedstawić w kilku etapach [2, 6, 11]:

- 1) losowa inicjalizacja pozycji i prędkości początkowych cząstek;
- 2) ocena położenia cząstek za pomocą funkcji dopasowania;
- 3) porównanie zachowania każdej cząstki z jej najlepszym dotychczasowym zachowaniem;
- 4) uaktualnienie prędkości każdej cząstki w każdym kroku k :

$$v_i(k) = \omega v_i(k-1) + c_1 r_1 [p_i(k-1) - x_i(k-1)] + c_2 r_2 [p_d(k-1) - x_i(k-1)] \quad (8)$$

przy czym ω oznacza współczynnik inercji ruchu cząstki, c_1 to wskaźnik samooceny oznaczający zaufanie kierunkowi swojego najlepszego położenia, c_2 to wskaźnik społecznościowy określający jak bardzo cząstka ufa położeniom swoich sąsiadów, r_1 oraz r_2 to losowe liczby o rozkładzie równomiernym w przedziale $[0, 1]$;

5) uaktualnienie położenia każdej cząstki:

$$x_i(k) = x_i(k-1) + v_i(k) \tag{9}$$

W celu inicjalizacji cząstek można utworzyć wektor AC określający kolejność zadań na podstawie oszacowanego kosztu transportu zadania i do zadania $i+1$, w kolejności nierosnącej oraz wektor BC, który przedstawia kolejność lokacji na podstawie odległości z lokacji j do $j+1$, uszeregowanej nierosnąco. Wektor rozwiązań x_i polega więc na przypisaniu kolejno zadań z wektora AC do lokacji z wektora BC [8].

W pracy [7] przedstawiono przybliżenie optymalizacji rojem cząstek (*fuzzy particle swarm approach*), które można stosować do rozwiązania mniej skomplikowanych problemów QAP. Dla zbioru n obiektów $N = \{N_1, N_2, \dots, N_n\}$ i zbioru n lokacji $L = \{L_1, L_2, \dots, L_n\}$ można relację przypisania obiektów do lokalizacji wyrazić jako:

$$AS = \begin{bmatrix} as_{11} & as_{12} & \dots & as_{1n} \\ as_{21} & as_{22} & \dots & as_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ as_{n1} & as_{n2} & \dots & as_{nn} \end{bmatrix} \tag{10}$$

gdzie as_{ij} określa stopień przynależności j -tego elementu N_j do i -tego elementu L_i w relacji AS.

Oczywiście elementy rozwiązania muszą spełniać następujące warunki:

$$as_{ij} \in \{0,1\}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n$$

$$\sum_{i=1}^n as_{ij} = 1, \quad \sum_{j=1}^n as_{ij} = 1 \tag{11}$$

W celu zastosowania algorytmu PSO do kwadratowego problemu przydziału należy przeddefiniować pozycję X i prędkość V w następujący sposób [7]:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nn} \end{bmatrix}; \quad V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \dots & v_{nn} \end{bmatrix} \tag{12}$$

przy ograniczeniach:

$$\begin{aligned}
 x_{ij} &\in \{0,1\}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n \\
 \sum_{i=1}^n x_{ij} &= 1, \quad \sum_{j=1}^n x_{ij} = 1
 \end{aligned} \tag{13}$$

Aby macierz pozycji nie naruszała ograniczeń (11), należy ją znormalizować. Podlega więc ona następującemu przekształceniu [7]:

$$X_{znormalizowana} = \begin{bmatrix} x_{11} / \sum_{i=1}^n x_{i1} & x_{12} / \sum_{i=1}^n x_{i2} & \cdots & x_{1n} / \sum_{i=1}^n x_{in} \\ x_{21} / \sum_{i=1}^n x_{i1} & x_{22} / \sum_{i=1}^n x_{i2} & \cdots & x_{2n} / \sum_{i=1}^n x_{in} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} / \sum_{i=1}^n x_{i1} & x_{n2} / \sum_{i=1}^n x_{i2} & \cdots & x_{nn} / \sum_{i=1}^n x_{in} \end{bmatrix} \tag{14}$$

Macierz pozycji wskazuje potencjalne rozwiązanie przydziału. Wybieramy element maksymalny w kolumnie i przypisujemy mu wartość 1, pozostałe elementy przyjmują zerowe wartości. Po przeanalizowaniu wszystkich kolumn i wierszy otrzymujemy rozwiązanie problemu przydziału bez naruszenia ograniczeń (11) [7].

3.3. Zastosowanie algorytmu pszczelego do rozwiązania QAP

Algorytm pszczeli jest algorytmem iteracyjnym, który można zrealizować w kilku etapach [4, 9]:

1. Losowa inicjalizacja populacji początkowej (permutacji).
2. Obliczenie funkcji celu dla całej populacji i przejście do kolejnego etapu części rozwiązań.
3. Dopóki niespełnione jest kryterium stopu, należy przeprowadzić:
 - wybór miejsc do przeszukiwania sąsiedztwa (zdefiniowanie sąsiedztwa rozwiązania),
 - rekrutacja pszczoł do najlepszych miejsc (proporcjonalnie do jakości miejsca),
 - wyliczenie funkcji celu (posortowanie rozwiązań populacji według funkcji celu),
 - wybranie najlepszej pszczoły w danym miejscu (każde z przeszukiwań lokalnych generuje najlepsze lokalne rozwiązanie),
 - przeszukiwanie przestrzeni rozwiązań niezatrudnionymi pszczołami – przypisanie pozostałych pszczoł do losowych poszukiwań i wyliczenie ich funkcji dopasowania.
4. Spełnione kryterium stopu (np. zadana liczba iteracji) – wyznaczenie najlepszego rozwiązania.

