

**Grzegorz BIEŚ**KOMENDA POWIATOWA POLICJI W DĄBROWIE TARNOWSKIEJ  
Berka Joselewicza 4, 33-200 Dąbrowa Tarnowska**Komputerowy algorytm optymalizacji oparty na teorii „małych światów”**

Mgr inż. Grzegorz BIEŚ

Absolwent Wydziału Elektrotechniki i Informatyki Politechniki Rzeszowskiej im. Ignacego Łukasiewicza na kierunku Informatyka. Posiada międzynarodowe certyfikaty informatyczne MCSA, MCITP, MCTS, MCP. Aktualnie pracuje w Komendzie Powiatowej Policji w Dąbrowie Tarnowskiej. Interesuje się algorytmami ewolucyjnymi, optymalizacją oraz przestępczością komputerową.



e-mail: g.bies@poczta.fm

**Streszczenie**

W opracowaniu przedstawiono komputerowy algorytm optymalizacji o nazwie *Algorytm „Małe Światy”* (AMS), wykorzystujący mechanizm przeszukiwania przestrzeni dopuszczalnych rozwiązań problemu optymalizacyjnego opartego na fenomenie zjawiska „małych światów”. W algorytmie zaproponowano metody wyznaczania lokalnych i dalekich kontaktów dla modelu sieci o właściwościach „małego świata”, korzystając z teorii rachunku prawdopodobieństwa połączonego z pozycyjnym systemem liczbowym. Uzyskane wyniki testów wskazują, że algorytm skutecznie przeszukuje znaczne obszary dopuszczalnych wartości funkcji celu, znajdując rozwiązania optymalne. Właściwości te sugerują, że zaprezentowane rozwiązania mogą być przydatną metodą optymalizacyjną.

**Słowa kluczowe:** teoria „małych światów”, algorytm ewolucyjny, optymalizacja funkcji.

**Computer optimization algorithm based on "small-worlds" phenomenon****Abstract**

The paper presents a computer algorithm optimization, called *Algorithm „Small-Worlds”* (ASW), that uses mechanism of search the space of feasible solutions the optimization problem based on the "small-worlds" phenomenon. In the algorithm the determination of local and distant contacts for the "small-world" network model, using theory of probability combined with positional numeral system is used. In order to validate the algorithm with respect to the theory of J.M. Kleinberg, conducted test by finding the shortest path between any two points in two-dimensional networks of different sizes [Tab. 1]. Benchmarks of algorithm (ASW) as compared with genetic algorithm (GA) for finding an extremum functions and polynomial coefficients are given. The tests results indicate [Tab. 2] that algorithm provides effective mechanisms to search large areas of the limit values objective function and finds optimal solution. These properties suggest that presented solutions can be a useful method for solving optimization problems.

**Keywords:** "small-worlds" phenomenon, evolutionary algorithm, optimization functions.

**1. Wyjaśnienie oznaczeń i symboli**

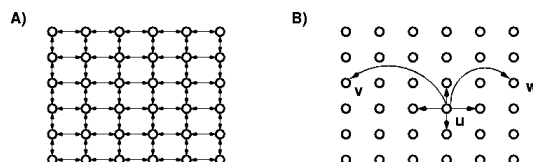
AMS - Algorytm „Małe Światy”

