AUTOMATYKA • 2007 • Tom 11 • Zeszyt 1–2

Piotr Kadłuczka*, Wojciech Chmiel*

Wykorzystanie własności zagadnienia QAP w konstrukcji algorytmów ewolucyjnych

1. Wprowadzenie

Problem właściwego doboru zbieżności algorytmu ewolucyjnego EA (*Evolutionary Algorithm*) jest bardziej istotny i złożony niż w przypadku innych typów algorytmów przybliżonych. Powszechnie opisywane w literaturze [1, 2]: brak zbieżności oraz przedwczesna zbieżność EA są zjawiskami skrajnymi, pomiędzy którymi znajduje się optymalna realizacja tego mechanizmu. Brak jednak gotowej recepty na jej realizację, gdyż uwarunkowane to jest zbyt wieloma czynnikami i ściśle związane ze specyfiką rozwiązywanego zagadnienia. Zbieżność EA jest zależna między innymi od sposobu realizacji pokolenia, przetwarzania populacji, występowania elit, mechanizmu selekcji, stosowanych operatorów genetycznych oraz szeregu parametrów, np. wielkości populacji, elity, prawdopodobieństw stosowania operatorów itp. Wyżej wymienione elementy realizują dwa przeciwstawne mechanizmy: różnicowania i intensyfikacji, których właściwa proporcja ma gwarantować wysoką efektywność algorytmu. Mechanizmy te jednak nie działają w jednej „płaszczyźnie" – znaczenie ma nie tyle ich „sumaryczna wartość" ale również wielorakie, wzajemne zależności. Problemowi temu jest poświęcony poniższy artykuł.

2. Kwadratowe zagadnienie przydziału

Wykorzystywane przez nas w badaniach eksperymentalnych kwadratowe zagadnienie przydziału QAP (*Quadratic Assignment Problem*) należy do klasy zagadnień NP-trudnych. Wymusza to stosowanie do jego rozwiązania metod przybliżonych już dla zadań o niewielkim rozmiarze (powyżej 30). Mimo że jest ono znacznie trudniejsze niż inne zagadnienia optymalizacji kombinatorycznej, to cieszy się powszechnym zainteresowaniem ponieważ modeluje ważną klasę problemów decyzyjnych dla których rozwiązania można przedstawić w postaci permutacji. Dzięki temu, uzyskane w pracy wyniki mogą zostać wykorzystanie

* Katedra Automatyki, Akademia Górniczo-Hutnicza w Krakowie

w konstrukcji algorytmów przybliżonych dla wielu innych ważnych zagadnień permutacyjnych. Przykładami zastosowań praktycznych zagadnienia są m.in. [3]:

- problem lokalizacji kooperujących ze sobą zakładów produkcyjnych, maszyn w hali w zautomatyzowanych systemach produkcji, planowanie infrastruktury w projektowaniu miast;
- organizacja biur, terminali przeładunkowych, lotnisk, oddziałów szpitalnych;
- projektowanie rozmieszczenia elementów elektronicznych w układach o wielkiej skali integracji (VLSI), problem połączeń Steinberga;
- wyważanie turbin silników odrzutowych;
- problem dopasowania molekularnego;
- projektowanie klawiatur, przełączników ATM.

2.1. Model matematyczny zagadnienia QAP

Model matematyczny zagadnienia QAP możemy zdefiniować następująco:

Dany jest zbiór $N = \{1, ..., n\}$ oraz dwie $(n \times n)$-wymiarowe macierze $D = [d_{i,k}]$, $F = [f_{j,l}]$. W terminologii alokacji obiektów: zbiór N jest zbiorem numerów obiektów, a $\pi(i) \in N$, $i = 1, ..., n$ określa numer obiektu przydzielonego do pozycji i. Macierz D jest wtedy macierzą odległości pomiędzy pozycjami rozmieszczenia obiektów, podczas gdy macierz F opisuje powiązania (np. liczbę połączeń lub wielkość przepływu) występujące pomiędzy obiektami.

