

Dr inż. Joanna Kwiecień

AGH Akademia Górniczo-Hutnicza
Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej
Al. Mickiewicza 30, 30-059 Kraków, Polska
E-mail: kwiecien@agh.edu.pl

Prof. dr hab. inż. Bogusław Filipowicz

Państwowa Wyższa Szkoła Zawodowa w Tarnowie
Instytut Politechniczny
Ul. Mickiewicza 8, 33-100 Tarnów, Polska
E-mail: fil@agh.edu.pl

Optymalizacja niezawodności złożonych systemów za pomocą algorytmu świetlika

Streszczenie. Algorytmy bazujące na inteligencji stadnej są coraz częściej stosowane w problemach niezawodności systemów. Artykuł prezentuje zastosowanie algorytmu świetlika do optymalizacji niezawodności dwóch systemów: mostkowego i 10-elementowego, z wykorzystaniem metod zbioru minimalnych ścieżek, minimalnych cięć oraz metody dekompozycji. Uzyskane rezultaty zostały przedstawione i porównane z dostępnymi danymi literaturowymi.

Słowa kluczowe: *problemy optymalizacji niezawodności systemów, metody optymalizacji niezawodności, system RRAP, algorytm świetlika*

1. Wprowadzenie

Problem badania niezawodności grupy urządzeń wpływających na jakość pracy obiektu technicznego można rozpatrywać jako analizę systemu tj. celowo wyodrębnionej zbiorowości podsystemów powiązanych zależnościami lub oddziaływaniami. Model systemu można przedstawić jako parę uporządkowaną $\langle N, f \rangle$, gdzie N oznacza zbiór liczb naturalnych przyporządkowanych danym elementom, a f to funkcja nosząca nazwę struktury systemu, która określa stan systemu zależnie od stanu jego elementów. Jeśli obiekt jest dwustanowy $\{\text{zdatny, niezdatny}\}$ to funkcja ta przyjmuje wartości binarne, przy czym „1” przyporządkowujemy stanowi zdatności, a wartość „0” - stanowi niezdatności. W celu uzyskania wymaganej niezawodności całego systemu należy zapewnić odpowiednie wartości niezawodności poszczególnych jego komponentów. Należy przy tym uwzględnić pewne narzucone ograniczenia, takie jak łączny koszt komponentów wchodzących w skład urządzenia, ich sumaryczną masę i objętość.

Istnieje wiele różnych podejść, które pozwalają rozwiązać problem optymalizacji niezawodności złożonych systemów. Wiele prac dotyczy zastosowania algorytmów należących do grupy algorytmów stadnych tj. bazujących na zachowaniach owadów społecznych czy stad zwierząt. Analizując dotychczasową literaturę, można zauważyć, że głównie dotyczy ona badania efektywności algorytmów mrówkowych [1], optymalizacji

rojem cząstek [4, 10, 15], pszczelich [18] i kukułki [5, 6, 13, 14], ze wskazaniem jednocześnie przewagi algorytmu kukułki nad innymi algorytmami stadnymi. W niniejszej pracy skupiono się na zbadaniu przydatności algorytmu świetlika do systemów złożonych z 5 i 10 elementów, uwzględniając kilka metod wyznaczania niezawodności systemów.

2. Niezawodność systemów złożonych

Podczas projektowania wysoce niezawodnego systemu bardzo ważne jest uzyskanie równowagi między niezawodnością, a innymi zasobami, takimi jak: cena, objętość czy waga. Problem optymalizacji niezawodności z uwzględnieniem redundancji (*RRAP*, ang. *reliability redundancy allocation problem*) jest traktowany jako problem programowania nieliniowego z jednym lub wieloma ograniczeniami zasobów. Spośród znanych systemów tego typu w pracy rozpatrzono dwa systemy: system mostkowy oraz system złożony z 10 elementów.

2.1. System mostkowy

System mostkowy przedstawiony na rysunku 1 można sformułować następująco [12, 14]:

$$\begin{aligned} \text{Max } f(r, n) = & R_1 R_2 + R_3 R_4 + R_1 R_4 R_5 + R_2 R_3 R_5 - R_1 R_2 R_3 R_4 - R_1 R_2 R_3 R_5 + \\ & - R_1 R_2 R_4 R_5 - R_1 R_3 R_4 R_5 - R_2 R_3 R_4 R_5 + 2R_1 R_2 R_3 R_4 R_5 \end{aligned} \quad (1)$$

z ograniczeniami uwzględniającymi górny limit łącznej objętości i wagi (V), kosztu (C) i wagi systemu (W):

