

Stanisław Kozdrowski *

Jarosław Arabas †

Zastosowanie Algorytmów Ewolucyjnych w odpornym wymiarowaniu sieci telekomunikacyjnych

Streszczenie

Artykuł przedstawia zastosowanie algorytmu ewolucyjnego do problemu wymiarowania sieci telekomunikacyjnych odpornych na awarie. Rozważane zadania sformułowane są w języku optymalizacji przepływów wielotowarowych i należą do zadań całkowitoliczbowych bez ustalonych pojemności krawędzi. Przedstawione jest porównanie algorytmu ewolucyjnego z metodą programowania liniowego, zaimplementowaną w pakiecie CPLEX.

1 Wprowadzenie

Artykuł omawia zagadnienia dotyczące wymiarowania zasobów sieci telekomunikacyjnych odpornych na awarie. Sformułowane są warianty ogólnego problemu wymiarowania odpornego w postaci pozwalającej na ich zastosowanie do dowolnej warstwy zasobów sieci (np. do sieci transmisyjnej SDH, sieci cyfrowych grup łączy, warstwy ścieżek wirtualnych sieci ATM) [6].

Problem projektowania sieci odpornych może być sformułowany jako problem programowania liniowego (LP) i rozwiązany za pomocą pakietu CPLEX [2]. W przypadku gdy pojemności krawędzi są liczbami całkowitymi stosuje się wtedy LP z uwzględnieniem całkowitoliczbowości zmiennych (ang. Mixed Linear Programming (MIP)). Metoda ta wykorzystuje algorytm podziału i oszacowań, znajdujący dokładne rozwiązanie w akceptowalnym czasie. Okazuje się, że w przypadku dużych (nawet kilkunastowęzłowych) sieci metoda MIP jest czasochłonna i nie zawsze znajduje rozwiązanie problemu w akceptowalnym czasie. Dlatego stosowanie metod heurystycznych jest szczególnie uzasadnione. Jako podejście do rozwiązywania zadań wymiarowania odpornego wprowadzono specjalizowany algorytm ewolucyjny (EA) [1]. Pokrewne zastosowanie EA do rozmieszczania zapotrzebowań w istniejącej sieci o zadanych pojemnościach krawędzi zostało opisane w [4]. W literaturze spotyka się też zastosowanie innych metod optymalizacji dyskretnych do rozwiązania problemu wymiarowania sieci ([8] i [9]).

W niniejszym artykule zamieszczono opis algorytmu ewolucyjnego, specjalizowanego do problemu wymiarowania sieci telekomunikacyjnych. Przeprowadzono szereg eksperymentów na przykładowych sieciach. Wyniki porównano z uzyskanymi za pomocą CPLEX-a. Porównano rozwiązania uzyskane za pomocą obu metod, jak również ich szybkość działania.

* Instytut Telekomunikacji, Politechnika Warszawska, e-mail: *stko@tele.pw.edu.pl*.

Praca była finansowana w ramach Programu Automatyki, Technik Informacyjnych i Automatyzacji (PATIA) na Politechnice Warszawskiej.

† Instytut Podstaw Elektroniki, Politechnika Warszawska.

Praca była finansowana ze środków KBN w ramach grantu 8T11C04611.

2 Model sieci i sformułowanie problemu

2.1 Model sieci

Sieci przepływowe rozważane w tym artykule modelowane są za pomocą *multigrafu* $G = (V, E, \mathcal{P})$ składającego się ze zbioru *węzłów* V , zbioru *krawędzi* E oraz *relacji incydencji* węzłów i krawędzi $\mathcal{P}$ określonej na iloczynie kartezjańskim $V \times E \times V$. W grafie G rozważać będziemy tylko *ścieżki (drogi) proste* (przechodzące przez węzły co najwyżej jeden raz), identyfikując je ze zbiorem krawędzi $p = \{e_1, e_2, \dots, e_n\}$. Jako multigraf, G dopuszcza istnienie wielu krawędzi pomiędzy dowolną parą węzłów. Zbiór wszystkich krawędzi pomiędzy daną parą węzłów nazywa się *multikrawędzią*. Możliwość taka jest użyteczna przy modelowaniu sytuacji awaryjnych w sieci: w danej sytuacji awarii ulegają tylko niektóre z krawędzi wchodzących w skład multikrawędzi pomiędzy poszczególnymi parami węzłów.