Należy znaleźć permutację $\pi = (\pi(1), ..., \pi(n))$ elementów zbioru N, która minimalizuje funkcję celu $\phi(\pi)$ o następującej postaci

$$\phi(\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{i,j} d_{\pi(i)\pi(j)} \tag{1}$$

Funkcja celu $\phi(\pi)$, $\pi \in \Pi$, określa globalny koszt realizacji lub eksploatacji systemu, natomiast Π jest zbiorem permutacji zbioru N.

2.2. Warunkowa wartość oczekiwana funkcji celu dla rozwiązania częściowo ustalonego

Warunkowa wartość oczekiwana funkcji celu pozwala na ocenę jakości rozwiązań częściowo ustalonych, tzn. takich w których dokonano przydziału na niektórych pozycjach rozwiązania, pozostawiając nieustalone pozostałe pozycje. Uzyskana ocena określa średnią wartość funkcji celu rozwiązań, które możemy utworzyć w oparciu o dokonany częściowy przydział.

Określmy zbiór liczb naturalnych z zakresu od 1 do n jako $L = [1, ..., n]$, zbiór $M = [1, ..., n] \backslash \{c(s_1), ..., c(s_i), ..., c(s_k)\}$ (numery nieprzydzielonych obiektów, gdzie $c(s_i)$ – jest numerem obiektu przydzielonym do pozycji s_i) oraz zbiór $H = \{s_1, ..., s_i, ..., s_k\}$ (numery zajętych pozycji). W takim przypadku warunkowa wartość oczekiwana funkcji celu dla problemu QAP przy k ustalonych pozycjach, wyniesie [3]:

$$
\begin{aligned}
&\mathrm{E}(\phi / \pi(s_1) = c(s_1), ..., \pi(s_i) = c(s_i), ..., \pi(s_k) = c(s_k)) = \sum_{i \in H} \sum_{j \in H} f_{ij} d_{\pi(i)\pi(j)} + \\
&+ \frac{1}{n-k} \sum_{i \in H} \sum_{j \in L \backslash H} f_{ij} \sum_{m \in M} d_{\pi(i)m} + \frac{1}{n-k} \sum_{i \in L \backslash H} \sum_{j \in H} f_{ij} \sum_{m \in M} d_{m\pi(j)} + \\
&+ \frac{1}{n-k} \sum_{i \in L \backslash H} f_{ii} \sum_{m \in M} d_{mm} + \frac{1}{n-k} \frac{1}{n-k-1} \sum \left\{ f_{ij} : j \neq i \in L \setminus H \right\} \sum \left\{ d_{ij} : j \neq i \in M \right\}
\end{aligned} \tag{2}
$$

3. Algorytm ewolucyjny wykorzystujący własności zagadnienia QAP

3.1. Sterowanie zbieżnością algorytmu ewolucyjnego

W konstrukcji EA (*Evolutionary Algorithm*) należy pogodzić dwa sprzeczne podejścia:

1) swobodne poszukiwanie optimum globalnego w całej przestrzeni rozwiązań (eksploracja),
2) możliwie dokładne przeszukiwanie otoczenia optimum lokalnego (eksploatacja).

Kompromis pozwalający podążać w stronę optimum lokalnego, a następnie opuścić jego obszar przyciągania realizowany jest za pomocą przeciwstawnych mechanizmów: intensyfikacji i różnicowania. Precyzyjne wskazanie miejsca występowania tych mechanizmów w algorytmie ewolucyjnym jest kłopotliwe, ponieważ odpowiada za nie sposób implementacji wielu elementów wchodzących w skład algorytmu. Podstawowe znaczenie ma tu rodzaj stosowanej selekcji oraz sam schemat algorytmu. Elementy te odpowiadają za wystąpienie tendencji do poprawy (mniej lub bardziej zdecydowanej) średniej wartości funkcji przystosowania w populacji.

