

KAZIMIERZ KIELKOWICZ\*, ZBIGNIEW KOKOSIŃSKI\*\*

## ALGORYTM HYBRYDOWY DLA PROBABILISTYCZNEGO PROBLEMU KOMIWOJAŻERA

### A HYBRID ALGORITHM FOR PROBABILISTIC TRAVELING SALESMAN PROBLEM

#### Streszczenie

W artykule rozważono Probabilistyczny Problem Komiwojażera (PTSP), dla którego został zaproponowany algorytm hybrydowy, łączący algorytm ewolucyjny z metodami optymalizacji lokalnej i obliczeniami równoległymi. Metody optymalizacji lokalnej obejmują operatory 1-shift i 2-p-opt. Przebadano eksperymentalnie kilka wariantów algorytmu ewolucyjnego i hybrydowego oraz wpływ zastosowanych metod optymalizacji lokalnej i metod zrównoleglenia obliczeń na jakość znajdowanych rozwiązań.

*Słowa kluczowe: PTSP, algorytm ewolucyjny, równoległy algorytm ewolucyjny, algorytm hybrydowy*

#### Abstract

In this paper Probabilistic Traveling Salesman Problem (PTSP) is considered and a hybrid algorithm is proposed, in which an evolutionary algorithm is combined with local optimization and parallelization techniques. Local optimization methods include 1-shift and 2-p-opt operators. Several basic variants of evolutionary and hybrid algorithms are experimentally tested and compared.

*Keywords: PTSP, evolutionary algorithm, parallel evolutionary algorithm, hybrid algorithm*

\* Mgr inż. Kazimierz Kielkowicz, Studium Doktoranckie, Instytut Badań Systemowych, Polska Akademia Nauk.

\*\* Dr inż. Zbigniew Kokosiński, Katedra Automatyki i Technik Informatycznych, Wydział Inżynierii Elektrycznej i Komputerowej, Politechnika Krakowska.

## 1. Wstęp

W artykule rozważono Probabilistyczny Problem Komiwożacza (PTSP), który stanowi uogólnienie klasycznego Problemu Komiwożacza (TSP). W problemie PTSP miasta na trasie komiwożacza są odwiedzane z pewnym, z góry określonym, prawdopodobieństwem. Zadanie polega na znalezieniu takiej trasy *a priori*, która minimalizuje wartość oczekiwaną kosztu ścieżki. PTSP jest problemem NP-trudnym, co wynika z NP-trudności TSP. TSP jest szczególnym przypadkiem PTSP, gdzie prawdopodobieństwo odwiedzenia każdego miasta jest równe 1. Problem PTSP może zostać wykorzystany do modelowania zadań w logistyce i planowaniu transportu, takich jak: dzienna dostawa towarów z losowym zapotrzebowaniem, sekwencjonowanie zadań ze zmiennym kosztem czy planowanie trasy w domach spokojnej starości [9].

Do rozwiązywania problemów optymalizacyjnych z klasy NP stosowane są często uniwersalne algorytmy iteracyjne, jakimi są metaheurystyki. W wyniku połączenia dwóch lub więcej takich samych (lub różnych) metaheurystyk powstaje homogeniczny (lub heterogeniczny) algorytm hybrydowy. Połączenie to może mieć charakter jednorodny lub niejednorodny i zostać zrealizowane sekwencyjnie albo w modelu obliczeń równoległych. W literaturze spotykane są opisy algorytmów hybrydowych zbudowanych z połączenia algorytmu optymalizacji globalnej, takiego jak algorytm ewolucyjny (EA), z heurystyką przeszukiwania lokalnego. Można tego dokonać np. przez wykorzystanie algorytmu wyszukiwania z tabu (TS) jako mutacji [12] lub przez zastosowanie mechanizmu dekodowania chromosomu [7].

W artykule przedstawiono algorytm hybrydowy, w którym zastosowano metody optymalizacji lokalnej w algorytmie ewolucyjnym oraz równoległy model migracyjny EA. Badania eksperymentalne kilku wariantów algorytmu ewolucyjnego i hybrydowego dla problemu PTSP wykonano na podstawie odpowiednio zmodyfikowanych benchmarków dla problemu TSP [13].

Dalsza część artykułu zorganizowana jest następująco. W rozdziale 2 przedstawiono formalną definicję problemu PTSP oraz jego ogólną interpretację. W rozdziale 3 zawarto przegląd literatury. W rozdziale 4 przedstawiono istniejące metody optymalizacji lokalnej, tj. 1-shift i 2-p-opt. W rozdziale 5 zawarto opis algorytmu hybrydowego, a w rozdziale 6 przeprowadzono badania symulacyjne. W rozdziale 7 podsumowano wyniki pracy i wskazano kierunki dalszych badań.

