

Stanisław KRENICH¹

OPTYMALNA ALOKACJA OBIEKTÓW Z WYKORZYSTANIEM ALGORYTMÓW EWOLUCYJNYCH

W artykule opisano metodę rozwiązywania zagadnienia optymalnej alokacji obiektów z wykorzystaniem algorytmów ewolucyjnych. Przedstawiono podejście wykorzystujące teorię sieci jako narzędzia do opisu zagadnienia alokacji podając sposób zapisu problemu o własnościach lub bez własności Markowa. Jednocześnie na przykładach zdefiniowano i opisano problem optymalizacji sieci jedno lub wielokryterialnej. Następnie przeprowadzono obliczenia optymalizacyjne dla dwóch przykładów jedno i dwukryterialnej alokacji obiektów opisanych sieciami odpowiednio bez własności i z własnościami Markowa. Wyniki porównano z metodą przeszukiwania systematycznego. Uzyskane wyniki dają podstawę do stwierdzenia dużej przydatności algorytmów ewolucyjnych w rozwiązywaniu problemów alokacji.

OPTIMUM OBJECTS ALLOCATION USING EVOLUTIONARY ALGORITHMS

The paper presents an approach to optimal allocation of objects (devices). Based on the network theory the single and multicriteria allocation procedure, including Markov and non-Markov pattern of networks, was introduced. In order to solve the problem an evolutionary algorithms based method was proposed. The results obtained for two different examples of allocation were compared with the systematic search method and they indicate that the proposed approach is going to be a very efficient method for solving wide range of allocation problems.

1. WSTĘP

Jednym z bardziej racjonalnych podejść do problemu alokacji obiektów jest wykorzystanie teorii sieci, w której możliwe rozwiązania reprezentowane są przez łuki sieci, natomiast optymalna alokacja sprowadza się do znalezienia najlepszej drogi w sieci. Sieci reprezentujące problem alokacji mogą mieć różny charakter tj. mogą wystąpić sieci posiadające tzw. własność Markowa lub bez tej własności, alokacja może się odbywać na podstawie jednego kryterium lub wielu kryteriów dając w efekcie odpowiednio jedno rozwiązanie lub zbiór rozwiązań Pareto optymalnych [3]. Jest to problem trudny do rozwiązania tradycyjnymi metodami optymalizacji, szczególnie dla problemów opisanych

¹ Instytut Technologii Maszyn i Automatyzacji Produkcji, Politechnika Krakowska,
Al. Jana Pawła II 37, 31-864 Kraków, tel. 12-374-32-18, email: krenich@mech.pk.edu.pl

sieciami bez własności Markowa, o dużej liczbie potencjalnych kombinacji rozwiązań. W artykule przedstawiono metodę rozwiązania problemu optymalnej alokacji przy wykorzystaniu algorytmów ewolucyjnych [1, 2]. Algorytm ewolucyjny generuje populację rozwiązań i ocenia je na podstawie funkcji dopasowania. W przypadku sieci generuje drogę a suma obciążeń łuków sieci stanowi funkcję dopasowania. Stąd dla algorytmu nie ma zupełnie znaczenia fakt czy sieć ma własność Markowa, czy dotyczy to optymalizacji jedno czy wielokryterialnej. Należy tu zwrócić uwagę na fakt, że jakkolwiek poszukiwanie dróg Pareto optymalnych w sieci, dla sieci posiadającej własność Markowa, jest możliwe przy wykorzystaniu metody programowania dynamicznego to jego realizacja komputerowa także jest trudna do realizacji gdyż wymaga dużych pamięci komputera i jest nieefektywna ze względu na czas obliczeń.

2. OGÓLNE SFORMUŁOWANIE PROBLEMU OPTIMALIZACJI SIECI

2.1 Definicja problemu

Niech $S = \langle X, \bar{R}, f \rangle$ oznacza sieć, której graf $G = \langle X, \bar{R} \rangle$ należy do klasy grafów bez pętli, gdzie:

X - zbiór wierzchołków grafu,

\bar{R} - relacja dwuargumentowa określona na zbiorze X ,

f - funkcja wektorowa opisana na łukach grafu.