Selekcja jest realizowana w dwóch etapach:

1. wyboru rozwiązań rodzicielskich (reprodukcja), poddawanych następnie krzyżowaniu i mutacji,
2. utworzeniu nowej populacji (sukcesja), na podstawie starej populacji i powstałych rozwiązań potomnych.

Problem przedwczesnej zbieżności algorytmu ewolucyjnego jest konsekwencją stosowanego w algorytmie mechanizmu selekcji oraz sposobu przetwarzania populacji (zjawisko naporu selekcyjnego, wzgl. nacisku selektywnego). Prowadzi on do sytuacji, w której populacja zawiera rozwiązania będące w obszarze przyciągania optimów lokalnych, a zastosowane w algorytmie ewolucyjnym mechanizmy nie pozwalają na dalszą eksplorację przestrzeni rozwiązań. W celu poprawy efektywności mechanizmów selekcji należy stworzyć odpowiednie proporcje między naporem selekcyjnym a różnorodnością populacji [4]. Zwiększenie naporu selekcyjnego zmniejsza różnorodność populacji, co prowadzi na ogół do przedwczesnej zbieżności algorytmu.

Intensyfikującymi będą te metody selekcji (reprodukcji), które w bardziej zdecydowany sposób preferują rozwiązania o lepszej wartości funkcji przystosowania. Należą do nich metody (wymienione malejąco):

- proporcjonalna – „koła ruletki”,
- oparta na rankingu,
- turniejowa,
- progowa.

Przykładem typowego mechanizmu różnicującego jest schemat przetwarzania, w którym populacja potomna zastępuje rodzicielską (sukcesja z całkowitym zastępowaniem – *generational replacement*). Zastępowanie częściowe (np. ochrona elity) czy wielokrotne powtórzenia rozwiązań w populacji stanowią osłabienie mechanizmu dywersyfikacji i wzmocnienie naporu selekcyjnego.

Stosowana przez nas selekcja z częściowym zastępowaniem (*Steady-State*), w której rozwiązania potomne konkurują z rodzicielskimi o miejsce w populacji należy do metod silnie intensyfikujących. Zależnie od typu stosowanego operatora, w każdym pokoleniu w populacji wymienione zostanie jedno lub dwa rozwiązania, które zastąpią najgorsze. Wybór osobnika (osobników) realizowany jest po dokonaniu wyboru operatora, za pomocą procedury losowania o rozkładzie równomiernym.

Zadanie selekcji – stworzenie naporu selekcyjnego, dokonuje się tu przez zwielokrotnienie szansy wyboru rozwiązania lepszego – utrzymującego się w populacji przez większą liczbę pokoleń. Zadbanie o różnorodność populacji skutkuje zabronieniem wstawienia do populacji duplikatu już istniejącego rozwiązania. Przy tak prosto realizowanej selekcji sterowanie zbieżnością optymalizacji przesuwa się z selekcji na zbiór operatorów, organizację populacji i inne mechanizmy hybrydowe.

Realizacja mechanizmu intensyfikacji i różnicowania za pomocą operatorów genetycznych (pseudogenetycznych) jest drugim, ważnym obszarem badań. Intensyfikacja, częściej stosowana, polega na włączeniu do AE algorytmów optymalizacji lokalnej jako dodatkowego operatora (lub elementu konstrukcji algorytmu hybrydowego). Podejście to jest wykorzystywane najczęściej w sytuacji gdy brak jest wydajnych operatorów krzyżowania.

Kolejnym zrealizowanym pomysłem [5] jest próba poprawy efektywności działania operatorów (krzyżowania *PMX*, *OX*) poprzez wykorzystanie warunkowej wartości oczekiwanej funkcji celu rozwiązań częściowo ustalonych. Koncepcja ta opiera się na założeniu, że dodatkowy parametr niesie w sobie informację na temat perspektywiczności dokonywanych zmian w rozwiązaniach.