## 2. Problem PTSP

Probabilistyczny Problem Komiwożacza (PTSP) jest rozszerzeniem klasycznego Problemu Komiwożacza (TSP). Oprócz odległości między parami miast zadane jest prawdopodobieństwo odwiedzenia danego miasta. Celem optymalizacji jest znalezienie takiej trasy *a priori*, która minimalizuje całkowity oczekiwany koszt ścieżki komiwożacza. Klasyczny problem komiwożacza polega na znalezieniu najkrótszej trasy pomiędzy miastami, w której każde miasto musi zostać odwiedzone tylko jeden raz, w wersji probabilistycznej każde miasto może zostać odwiedzone tylko z pewnym prawdopodobieństwem. Komiwożacz w pierwszej kolejności decyduje o kolejności odwiedzenia miast (tworzy trasę *a priori*), a następnie dostaje informacje, które miasta mają zostać odwiedzone, a które nie. Miasta są odwiedzane w kolejności wyznaczonej przez trasę *a priori* z pominięciem miast, których nie trzeba odwiedzić – tworząc w ten sposób trasę „wyciętą”. Zadaniem optymalizacji jest znalezienie takiej trasy *a priori*, która minimalizuje średnią długość wszystkich możliwych „wyciętych” tras komiwożacza.

## 2.1. Formalna definicja PTSP

Rozważmy pełny, symetryczny graf ważony  $G = (V, E)$ , gdzie  $V = \{v_1, v_2, v_3, \dots, v_n\}$  jest zbiorem wierzchołków,  $E \subseteq V \times V$  jest zbiorem krawędzi. Do każdego wierzchołka  $v_i$  ( $i = 1, 2, 3, \dots, n$ ) przypisane jest prawdopodobieństwo  $p_i$  wystąpienia wierzchołka  $v_i$ . Zadaniem PTSP jest znalezienie takiego cyklu Hamiltona w grafie  $G$ , który minimalizuje funkcję celu:

$$E[\lambda] = \sum_{i=1}^n \sum_{r=1}^{n-1} d(\lambda(i), \lambda(i+r)) p_{\lambda(i)} p_{\lambda(i+r)} \prod_{t=i+1}^{i+r-1} q_{\lambda(t)} \quad (1)$$

$$q_{\lambda(t)} = 1 - p_{\lambda(t)}$$

Przez  $d(i, j)$ ,  $i, j = 1, 2, 3, \dots, n$ , rozumieć należy wagę krawędzi pomiędzy wierzchołkami  $v_i$  oraz  $v_j$ , gdzie  $v_p, v_j \in V$ . Przez  $\lambda = (\lambda(1), \lambda(2), \lambda(3), \dots, \lambda(n))$  rozumieć należy permutację bez powtórzeń nad zbiorem  $V$ , w dalszej części artykułu zwaną naprzemiennie trasą *a priori* lub osobnikiem. Przez krawędź lub łuk rozumiana będzie trasa od  $\lambda(i)$ , do  $\lambda(j)$ ,  $\forall (\lambda(i), \lambda(j)) \in \lambda$ , Problem PTSP jest problemem NP-trudnym, wynika to wprost z NP-trudności TSP, zostało to udowodnione w [4].

W przypadku, gdy  $\forall p_i = p$  mówimy o homogenicznym, a w przeciwnym przypadku o heterogenicznym Probabilistycznym Problemie Komijóźera.

## 2.2. Funkcja celu

Funkcja celu (oczekiwany koszt trasy  $\lambda$ ) obliczany jest w czasie  $O(n^2)$  [4]:

W przypadku heterogenicznym:

$$\prod_{t=i}^j q_{\lambda(t)} = \begin{cases} \prod_{t=i}^j q_{\lambda(t)}, & 0 \leq j - i \leq n - 1 \\ \prod_{t=1}^n q_{\lambda(t)} \prod_{u=1}^j q_{\lambda(u)}, & i - j \geq 1 \\ 1, & \text{w przeciwnym przypadku} \end{cases} \quad (2)$$

$$q_i = 1 - p_i$$

Oczekiwany koszt trasy jest sumą wag wszystkich łuków w trasie  $\lambda$  pomiędzy miastem  $\lambda(i)$ , a  $\lambda(i+r)$ , pomnożonych przez prawdopodobieństwo wystąpienia tych miast, tj,  $p_{\lambda(i)} p_{\lambda(i+r)}$  oraz prawdopodobieństwo nie wystąpienia między nimi innych miast  $q_{\lambda(t)}$ .