$$g_1(r, n) = \sum_{i=1}^m w_i v_i^2 n_i^2 - V \leq 0$$

$$g_2(r, n) = \sum_{i=1}^m \alpha_i \left(-\frac{1000}{\ln(r_i)} \right)^{\beta_i} [n_i + e^{0.25n_i}] - C \leq 0$$

$$g_3(r, n) = \sum_{i=1}^m w_i n_i e^{0.25n_i} - W \leq 0$$

$$0 \leq i \leq m, \quad 0 \leq r_i \leq 1, \quad n_i \in \mathbb{Z}^+$$

gdzie:

m – liczba podsystemów w całym systemie,

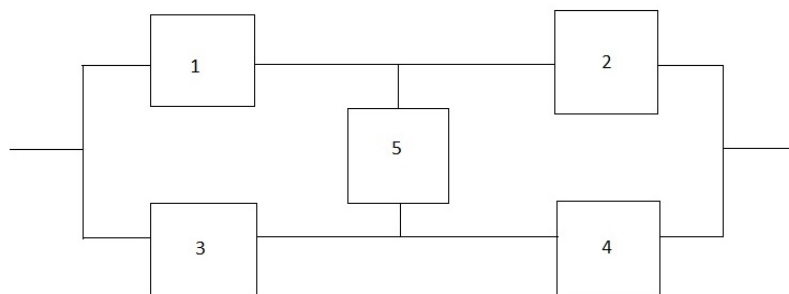
n_i – liczba elementów w podsystemie i ,

r_i – niezawodność każdego elementu w podsystemie i ,

R_i – niezawodność podsystemu i ,

α_i, β_i – fizyczne cechy elementów,

w_i, v_i, c_i – waga, objętość, koszt elementu w podsystemie i .



Rys.1. Schemat systemu mostkowego

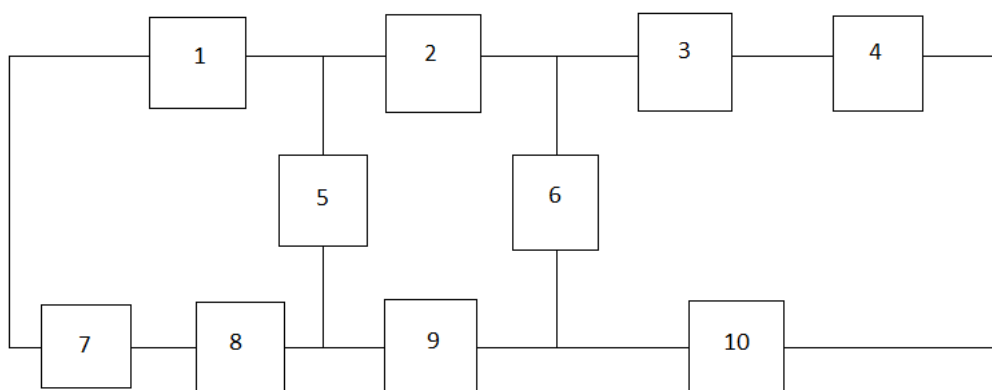
W literaturze można znaleźć wartości ustawień parametrów systemu mostkowego. Korzystając z pracy [14] wybrane zostały następujące wartości przedstawione w tabeli 1.

Tabela 1. Dane wykorzystane w systemie mostkowym

i	$10^5 a_i$	β_i	$w_i v_i^2$	w_i	V	C	W
1	2.330	1.5	1	7	110	175	200
2	1.450	1.5	2	8			
3	0.541	1.5	3	8			
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

2.2. System złożony z 10 elementów

Struktura niezawodnościowa systemu 10-cio elementowego jest przedstawiona na rysunku 2 [14].



Rys.2. Diagram systemu 10-cio elementowego

Zakładając, przez $R_i(x_i)$ niezawodność podsystemu i równą $1 - (1 - r_i)^{x_i}$ oraz $Q_i = 1 - R_i$, problem ten można sformułować następująco [1]:

$$\begin{aligned}
 \text{Max } f(x) = & R_1 R_2 R_3 R_4 + R_1 R_2 R_6 R_{10} (Q_3 + R_3 Q_4) + R_1 R_5 R_9 R_{10} (Q_2 + R_2 Q_3 Q_6 + \\
 & + R_2 R_3 Q_4 Q_6) + R_7 R_8 R_9 R_{10} (Q_1 + R_1 Q_2 Q_5 + R_1 R_2 Q_3 Q_5 Q_6 + R_1 R_3 Q_4 Q_5 Q_6) + \\
 & + R_2 R_3 R_4 R_5 R_7 R_8 Q_1 (Q_9 + R_9 Q_{10}) + Q_1 R_3 R_4 R_6 R_7 R_8 R_9 Q_{10} (Q_2 + R_2 Q_5) + \\
 & + R_1 Q_2 R_3 R_4 R_6 R_7 R_8 R_9 Q_{10} + R_1 Q_2 R_3 R_4 R_5 R_6 R_9 Q_{10} (Q_7 + R_7 Q_8) + \\
 & + Q_1 R_2 R_5 R_6 R_7 R_8 Q_9 R_{10} (Q_3 + R_3 Q_4)
 \end{aligned} \quad (2)$$