3.2. Schemat algorytmu

W eksperymentach obliczeniowych do rozwiązywania zagadnienia QAP wykorzystano zmodyfikowany algorytm genetyczny modGA zaproponowany przez Michalewicza [2]. Wprowadzono w nim uproszczenia, polegające na losowym wyborze rodziców z populacji, za pomocą rozkładu równomiernego, oraz na współzawodnictwie nowo utworzonych rozwiązań potomnych o miejsce w populacji, na podstawie wartości funkcji celu.

Parametrami algorytmu (rys. 1) są:

O – zbiór operatorów pseudogenetycznych (U – podzbiór operatorów unarnych),
$ZP = \{p_i\}$ – zbiór prawdopodobieństw wyboru operatorów ($p_i \geq 0, i \in O, \sum_{i \in O} p_i = 1$),
λ – rozmiar populacji,
$I_{\max}$ – zadana liczba wygenerowanych przez algorytm potomków.

W algorytmie U jest podzbiorem zbioru operatorów pseudogenetycznych O, do którego należą operatory unarne, tzn. takie, które do utworzenia potomka potrzebują jednego rodzica (tworząc na jego podstawie także jednego potomka). Przez P oznaczono populację rozwiązań przetwarzaną przez algorytm, natomiast przez R wygenerowany zbiór rozwiązań wstępnych.

Opis zastosowanych w algorytmie procedur:

GenerujRozwiązaniaWstępne() – generuje losowo, na podstawie rozkładu równomiernego, populację rozwiązań początkowych.

Oceń() – określa wartości funkcji przystosowania (celu) dla rozwiązania lub rozwiązań.

UtwórzPopulacjęPoczątkową() – tworzy uporządkowaną wg wartości funkcji przystosowania (od najlepszego do najgorszego) populację początkową P o rozmiarze λ, na podstawie zbioru R zawierający rozwiązania wstępne.

LosujOperator() – losuje, na podstawie rozkładu równomiernego, operator pseudogenetyczny ze zbioru O z prawdopodobieństwem $p_i \geq 0,\ i \in O:\ \sum_{i \in O} p_i = 1$.

LosujRodzica() – losuje, dla wybranego operatora, zgodnie z rozkładem równomiernym ze zbioru P, jedno (w przypadku operatora unarnego) lub dwa (w przypadku operatora krzyżowania) rozwiązania-rodziców.

GenerujPotomka() – na podstawie wylosowanego operatora dokonuje modyfikacji rodzica / rodziców i zwraca potomka / potomków.

Wstaw() – wstawia potomka / potomków do populacji, na podstawie wartości funkcji przystosowania. Jeżeli wartość funkcji przystosowania dla potomka jest lepsza niż dla najgorszego rozwiązania ze zbioru P, to umieszcza go w zbiorze P, usuwając rozwiązanie najgorsze. W przeciwnym przypadku potomek nie jest wstawiany do P.

ZwróćNajlepsze() – zwraca najlepsze rozwiązania w populacji P.

```
Algorytm modGA(I_max, O, ZP, λ)
R ← GenerujRozwiązaniaWstępne(λ)
Oceń(R)
P ← UtwórzPopulacjęPoczątkową(R, λ)
I ← 0
while I < I_max
do {
    o_i ← LosujOperator(O, ZP)
    if o_i ∈ U
    then {
        π_1 ← LosujRodzica(P)
        π^1 ← GenerujPotomka(π_1, o_i)
        Oceń(π^1)
        Wstaw(π^1)
        I ← I + 1
    }
    else {
        π_1 ← LosujRodzica(P)
        π_2 ← LosujRodzica(P)
        {π^1, π^2} ← GenerujPotomka(π_1, π_2, o_i)
        Oceń(π^1, π^2)
        Wstaw(π^1, π^2)
        I ← I + 2
    }
}
π_best ← ZwróćNajlepsze(P)
return (π_best);
end
```