Założmy, że zbiór wszystkich wierzchołków grafu G można podzielić na podzbiory $X_1, \dots, X_i, \dots, X_I$, z których i -ty podzbiór zawiera wierzchołki $x_{i_1}, \dots, x_{i_{e_i}}, \dots, x_{i_{E_i}}$. Dla tak uporządkowanych i oznaczonych wierzchołków grafu G wymagane jest, aby były spełnione następujące funkcje zdaniowe:

$$\forall 1 \leq i \leq I-1 \quad \forall 1 \leq e_i \leq E_i \quad \forall 1 \leq e_{i+1} \leq E_{i+1} \quad (x_{i,e_i} \bar{R} x_{i+1,e_{i+1}}) \quad (2.1)$$

$$\exists i+1 \leq i+p \leq I \quad \forall 1 \leq e_i \leq E_i \quad \forall 1 \leq e_{i+p} \leq E_{i+p} \quad (x_{i,e_i} \bar{R} x_{i+p,e_{i+p}}) \quad (2.2)$$

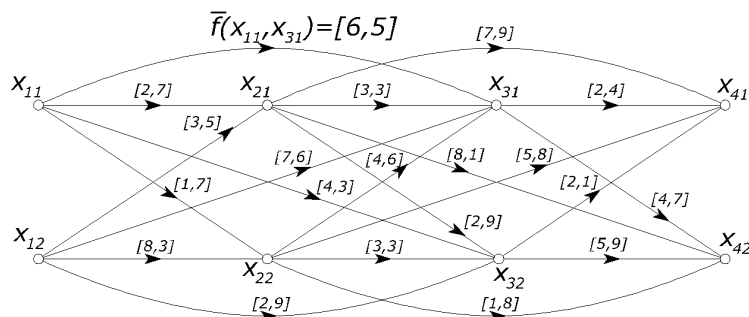
Wzór (2.1) oznacza, że muszą istnieć relacje między kolejnymi podzbiarami wierzchołków tj. X_1 - X_2 , X_2 - X_3 , ..., X_{I-1} , X_I . Natomiast wzór (2.2) oznacza, że mogą istnieć również relacje między oddalonymi od siebie zbiorami np. X_1 - X_3 , X_2 - X_5 itp. Sieć posiadająca takie relacje traci własność Markowa. Dodatkowo założmy, że zbiór X_1 posiada tylko łuki wychodzące z niego, natomiast zbiór X_I posiada tylko łuki wchodzące do niego. Niech

$$f(x_{i,e_i}, x_{i+p,e_{i+p}}) = [f_1(x_{i,e_i}, x_{i+p,e_{i+p}}), f_2(x_{i,e_i}, x_{i+p,e_{i+p}}), \dots, f_K(x_{i,e_i}, x_{i+p,e_{i+p}})]^T \quad (2.3)$$

będzie funkcją wektorową opisaną na łukach grafu G . Wówczas zadanie optymalizacji sieci sprowadza się do znalezienia takiego zbioru wierzchołków $\bar{x}^* = [x_{1,e_1}^*, x_{2,e_2}^*, \dots, x_{I,e_I}^*]$, dla którego suma obciążeń łuków sieci jest minimalna. Dla k -tego kryterium suma ta jest wyznaczana z wzoru:

$$F_k = \sum_{i=1}^{i=I-1} \sum_{p=1}^{p=I-i} f_k(x_{i,e_i}, x_{i+p,e_{i+p}}) \quad \text{dla } k = 1, \dots, K \quad (2.4)$$

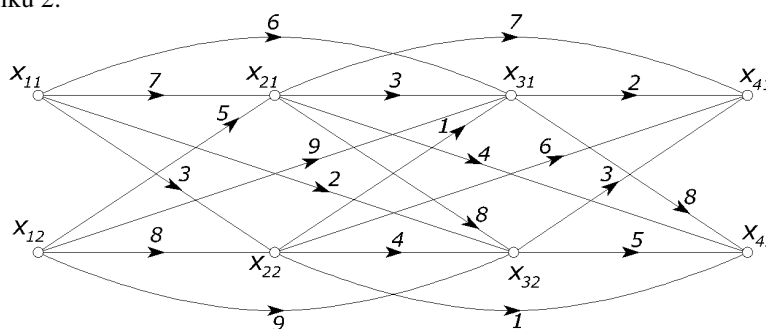
Ten złożony zapis formalny ma łatwą interpretację graficzną, którą przedstawiono na rysunku 1.



Rys. 1. Ogólny przypadek sieci bez własności Markowa i dwóch kryteriach oceny

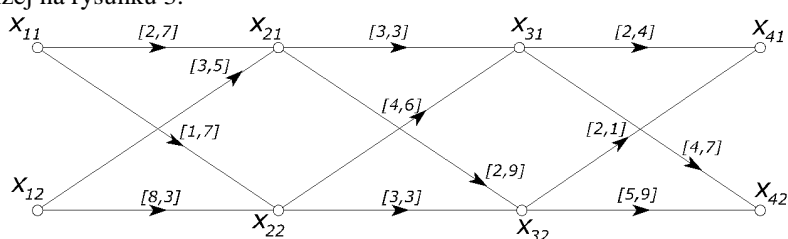
Przypadki szczególne:

1. $K=1$ oznacza, że mamy tylko jedno kryterium i mamy do czynienia z optymalizacją jednokryterialną sieci bez własności Markowa. Przykład takiej sieci podano poniżej na rysunku 2.