przy m ograniczeniach:

$$g_y(r, n) = \sum_{i=1}^{10} c_{yi} x_i \leq b_y, \quad y = 1, 2, \dots, m, \quad x_i \in Z^+$$

Tabela 2. Dane wykorzystane w systemie 10-cio elementowym

i	r	c_1	c_2	c_3	c_4	c_5
1	0.6796	33.2468	35.6054	13.7848	44.1345	10.9891
2	0.7329	27.5668	44.9520	96.7365	25.9855	68.0713
3	0.6688	13.3800	28.6889	85.8783	19.2621	1.0164
4	0.6102	0.4710	0.4922	63.0815	12.1687	29.4809
5	0.7911	51.2555	39.6833	78.5364	23.9668	59.5441
6	0.8140	82.9415	59.2294	11.8123	28.9889	46.5904
7	0.8088	51.8804	78.4996	97.1872	47.8387	49.6226
8	0.7142	77.9446	86.6633	45.0850	25.0545	59.2594
9	0.8487	26.8835	7.8195	3.6722	76.9923	87.4070
10	0.7901	85.8722	27.7460	55.3950	53.3007	55.3175

Dla tego systemu współczynniki c_{yi} to liczby losowe z przedziału $[0, 100]$, r_i generowane są z przedziału $[0.65, 0.85]$, natomiast parametr $b_y = rand(1.5, 3.5) \cdot \sum_{i=1}^{10} c_{yi}$. Wartości ustawień parametrów tego modelu zestawiono w tabeli 2, na podstawie danych dostępnych w literaturze [14].

3. Metody wyznaczania niezawodności systemów

W systemach określonych jako mających nadmiarową strukturę niezawodnościową, przypadek niespełnienia zgodności niektórych cech systemu z określonymi wymaganiami nie powoduje jego niezdatności. Można wyróżnić dwa minimalne podzbiory elementów, dzięki którym można oszacować od dołu i od góry niezawodność systemu [2, 3, 8, 11]:

- minimalna ścieżka - zbiór elementów systemu, które poprawnie funkcjonując (wszystkie) zapewniają poprawną pracę całego systemu, jednak uszkodzenie choćby jednego z tych elementów spowoduje, że system jako całość nie będzie w stanie zdatności; elementy w minimalnej ścieżce są połączone szeregowo a rzeczywista struktura niezawodnościowa systemu może być odwzorowana strukturą równoległo-szeregową, w której minimalne ścieżki są połączone równoległe [3]; oznaczając przez P_1, \dots, P_r zbiór minimalnych ścieżek systemu, funkcja strukturalna systemu dana jest w postaci [2]:

$$f(x) = 1 - \prod_{j \in \{1, \dots, r\}} \left(1 - \prod_{i \in P_j} R_i \right) \quad (3)$$

- minimalny przekrój (cięcie) - stanowi zbiór elementów, które będąc w stanie niezdatności powodują, że system jest w stanie niezdatności, jednak uszkodzenie dowolnego podzbioru elementów tego zbioru nie powoduje uszkodzenia systemu; elementy w minimalnym przekroju są połączone równoległe a rzeczywista struktura niezawodnościowa systemu może być odwzorowana ekwiwalentną strukturą szeregowo-równoległą, w której minimalne przekroje są połączone szeregowo [3]. Jeśli C_1, \dots, C_s oznaczają zbiór minimalnych cięć to otrzymujemy:

$$f(x) = \prod_{j \in \{1, \dots, s\}} \left(1 - \prod_{i \in C_j} (1 - R_i) \right) \quad (4)$$