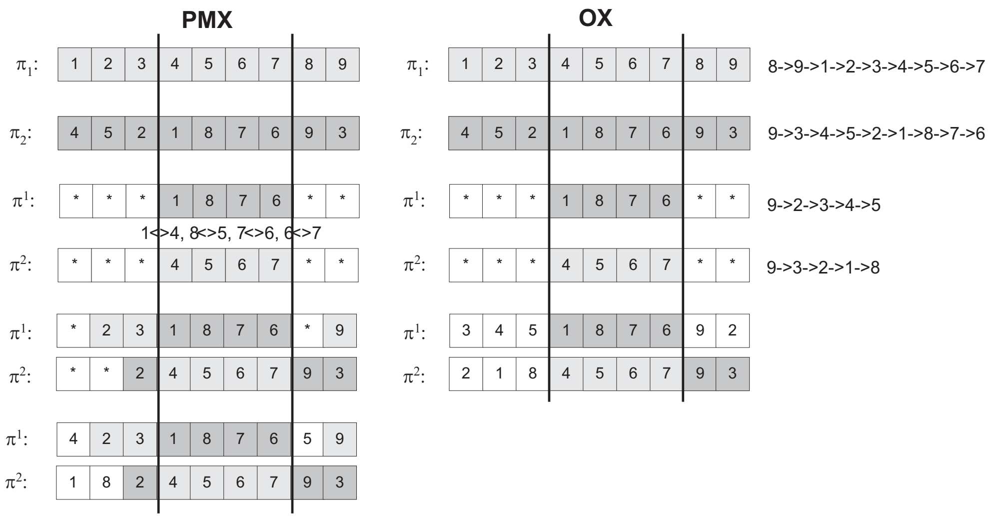
Rys. 1. Pseudokod algorytmu modGA

3.3. Specjalizowane operatory genetyczne

Specjalizowane operatory wykorzystujące warunkową wartość oczekiwaną realizują mechanizm sterowania zbieżnością algorytmu ewolucyjnego. Uzyskuje się to poprzez poprawę efektywności działania operatorów w zakresie eksploatacji (intensyfikacja) oraz eksploracji (różnicowanie) przestrzeni rozwiązań.

Warunkową wartość oczekiwaną funkcji celu – $E(\phi(\pi)\,|_{i\in D})$, w uproszczeniu oznaczoną $E(\phi(\pi))$, obliczamy dla rozwiązań częściowo ustalonych, na pozycjach różniących dwa rozwiązania. Podejście stosowane wcześniej, w którym wyznaczano dla rozwiązania: rodzica – π_1 i potomka – π^1, zbiór pozycji różnicujących oba rozwiązania $D=\left\{i:\pi_1(i)\neq\pi^1(i)\right\}$ okazało się mało efektywne obliczeniowo, gdyż wymagało rzeczywistej realizacji operacji krzyżowania lub mutacji.

Zastosowanie dla operatorów krzyżowania uproszczonej oceny perspektywiczności tworzonych rozwiązań przez ustalenie pozycji sekcji kojarzenia pierwszego rozwiązania oraz pozycji leżących poza sekcją kojarzenia drugiego rozwiązania rodzicielskiego wymaga mniejszego nakładu obliczeniowego. Jest to jednak ocena przybliżona, gdyż jak zilustrowano to na rysunku 2, użycie operatora krzyżowania, zawierającego wbudowany mechanizm usuwający „kolizje” i „dziury”, może zmienić wartości niektórych pozycji ustalonych rozwiązań potomnych poza sekcją kojarzenia (pozycje zaznaczone kolorem białym w rozwiązaniach końcowych).