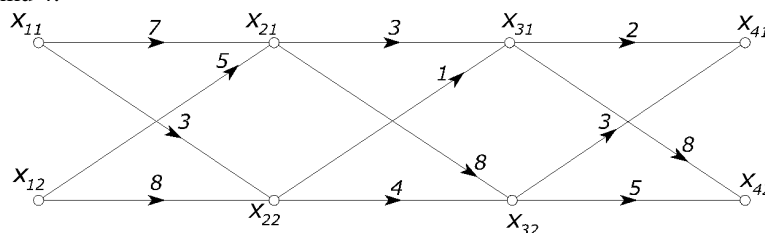
Rys. 2 Przypadek sieci bez własności Markowa i jednym kryterium oceny

2. Zależność (2.2) nie występuje i $K > 1$ oznacza, że mamy do czynienia z optymalizacją wielokryterialną sieci posiadającej własność Markowa. Przykład takiej sieci podano poniżej na rysunku 3.



Rys.3. Przypadek sieci mającej własność Markowa i wiele kryteriów

3. Zależność (2.2) nie występuje i $K=1$ oznacza, że mamy do czynienia z optymalizacją jednokryterialną sieci posiadającą własność Markowa. Jest to typowy problem poszukiwania najkrótszej drogi w sieci. Przykład takiej sieci podano poniżej na rysunku 4.



Rys. 4. Przypadek sieci posiadającej własność Markowa i jednym kryterium

Rozwiązanie problemu ogólnego pozwala ująć wszystkie podane powyżej trzy przypadki. Należy zwrócić uwagę na fakt, że nie istnieją dotychczas algorytmy pozwalające na ogólne rozwiązanie zadań optymalizacji sieci bez własności Markowa.

2.2 Modelowanie problemu alokacji obiektów metodą sieciową

Poniżej przedstawiono sposób zapisu problemu alokacji obiektów metodą sieciową [3], umożliwiając późniejsze rozwiązanie z wykorzystaniem algorytmów ewolucyjnych [1, 2]. Metodę tą można opisać następująco - dowolną sieć mającą własność Markowa można opisać następującymi parametrami:

- K – liczba kryteriów decyzyjnych (odpowiada ilości funkcji dopasowania F). Jeśli $K=1$ to mamy do czynienia z optymalizacją jednokryterialną, jeśli $K>1$ to z optymalizacją wielokryterialną. Zmienna określająca numer kryterium: k .
- L – liczba etapów występujących w sieci (odpowiada ilości zmiennych decyzyjnych X). Zmienna określająca numer etapu: l .
- $I[l]$ – wektor $I[0, \dots, l, \dots, L]$ zawiera liczbę stanów sieci na danym etapie.
- $M[K, L, I[l], I[l+1]]$ – macierz obciążenia łuków sieci. W postaci ogólnej $M[k, l, i, j]$ zawiera wartość liczbową łuku dla kryterium $k \in (0 \dots K)$, etapu $l \in (0 \dots L)$, stanu z bieżącego etapu $i \in (0 \dots I[l])$ – numer wierzchołka z którego wychodzi krawędź i stanu docelowego w etapie następnym $j \in (0 \dots I[l+1])$ – numer wierzchołka do którego krawędź dochodzi.