W przypadku homogenicznym:

$$E[\lambda] = \prod_{i=1}^n \prod_{r=1}^{n-1} p^2 q^{r-1} d(\lambda(i), \lambda(i+1)) \quad (3)$$

$$q = 1 - p$$

Oczekiwany koszt trasy (3) jest ważoną sumą wag wszystkich łuków w trasie  $\lambda$ .

### 3. Przegląd literatury

W 1985 roku Jaillet [4] zaprezentował Probabilistyczny Problem Komiwożazera (PTSP), który jest probabilistyczną wersją klasycznego problemu komiwożazera (TSP). W następnej pracy [5] autor wykazał, że optymalne rozwiązanie dla TSP nie jest optymalnym rozwiązaniem dla PTSP, co wskazało potrzebę stworzenia nowych narzędzi, zbudowanych z uwzględnieniem probabilistycznej natury problemu.

W chwili powstawania niniejszego artykułu dostępnych było kilka publikacji implementujących inteligentne metody obliczeniowe dla problemu PTSP: *Scatter Search* z algorytmem symulowanego wyżarzania (SA) [8], algorytm ewolucyjny (EA) [10] oraz algorytm optymalizacji mrowiskowej (ACO) [2, 3]. W pracach [8, 10] wykazano efektywność operatora krzyżowania ERX, w porównaniu z innymi operatorami rekombinacji. Fakt ten został również potwierdzony w [6], gdzie operator ERX został porównany z operatorem UOBX dla różnych rozkładów gęstości prawdopodobieństwa.

W [6] zostały rozważone różne schematy migracyjne dla algorytmu genetycznego. Stwierdzono, że migracja nie może występować zbyt często, gdyż doprowadza to nasycania się populacji osobnikami o zbliżonych cechach, co jest szczególnie niekorzystne w połączeniu z operatorem ERX.

We wspomnianych pracach metody optymalizacji lokalnej (o ile je wykorzystywano) ograniczały się do jednej z dwóch metod zaprezentowanych w [1]. Na uwagę zasługuje fakt, że owe metody, a właściwie równania rekurencyjne z nimi związane, zawierały błąd, co zostało wykazane w [2]. Tam też przedstawiono poprawne równania operatorów 2-p-opt i 1-shift dla przypadku heterogenicznego, jak i homogenicznego.

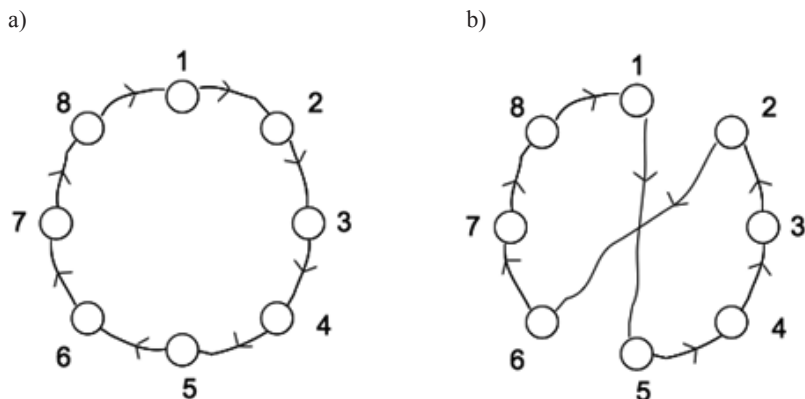
### 4. Metody optymalizacji lokalnej

Istnieją dwie podstawowe metody, 2-p-opt i 1-shift, które służą optymalizacji lokalnej w PTSP. Są to odpowiedniki metod stosowanych w TSP dostosowane do probabilistycznej natury PTSP. Dostosowanie to polega na uproszczeniu złożoności obliczeniowej funkcji celu (1) poprzez wyprowadzenie równań rekurencyjnych. Dokładny opis metod optymalizacji lokalnej, wraz z odpowiadającymi im równaniami rekurencyjnymi uproszczającymi obliczenia, znajduje się w [6] – tutaj przedstawimy jedynie istotę tych operatorów.