Jedną z metod wyznaczania niezawodności systemów złożonych jest tzw. metoda dekompozycji polegającą na wykonaniu kolejnych operacji strukturalnych przekształcających obiekt n -elementowy o dowolnej strukturze w pewną liczbę obiektów prostych o szeregowo – równoległych strukturach, dla których niezawodność można wyznaczyć znanymi metodami [3, 11]. W każdej operacji taką dekompozycję wykonujemy zawsze względem jednego wybranego elementu i o niezawodności R_i . Rozpatrywane są dwie struktury składające się z $n-1$ elementów. W jednej z nich wybrany element zostaje zastąpiony elementem absolutnie niezawodnym – „zwarciem” ($R_i = 1$), a w drugiej „przerwą” ($R_i = 0$) tj. elementem całkowicie niesprawnym. Niezawodność całego systemu n -elementowego $R^{(n)}$ może zostać obliczona za pomocą wzoru rekurencyjnego:

$$R^{(n)} = R_i R_i^{(n-1)} + (1 - R_i) R_i^{(n-1)} \quad (5)$$

gdzie $R_i^{(n-1)}$, $R_i^{(n-1)}$ oznaczają odpowiednio niezawodność zdekomponowanej struktury ze „zwarciem” ($R_i = 1$) oraz z „przerwą” ($R_i = 0$).

3.1 Przypadek struktury mostkowej

Dla struktury mostkowej przedstawionej na rysunku 1 istnieją cztery minimalne ścieżki: $P_1 = \{1, 2\}$, $P_2 = \{3, 4\}$, $P_3 = \{1, 4, 5\}$ i $P_4 = \{2, 3, 5\}$. Dla tych minimalnych ścieżek

funkcja opisująca niezawodność systemu z uwzględnieniem jego elementów przyjmuje postać:

$$f = 1 - (1 - R_1 R_2)(1 - R_3 R_4)(1 - R_1 R_4 R_5)(1 - R_2 R_3 R_5) \quad (6)$$

Analizując metodę zbioru minimalnych cięć, struktura mostkowa scharakteryzowana jest przez następujące cięcia: $C_1 = \{1, 3\}$, $C_2 = \{2, 4\}$, $C_3 = \{2, 3, 5\}$, $C_4 = \{1, 4, 5\}$, przy czym funkcja na podstawie równania (4) jest w postaci:

$$f = [1 - (1 - R_1)(1 - R_3)][1 - (1 - R_2)(1 - R_4)][1 - (1 - R_2)(1 - R_3)(1 - R_5)][1 - (1 - R_1)(1 - R_4)(1 - R_5)] \quad (7)$$

Jak można zauważyć, system mostkowy zawiera zbiory $\{2, 3, 5\}$, $\{1, 4, 5\}$, które zawierają zarówno minimalną ścieżkę oraz minimalne cięcie.

Korzystając z metody dekompozycji, niezawodność systemu mostkowego można wyznaczyć w oparciu o niezawodności zdekomponowanych struktur ze względu na element 5, które wynoszą:

$$R_5^{(4)} = [1 - (1 - R_1)(1 - R_3)][1 - (1 - R_2)(1 - R_4)]$$

$$R_5^{(4)} = R_1 R_2 + R_3 R_4 - R_1 R_2 R_3 R_4$$

Dla tak określonych niezawodności, całkowita niezawodność systemu mostkowego dana jest zależnością:

$$R^{(5)} = R_5 R_5^{(4)} + (1 - R_5) R_5^{(4)} = R_1 R_2 + R_3 R_4 + R_1 R_4 R_5 + R_2 R_3 R_5 - R_1 R_2 R_3 R_4 +$$

$$- R_1 R_2 R_4 R_5 - R_1 R_2 R_3 R_5 - R_1 R_3 R_4 R_5 - R_2 R_3 R_4 R_5 + 2 R_1 R_2 R_3 R_4 R_5 \quad (8)$$

W przypadku systemu jednorodnego ($R_i = r$) całkowita niezawodność sprowadza się do postaci:

$$R^{(5)} = 2r^2 + 2r^3 - 5r^4 + 2r^5$$

3.2 Przypadek struktury 10-cio elementowej

W celu oszacowania całkowitej niezawodności struktury przedstawionej na rysunku 2 wykorzystano również metody zbioru minimalnych cięć i minimalnych ścieżek. Korzystając z metody zbioru minimalnych ścieżek można wyróżnić 8 ścieżek. Zatem mamy 4 ścieżki rzędu 4: $P_1 = \{1, 2, 3, 4\}$, $P_2 = \{7, 8, 9, 10\}$, $P_3 = \{1, 5, 9, 10\}$, $P_4 = \{1, 2, 6, 10\}$ oraz odpowiadające im niezawodności: $\pi(P_1) = R_1 R_2 R_3 R_4$, $\pi(P_2) = R_7 R_8 R_9 R_{10}$, $\pi(P_3) = R_1 R_5 R_9 R_{10}$, $\pi(P_4) = R_1 R_2 R_6 R_{10}$. Z kolei dla 4 ścieżek rzędu 6: $P_5 = \{7, 8, 5, 2, 3, 4\}$, $P_6 = \{1, 5, 9, 6, 3, 4\}$, $P_7 = \{7, 8, 9, 6, 3, 4\}$, $P_8 = \{7, 8, 5, 2, 6, 10\}$ mamy $\pi(P_5) = R_7 R_8 R_5 R_2 R_3 R_4$, $\pi(P_6) = R_1 R_5 R_9 R_6 R_3 R_4$, $\pi(P_7) = R_7 R_8 R_9 R_6 R_3 R_4$, $\pi(P_8) = R_7 R_8 R_5 R_2 R_6 R_{10}$.