4. Wyniki przeprowadzonych eksperymentów

Do przetestowania działania trzech algorytmów wybrano instancje testowe z biblioteki QAPLIB, dostępnej on-line [12], w której zawarte są różnorodne zagadnienia przydziału przy kwadratowym wskaźniku jakości. Wyniki przykładowych eksperymentów podano w tabeli 1.

Tabela 1

Wyniki eksperymentów dla algorytmów mrówkowych, algorytmów optymalizacji rojem cząstek i algorytmów pszczelich dla wybranych instancji testowych z biblioteki QAPLIB

Nazwa problemu	Algorytmy mrówkowe	Algorytm PSO	Algorytmy pszczele	Znane najlepsze rozwiązanie
BUR26A	5473280	5527047	5466244	5426670
BUR26H	7182482	7292985	7098658	7098658
ESC32C	642	642	642	642
ESC32F	2	2	2	2
ESC32G	6	6	6	6
LIPA40A	31538	32412	31932	31538
LIPA50B	1460852	1210244	1422472	1210244
LIPA70B	5379780	4603200	5503244	4603200
LIPA90A	363379	366396	363095	360630
NUG21	2764	2818	2464	2438
NUG25	4280	4370	3764	3744
SKO42	16788	17622	16014	14934
SKO49	26624	27312	23644	22004
SKO81	102258	104060	91746	86072
WIL50	51388	53044	49086	48816

Na podstawie przeprowadzonych eksperymentów można stwierdzić, że najlepszym algorytmem do rozwiązania problemów przydziału przy kwadratowym wskaźniku jakości jest algorytm pszczeli. W większości przetestowanych przypadków znajdował rozwiązanie najbliższe najlepszemu znanemu rozwiązaniu. Przyjęte parametry algorytmu pszczelego to 500 iteracji oraz 100 pszczół.

W przypadku algorytmu optymalizacji rojem cząstek w instancjach z rodziny bur26 uzyskano co prawda wyniki gorsze, to jednak dla problemów z rodziny lipa osiągały bardzo dobre rezultaty, włącznie ze znajduwaniem optymalnego rozwiązania. Na wyniki obliczeń wpływają parametry algorytmu, w tym współczynniki c_1 i c_2 . Najlepsze wyniki otrzymane zostały dla parametrów $c_1 = c_2 = 0,2$. Przyjęto 500 iteracji algorytmu. Wpływ

współczynnika inercji jest zależny od rozmiaru instancji problemu – dla problemu bur26a.dat najlepsze wyniki dawała wartość inercji równa 0,5, dla większego lipa40a.dat – niższa wartość (0,25). W badanych instancjach wyższy współczynnik inercji pogarszał otrzymane wyniki.

Algorytm mrówkowy wykonywany był również dla kilku możliwych ustawień. Niestety trudno jest ustalić najlepsze parametry, przetestowano więc go dla wybranych ustawień: 500 iteracji, współczynnik wyparowywania feromonu 0,1, maksymalny poziom feromonu $\tau_{\max} = 10$ i minimalny poziom feromonu $\tau_{\min} = 0,1$.

5. Podsumowanie

W artykule przedstawiono zastosowanie trzech algorytmów inspirowanych przez naturę do rozwiązywania kwadratowego problemu przydziału. Spośród przedstawionych algorytmów stadnych, najlepsze wyniki zostały uzyskane dla algorytmu pszczelego. Głównym problemem w przetestowaniu algorytmów mrówkowych i optymalizacji rojem cząstek było dobranie odpowiednich ustawień parametrów.

Literatura

- [1] Burkard R.E., Karisch S.E., Rendl F., *QAPLIB – A Quadratic Assignment Problem Library*. European Journal of Operational Research, 55, 1991, 115–119.
- [2] Eberhart R., Shi Y., Kennedy J., *Swarm Intelligence*. Morgan Kaufman, San Francisco 2001.
- [3] Filipowicz B., Wala K., *Algorytmy optymalizacji kwadratowego zagadnienia przydziału*. Elektrotechnika (kwartalnik AGH), z. 1, 1992.
- [4] Filipowicz B., Chmiel W., Kadłuczka P., *Ukierunkowane przeszukiwanie przestrzeni rozwiązań w algorytmach rojowych*. Automatyka (półrocznik AGH), 13, 2, 2009.
- [5] Gong T., Tuson A.L., *Particle swarm optimization for quadratic assignment problems – a forma analysis approach*. International Journal of Computational Intelligence Research, 4, 2008, 177–185.
- [6] Kennedy J., Eberhart R., *Particle Swarm Optimization*. Materiały IEEE International Conference on Neural Networks, 4, 1942–1948, 1995
- [7] Liu H., Abraham A., Zhang J., *A particle swarm approach to quadratic assignment problems*. Soft Computing in Industrial Applications. Advances in Intelligent and Soft Computing, 39, 2007, 213–222.
- [8] Nędza T., Szpat K., *Zastosowanie algorytmu ptasiego do rozwiązania problemów optymalizacji kombinatorycznej*. Praca inżynierska (niepublikowana), AGH, 2011 (promotor B. Filipowicz).
- [9] Pham D.T., Ghanbarzadeh A., Koc E., Otri S., Rahim S., Zaidi M., *The Bees Algorithm – A Novel Tool for Complex Optimisation Problems*. Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
- [10] Stützle T., Dorigo M., *ACO Algorithms for the Quadratic Assignment Problem*. [w:] D. Corne, M. Dorigo, F. Glover, New Ideas for Optimization, McGraw-Hill, 1999, 33–50.
- [11] Trojanowski K., *Metaheurystyki praktycznie*. Wydawnictwo WIT, Warszawa 2005.
- [12] Zbiór instancji testowych problemu QAP: <http://www.seas.upenn.edu/qaplib/>.