Rys. 2. Schemat działania operatorów krzyżowania *PMX* oraz *OX*

W realizowanym obliczeniowo algorytmie wprowadzono do operatorów krzyżowania *PMX* i *OX* wybór, na podstawie $E(\phi(\pi))$, punktów przecięcia rozwiązań rodzicielskich, wyznaczających sekcję kojarzenia, ze zbioru losowo ustalonych K zestawów punktów (a przez to zbioru pozycji ustalonych rozwiązania $\pi_1 - D_k$). Dla operatora krzyżowania intensyfikującego (*PMX-I* oraz *OX-I*) dobór sekcji kojarzenia minimalizuje wartość oczekiwaną:

$$D^* = \min_{k=1,\ldots,K} \left\{ E(\phi(\pi_1)\,|_{i \in D_k}) + E(\phi(\pi_2)\,|_{i \notin D_k}) \right\} \tag{3}$$

natomiast dla operatora różnicującego (*PMX-D* oraz *OX-D*):

$$D^* = \max_{k=1,\ldots,K} \left\{ E(\phi(\pi_1)\,|_{i \in D_k}) + E(\phi(\pi_2)\,|_{i \notin D_k}) \right\} \tag{3a}$$

4. Wyniki badań obliczeniowych

Główny nacisk w eksperymentach obliczeniowych położono na wybór operatorów, realizowanych w postaci klasycznej lub wykorzystujących mechanizm intensyfikacji i różnicowania (*PMX*, *OX*) oraz dobór prawdopodobieństw ich stosowania. Wnioskiem z wcześniejszych badań [5, 6] było:

- ograniczenie rozważań do operatorów krzyżowania *PMX*, *OX*, optymalizacji lokalnej *LO* oraz mutacji *RM*,
- wskazanie operatorów krzyżowania jako najbardziej obiecującego elementu realizacji mechanizmów intensyfikacji i różnicowania,
- wstępne określenie zakresu wartości prawdopodobieństw efektywnego stosowania operatorów.

Na podstawie algorytmu modGA wykonano eksperymentalne oprogramowanie dla zagadnienia QAP, zaimplementowane w języku C++. Przeprowadzone doświadczenia obliczeniowe wykonano na podstawie różnych zbiorów operatorów $O = \{RM, PMX, OX, LO\}$, gdzie operatory *PMX* oraz *OX* występują w trzech wersjach: klasycznej, intensyfikującej i różnicującej (dywersyfikującej) – oznaczone odpowiednio *PMX*, *PMX-I*, *PMX-D* oraz *OX*, *OX-I*, *OX-D*. Zastosowano dla nich, podane poniżej, zróżnicowane wartości prawdopodobieństw ich losowania. Pozostałe nie zmieniane parametry algorytmu wynoszą: $I_{\max} = 10\,000$ iteracji, $\lambda = 100$ (rozmiar populacji). W tabeli 1 przedstawiono wyniki badań eksperymentalnych zagadnienia QAP, dla 39 zagadnień testowych o rozmiarze $n = 22–64$, zaczerpniętych z biblioteki QAPLIB-A [7]. Została ona stworzona w 1991 r. przez: R. Burkarda, S. Karischa, F. Rendla i zawiera instancje testowe opisujące rzeczywiste oraz wygenerowane zadania kwadratowego zagadnienia przydziału.

Eksperymenty obliczeniowe przeprowadzono dla algorytmu ewolucyjnego korzystającego z następujących zbiorów operatorów pseudogenetycznych $O = \{ O_i = \{o_1, o_2, o_3, o_4\}, i = 1, ..., 7\}$:

$O_1 = (M, PMX\text{-}I, OX, LO)$, $O_2 = (M, PMX\text{-}I, OX\text{-}D, LO)$, $O_3 = (M, PMX\text{-}I, OX\text{-}I, LO)$,

$O_4 = (M, PMX, OX, LO)$, $O_5 = (M, PMX\text{-}D, OX\text{-}D, LO)$, $O_6 = (M, PMX\text{-}D, OX\text{-}I, LO)$,

$O_7 = (M, PMX, OX\text{-}I, LO)$.