#### 4.1. Operator 2-p-opt

Metoda 2-p-opt polega na odwróceniu sekwencji miast w trasie  $\lambda$  między miastem  $\lambda(i)$  i  $\lambda(j)$ . Jest to specjalna odmiana algorytmu 2-opt, stosowanego w klasycznej wersji problemu TSP, dostosowana do stochastycznej natury problemu PTSP poprzez wprowadzenie równań rekurencyjnych wyznaczających zmianę wartości funkcji celu. Dokładna forma równań, jak i dokładny opis działania operatora został przedstawiony w [6]. Przykład zastosowania operatora 2-p-opt dla trasy przedstawionej na rys. 1a zaprezentowano na rys. 1b.

Całkowity czas potrzebny na optymalizację osobnika z wykorzystaniem metody 2-p-opt wynosi  $O(n^2)$  w przypadku eksploracji całego otoczenia osobnika (tzn, wszystkich możliwych rozwiązań osiągalnych przez zadanie operatora 2-p-opt).

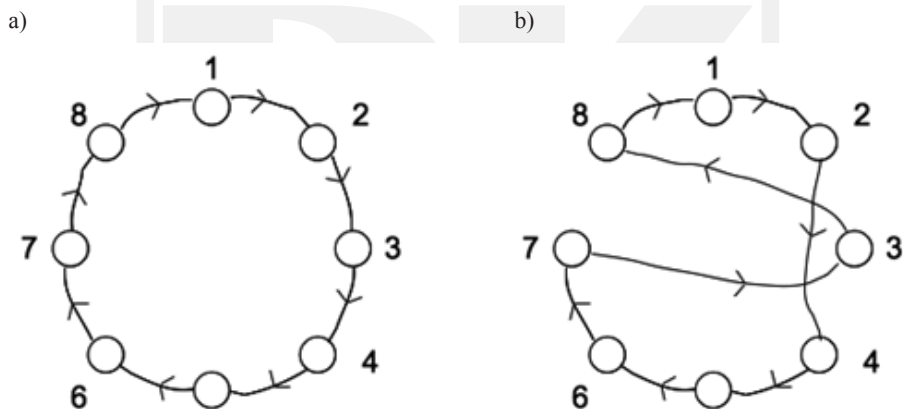


Rys. 1. Zastosowanie operatora 2-p-opt dla  $i = 2, j = 7$

Fig. 1. 2-p-opt operator used for  $i = 2$  and  $j = 7$

#### 4.2. Operator 1-shift

Metoda 1-shift polega na przesunięciu sekwencji miast od pozycji  $i$  do  $j$  w trasie  $\lambda$  o jedną pozycję w lewo, tak aby miasto na pozycji  $i + 1$  znalazło się na pozycji  $i$ , a miasto na pozycji  $i$  na miejscu miasta  $j$ ; dokładny opis metody wraz z odpowiednimi równaniami znajduje się w [6]. Przykład zastosowania operatora 1-shift dla trasy na rys. 2a pokazano na rys. 2b.



Rys. 2. Zastosowanie operatora 1-shift dla  $i = 3, j = 7$

Fig. 2. 1-shift operator used for  $i = 3$  and  $j = 7$

Całkowity czas potrzebny na optymalizację osobnika z wykorzystaniem metody 1-shift w przypadku eksploracji całego otoczenia (tzn. wszystkich osobników osiągalnych za pomocą metody 1-shift) wynosi  $O(n^2)$ .

## Algorytm hybrydowy

```

1:   inicjalizuj populację początkową  $P$  algorytmem  $NN$ 
2:   wyznacz dopasowanie  $\forall \lambda \in P$  jako  $\frac{1}{\varepsilon[\lambda]}$ 
3:   if optymalizacja początkowa then
4:      $\forall \lambda \in P$  zastosuj operator 1-shift i/lub 2-p-opt
5:   end if
6:   while warunek stopu nie jest spełniony do
7:      $P' \leftarrow$  selekcja turniejowa z  $P$  (z elitaryzmem)
8:     if uzupełnij populację then
9:       zastąp 5% najsłabszych osobników z  $P'$ , osobnikami wygenerowanymi z  $NN$ 
10:    end if
11:    krzyżuj osobniki z  $P'$  metodą ERX
12:    mutuj osobniki z  $P'$  metodą Transposition
13:    if optymalizacja lokalna then
14:       $\forall \lambda \in P$  zastosuj operator 1-shift i/lub 2-p-opt
15:    end if
16:    if migracja osobników then
17:      wybierz reprezentantów z  $P'$  i dokonaj migracji „do następnego”
18:    end if
19:     $P \leftarrow P'$ 
20:  end while