Pojemność krawędzi i wielkości zapotrzebowań są wyrażane jako wielokrotność *absolutnej jednostki zapotrzebowania* (ACU). Pojemność (przepustowość) krawędzi $e \in E$ jest oznaczana przez $y(e)$ i wyrażona w *jednostkach pojemności krawędzi* (LCU); jedna jednostka LCU jest równa m jednostek ACU. Koszt jednej jednostki LCU oznaczony jest przez $\xi(e)$, koszt stały związany z ‘otwarcie’ nowej krawędzi oznaczony jest przez $\kappa(e)$, a koszt związany z długością krawędzi — przez $\delta(e)$. W sumie koszt krawędzi sieci wyrażamy jako:

$$C(y) = \sum_{e \in E} \xi(e)y(e) + \sum_{e \in E, y(e) > 0} \kappa(e) + \sum_{e \in E} \delta(e)y(e). \quad (1)$$

gdzie $y: E \rightarrow \mathbf{N}$ jest funkcją przypisującą pojemność krawędziom ($\mathbf{N} = \{0, 1, \dots\}$ jest zbiorem liczb naturalnych); $\kappa, \xi: E \rightarrow \mathbf{R}^+$ są funkcjami kosztu ($\mathbf{R}^+$ jest zbiorem nieujemnych liczb rzeczywistych). Natomiast $\delta(e)$ jest funkcją opisującą długość krawędzi.

Niech D oznacza zbiór *zapotrzebowań*. Każde zapotrzebowanie $d \in D$ charakteryzują poniższe atrybuty:

- para węzłów $(s(d)$ i $t(d))$ pomiędzy którymi istnieje zapotrzebowanie,
- *wielkość zapotrzebowania* $n(d)$, będąca liczbą *jednostek przepustowości* (DCU),
- Zbiór *ścieżek dopuszczalnych* $P(d)$; każda ścieżka $p \in P(d)$ jest ścieżką prostą łączącą węzły $s(d)$ i $t(d)$, służącą do realizacji części zapotrzebowania DCU.

W praktyce często istnieje wymaganie na agregację alokowanych w sieci jednostek przepustowości, aby każde zapotrzebowanie było realizowane w kilku (np. dwóch) niepodzielnych *modułach* grupujących jednostki przepustowości. Wszystkie jednostki przepustowości modułu są realizowane zawsze razem (na jednej ścieżce). Wymaganie tego typu będziemy opisywać za pomocą *współczynnika agregacji* W . I tak $W = 100\%$ oznacza pełną agregację (każde zapotrzebowanie realizowane jest jako jeden moduł), $W = 50\%$ (każde zapotrzebowanie realizowane jest co najwyżej w dwóch modułach) itd. ($W = 0\%$ oznacza pełną dezagregację — każda jednostka przepustowości zapotrzebowania może być realizowana i odtwarzana na ‘swojej’ ścieżce).

S oznacza rozpatrywany zbiór *sytuacji (stanów)* awaryjnych. Każda sytuacja $s \in S$ jest scharakteryzowana poprzez współczynnik $\alpha(s, e)$ ($e \in E$), który oznacza dostępną część pojemności krawędzi $e \in E$ w sytuacji s . Jeśli $s_0 \in S$ jest stanem nominalnym (bezawaryjnym), wówczas wszystkie współczynniki $\alpha(s_0, e)$ są równe 1. Dla każdej sytuacji s i krawędzi e , $\alpha(s, e) \in \{0, 1\}$. Oznacza to, że w sytuacji awaryjnej s krawędź albo w całości ulega awarii ($\alpha(s, e) = 0$), albo w całości jest sprawna ($\alpha(s, e) = 1$).