AG - Algorytm Genetyczny

 $R^m$  - przestrzeń wartości rzeczywistych $I$  - przestrzeń dopuszczalnych rozwiązań problemu $T$  - określony cel optymalizacji $f(x)$  - funkcja celu $f^o$  - optymalna (minimalna) wartość funkcji celu  $f(x)$  $m$  - liczba zmiennych decyzyjnych $x = \{x_1, x_2, \dots, x_m\}$  - wektor zmiennych decyzyjnych $d = \{d_1, d_2, \dots, d_m\}$  - dolny zakres wartości  $x$  $g = \{g_1, g_2, \dots, g_m\}$  - górny zakres wartości  $x$  $q(x)$  - funkcja kodująca  $x$  na liczbę binarną  $s$  typu *String* $q^{-1}(s)$  - funkcja dekodująca liczbę binarną  $s$  typu *String* na  $x$  $SubStr(s, a, b)$  - funkcja zwracająca fragment łańcucha  $s$  od pozycji  $a$  do  $b$  $Random[a, b]$  - funkcja losowa przyjmująca wartości z określonego przedziału $s_j$  - zakodowana wartość  $x_j$  ( $j$ -ty segment osobnika) $l_j$  - długość segmentu $s = s_1 + s_2 + \dots + s_m$  - osobnik (suma łańcuchów  $s_j$ ) $l$  - długość osobnika $S$  - zbiór osobników (populacja) $s[i]$  -  $i$ -ty osobnik populacji $op$  - liczba osobników populacji $s_{best}$  - najlepszy osobnik aktualnej iteracji algorytmu $f^{best}$  - wartość funkcji  $f(x)$  dla najlepszego osobnika aktualnej iteracji algorytmu $\varepsilon$  - dopuszczalny błąd $h$  - współczynnik odległości kontaktów $wsp$  - współczynnik rodzaju dalekiego kontaktu $max$  - maksymalna liczba iteracji algorytmu**2. Założenia teoretyczne algorytmu AMS**

Zasada działania prezentowanego algorytmu oparta jest na teorii „małych światów”, która zakłada istnienie powiązań pomiędzy dowolnymi osobnikami jakiegokolwiek dużej populacji, poprzez sieć pośrednich znajomych. Zjawisko istnienia takich powiązań zostało opisane przez amerykańskiego psychologa Stanleya Milgrama, który w 1967 roku przeprowadził eksperyment znany pod nazwą „Świat jest mały” [1]. Doświadczenie miało potwierdzić hipotezę, że członkowie nawet bardzo dużej społeczności mogą być pokrewni sobie dzięki istnieniu relacji międzyludzkich oraz posiadaniu wspólnych znajomych. Milgram posłużył się metodą wysyłania listów do kilkuset losowo wybranych osób w Kansas i Nebrasce, prosząc ich by przekazali list do jego przyjaciela. Gdyby go jednak nie znali, to musieli przekazać list komuś, kto według ich wiedzy mógł go znać. Okazało się, że większość listów Milgrama, która dotarła do jego przyjaciela, została przekazana średnio sześciu kolejnym osobom.

Hipoteza o „małym świecie” była wielokrotnie badana i jak się okazuje jest cechą wielu sieci powstałych w przyrodzie. Wskazuje po pierwsze, że „krótkie ścieżki” są wszechobecne, a po drugie posiadając jedynie informacje na temat lokalnej struktury sieci, można te ścieżki odnaleźć. Okazuje się, że informacje niezbędne do odkrywania tego typu połączeń pojawiają się w bardzo prostym modelu sieci, zaproponowanym przez Jon M. Kleinberga (rys. 1) [2]. Osoby w sieci społecznej reprezentowane są przez węzły siatki o wymiarach  $n \times n$ ,  $\{(i, j) : i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, n\}\}$ . Odległość pomiędzy dwoma dowolnymi węzłami siatki  $(i, j)$  i  $(k, l)$  została określona jako  $d((i, j), (k, l)) = |k - i| + |l - j|$ .



Rys. 1. Model sieci Jon M. Kleinberga. a) siatka dwuwymiarowa o  $n = 6, p = 1, q = 0$ , b) kontakty węzła  $U, p = 1$  (lokalne) i  $q = 2$  (dalekie)

Fig. 1. The network model Jon M. Kleinberg. a) two-dimensional grid with  $n = 6, p = 1, q = 0$ , b) the contacts a node  $U, p = 1$  (local) and  $q = 2$  (long)