Dla wszystkich zbiorów operatorów O_i ustalono osiem zestawów prawdopodobieństw losowania kolejnych operatorów $ZP = \{ ZP_i = (p_1, p_2, p_3, p_4), i = 1, ..., 8\}$, gdzie:

$ZP_1 = (0{,}1, 0{,}2, 0{,}2, 0{,}5)$, $ZP_2 = (0{,}1, 0{,}4 , 0{,}2, 0{,}3)$, $ZP_3 = (0{,}1, 0{,}5, 0{,}2, 0{,}2)$,

$ZP_4 = (0{,}1, 0{,}4, 0{,}1, 0{,}4)$, $ZP_5 = (0{,}1, 0{,}5, 0{,}0, 0{,}4)$, $ZP_6 = (0{,}1, 0{,}1, 0{,}4, 0{,}4)$,

$ZP_7 = (0{,}1, 0{,}2, 0{,}3, 0{,}4)$, $ZP_8 = (0{,}0, 0{,}5, 0{,}0, 0{,}5)$.

Tabela 1
Średnie procentowe błędy – $E_{śr}$ uzyskane w eksperymentach obliczeniowych

	ZP_1	ZP_2	ZP_3	ZP_4	ZP_5	ZP_6	ZP_7	ZP_8	**Śred.**	**Min.**
O_1	1,14	1,94	2,20	1,37	1,98	1,91	2,20	2,02	1,84	1,14
O_2	2,28	1,56	2,17	1,80	1,98	1,89	2,47	2,02	2,02	1,56
O_3	1,89	2,56	2,22	1,99	1,98	2,52	1,89	2,02	2,13	1,89
O_4	1,84	2,58	2,22	1,41	1,59	2,19	2,39	2,17	2,05	1,41
O_5	1,84	2,12	2,36	2,36	1,81	2,80	1,49	1,58	2,04	1,49
O_6	2,03	2,55	2,55	1,20	2,80	1,97	1,99	1,89	2,12	1,20
O_7	1,74	2,14	2,14	1,58	1,59	2,26	1,91	1,91	1,91	1,58

W tabeli 1 przedstawiono uśrednione wartości błędów $E_{śr}$ uzyskane dla całego zestawu 39 zadań testowych (dla pięciu przebiegów algorytmu):

$$E_{Śr} = 100\% * \left(\sum_{i \in I} (\phi_i - \phi_{i_{QAPLIB-A}}) / \phi_{i_{QAPLIB-A}} \right) / |I| \tag{4}$$

gdzie:

I – zbiór instancji testowych dla zagadnienia QAP, zaczerpniętych z biblioteki QAPLIB-A;
$|I|$ – liczebność zbioru testowego (39 instancji testowych);
ϕ_i – wartość średnia funkcji przystosowania najlepszego znalezionego rozwiązania – π_{best}, dla i-tej instancji testowej z pięciu przebiegów algorytmu;
$\phi_{iQAPLIB\text{-}A}$ – najlepsza znana wartość funkcji przystosowania dla i-tej instancji testowej, zaczerpnięta z biblioteki QAPLIB-A.

5. Podsumowanie

Zamieszczone w tabeli 1 wyniki pozwalają zauważyć, że:

– Zastosowanie operatora *PMX-I* z *OX*, *PMX-I* z *OX-D* lub *PMX* z *OX-I* pozwala poprawić uzyskiwane wyniki przez algorytm stosujący klasyczne operatory *PMX* i *OX*.