2.2 Zadania projektowania odpornego

W celu zrealizowania zapotrzebowania na przepustowość $d \in D$ (które wynosi $n(d)$ jednostek) należy wskazać, jaka część $x(d, p)$ tego zapotrzebowania (przepływu) realizowana jest na każdej

dostępnej ścieżce $p \in P(d)$. Funkcję całkowitoliczbową $x : Q \rightarrow N$ (gdzie $Q = \{(d, p) : d \in D, p \in P(d)\}$) nazywamy *funkcją alokacji* lub *przepływu* a jej wartości — *przepływami* wyrażonymi w DCU. Dla ustalonego x , wartość $r(d, x) = \sum_{p \in P(d)} x(d, p)$ jest nazywana *stopniem realizacji* zapotrzebowania $d \in D$, a wartość $o(e, x) = \sum_{d \in D, p \in P(d), e \in p} x(d, p)$ *obciążeniem* krawędzi $e \in E$ (wyrażona w modułach DCU). Wreszcie przez $f(o)$ oznaczamy minimalną pojemność krawędzi e wyrażoną w LCU, niezbędną do realizacji obciążenia o (funkcje $f(\cdot)$ nie muszą być liniowe).

Zadanie 0: Wymiarowanie dla stanu nominalnego

Dla zadanego zbioru zapotrzebowań D znaleźć:

- funkcję pojemności krawędzi y
- funkcję alokacji x

minimalizujące koszt sieci (1) przy ograniczeniach:

$$r(d, x) = n(d), \quad d \in D \quad (2)$$

$$x(d, p) \leq b(d) \cdot n(d), \quad d \in D, p \in P(d) \quad (3)$$

$$f(o(e, x)) \leq y(e), \quad e \in E. \quad (4)$$

Ograniczenie (2) oznacza realizowalność zapotrzebowania, natomiast warunek (3) ma charakter niezawodnościowy — nie pozwala realizować więcej niż część $b(d)$ (liczby $b(d)$ są ustalone) przepustowości na jednej ścieżce. Ograniczenie (4) zapewnia dostateczną pojemność krawędzi. Ze względu na całkowitoliczbowość (modularność) funkcji x oraz y , nawet przy liniowych funkcjach $f(\cdot)$ powyższy problem jest *NP-zupełny* [3]. Zagadnienie modularności ilustruje następujący przykład.

Przykład: Załóżmy, że zapotrzebowania na przepustowość $n(d)$ wyrażone są liczbą strumieni 2Mb/s, a pojemność krawędzi $y(e)$ — liczbą modułów STM-1; jeden moduł STM-1 odpowiada jednej jednostce LCU; jedna jednostka LCU jest równa $m = 63 \cdot \text{ACU}$, gdyż jeden moduł STM-1 może transportować 63 kontenery VC-12, a każdy kontener zawiera jeden 2Mb/s strumień [10]). Dla każdej krawędzi mamy $f(o) = \lceil o/63 \rceil$. Zauważmy, że dla funkcji f ograniczenie (4) można zapisać w postaci: $o(e, x) \leq m \cdot y(e)$, $e \in E$.

Przy minimalnej realokacji istotne jest wyróżnienie stanu nominalnego. Spowodowane jest to wymaganiem, aby na wypadek awarii następowała realokacja tylko tych przepływów, które uległy uszkodzeniu; nie wolno natomiast zmieniać tych przepływów, które przetrwały awarię.