Stała  $p > 0$  reprezentuje odległość lokalnych kontaktów węzła  $U$ , natomiast stała  $q \geq 0$  liczbę połączeń do dalekich kontaktów. Strukturę sieci określa współczynnik  $r \geq 0$  kształtujący prawdopodobieństwo wyboru dalekiego kontaktu  $V$  węzła  $U$  w zależności od odległości pomiędzy węzłami, określone jako  $[d(U, V)]^{-r}$ . Za-

prezentowany model ma prostą interpretację: osoby znają swoich sąsiadów, którzy mieszkają obok nich oraz posiadają pewną liczbę znajomych, którzy mieszkają w pewnej odległości [3]. Celem jest znalezienie „krótkiej ścieżki”, określonej liczbą wykonanych kroków pomiędzy dwoma dowolnie wybranymi węzłami. Można to porównać do przekazywania wiadomości, w każdym kroku węzeł posiadający wiadomość musi ją przekazać przez jednego z lokalnych kontaktów lub poprzez daleki kontakt [2]. Najważniejsze jest to, że aktualny posiadacz wiadomości nie ma informacji o tym, który z poprzednich węzłów już ją przekazał. Teoretyczne założenia algorytmu przedstawił Jon M. Kleinberg w swojej pracy *The Small-World Phenomenon: An Algorithmic Perspective* [3].

Według Jona M. Kleinberga (2000):

Niezależnie od  $n$  oraz  $p = q = 1$ , gdy:

- $0 \leq r < 2$ , przewidywana liczba kroków dostarczenia wiadomości wynosi co najmniej  $\beta_1 n^{(2-r)/3}$ , dla określonej stałej  $\beta_1$ ,
- $r = 2$ , istnieje zdecentralizowany algorytm którego przewidywana liczba kroków dostarczenia wiadomości wynosi co najwyżej  $\beta_2 (\log_2(n))^2$ , dla określonej stałej  $\beta_2$ ,
- $r > 2$ , przewidywana liczba kroków dostarczenia wiadomości wynosi co najmniej  $\beta_3 n^{(r-2)/(r-1)}$ , dla określonej stałej  $\beta_3$ .

Można zauważyć, że dla  $r \neq 2$  liczba kroków potrzebnych do dostarczenia wiadomości jest funkcją wykładniczą i rośnie bardzo szybko wraz ze wzrostem rozmiaru sieci, określonej parametrem  $n$ . Jedynie dla wartości  $r=2$  istnieje zdecentralizowany algorytm, zdolny dostarczyć wiadomość w co najwyżej  $\log_2(n)^2$  krokach. Pomimo tego, że powyższe rozważania dotyczą sieci dwuwymiarowej, twierdzenie może być stosowane w szerszym zakresie. Jak podaje Jon M. Kleinberg w swojej pracy [3] możemy uogólnić dowód istnienia zdecentralizowanego algorytmu do  $k$ -wymiarowej sieci. W tym przypadku współczynnik  $r$  kształtujący prawdopodobieństwo wyboru dalekiego kontaktu musi być równy liczbie wymiarów  $k$ . Jedynie dla takiej wartości istnieją obszary znormalizowanego rozkładu prawdopodobieństwa wyboru dalekich kontaktów o wielkości proporcjonalnej do funkcji logarytmicznej, w zależności od odległości węzła źródłowego  $U$ . Zatem odnalezienie „krótkiej ścieżki” pomiędzy dwoma dowolnie wybranymi węzłami staje się możliwe, gdy średnia odległość generowana dla dalekich kontaktów wzrasta logarytmicznie wraz ze wzrostem rozmiaru sieci, określonej parametrem  $n$ .

Zjawiska zachodzące w sieciach o właściwościach „małego świata” wykazują jedną z najbardziej skutecznych metod przekazywania wiadomości w złożonych strukturach sieci. Można zatem postawić tezę, iż na podstawie przedstawionej teorii możliwe jest stworzenie komputerowego algorytmu przeszukującego przestrzeń alternatywnych rozwiązań problemu w celu wyszukania rozwiązań najlepszych, a tym samym opracowanie nowej metody optymalizacji.