Niezawodność całego systemu można przedstawić następującą zależnością:

$$f = 1 - \prod_{i=1}^n (1 - \pi(P_i)) = 1 - (1 - R_1 R_2 R_3 R_4)(1 - R_7 R_8 R_9 R_{10})(1 - R_1 R_5 R_9 R_{10})(1 - R_1 R_2 R_6 R_{10})$$

$$(1 - R_7 R_8 R_5 R_2 R_3 R_4)(1 - R_1 R_5 R_9 R_6 R_3 R_4)(1 - R_7 R_8 R_9 R_6 R_3 R_4)(1 - R_7 R_8 R_5 R_2 R_6 R_{10}) \quad (9)$$

Struktura 10-cio elementowa scharakteryzowana jest przez 16 minimalnych cięć, w tym:

- 5 drugiego rzędu: $C_1 = \{1, 7\}$, $C_2 = \{1, 8\}$, $C_3 = \{2, 9\}$, $C_4 = \{3, 10\}$, $C_5 = \{4, 10\}$,
- 6 trzeciego rzędu: $C_6 = \{1, 5, 9\}$, $C_7 = \{2, 6, 10\}$, $C_8 = \{2, 5, 8\}$, $C_9 = \{3, 6, 9\}$, $C_{10} = \{2, 5, 7\}$, $C_{11} = \{4, 6, 9\}$,
- 5 czwartego rzędu: $C_{12} = \{7, 6, 5, 3\}$, $C_{13} = \{8, 6, 5, 3\}$, $C_{14} = \{7, 5, 6, 4\}$, $C_{15} = \{8, 5, 6, 4\}$, $C_{16} = \{1, 5, 6, 10\}$.

Niezawodność całej struktury podana z wykorzystaniem równania (4) jest wyznaczona z następującej zależności:

$$\begin{aligned}
f = & [1 - (1 - R_1)(1 - R_7)][1 - (1 - R_1)(1 - R_8)][1 - (1 - R_2)(1 - R_9)][1 - (1 - R_3)(1 - R_{10})][1 + \\
& - (1 - R_4)(1 - R_{10})][1 - (1 - R_1)(1 - R_5)(1 - R_9)][1 - (1 - R_2)(1 - R_6)(1 - R_9)][1 - (1 - R_2) \\
& (1 - R_5)(1 - R_8)][1 - (1 - R_3)(1 - R_6)(1 - R_9)][1 - (1 - R_2)(1 - R_5)(1 - R_7)][1 - (1 - R_4)(1 + \\
& - R_6)(1 - R_9)][1 - (1 - R_7)(1 - R_6)(1 - R_5)(1 - R_3)][1 - (1 - R_8)(1 - R_6)(1 - R_5)(1 - R_3)][1 + \\
& - (1 - R_7)(1 - R_6)(1 - R_5)(1 - R_4)][1 - (1 - R_8)(1 - R_5)(1 - R_6)(1 - R_4)][1 - (1 - R_1)(1 - R_5) \\
& (1 - R_6)(1 - R_{10})]
\end{aligned} \quad (10)$$

W przypadku metody dekompozycji, przeprowadzana analiza na nowych elementach powtarzana jest tak długo, aż struktury wynikłe z podziału są wystarczająco proste do obliczenia. Stąd początkową strukturę $R^{(10)}$ można zdekomponować ze względu na element 5, zastępując go „zwarciem” oraz „przerwą”, dla której niezawodność określona jest jako:

$$R^{(10)} = R_5 R_5^{(9)} + (1 - R_5) R_5^{(9)} \quad (11)$$

W kolejnym kroku struktury $R_5^{(9)}$ i $R_5^{(9)}$ dekomponowane są ze względu na element 6, dla których:

$$\begin{aligned}
R_5^{(9)} &= R_6 R_{5,6}^{(8)} + (1 - R_6) R_{5,6}^{(8)} \\
R_5^{(9)} &= R_6 R_{5,6}^{(8)} + (1 - R_6) R_{5,6}^{(8)}
\end{aligned} \quad (12)$$

W ten sposób otrzymane struktury $R_{5,6}^{(8)}$, $R_{5,6}^{(8)}$, $R_{5,6}^{(8)}$, $R_{5,6}^{(8)}$ są już strukturami prostymi, dla których można łatwo obliczyć niezawodność.