Zadanie 1: Wymiarowanie odporne z minimalną realokacją

Dla danych:

- zbioru zapotrzebowań D
- zbioru sytuacji awaryjnych S ze współczynnikami $\alpha(s, e)$

znaleźć:

- pojemność krawędzi y
- funkcję alokacji x_s dla wszystkich sytuacji $s \in S$ (x_0 - funkcja alokacji dla stanu nominalnego $s_0 \in S$)

minimalizujące koszt sieci (1) przy ograniczeniach:

$$r(d, x_s) = n(d), \quad d \in D, s \in S \quad (5)$$

$$x_s(d, p) \geq a_s(d, p) \cdot x_0(d, p), \quad d \in D, p \in P(d), s \in S \setminus \{s_0\} \quad (6)$$

$$f(o(e, x_s)) \leq \alpha(s, e) \cdot y(e), \quad e \in E, s \in S, \quad (7)$$

gdzie

$$a_s(d, p) = \prod_{e \in p} \alpha(s, e). \quad (8)$$

Rozwiązanie zadania wyznacza przepływy we wszystkich sytuacjach $s \in S$, włącznie ze stanem nominalnym s_0 (dla którego $\alpha(s_0, e) \equiv 1$). Zadanie 1 wymusza zachowanie przepływów nominalnych nieuszkodzonych w każdej sytuacji awaryjnej.

Zadanie 2: Zabezpieczenie sieci istniejącej z minimalną realokacją.

Zadanie to polega na zabezpieczaniu nominalnej przepustowości w sieci istniejącej, zaprojektowanej dla stanu nominalnego. Pojemność każdej krawędzi składa się z części istniejącej (podstawowej) — $y_0(e)$ i części dodawanej (zabezpieczającej) — $y'(e)$. Zakładamy, że przepływy $x_0(d, p)$ w stanie nominalnym są zadane i że realizują one zapotrzebowania $n(d)$ wyłącznie przy użyciu pojemności podstawowych $y_0(e)$. Ograniczenie (7) przyjmuje w tym zadaniu postać $f(o(e, x_s)) \leq \alpha(s, e) \cdot (y_0(e) + y'(e))$, $e \in E, s \in S \setminus \{s_0\}$. Minimalizowana funkcja kosztu (1) przybiera tutaj następującą postać

$$C'(y) = \sum_{e \in E} \xi(e)y'(e) + \sum_{e \in E, y(e) > 0} \kappa(e) + \sum_{e \in E} \delta(e)y'(e). \quad (9)$$

3 Algorytm Ewolucyjny

Wykorzystany przez nas Algorytm Ewolucyjny oparty jest na Strategii Ewolucyjnej $(\mu + \lambda)$ [7]. Metoda kodowania chromosomu i zastosowane operatory genetyczne są ściśle związane z rozpatrywanym problemem i różnią się znacznie od typowej Strategii Ewolucyjnej.

3.1 Kodowanie chromosomu

Chromosom (oznaczony przez x) jest stabilizowaną funkcją alokacji x i ma następującą strukturę. Każde zapotrzebowanie d_i jest alokowane na predefiniowanych ścieżkach (zapotrzebowania są ponumerowane, tzn. $D = \{d_1, d_2, \dots, d_K\}$). Dla każdego zapotrzebowania d_i określony jest wektor alokacji x_i , którego elementy x_{ij} są przepływami zapotrzebowania d_i realizowanymi na dostępnych ścieżkach dla tego zapotrzebowania (ścieżki te należą do zbioru $P(d_i) = \{p_{i1}, p_{i2}, \dots, p_{ih(i)}\}$). W algorytmie *EA* rozpatrujemy tylko stany maksymalne, tzn. pełnej alokacji:

$$\sum_{j=1}^{h(i)} x_{ij} = n(d_i), \quad x_{ij} \in \mathbf{N}. \quad (10)$$

Wektory x_i tworzą wzorec alokacji x i w sumie tworzą chromosom. Przykładowy fragment chromosomu jest przedstawiony poniżej. W przykładzie poniżej rozpatrywane są cztery zapotrzebowania. I tak, drugie zapotrzebowanie (dla którego $n(d_2) = 5$) na pierwszej dostępnej ścieżce ma 2 jednostki pojemności DCU i 3 jednostki DCU na drugiej dostępnej ścieżce. Trzecie zapotrzebowanie zostało w całości rozłożone na czwartej dostępnej ścieżce.