### 3. Algorytm „Małe Światy” (AMS)

W rozdziale zaprezentowano zasadę działania algorytmu optymalizacji, który jak pokazują przeprowadzone testy skutecznie wyszukuje optymalne rozwiązania problemu, w obrębie przestrzeni dopuszczalnych wartości. Inspiracją do stworzenia algorytmu była publikacja „*An Optimization Algorithm Based on the Small-World Phenomenon*” [4] przedstawiająca propozycję dwóch algorytmów opartych na teorii Jon M. Kleinberga, binarnego BSWA (*Binary-coding small-world algorithm*) i dziesiętnego DSWA (*Decimal-coding small-world algorithm*). Autorzy wspomnianej publikacji stworzyli jednak algorytmy, które raczej swoim działaniem są bardziej zbliżone do teorii „small-world” prezentowanej przez Watts’a i Strogatz’a [5], niż do teorii Jon M. Kleinberga [3]. Powyższą tezę potwierdza test algorytmu BSWA w porównaniu z algorytmem AMS, którego wyniki zostały zamieszczone w tabeli 1 opracowania i opisane w punkcie 4.

Założeniem algorytmu AMS jest znalezienie optymalnego rozwiązania  $m$ -wymiarowego problemu w jak najmniejszej liczbie kroków poprzez [4]:

minimalizację  $f(x)$   
dla  $x$  takiego, że  $d \leq x \leq g$ .

gdzie:  $x = \{x_1, x_2, \dots, x_m\} \in R^m$ ,  $d = \{d_1, d_2, \dots, d_m\}$  i  $g = \{g_1, g_2, \dots, g_m\}$  są zakresami dolnym oraz górnym dla  $x$ .

Każdy z punktów w przestrzeni  $I$  dopuszczalnych rozwiązań problemu zdefiniowano jako łańcuch binarny  $s = s_1 + s_2 + \dots + s_m$  typu *String* (osobnik), który składa się z  $m$  segmentów  $s_j$  i reprezentuje zakodowany wektor wartości rzeczywistych  $x$  taki, że  $s = q(x)$ , natomiast  $x$  jest dekodowane przez funkcję  $x = q^{-1}(s)$ . Funkcja, która wyznacza poszczególne zmienne  $x_j$  na podstawie łańcuchów  $s_j$ , oblicza wartości zmiennych według poniższego wzoru:

$$x_j = d_j + \frac{\sum_{k=1}^{l_j} s_{jk} \cdot 2^{l_j-k}}{2^{l_j} - 1} \cdot |g_j - d_j|, \quad (1)$$

gdzie:  $s_{jk}$  - bit na pozycji  $k$  segmentu  $s_j$ ,  $l_j$  - długość segmentu,  $d_j, g_j$  - zakres dolny i górny zmiennej  $x_j$ .

Na podstawie przyjętych założeń zdefiniowano cel optymalizacji problemu [4]:

$$T = \left\{ s \in I : f(q^{-1}(s)) = f^0 \equiv \min(f(x) : d \leq x \leq g) \right\} \quad (2)$$

Algorytm AMS wyszukuje optymalne rozwiązania problemu poprzez generowanie lokalnych i dalekich kontaktów dla poszczególnych segmentów osobnika, a tym samym tworzenie nowych osobników, reprezentujących zakodowane wartości zmiennych decyzyjnych. W algorytmie zastosowano pozycyjny system kodowania [6] oraz elementy probabilistyki do wyznaczania odpowiednich odległości dla poszczególnych kontaktów. Przyjęto, że baza systemu jest równa 2, natomiast wykładnik określa tzw. współczynnik odległości  $h$  generowany losowo. Wyznaczanie odległości  $z_j$  dla lokalnych lub dalekich kontaktów segmentu  $s_j$  realizuje funkcja *przesunięcie*( $l_j, h, t$ ) prezentowanego algorytmu. Działanie funkcji jest następujące: generowana jest losowa wartość współczynnika  $1 \leq h \leq l_j$ , następnie na pozycji  $(l_j - h) + 1$  zerowego łańcucha binarnego o długości  $l_j$  ustawiana jest wartość „1”. Dla  $h > 1$  oraz  $t = 1$  na pozycjach  $(l_j - h) + 2$  do  $l_j$  generowane są losowe wartości  $\{0, 1\}$  (rys. 2a), natomiast dla  $t = 0$  pozycje łańcucha pozostają „0” (rys. 2b). Wybór sposobu a) lub b) uzależniony jest od współczynnika rodzaju dalekiego kontaktu *wsp*. Jak łatwo zauważyć lokalne odległości wyznaczane są dla współczynnika  $h = 1$ .