4. Algorytm świetlika

Algorytm świetlika (FA, ang. *firefly algorithm*) bazujący na zachowaniu świetlików dążących do źródła światła i ich wzajemnym oddziaływaniu za pomocą sygnałów bioluminescencyjnych jest jednym z algorytmów należących do grupy algorytmów stadnych. Podstawą jego działania jest zjawisko podążania świetlika w stronę jaśniejszego osobnika. Jedną z zasad wykorzystywanych w algorytmie świetlika jest brak zróżnicowania świetlików pod względem płci. Ponadto atrakcyjność świetlików jest proporcjonalna do intensywności emitowanego przez nich światła, przy czym intensywność światła określona przez wartość funkcji celu (dla problemów maksymalizacji jest proporcjonalna do wartości funkcji celu) maleje wraz ze wzrostem odległości między świetlikami. Jeśli w otoczeniu danego świetlika nie ma osobnika atrakcyjniejszego, to porusza się on losowo [16, 17]. Każdy świetlik ma określoną intensywność światła I zmieniającą się wraz z odległością r pomiędzy dwoma osobnikami oraz atrakcyjność β , która jest proporcjonalna do intensywności świecenia widzianej przez sąsiednie świetliki, zatem jest wielkością zależną od odległości oraz od współczynnika absorpcji światła γ [17]:

$$\beta(r) = \beta_0 e^{-\gamma r^m}, \quad m \geq 1 \quad (13)$$

gdzie β_0 oznacza atrakcyjność w $r = 0$.

Ruch, podczas którego świetlik i będący w pozycji x_i stara się zbliżyć do atrakcyjniejszego osobnika j w pozycji x_j wykonywany jest według zależności [17]:

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha (rand - \frac{1}{2}) \quad (14)$$

gdzie x_i to bieżąca pozycja świetlika i , drugi składnik określa atrakcyjność, zaś trzeci stanowi o losowym przemieszczaniu ($rand$ jest wygenerowaną liczbą o rozkładzie jednostajnym w przedziale $[0, 1]$, zaś $\alpha \in [0, 1]$).

Ogólna struktura algorytmu jest następująca [7, 16, 17]:

1. Inicjalizacja parametrów algorytmu (β_0 , γ , warunek stopu) i losowe wygenerowanie początkowej populacji n świetlików; zdefiniowanie funkcji celu $f(x)$.
2. Wyznaczenie intensywności światła każdego osobnika, przy czym intensywność światła i -tego świetlika I_i jest określona przez wartość funkcji celu $f(x_i)$.
3. Dopóki nie jest spełniony warunek stopu, należy:
 - porównać parami wszystkie świetliki pod względem intensywności światła: jeśli ($I_j > I_i$), to przesunąć świetlika i w kierunku świetlika j ,
 - wyznaczyć nowe wartości funkcji celu, ocenić nowe rozwiązania, uaktualnić intensywność światła.
4. Jeśli jest spełnione kryterium stopu, wyznaczyć najlepsze rozwiązanie.

Algorytm świetlika pierwotnie przeznaczony był do optymalizacji problemów ciągłych. Jego zastosowanie do optymalizacji niezawodności wybranych struktur, w których występują zmienne decyzyjne ciągłe i dyskretne, wymaga pewnych dodatkowych działań. Właściwe określenie odległości czy sposobu poruszania się osobników, które zapewnią ważność rozwiązania, to główne elementy algorytmu, które należy dopasować. Zakładamy, że odległość między dwoma świetlikami wyznaczona jest jako norma z różnicy wartości zmiennych decyzyjnych przyporządkowanych obu osobnikom. Ruch każdego świetlika w kierunku jaśniejszego osobnika, polega na wykonaniu określonej liczby kroków, przy czym długość kroku nie może przekroczyć predefiniowanych maksymalnych zmian wartości zmiennych ciągłych (STEP_MAX_CV) oraz dyskretnych (STEP_MAX_DV). Jeśli po wykonaniu kroku świetlik znajdzie się poza obszarem dopuszczalnym to maksymalna długość kroku jest zmniejszona (mnożona przez liczbę losową z przedziału [0.5, 0.99]). Jeśli po określonej liczbie prób (MAX_P) nie uda się zmodyfikować rozwiązania tak, aby znajdowało się ono w obszarze dopuszczalnym, świetlik nie wykonuje żadnego kroku.