1	2	0	12
0	3	0	0
10	0	3	
		15	

3.2 Inicjacja populacji

Podczas inicjalizacji populacji każdy chromosom jest generowany w sposób losowy, przy czym muszą być spełnione ograniczenia (10). W praktyce populację chromosomów generujemy poprzez alokację wszystkich modułów pojemności DCU każdego zapotrzebowania d_i na losowo wybraną ścieżkę, dostępną dla tego zapotrzebowania.

3.3 Operatory genetyczne**3.3.1 Mutacja rozpraszająca**

Mutacja rozpraszająca jest typem mutacji, która losowo wybiera wektor alokacji x_i , realokując jego przepływy, niezależnie od ich poprzedniej wartości.

$$x'_{ij} = \rho_j, \quad j = 1, 2, \dots, h(i) \quad (11)$$

gdzie $\rho = \rho_1, \rho_2, \dots, \rho_{h(i)}$ jest losowo wygenerowanym wektorem o właściwościach:

$$\sum_{j=1}^{h(i)} \rho_j = n(d_i), \quad \rho_j \in N. \quad (12)$$

3.3.2 Mutacja z poprawką

Mutacja z poprawką dla wektora alokacji x_i składa się z trzech kroków. W pierwszym kroku jest wybierana ścieżka $p_k, k \in \{1, 2, \dots, h(i)\}$ zgodnie z regułą:

1. $x_{ik} \neq 0$,
2. p_k zawiera krawędzie o największej minimalnej szczałkowej pojemności krawędzi.

Pojemność resztkową (pozostałość pojemności) $res(e)$ krawędzi e definiujemy następująco:

$$res(e, x) = m \cdot f(o(e, x)) - o(e, x) \quad (13)$$

gdzie m jest modularnością krawędzi. Całkowitoliczbowy wektor ρ tworzony jest losowo w następujący sposób:

$$\forall j \neq k \quad \rho_j \in N \quad (14)$$

$$-x_{ik} \leq \rho_k < 0 \quad (15)$$

$$\sum_{j=1}^{h(i)} \rho_j = 0. \quad (16)$$

W trzecim kroku generowany jest nowy wektor alokacji x'_i , zgodnie z regułą:

$$x'_{ij} = x_{ij} + \rho_j, \quad j = 1, 2, \dots, h(i). \quad (17)$$

3.3.3 Krzyżowanie równomierne

W wyniku krzyżowania równomiernego para chromosomów $x^{(P1)}, x^{(P2)}$ ustępuje miejsce chromosomom potomnym $x^{(C1)}, x^{(C2)}$ zgodnie z regułą:

$$\begin{aligned} x_i^{(C1)} &= x_i^{(P1)} & x_i^{(C2)} &= x_i^{(P2)} & \text{z prawdopodobieństwem } 0.5 \\ x_i^{(C1)} &= x_i^{(P2)} & x_i^{(C2)} &= x_i^{(P1)} & \text{z prawdopodobieństwem } 0.5. \end{aligned}$$

Warto zauważyć, że ze spełnialności ograniczeń rodziców wynika spełnialność ograniczeń przez potomków. Zatem ograniczenia nie będą nigdy przekroczone i przestrzeń rozwiązań ogranicza się tylko do zbioru rozwiązań dopuszczalnych. Przestrzeń rozwiązań redukuje się znacznie do zbioru rozwiązań dopuszczalnych.

3.4 Funkcja celu

Wartość funkcji celu w dużym stopniu zależy od rozpatrywanego zadania. Skoncentrowaliśmy się na dwóch problemach (i) wymiarowanie dla stanu nominalnego — Zadanie 0, i (ii) wymiarowanie odporne (Zadanie 1 i Zadanie 2). Należy podkreślić, że dla wszystkich tych zadań struktura chromosomu pozostaje niezmienną.

3.4.1 Projektowanie dla stanu nominalnego — Zadanie 0

W stanie nominalnym $s_0 \in S$, wartość funkcji celu chromosomu x obliczana jest jako koszt sieci (1), podczas gdy pojemności krawędzi $y(e)$ są liczbami naturalnymi takimi, że warunek (4) jest spełniony.