	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
a)	0	0	0	0	0	1	1	0	1	0
	Wypełnienie „0”					Losowe wartości {0,1}				
b)	0	0	0	0	0	1	0	0	0	0
	Wypełnienie „0”					Wypełnienie „0”				

Rys. 2. Sposoby generowania dalekich odległości dla  $h=5$ , a)  $t=1$ , b)  $t=0$   
Fig. 2. Ways of generating of long distance for  $h=5$ , a)  $t=1$ , b)  $t=0$

Po wyznaczeniu odległości  $z_j$  dla kontaktu danego segmentu  $s_j$ , za pomocą funkcji *operacja*( $t, s_j, z_j$ ) wykonywana jest losowa operacja dodawania ( $t=0$ ) lub odejmowania ( $t=1$ ) w systemie binarnym, celem określenia pozycji kontaktu w sieci. W kolejnym kroku algorytm sprawdza „jakość” kontaktu poprzez umieszczenie jego kodu w osobniku i obliczenie wartości funkcji celu  $f(x)$  w odniesieniu do całego osobnika. W przypadku, gdy wartość funkcji jest „bliżej rozwiązania” problemu, wygenerowany kontakt staje się aktualnym segmentem osobnika. Następnie algorytm przechodzi do wyznaczania kontaktu dla kolejnego segmentu i operacja jest powtarzana, aż do ostatniego segmentu, a tym samym utworzenia nowego osobnika. Algorytm może również generować kontakty w odniesieniu do całego osobnika (równoległe dla wszystkich segmentów), jednak w prezentowanych problemach optymalizacyjnych lepsze wyniki uzyskano stosując podział segmentowy. Celem zwiększenia skuteczności algorytmu do odkrywania najlepszych rozwiązań problemu, wykorzystano

zbiór kandydatów  $S=\{s[1],s[2],\dots,s[op]\}$ , a nie pojedynczego osobnika  $s$ , podobnie jak Milgram używał wiele ścieżek jednoczesnego przesyłania listu do celu [4].

Sprawdzenie skuteczności algorytmu AMS oraz możliwość porównania z algorytmem genetycznym AG określono poprzez wprowadzenie kryterium zakończenia działania jako:

$$\left|f^0 - f^{best}\right| \leq \varepsilon \quad (3)$$

W celu przetestowania działania przygotowywano aplikację testową wykorzystując środowisko programistyczne NetBeans IDE 6.7.1 wraz z biblioteką GALib, zawierającą podstawowe klasy algorytmu genetycznego. Poniżej zaprezentowano pseudokod najważniejszych modułów aplikacji, prezentujący zasadę działania algorytmu.

#### a) Funkcja generująca przesunięcia (odległości dla kontaktów)

```
przesuniecie(lj, h, t)
begin
  z:="1";
  for k:=1 to (h-1) do
    begin
      if (t=0) then z:=z+"0"
      else z:=z+Random["0", "1"];
    end;
  for k:=1 to (lj-h) do z:="0"+z;
  przesuniecie:=z;
end;
```

#### b) Funkcja generująca nowego osobnika

```
nowy_osobnik(osobnik)
begin
  j:=0;
  while (j<m) do
    begin
      segment:=SubStr(osobnik, (j*lj)+1,
        (j+1)*lj);
      h:=Random[1, lj];
      if (Random[0, 9]<wsp) then
        segment:=operacja(Random[0, 1],
          segment, przesuniecie(lj, h, 0))
      else
        segment:=operacja(Random[0, 1],
          segment, przesuniecie(lj, h, 1));
      nowy:=SubStr(osobnik, 1, j*lj)+segment+
        SubStr(osobnik, (j+1)*lj+1, l);
      if (f(q-1(nowy))<f(q-1(osobnik))) then
        osobnik:=nowy;
      j++;
    end;
  nowy_osobnik:=osobnik;
end;
```