```

## 5. Algorytm hybrydowy

Jednym ze stosowanych schematów zrównoleglenia algorytmu genetycznego jest prowadzenie obliczeń w ramach współbieżnych procesów ewolucyjnych – zwanymi wyspami – w tzw. modelu migracyjnym, w którym występuje wymiana osobników między wyspami. „Algorytm 1” przedstawia schemat algorytmu hybrydowego, opartego na równoległym algorytmie ewolucyjnym w modelu migracyjnym, w którym hybrydyzacja została zrealizowana przez zastosowanie metod optymalizacji lokalnej (2-p-opt i 1-shift). W modelu migracyjnym współbieżne procesy ewolucyjne wymieniają się informacją o najlepszych znalezionych rozwiązaniach, z pewnym niewielkim prawdopodobieństwem, w każdej iteracji i według zdefiniowanego schematu migracyjnego. Do kodowania osobników wykorzystano reprezentację ścieżkową. Każdy osobnik populacji początkowej algorytmu inicjowany jest za pomocą metody zachłannej  $NN$  (*Nearest Neighbour*), która buduje rozwiązanie, zaczynając od losowo wybranego wierzchołka, a następnie sukcesywnie wybiera ze zbioru jeszcze niewybranych wierzchołków następny – najbliższy. Skuteczność algorytmu  $NN$  dla zadania PTSP została potwierdzona w [6]. W razie potrzeby populacja poddawana jest optymalizacji lokalnej.

Linie 8–10 prezentują mechanizm uzupełniania populacji rodzicielskiej nowo wygenerowanymi osobnikami w celu uniknięcia przedwczesnej zbieżności algorytmu. Linie 16–18 realizują migrację osobników między procesami ewolucyjnymi.

## 6. Badania symulacyjne

Trzy warianty „Algorytmu 1”, w którym populacja początkowa została zoptymalizowana i występowała migracja między wyspami, zostały porównane z trzema wariantami algorytmu ewolucyjnego (tab. 4).

Dla danych testowych pochodzących z bibliotek TSPLib [13] zostały wygenerowane wektory prawdopodobieństw z rozkładem jednostajnym, Zmodyfikowane instancje testowe oznaczono dla odróżnienia od instancji standardowych sufiksem „\*”. Zadanie optymalizacyjne polegało na minimalizacji długości trasy *a priori*.

Implementacja modelu migracyjnego zrealizowana została w architekturze „master-slave”, gdzie każdy z dwóch procesów „slave” odpowiadał jednej podpopulacji, a „master” odpowiedzialny był za komunikację między podpopulacjami. Ewolucja w każdej podpopulacji stanowiła oddzielny proces. Procesy komunikowały się za pomocą wymiany wiadomości (ang. *message passing*), wykorzystano biblioteki MPICH2 [11].

Badania zostały przeprowadzone na procesorze dwurdzeniowym Intel Core 2 Duo 2.1 GHz, wszystkie algorytmy użyte w testach zostały zaimplementowane w języku C.

### 6.1. Wpływ metody optymalizacji lokalnej

W celu zbadania jakości generowanych rozwiązań przez operatory optymalizacji lokalnej, utworzono za pomocą algorytmu NN jednego osobnika, a następnie poddano go wielokrotnej optymalizacji, za każdym razem losując punkty  $i, j$  z rozkładem jednostajnym. Kryterium stopu stanowiło 100 iteracji. Wpływ metod optymalizacji przedstawiono w tabelach 1, 2 i 3, odpowiednio dla metod 1-shift, 2-p-opt oraz sekwencyjnego połączenia obu operatorów, Tabele zawierają informacje o najkrótszej, najdłuższej oraz średniej znalezionej trasie, średnim czasie potrzebnym na wykonanie symulacji oraz odchyleniu standardowym długości trasy. Test powtórzono 10 razy dla każdej instancji.