3.4.2 Projektowanie odporne

W przypadku wymiarowania odpornego wartość funkcji celu dla chromosomu jest obliczana w nieco bardziej skomplikowany sposób. Podstawowym zadaniem jest znalezienie dla każdej krawędzi e , maksymalnego obciążenia ($o(e, x_s)$) ze wszystkich sytuacji awaryjnych s , i przyporządkowanie danej krawędzi e . Pojemności krawędzi są minimalizowane, tzn.

$$y(e) = f \left(\max_{s \in S} o(e, x_s) \right), \quad e \in E, s \in S. \quad (18)$$

Wartości pojemności krawędzi $y(e)$ (a w przypadku Zadania 2 — $y'(e)$) wykorzystane są do obliczenia całkowitego kosztu sieci (1) (a w przypadku Zadania 2 — (9)), który podlega optymalizacji. Podstawowym problemem jest tutaj obsługa sytuacji awaryjnych. Uszkodzony przepływ jest rozprowadzany w sposób losowy na wszystkie dostępne ścieżki, które są rozłączne z uszkodzoną krawędzią. Podczas zabezpieczania sieci istniejącej pojemności krawędzi w stanie nominalnym $y_0(e)$ są ustalone. Optymalizacji podlega część zabezpieczająca $y'(e)$. Jeśli przyjmiemy C_0 za wyliczony koszt Zadania 0, C_1 — koszt Zadania 1, natomiast C_2 — koszt Zadania 2, wtedy zachodzi następująca zależność: $C_1 \leq C_0 + C_2$. Łatwo zauważyć, że przestrzeń rozwiązań Zadania 0 i Zadania 2 jest podzbiorem przestrzeni rozwiązań Zadania 1.

3.5 Algorytm Ewolucyjny

Schemat EA wygląda następująco:

```

begin
  t := 0;
  initialize Pop(t);
  evaluate Pop(t);
  while (not termination_condition) do
    begin
      Offs(t) := reproduce Pop(t);
      recombine Offs(t);
      evaluate Offs(t);
      Pop(t + 1) := select_best (Pop(t) ∪ Offs(t));
      t := t + 1;
    end
  end
end

```

W kroku 'initialize Pop(t)' generowanych jest w sposób losowy μ chromosomów, które stanowią populację początkową. W następnym kroku 'evaluate Pop(t)', dla każdego chromosomu z populacji początkowej obliczana jest wartość funkcji celu. W procesie reprodukcji 'reproduce Pop(t)' tworzona jest populacja chromosomów potomnych Offs(t), która zawiera λ chromosomów, wybranych losowo z populacji Pop(t). Podczas procesu rekombinacji 'recombine Offs(t)' chromosomy z populacji potomnej Offs(t) podlegają procesom mutacji i krzyżowania. W kroku 'select_best', μ najlepszych chromosomów (o najlepszej funkcji celu) jest wybieranych spośród $\mu + \lambda$ chromosomów, reprezentujących populację Pop(t) ∪ Offs(t).

4 Przykłady numeryczne

W rozdziale tym omówione są wyniki optymalizacji przykładowych sieci. Rozpatrywaliśmy cztery topologie sieci.

1. G_1 : Sieć posiada węzeł usytuowany centralnie, zawiera 12 krawędzi, 7 węzłów i 21 zapotrzebowań — a więc jest przykładem sieci pełnej tzn. pomiędzy każdymi dwoma węzłami istnieje zapotrzebowanie. Rozważane są trzy typy obciążenia krawędzi: $G_1(l)$ — małe, $G_1(m)$ — średnie i $G_1(h)$ — duże.
2. G_2 : Następny przykład oparty jest na strukturze polskiej sieci transmisyjnej SDH. Sieć posiada 12 węzłów, 18 krawędzi i 66 zapotrzebowań. W sumie należy zaalokować 1705 jednostek przepustowości DCU.