#### c) Główny moduł aplikacji

```
Generowanie populacji początkowej
S = {s[1], s[2], ..., s[op]}
atr:=0;
blad:=ε*103;
while (atr<max) and (blad>ε) do
begin
  atr++;i:=0;
  repeat
    i++;
    s[i]:=nowy_osobnik(s[i]);
    if (Abs(f(q-1(s[i]))-fo)<blad) then
      begin
        sbest:=s[i]; fbest:=f(sbest);
        blad:=Abs(fbest-fo);
      end;
  until (i=op) or (blad<=ε);
end;
```

## 4. Uzyskane wyniki testów

Obliczenia wykonane zostały na komputerze klasy PC o parametrach: procesor Pentium D 3.0 GHz, pamięć RAM 2048 MB, karta graficzna GeForce 8600, system operacyjny Microsoft Windows XP Professional. Długość osobnika dla poszczególnych testów wyznaczano na podstawie liczby segmentów, maksymalnego zakresu wartości zmiennych decyzyjnych oraz założonej dokładności.

#### a) Porównanie algorytmów BSWA i AMS

Test przeprowadzono przy następujących wartościach parametrów algorytmów: liczba wymiarów – 2, liczba prób – 100, liczba osobników – 20, maksymalna liczba generacji – 3000, dla algorytmu BSWA: odległość lokalna – 1, liczba lokalnie przeszukiwanych osobników – 3, dla algorytmu AMS: współczynnik rodzaju dalekiego kontaktu – 1.

Przeprowadzony test polegał na minimalizacji funkcji  $f(x_1, x_2)=|x_1-a_1|+|x_2-a_2|$  dla liczb całkowitych:  $x_1, x_2 \in [0, 10^k]$  oraz losowo wybranych  $a_1, a_2 \in [0, 10^k]$ , gdzie  $k=\{2, 3, \dots, 8\}$ . Funkcja posiada minimum równe 0 w punkcie  $(a_1, a_2)$ . Rodzaj testu odzwierciedla zadanie dostarczania wiadomości od źródła do celu w sieciach dwuwymiarowych o różnej wielkości, a tym samym umożliwia zweryfikowanie działania algorytmów w odniesieniu do teorii Kleinberga. Wyniki uzyskane podczas testu zamieszczono w tabeli 1.

#### b) Testy porównawcze algorytmów AMS i AG dla problemów m-wymiarowych

Testy przeprowadzono przy następujących wartościach parametrów algorytmów: liczba wymiarów  $m$  – 2-5, liczba prób – 50, maksymalna liczba generacji – 1500, dla algorytmu AMS: liczba osobników – 20, współczynnik rodzaju dalekiego kontaktu – 1, dla algorytmu genetycznego AG: rozmiar populacji – 100, współczynnik krzyżowania – 0.7, współczynnik mutacji – 0.02. Wyniki uzyskane w przeprowadzonych testach zamieszczono w tabeli 2.

W pierwszym teście algorytmy wykorzystano do minimalizacji funkcji postaci:

$$f_1(x_i) = \sum_{i=1}^m |x_i - a_i|, \quad (4)$$

dla liczb całkowitych:  $x_i \in [0, 100]$  oraz losowo wybranych  $a_i \in [0, 100]$ . Funkcja posiada minimum równe 0 w punkcie  $(a_1, a_2, \dots, a_m)$ . Test odzwierciedla zadanie dostarczania wiadomości od źródła do celu w sieci wielowymiarowej.