- Połączenie działania operatora *PMX-D* oraz *OX-I*, pomimo że statystycznie pogorszyło uzyskane wyniki, to pozwoliło poprawić najlepsze dotąd znane rozwiązania dla zadań testowych: BUR26B (z 3.117.852 na 3.117.850), BUR26F (z 3.782.044 na 3.782.040), LIPA50B (z 1.210.244 na 1.210.240).
- Najlepsze statystycznie wyniki uzyskano w przypadku połączenia działania operatorów *RM*, *PMX-I*, *OX* i *LO*, przy prawdopodobieństwach ich stosowania wynoszących odpowiednio: 0,1, 0,2, 0,2, i 0,5.
- Łączne zastosowanie operatorów *PMX-I* oraz *OX-I* powoduje wyraźne pogorszenie uzyskanych wyników w porównaniu do ich wersji klasycznych.
- W przypadku małego prawdopodobieństwa losowania operatora *LO* spada wydajność algorytmu.

Podsumowując, można wyciągnąć następujące ogólne wnioski:

- Mechanizm intensyfikacji lepiej ukierunkowuje proces przeszukiwania przestrzeni rozwiązań.
- Zastosowanie warunkowej wartości oczekiwanej w konstrukcji operatora *PMX* jest bardziej skuteczne niż w przypadku operatora *OX*. Można to uzasadnić tym, że operator *OX* wprowadza większe zaburzenia podczas tworzenia rozwiązań-potomków, związane z koniecznością zachowania dopuszczalności rozwiązania. Dlatego warunkowa wartość oczekiwana dla podciągów rozwiązań, na podstawie której wybrano sekcję kojarzenia u rodziców, nie odpowiada dokładnie wartości oczekiwanej wymienionych podciągów znajdujących się w rozwiązaniach potomnych.
- Istotnym warunkiem uzyskania wysokiej efektywności algorytmu jest stosowanie dużego prawdopodobieństwa użycia operatora optymalizacji lokalnej *LO* oraz równomiernych prawdopodobieństw dla operatorów *PMX-I* i *OX*. Intensywne wykorzystanie optymalizacji lokalnej daje możliwość wydajniejszej eksploatacji przestrzeni rozwiązań w rejonach perspektywicznych „wskazanych” przez operatory krzyżowania.
- Zastosowanie różnicowania, szczególnie dla *PMX*, pozwala na przeniesienie procesu przeszukiwania w obszary dotąd nie eksplorowane przy zastosowaniu innych operatorów. Dzięki temu możliwa jest poprawa najlepszych, dotychczas znanych rozwiązań, dla niektórych instancji testowych zagadnienia QAP.

Literatura

[1] Goldberg D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, 1989; tłum. na jęz. polski: *Algorytmy genetyczne i ich zastosowania*. WNT, Warszawa 1995

[2] Michalewicz Z.: *Genetic Algorithms + Data Strructures = Evolution Programs*. Springer Verlag, 1995; tłum. na jęz. polski: *Algorytmy genetyczne + struktury danych = programy ewolucyjne*, WNT, Warszawa 1996

[3] Chmiel W.: *Algorytmy ewolucyjne w optymalizacji przydziału zadań z kwadratową funkcją celu.* Kraków, AGH 2005 (Praca doktorska)

[4] Nissen V.: *Evolutionäre Algorithmen Darstellung, Beispiele, betriebswirtschaftliche Anwendungs-möglichkeiten*. Wiesbaden, Deutscher Universitäts Verlag 1994

[5] Kadłuczka P., Chmiel W.: *Efektywność algorytmu ewolucyjnego wykorzystującego warunkową wartość oczekiwaną funkcji celu.* Automatyka, AGH, Kraków, 2005, 105–114

[6] Chmiel W., Kadłuczka P., Jędrusik S.: *Efektywność algorytmu ewolucyjnego wykorzystującego mechanizm różnicowania*. Zeszyty Naukowe Politechniki Śląskiej, seria: AUTOMATYKA, Gliwice 2006, z. 143, 33–44

[7] Burkard R.E., Karisch S.E., Rendl F.: *QAPLIB-A Quadratic Assignment Problem Library.* European Journal of Operational Resarch, 55, 1991, 115