3. G_3 : Sieć ta jest przykładem sieci transmisyjnej SDH pewnego kraju. Zawiera 9 węzłów, 23 krawędzie i 23 zapotrzebowania.
4. G_4 : Sieć ta jest przykładem większej sieci transmisyjnej SDH również pewnego kraju. Zawiera 28 węzłów, 45 krawędzi i 378 zapotrzebowań.

Poniżej przedstawimy wyniki uzyskane za pomocą metody EA i metody MIP. W przedstawionych przykładach rozważa się sytuacje awaryjne polegające na pojedynczych awariach całych krawędzi ($\alpha(s, e) \in \{0, 1\}$). Modularność krawędzi wynosi $m = 63$ (pojemność kontenera STM-1), co oznacza, że krawędź o pojemności $y(e)$ realizuje $63 \cdot y(e)$ jednostki zapotrzebowań DCU, które odpowiadają kontenerom VC-12 (DCU) [10]. Koszt jednostkowy $\xi(e)$ każdej krawędzi wynosi 1, a koszt stały $\kappa(e) = 0$. Długości krawędzi nie są rozważane ($\delta(e) = 0$).

W Tabeli 1 rozpatrywany jest stan nominalny (Zadanie 0). Wyniki Zadania 1 przedstawione są w Tabeli 2. Natomiast w Tabeli 3 — wyniki Zadania 2. We wszystkich przypadkach $m = 63$.

	EA		CPLEX
	$W = 0\%$	$W = 100\%$	$W = 0\%$
$G_1(l)$	7	7	7
$G_1(m)$	17	18	17
$G_1(h)$	46	47	45
G_2	60	61	59
G_3	26	27	25
G_4	403	406	413

 Tabela 1. Wyniki Zadania 0; $m = 63$

W przypadku stanu nominalnego (Zadanie 0) dla niedużych sieci (G_1 , G_2 i G_3) metoda MIP jest w stanie doprowadzić do rozwiązania optymalnego w dość krótkim czasie (ponieważ przestrzeń przeszukiwań jest niewielka). W przypadku sieci G_1 za pomocą metody EA również otrzymano optymalne rozwiązanie (oprócz ostatniego przypadku) w krótkim czasie (rzędu kilku sekund). Dla sieci G_4 (gdzie przestrzeń przeszukiwań jest bardzo duża) metoda EA okazała się konkurencyjna w stosunku do metody MIP.

W Zadaniu 1 dla sieci G_1 metodą MIP (podobnie jak dla Zadania 0) uzyskuje się optymalne rozwiązanie w krótkim czasie (rzędu kilkudziesięciu sekund). Metodą EA uzyskano nieco gorsze wyniki. Różnica wynosi około 3% na korzyść metody MIP. W przypadku braku agregacji ($W = 0\%$) EA jest bardziej skuteczna w porównaniu do pełnej agregacji ($W = 100\%$). Wynika to z faktu, że pełna agregacja silnie redukuje przestrzeń przeszukiwań. W przypadku dużych sieci (G_4) za pomocą metody EA uzyskuje się suboptymalne wyniki po czasie rzędu kilkunastu minut. Dla sieci G_4 metodą MIP przerwano eksperyment po dwunastu godzinach, nie uzyskując wyniku całkowitoliczbowego.

	EA		CPLEX
	$W = 0\%$	$W = 100\%$	$W = 0\%$
$G_1(l)$	10	12	10
$G_1(m)$	25	26	24
$G_1(h)$	70	72	67
G_2	95	97	95
G_3	40	41	38
G_4	684	690	—

 Tabela 2. Wyniki Zadania 1; $m = 63$

	EA		CPLEX
	$W = 0\%$	$W = 100\%$	$W = 0\%$
$G_1(l)$	5	5	5
$G_1(m)$	10	10	10
$G_1(h)$	26	27	26
G_2	37	38	37
G_3	18	18	18
G_4	306	311	—

 Tabela 3. Wyniki Zadania 2; $m = 63$

W przypadku zabezpieczania sieci istniejącej (Zadanie 2) za pomocą metody EA (przy pełnej dezagregacji) uzyskano optymalne wyniki. Wyjątek stanowi G_4 , dla której uzyskano wynik tylko

za pomocą metody EA. W przypadku pełnej agregacji ($W = 100\%$) w niektórych przypadkach wyniki są gorsze.