Tabela 1

**Długość trasy wyznaczona metodą 1-shift**

Instancja	Znaleziona trasa			Średni czas obliczeń [s]	Odchylenie standardowe długości trasy
	najkrótsza	średnia	najdłuższa		
eil51*	386,51	402,88	413,79	0,13	13,37
berlin52*	7243,87	7471,85	7813,81	0,18	279,21
rd100*	7153,91	7571,71	8076,91	1,90	322,29
kroA100*	18848,70	20421,69	21803,60	1,97	862,90
kroC100*	19527,20	21394,91	24820,10	1,89	1825,36
eil101*	550,10	573,54	618,87	2,18	19,61
lin105*	13008,50	14635,13	15882,00	2,62	933,25
ch150*	5625,81	6092,12	6341,78	9,84	223,87

Długość trasy wyznaczona metodą 2-p-opt

Instancja	Znaleziona trasa			Średni czas obliczeń [s]	Odchylenie standardowe długości trasy
	najkrótsza	średnia	najdłuższa		
eil51*	378,84	405,79	421,82	0,46	14,56
berlin52*	6763,82	7179,45	7763,94	0,44	354,81
rd100*	7163,98	7533,50	7862,07	6,27	211,27
kroA100*	20508,90	21374,58	21953,90	6,32	536,96
kroC100*	19971,30	21532,96	23472,90	5,74	1211,05
eil101*	556,61	578,66	601,74	6,53	14,61
lin105*	13100,0	14641,66	16137,20	7,66	1190,46
ch150*	5676,87	6194,46	6596,17	29,18	370,47

Przeważnie najlepsze rezultaty daje połączenie dwóch metod optymalizacji, jednak jest to rozwiązanie bardziej kosztowne. Nie ma natomiast większej różnicy pomiędzy metodami 1-shift czy 2-p-opt, obie generują podobnej jakości rozwiązanie mimo różnego schematu eksploracji przestrzeni dopuszczalnych rozwiązań.

Tabela 3

Długość trasy wyznaczona metodami 2-p-opt i 1-shift

Instancja	Znaleziona trasa			Średni czas obliczeń[s]	Odchylenie standardowe długości trasy
	najkrótsza	średnia	najdłuższa		
eil51*	381,90	407,41	420,95	0,59	10,70
berlin52*	6716,67	7125,82	7729,57	0,68	362,25
rd100*	7346,66	7657,58	7998,08	8,24	191,26
kroA100*	18678,80	20002,50	21792,60	8,35	1205,67
kroC100*	18762,20	20901,85	23794,7	8,72	1501,43
eil101*	538,82	578,84	631,83	8,43	28,89
lin105*	13050,60	14706,53	15812,80	10,30	814,63
ch150*	5598,42	5977,07	6552,47	39,34	298,08

## 6.2. Badania eksperymentalne algorytmów

W celu zbadania jakości generowanych rozwiązań przez algorytm hybrydowy zbadano go w czterech wariantach, zbudowanych z ogólnego schematu (algorytm 1) oraz porównano jakość generowanych rozwiązań z implementacją algorytmu ewolucyjnego oraz równoległego algorytmu ewolucyjnego. Zestawienie konfiguracji testowanych algorytmów zawarto w tabeli 4, natomiast średnią długość znalezionej trasy oraz średni czas obliczeń z 5 prób symulacyjnych – w tab. 5 (najlepsze znalezione rozwiązanie zostało pogrubione) oraz tab. 6. W tab. 7 przedstawiono procentowy odstęp od średnio najlepszego znalezionej trasy (wyróżnionego pogrubieniem), obliczany według wzoru (4):



$$r_{i,j} = \left( \frac{v_{i,j}}{\min_i \{v_{i,j}\}} \cdot 100\% \right) - 100\% \quad (4)$$

gdzie:

- $i \in I, j \in J$   
 $I$  – zbiór testowych instancji,  
 $J$  – zbiór testowanych algorytmów,  
 $r_{i,j}$  – procentowa różnica wyniku w testu  $(i, j)$ ,  
 $v_{i,j}$  – rozwiązanie znalezione przez test  $(i, j)$ ,  
 $\min_i \{v_{i,j}\}$  – najlepsze rozwiązanie znalezione we wszystkich testach  $i$  dla  $j$ .

Algorytm hybrydowy „H4” generował przeważnie najlepsze rozwiązania ze wszystkich przetestowanych algorytmów. Porównanie algorytmów „H1” i „H2” sugeruje, że bardziej opłacalną strategią hybrydyzacji jest stosowanie metody 2-p-opt jako metody optymalizacji lokalnej. Algorytm „H2” średnio znajdował lepsze rozwiązania od swojego bezpośredniego konkurenta algorytmu „H1”, ale potrzebował na to więcej czasu. Algorytm „H3”, który używał dwóch różnych metod optymalizacji na wyspach okazał się równie szybki, jak algorytm „H2”, wynika to z faktu synchronizacji między procesami. Jednocześnie okazało się, że takie podejście (stosowanie dwóch różnych metod optymalizacji na dwóch różnych wyspach) średnio znajduje lepsze rozwiązania niż używanie tylko jednej metody.