Kolejnym testem było znalezienie minimum funkcji Rastrigina, która jest typowym przykładem liniowo niezależnej funkcji wielomodalnej i podstawową funkcją testową dla algorytmów ewolucyjnych. Posiada minimum globalne znajdujące się w punkcie  $(0, 0, \dots, 0)$  w którym przyjmuje wartość 0.0. Dla  $m$ -wymiarów funkcja Rastrigina ma następującą postać:

$$f_2(x_i) = A \cdot m + \sum_{i=1}^m x_i^2 - A \cdot \cos(\omega \cdot x_i), \quad (5)$$

dla  $A=10$ ,  $\omega=2\pi$ ,  $x_i \in [-5.12, 5.12]$ .

Trzeci test polegał na znalezieniu minimum funkcji Griewanka, która jest funkcją trudną do optymalizacji z powodu silnej zależności wewnętrznej pomiędzy zmiennymi. Posiada minimum globalne w punkcie  $(0, 0, \dots, 0)$ , gdzie przyjmuje wartość 0.0. Funkcję Griewanka definiuje się następującym wzorem:

$$f_3(x_i) = \sum_{i=1}^m \frac{x_i^2}{4000} - \prod_{i=1}^m \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad (6)$$

dla  $x_i \in [-40, 40]$ .

Ostatnim z przeprowadzonych testów była minimalizacja funkcji postaci:

$$f_4(x_i) = \sum_{j=1}^{50} \left| \sum_{i=1}^m x_i \cdot y_j^{i-1} - \sum_{i=1}^m a_i \cdot y_j^{i-1} \right|, \quad (7)$$

dla  $x_i \in [-10, 10]$ .

W celu przeprowadzenia testu losowo wyznaczono wartości rzeczywiste współczynników  $a_i \in [-10, 10]$ , a następnie przygotowano tabelę zawierającą 50 losowo wygenerowanych wartości zmiennej  $y_j \in [-10, 10]$ . W tabeli 2 dla funkcji  $f_4$  zamieszczono średnią wartość błędu, określającą z jaką dokładnością zostały wyznaczone optymalne wartości  $x_i$ .

### c) Tabele wyników

Tab. 1. Wynik testu porównawczego algorytmów AMS i BSWA  
Tab. 1. The benchmark result of algorithms AMS and BSWA

Rozmiar $n$	$10^2$	$10^3$	$10^4$	$10^5$	$10^6$	$10^7$	$10^8$
<b>AMS</b>							
Maksymalna liczba kroków	14	30	63	106	131	196	255
Minimalna liczba kroków	1	6	14	28	63	92	124
Średnia liczba kroków	6,5	18,4	42,7	68	100,9	153,2	191,6
<b>BSWA</b>							
Maksymalna liczba kroków	55	571	3000	3000	3000	3000	3000
Minimalna liczba kroków	1	4	11	23	44	88	144
Średnia liczba kroków	6,7	34,3	309,9	1446,7	2174,2	2741,6	2894,4
$\log(n)^2$	44,1	99,3	176,5	275,8	397,2	540,7	706,2
Błąd $\varepsilon$	1,00						

Tab. 2. Testy porównawcze algorytmów AMS i AG  
Tab. 2. Benchmarks of algorithms AMS and AG

Algorytm	AMS				AG			
	2	3	4	5	2	3	4	5
<b>Funkcja <math>f_1</math></b>								
Maksymalna liczba kroków	17	25	27	29	1500	1500	1500	1500
Minimalna liczba kroków	1	6	9	13	2	9	14	43
Średnia	9,4	14,6	17,7	20,9	88,2	668,4	1046	1233
% rozwiązań	100	100	100	100	96	56	36	24
Błąd $\varepsilon$	0,00							
Zakres; dokładność	<0;100>;1,0							
<b>Funkcja <math>f_2</math></b>								
Maksymalna liczba kroków	59	108	137	143	1500	1500	1500	1500
Minimalna liczba kroków	8	28	42	69	16	48	86	112
Średnia	40,4	58,6	86,4	103,1	435,5	1109	1234	1453
% rozwiązań	100	100	100	100	74	24	20	4
Błąd $\varepsilon$	0,001							
Zakres; dokładność	<-5,12;5,12>;0,001							
<b>Funkcja <math>f_3</math></b>								
Maksymalna liczba kroków	371	661	1500	1500	1500	1500	1500	1500
Minimalna liczba kroków	14	28	58	133	15	57	153	1500