5. Wyniki eksperymentów

Korzystając z wartości ustawień różnych parametrów w wybranych dwóch systemach, zestawionych w tabelach 1 i 2, wykonano wiele eksperymentów mających na celu zbadanie przydatności algorytmu świetlika w rozwiązywaniu wybranych problemów niezawodności. Jak wiadomo niezawodność dla minimalnych cięć (tzw. dolne oszacowanie) jest zawsze mniejsza niż dla minimalnych ścieżek (tzw. górne oszacowanie), co stanowi bazę do poszukiwania optymalnych wartości. W ramach testów sprawdzono wydajność algorytmu świetlika przy wybranych, ustalonych parametrach i porównano osiągnięte wyniki z najlepszymi, dotychczas znanymi rozwiązaniami. Przedstawienie wyników zastosowania algorytmu do rozwiązania problemu niezawodności systemu złożonego z 10 elementów zostało ograniczone jedynie do omówienia wyników dla przypadku, gdy $m = 5$. Algorytm świetlika został zaimplementowany w środowisku Matlab 2015a. W trakcie przeprowadzonych badań dotyczących sprawdzenia jakości wyników algorytmu ustalono następujące wartości jego parametrów: kryterium stopu pojedynczego uruchomienia: 1000 iteracji, rozmiar populacji: 10 lub 30 osobników, MAX_P = 100, STEP_MAX_CV = 0.5, STEP_MAX_DV = 2, $\gamma = 0.1$. Dla każdego przypadku uruchomiono 50 niezależnych powtórzeń algorytmu.

Wyniki eksperymentów przedstawiono w tabelach 3 i 4, w których zestawiono najlepsze i najgorsze otrzymane rezultaty oraz wartość średnią z 50 powtórzeń. Wyniki badań sugerują przewagę prezentowanego algorytmu z metodą minimalnych ścieżek nad pozostałymi metodami.

Tabela 3. Wyniki dla systemu mostkowego dla 50 powtórzeń

Metoda	Liczba świateł	Wartość najlepsza	Wartość najgorsza	Wartość średnia
dekompozycji	30	0.999889027392830	0.999692113944072	0.999867770075797
	10	0.999882704854672	0.999535345770864	0.999789510970645
cięć	30	0.999887373640587	0.999709686550381	0.999839910069411
	10	0.999881295104186	0.999561725825725	0.999795740150406
ścieżek	30	0.999998825015460	0.999995999869590	0.999997854290027
	10	0.99999874719315	0.999992668983783	0.999997639615957

Tabela 4. Wyniki dla systemu zbudowanego z 10 elementów ($m = 5$)

Metoda	Liczba świateł	Wartość najlepsza	Wartość najgorsza	Wartość średnia
dekompozycji	30	0.999124934817144	0.998712767969089	0.999029684217294
	10	0.999124934817144	0.997639045897561	0.998706554859805
cięć	30	0.999123179843347	0.998518087543003	0.998951987420550
	10	0.999123179843347	0.997349605284400	0.998697533072659
Ścieżek	30	0.999999983601514	0.999999961168195	0.999999979369618
	10	0.999999983601514	0.999999967299759	0.999999978215736

Analiza danych literaturowych dotycząca najlepszych uzyskanych rozwiązań różnymi metodami inspirowanymi zachowaniami stadnymi, w tym algorytmami mrówkowymi (ACO), optymalizacji rojem cząstek (PSO) wraz z wersją zmodyfikowaną (MPSO), sztucznej kolonii pszczół (ABC), kukułki (CS) i nietoperza (BAT), pozwoliła na ich zbiorcze zestawienie (Tabela 5).

Tabela 5. Zestawienie najlepszych uzyskanych wyników z danych literaturowych

Struktura: system mostkowy		
<i>Algorytm</i>	<i>Najlepszy wynik</i>	<i>Wartość średnia</i>
PSO [4]	0.99988957	0.99988594
PSO [15]	0.99988963	-
MPSO [10]	0.9998896376	0.9998891423
ABC [18]	0.99988962	0.99988362
CS-GA [6]	0.99988964	0.9998854
CS [13, 14]	0.99988964	0.99987998
BAT [9]	0.9998896376	0.9998894767
Struktura: system 10-elementowy		
<i>Algorytm</i>	<i>Najlepszy wynik</i>	<i>Wartość średnia</i>
ACO [1]	0.999991	0.9980477
CS [13, 14]	0.67189992	0.67189992