Tabela 4

#### Zestawienie testowanych algorytmów

Parametr	H1	H2	H3	H4	E1	E2	E3
Zrównoleglenie	tak				nie		tak
Rozmiar populacji	50						
Inicjalizacja populacji	algorytm NN						
Optymalizacja początkowa	1-shift	2-p-opt	1-shift i 2-p-opt	1-shift i 2-p-opt	1-shift i 2-p-opt	–	1-shift i 2-p-opt
Selekcja	Turniej z elitaryzmem						
Krzyżowanie	ERX, z prawdopodobieństwem 0,4						
Mutacja	Transposition, z prawdopodobieństwem 0,1						
Optymalizacja osobników	1-shift	2-p-opt	1-shift/2-p-opt	1-shift i 2-p-opt	–	–	–
Model migracji	„do kolejnej”, 5 osobników, z prawdopodobieństwem 0,2						taki jak H1-H4
Kryterium stopu	25 iteracji bez poprawy rozwiązania						

Porównując algorytmy „E1” i „E3”, widać, jaki wpływ na średnie rozwiązanie ma zrównoleglenie algorytmów. Algorytm „E3” średnio znajduje dużo lepsze rozwiązanie od algorytmu „E1”. Najgorsze średnie rozwiązanie dla każdej instancji znajdował „czysty” (nie podle-

gający hybrydyzacji ani zrównolegleniu) algorytm ewolucyjny „E2”. Wszystkie hybrydyzowane algorytmy znajdowały lepsze średnie rozwiązania (tylko w dwóch testach równoległy algorytm ewolucyjny znajdował średnio lepsze rozwiązanie).

Tabela 5

## Wyznaczona długość trasy (średnia z 5 prób)

Instancja	H1	H2	H3	H4	E1	E2	E2
eil51*	337,92	333,86	328,58	<b>325,24</b>	363,43	364,40	358,941
berlin52*	6626,07	6337,99	6324,42	<b>6321,20</b>	6710,81	6625,75	6508,79
rd100*	6700,92	6680,92	6654,21	<b>6508,79</b>	6961,15	7102,21	6777,74
kroA100*	17833,00	17612,70	<b>17555,70</b>	18265,80	18555,70	19008,60	18769,1
kroC100*	18820,50	18951,00	18859,00	18751,00	18996,3	18969,20	<b>18723,9</b>
eil101*	514,12	510,532	512,95	<b>509,36</b>	521,566	529,62	519,459
lin105*	12890,30	12795,00	12845,00	<b>12437,50</b>	12899,00	13003,30	12897
ch150*	5574,97	5567,29	5572,74	<b>5565,42</b>	5588,78	5621,19	5577,74

Tabela 6

## Średni czas potrzebny na znalezienie rozwiązania [s] (średnia z 5 prób)

Instancja	H1	H2	H3	H4	E1	E2	E2
eil51*	396,98	461,31	415,35	338,64	23,69	2,93	22,88
berlin52*	401,82	435,59	336,68	555,63	26,93	0,86	24,92
rd100*	578,09	1232,54	169,063	1473,80	295,59	3,58	291,14
kroA100*	1461,34	1743,00	1510,61	4046,84	298,17	5,13	287,75
kroC100*	1397,84	1128,50	1549,19	1310,46	289,54	4,07	287,25
eil101*	1176,14	1516,32	1441,78	2913,26	306,01	11,52	305,05
lin105*	3192,48	3237,27	3649,86	40809,91	339,00	9,96	343,65
ch150*	5119,42	3818,10	4836,26	4788,78	1303,02	13,13	1333,06

Tabela 7

## Różnica procentowa długości trasy wyznaczona ze wzgl. na najlepsze rozwiązanie

Instancja	H1	H2	H3	H4	E1	E2	E2
eil51*	3,89	2,65	1,0272	<b>0</b>	11,7421	12,0398	10,3622
berlin52*	4,82	0,26	0,0509	<b>0</b>	6,1635	4,8179	2,9676
rd100*	2,95	2,64	2,2342	<b>0</b>	6,9500	9,1172	4,1321
kroA100*	1,57	0,32	<b>0</b>	4,0448	5,6962	8,2759	6,9117
kroC100*	0,51	1,21	0,7215	0,1447	1,4548	1,3101	<b>0</b>
eil101*	0,93	0,23	0,7068	<b>0</b>	2,3967	3,9787	1,9831
lin105*	3,64	2,87	3,2764	<b>0</b>	3,7106	4,5491	3,6945
ch150*	0,17	0,03	0,1315	<b>0</b>	0,4197	1,0021	0,2214
Średnia	2,31	1,27	1,0186	0,5237	4,8167	5,6364	3,7841