Średnia	94,1	269,3	537,3	790,8	1205	1446	1473	1500
% rozwiązań	100	100	90	74	20	4	2	0
Błąd $\varepsilon$	0,001							
Zakres; dokładność	<-40,0;40,0>; 0,001							
<b>Funkcja <math>f_4</math></b>								
Maksymalna liczba kroków	64	161	776	1500	1500	1500	1500	1500
Minimalna liczba kroków	18	64	225	298	14	1500	1500	1500
Średnia	44,2	114,4	595,9	1327	177,3	1500	1500	1500
% rozwiązań	100	100	100	24	30	0	0	0
Średni błąd	0,6	0,8	4,8	85,3	6,2	35,5	577,7	3377
Błąd $\varepsilon$	1,0	1,0	5,0	20,0	1,0	1,0	5,0	20,0
Zakres; dokładność	<-10,0;10,0>; 0,001							

## 5. Podsumowanie

Wstępnie analizując uzyskane wyniki można stwierdzić, że algorytm AMS w porównaniu z algorytmem BSWA wykazał się dużo lepszą skutecznością odnalezienia dwóch losowo wybranych liczb z określonego przedziału, a tym samym zgodnie z teorią Kleinberga dostarczania wiadomości od źródła do celu dla sieci dwuwymiarowej. Pomimo tego, że algorytm BSWA w początkowej fazie testu bardzo szybko odnajdywał optymalne rozwiązania, jednak po przekroczeniu rozmiaru sieci  $n > 10^3$  nie potrafił w niektórych próbach uzyskać satysfakcjonującego wyniku. Zgodnie z teorią Kleinberga taka sytuacja nie powinna mieć miejsca, gdyż z matematycznego dowodu wynika, że zdecentralizowany algorytm w sieci dwuwymiarowej powinien znaleźć „krótką ścieżkę” w co najwyżej  $\log_2(n)^2$  krokach. Porównując efekty przeprowadzonych testów algorytmów AMS i AG można zauważyć, że algorytm AMS uzyskał lepsze wyniki w prezentowanych problemach optymalizacyjnych, co może wskazywać na skuteczność jego działania przy rozwiązywaniu tego typu zadań. W szczególności należy zwrócić uwagę na stosunkowo niewielką liczbę kroków potrzebnych do znalezienia „krótkiej ścieżki” w sieciach o niekiedy bardzo dużym rozmiarze. Jednak właściwa ocena przydatności algorytmu i ewentualnego praktycznego wykorzystania może nastąpić po przeprowadzeniu dalszych testów oraz symulacji ukierunkowanych na inne znane problemy, w tym uwzględniające rzeczywisty czas wykonywanych operacji.

Na uwagę zasługuje jednak fakt, że wykorzystanie elementów probabilistyki w połączeniu z pozycyjnym systemem liczbowym pozwoliło stworzyć podstawy algorytmu opartego na teorii „małych światów”, który w sposób skuteczny przeszukuje znaczne obszary dopuszczalnych wartości funkcji celu, znajdując rozwiązania optymalne.

## 6. Literatura

- [1] Milgram S.: The Small World Problem. Psychology Today, 1967, pp 60–67.
- [2] Kleinberg J.: Navigation in a small world. Nature, 406: 845, 2000.
- [3] Kleinberg J.: The Small-World Phenomenon: An Algorithmic Perspective. Cornell Computer Science Technical Report, 1999.99-1776.
- [4] Haifeng Du, Jie Shao, Feldman Marcus W.: An Optimization Algorithm Based on the Small-World Phenomenon. 2008.
- [5] Watts, D. and Strogatz S.H.: Collective dynamics of 'small-world' networks. Nature. 393(4):440-42, 1998.
- [6] [http://pl.wikipedia.org/wiki/System\\_liczbowy](http://pl.wikipedia.org/wiki/System_liczbowy)

otrzymano / received: 25.03.2011

przyjęto do druku / accepted: 06.06.2011

artykuł recenzowany