Algorytm Ewolucyjny uruchamiany był przy następujących parametrach: $\mu = 100$, $\lambda = 300$, prawdopodobieństwo zajęcia krzyżowania wynosi 0.75, a prawdopodobieństwo mutacji 0.05. Algorytm jest zatrzymywany, jeśli po 100 kolejnych generacjach nie ma poprawy funkcji celu. Obliczenia przeprowadzono na komputerze firmy SUN (procesor Supersparc 36 MHz), pod systemem Solaris 2.5.

5 Wnioski końcowe

Wyniki optymalizacji przepływu wielotowarowego w sieci telekomunikacyjnej przedstawione w tym artykule i w [4], [5] potwierdzają skuteczność metody EA, która może być użyta w procesie efektywnego projektowania sieci telekomunikacyjnych. Oczywiście metoda EA, jak każde stochastyczne podejście do optymalizacji dyskretnej (kombinatorycznej), nie gwarantuje osiągnięcia rozwiązań optymalnych w rozsądnym czasie. Uzyskiwane wyniki oraz ich porównanie z innymi metodami świadczą jednak o zdolności proponowanej metody do dostarczenia dobrych rozwiązań suboptymalnych, przy krótkich czasach obliczeń. Co więcej, metoda EA pozwala w łatwy sposób uwzględnić następujące dodatkowe wymagania:

- agregacja jednostek zapotrzebowania DCU,
- nieliniowość funkcji kosztu i ograniczeń,
- udostępnienie wszystkich możliwych ścieżek dla realizacji przepływów.

Powyższe wymagania są niemożliwe do uwzględnienia w podejściu opartym na Programowaniu Liniowym, nawet z uwzględnieniem całkowitoliczbowości zmiennych. Warto podkreślić, że dla dużych sieci (o dużej liczbie węzłów — G_4) metoda podziału i ograniczeń zaimplementowana w pakiecie CPLEX, jest czasochłonna, wymaga dużo pamięci operacyjnej i nie zawsze znajduje rozwiązanie w akceptowalnym czasie.

Bibliografia

- [1] Arabas, J. *Algorytmy ewolucyjne ze zmienną liczebnością populacji i zmiennym zasięgiem krzyżowania*, Rozprawa Doktorska, Politechnika Warszawska, 1995.
- [2] *Using the CPLEX Callable Library*, CPLEX Optimization, Inc., 1994.
- [3] Garey, M.R., Johnson, D.S.: *Computers and Intractability - A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.
- [4] Kozdrowski, S.: "Design of Telecommunication Networks by Evolutionary Strategy", I Krajowa Konferencja Algorytmów Ewolucyjnych, Murzasichle, 1996.
- [5] Kozdrowski, S., Pióro, M., Arabas, J., Szcześniak, M.: "Robust Design of Multicommodity Integral Flow Networks" *Proceedings of the 7th International Conference on Genetic Algorithms*, East Lansing, University of Michigan, USA 1997.
- [6] Lubacz, J., Tomaszewski, A.: "A Lower Bound Based Approach to the Design of Meshed Networks with Protected Transmission Paths", *Proceedings of the 14th International Teletraffic Congress*, North Holland, 1995.
- [7] Michalewicz, Z., *Algorytmy Ewolucyjne + Struktury Danych = Programowanie Ewolucyjne*, Springer, 1996.
- [8] Pióro, M., Gajowniczek, P.: "Simulated Allocation - a Suboptimal Solution to the Multicommodity Flow Problem", *Proc. Eleventh UK Teletraffic Symposium*, Cambridge, 1994.
- [9] Pióro, M., Szcześniak, M.: "Robust Design Problems in Telecommunication Networks", *ITC Specialists Seminar on Control in Communications*, Lund, 1996.
- [10] Wu, T.: *Fiber Network Service Survivability*, Artech House, 1992.