## 7. Podsumowanie

W artykule przedstawiono Probabilistyczny Problem Komiwojażera (PTSP), który stanowi uogólnienie klasycznego Problemu Komiwojażera (TSP), przedstawiono jego formalną definicję i ogólną interpretację. Opisano dwie istniejące metody optymalizacji lokalnej oraz przedstawiono idee ich działania. Przebadano wpływ operatorów optymalizacji rozwiązań na jakość generowanych rozwiązań oraz stwierdzono, że żadna z dwóch porównywanych metod nie wyróżnia się. Zaprezentowano algorytmy hybrydyzowane metodami optymalizacji lokalnej oraz porównano je z trzema niehybrydyzowanymi algorytmami genetycznymi.

Z przeprowadzonych badań wynika, że najlepsze rezultaty dla problemu PTPS dał algorytm używający obu metod optymalizacji lokalnej (1-shift i 2-p-opt) jednocześnie. Optymalizacji osobników zawsze towarzyszy wzrost czasu potrzebnego na ukończenie obliczeń, dlatego zasadne zdaje się rozdzielenie metod optymalizacji pomiędzy wyspami (procesami), tak aby na każdej wyspie odbywała się optymalizacja inną metodą. Takie podejście umożliwia „szerszą” eksplorację przestrzeni rozwiązań, a – jak wykazano drogą symulacji – znalezione rozwiązania są wysokiej jakości.

Dalsze badania mogą być ukierunkowane na przeniesienie obliczeń do architektury GPU (CUDA), przebadanie sprawności algorytmu przy występowaniu innych rozkładów prawdopodobieństwa oraz rozważyć różne klasy zadań zeń powstałych i współzależność między nimi, śledzenie optimum w dynamicznie zmieniającym się otoczeniu lub znalezienie odpornego rozwiązania, które będzie nieczułe na drobne zmiany danych wejściowych zadania optymalizacyjnego.

## Literatura

- [1] Bertsimas D.J., Howell L., *Further results on the probabilistic traveling salesman problem*, European Journal of Operational Research, 65, 1993, 68-95.
- [2] Bianchi L., *Ant colony optimization and local search for the probabilistic travelling salesman problem: a case study in stochastic combinatorial optimization*, PhD thesis, Universite Libre de Bruxelles, Brussels 2006.
- [3] Branke J., Guntsch M., *Solving the Probabilistic TSP with Ant Colony Optimization*, Journal of Mathematical Modelling and Algorithms 3, 2004, 403-425.
- [4] Jaillet P., *Probabilistic Traveling Salesman Problems*, PhD thesis, MIT, Cambridge 1985.
- [5] Jaillet P., *A priori solution of a travelling salesman problem in which a random subset of the customers are visited*, Operations Research, 36(6), 1988, 929-936.
- [6] Kiełkiewicz K., *Hybrydowe algorytmy iteracyjnej optymalizacji kombinatorycznej*, praca dyplomowa, Politechnika Krakowska, Kraków 2009.
- [7] Kokosiński Z., Stuzienny Ł., *Hybrid Genetic Algorithms for the Open-Shop Scheduling Problem*, IJCSNS International Journal of Computer Science and Network Security, vol. 7, No. 9, 2007, 136-145.
- [8] Liu Y.-H., *A Scatter Search Based Approach with Approximate Evaluation for the Heterogeneous Probabilistic Traveling Salesman Problem*, Journal of Mathematical Modelling and Algorithms 3, 2004b, 403-425.

- [9] L i u Y.-H., A hybrid scatter search for the probabilistic traveling salesman problem, Elsevier Computers & Operations Research 34, 2007, 2949-2963.
- [10] L i u Y.-H., *Solving the Probabilistic Travelling Salesman Problem Based on Genetic Algorithm with Queen Selection Scheme*, September 2008, I-Tech, Vienna, Austria.
- [11] MPICH2, <http://www.mcs.anl.gov/research/projects/mpich2/>, (access 19.10.2012).
- [12] S u h J.Y., V a n G u c h t D., Incorporating Heuristic Information into Genetic Search, Proc. 2nd Int. Conf. on Genetic Algorithms, 1987, 100-107.
- [13] TSPLib, <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/tsp/> (access 27.10.2012).