Jak wynika z przeprowadzonych obliczeń, dla systemu mostkowego jedynie algorytm świateł z metodą zbioru minimalnych ścieżek zdolności pozwolił uzyskać wyniki (0.999998825015460) przewyższające rezultaty PSO, MPSO, ABC, CS, CS-GA i BAT. Niestety takiego wniosku nie można wysnuć porównując wyniki FA z pozostałymi metodami. W przypadku systemu 10-cio elementowego widać wyraźną przewagę algorytmu świateł

nad algorytmem kukułki. Należy jednak podkreślić, że dla rozpatrywanych przykładów wyniki otrzymywane podczas maksymalizacji przez metodę minimalnych ścieżek różnią się znacznie od wyników metod dekompozycji oraz minimalnych cięć. Zatem można stwierdzić, że przy projektowaniu różnych rzeczywistych systemów najbezpieczniejszym wyjściem jest przyjęcie dolnej wartości oszacowania niezawodności.

4. Wnioski

W pracy zaprezentowano wyniki badań uzyskanych przy zastosowaniu algorytmu świetlika w problemach optymalizacji niezawodności z uwzględnieniem redundancji. W celu zbadania efektywności algorytmu wybrano dwa systemy oraz trzy metody wyznaczania niezawodności tj. zbioru minimalnych ścieżek, zbioru minimalnych cięć oraz metodę dekompozycji. Analizując uzyskane wyniki można stwierdzić, że dla rozpatrywanych systemów uzyskano znaczną poprawę wyników dla algorytmu świetlika zastosowanego do metody zbioru minimalnych ścieżek. Warto zwrócić uwagę na fakt, że dotychczasowe wyniki dotyczące stosowania algorytmów stadnych prezentowane w literaturze okazały się gorsze niż te, które udało się uzyskać za pomocą zaproponowanej implementacji algorytmu świetlika.

Literatura

1. Agarwal M, Sharma V K. Ant colony approach to constrained redundancy optimization in binary systems. *Applied Mathematical Modelling* 2010; 34: 992-1003.
2. Barlow R E, Proschan F. *Statistical theory of reliability and life testing: probability models. To Begin With*, Silver Spring 1981.
3. Birolini A. *Reliability Engineering: Theory and Practice*. Sixth Edition, Springer-Verlag Berlin Heidelberg 2010.
4. Coelho L S. An efficient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications. *Reliability Engineering and System Safety* 2009; 94(4): 830-837.
5. Garg H. An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm. *Beni-Suef University Journal of Basic and Applied Sciences* 2015; 4(1): 14-25.
6. Kanagaraj G, Ponnambalam S G, Jawahar N. A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems. *Computers & Industrial Engineering* 2013; 66(4): 1115-1124.
7. Kwiecień J, Filipowicz B. Comparison of firefly and cockroach algorithms in selected discrete and combinatorial problems. *Bulletin of the Polish Academy of Sciences. Technical Science* 2014; 62(4): 797-804.
8. Lefebvre M. *Basic Probability Theory with Applications*. Springer-Verlag New York 2009.

9. Liu Y. Improved bat algorithm for reliability-redundancy allocation problems. *International Journal of Security and Its Applications* 2016; 10(2): 1-12.
10. Liu Y, Qin G. A modified particle swarm optimization algorithm for reliability redundancy optimization problem. *Journal of Computers* 2014; 9(9): 2124-2131.
11. Matuszak Z. *Badania rozkładów uszkodzeń systemów siłowni okrętowych*. Wyd. ADVSEO, Szczecin 2012.
12. Mutingi M, Mbohwa Ch, Kommula V P. Multi-criteria reliability optimization for a complex system with a bridge structure in a fuzzy environment: A fuzzy multi-criteria genetic algorithm approach. *Eksploatacja i Niezawodność – Maintenance and Reliability* 2016; 18 (3): 450-456.
13. Valian E, Tavakoli S, Mohanna S, Haghi A. Improved cuckoo search for reliability optimization problems. *Computers & Industrial Engineering* 2013; 64(1): 459-468.
14. Valian E. Solving reliability optimization problems by cuckoo search. [W:] X.S. Yang (ed.), *Cuckoo search and firefly algorithm. Theory and applications*, Springer 2014.
15. Wu P, Gao L, Zou D, Li S. An improved particle swarm optimization algorithm for reliability problems. *ISA Transactions* 2010; 50(1): 71-81.
16. Xing B, Gao W J. *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*. Intelligent Systems Reference Library 62, Springer International Publishing Switzerland 2014.
17. Yang X S. *Nature-Inspired Metaheuristic Algorithms*. Second Edition, Luniver Press 2010.
18. Yeh W C, Hsieh T J. Solving reliability redundancy allocation problems using an artificial bee colony algorithm. *Computers & Operations Research* 2011; 38(11): 1465-1473.