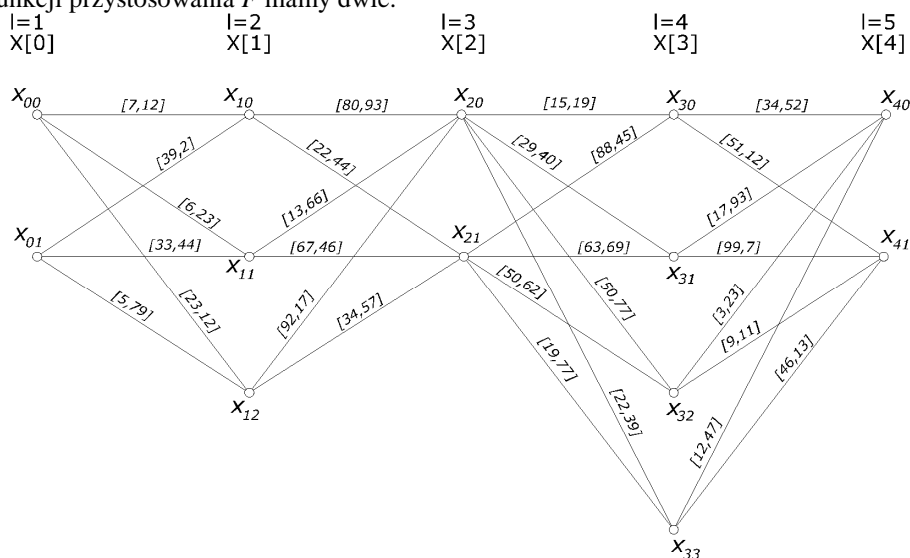
Poprzez dodanie dodatkowych krawędzi pomiędzy dalszymi etapami (pomijając pośrednie) otrzymujemy sieć, która traci własność Markowa. Do opisu tego typu sieci wykorzystujemy wcześniej przedstawiony sposób definiowania prostych sieci oraz musimy dodatkowo zdefiniować te krawędzie, które powodują utratę własności Markowa. W tym celu należy określić następujące parametry:

- m – numer etapu, z którego krawędź wychodzi,
- sl – numer stanu, z którego krawędź wychodzi,
- n – numer etapu, do którego krawędź dochodzi,
- el – numer stanu, do którego krawędź dochodzi,
- $N[k, m, sl, n, el]$ – macierz 5-wymiarowa opisująca sposób połączeń i obciążeń łuków dla dodatkowych krawędzi.

- E – ilość dodatkowych krawędzi.

2.3 Przykład zapisu sieci posiadającej własność Markowa

Na rysunku 5 pokazano przykład prostej sieci mającej własność Markowa. Sieć posiada pięć etapów $L=5$ oraz dwa kryteria $K=2$. Stąd ilość zmiennych decyzyjnych X wynosi pięć a funkcji przystosowania F mamy dwie.



Rys. 5. Przykład prostej sieci posiadającej własność Markowa

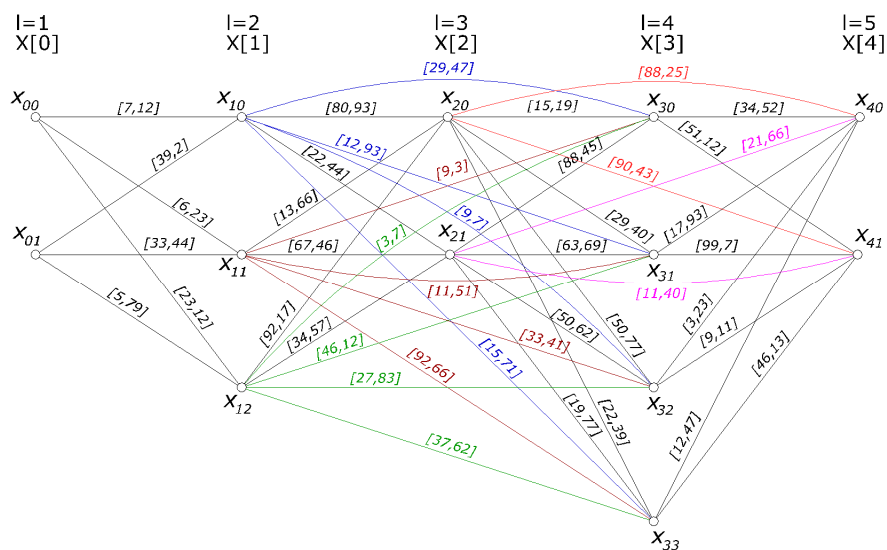
Dla kompletnego opisu takiej sieci należy podać, oprócz K i L , wektor ilości stanów, który dla naszego przykładu wygląda następująco²: $I[0]=2$, $I[1]=3$, $I[2]=2$, $I[3]=4$, $I[4]=2$ oraz macierz M , opisującą w przypadku ogólnym wektorowe obciążenia poszczególnych krawędzi. Ze względu na graficzną trudność przedstawienia macierzy 4-ro wymiarowej przyjęto przedstawiać przekroje 2-wymiarowe dla poszczególnych kryteriów i etapów:

² Uwaga: stosowanie indeksu zero wynika z późniejszego zapisu formalnego w języku ANSI C

$$\begin{array}{cccc}
 M[0,0,i,j] & M[0,1,i,j] & M[0,2,i,j] & M[0,3,i,j] \\
 \begin{bmatrix} 7 & 6 & 23 \\ 39 & 33 & 5 \end{bmatrix} & \begin{bmatrix} 80 & 22 \\ 13 & 67 \\ 92 & 34 \end{bmatrix} & \begin{bmatrix} 15 & 29 & 50 & 22 \\ 88 & 63 & 50 & 19 \end{bmatrix} & \begin{bmatrix} 34 & 51 \\ 17 & 99 \\ 3 & 9 \\ 12 & 46 \end{bmatrix} \\
 M[1,0,i,j] & M[1,1,i,j] & M[1,2,i,j] & M[1,3,i,j] \\
 \begin{bmatrix} 12 & 23 & 12 \\ 2 & 44 & 79 \end{bmatrix} & \begin{bmatrix} 93 & 44 \\ 66 & 46 \\ 17 & 57 \end{bmatrix} & \begin{bmatrix} 19 & 40 & 77 & 39 \\ 45 & 69 & 62 & 77 \end{bmatrix} & \begin{bmatrix} 52 & 12 \\ 93 & 7 \\ 23 & 11 \\ 47 & 13 \end{bmatrix}
 \end{array}$$

2.4 Przykład zapisu sieci bez własności Markowa

Na rysunku 6 pokazano przykład prostej sieci nie mającej własności Markowa. Sieć posiada pięć etapów $L=5$ oraz dwa kryteria $K=2$. Sieć ta jest modyfikacją sieci z przykładu wcześniejszego.



Rys.6. Prosta sieć bez własności Markowa

Podany sposób zapisu jest podejściem bardzo elastycznym i daje możliwość definiowania dowolnej liczby dodatkowych krawędzi pomiędzy dowolnie odległymi stanami. Ilość dodanych krawędzi $E=16$. Ze względu na 5-wymiarowy charakter macierzy oraz na niepełne jej wykorzystanie przyjęto przedstawić macierz N jako zbiór wektorów dla poszczególnych dodanych krawędzi w sposób następujący:

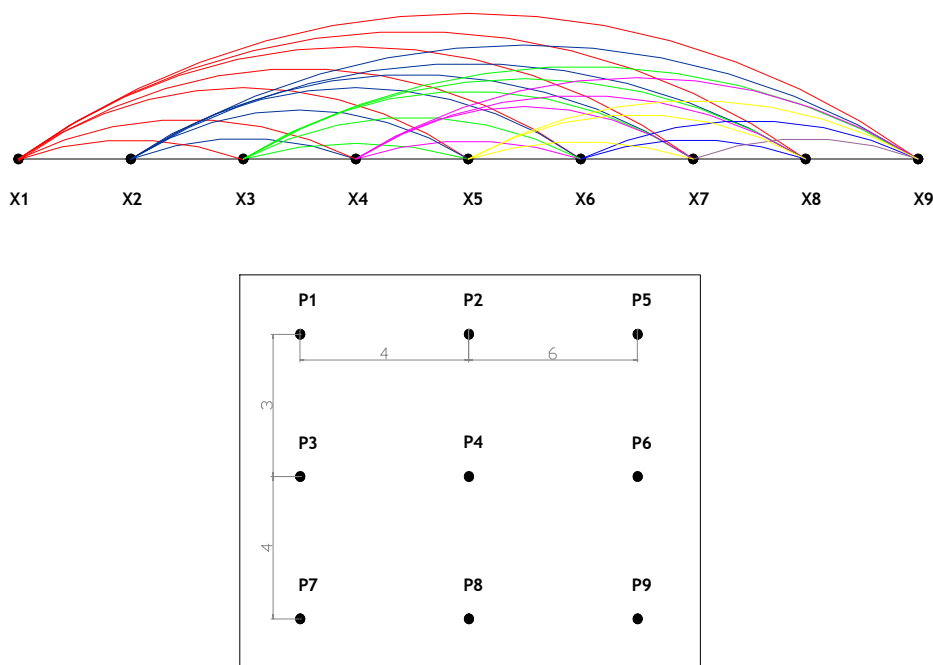
$N[0, 1, 0, 3, 0] = 29,$	$N[1, 1, 1, 3, 1] = 51,$	$N[0, 1, 2, 3, 3] = 37,$
$N[1, 1, 0, 3, 0] = 47,$	$N[0, 1, 1, 3, 2] = 33,$	$N[1, 1, 2, 3, 3] = 62,$
$N[0, 1, 0, 3, 1] = 12,$	$N[1, 1, 1, 3, 2] = 41,$	$N[0, 2, 0, 4, 0] = 88,$
$N[1, 1, 0, 3, 1] = 93,$	$N[0, 1, 1, 3, 3] = 92,$	$N[1, 2, 0, 4, 0] = 25,$
$N[0, 1, 0, 3, 2] = 9,$	$N[1, 1, 1, 3, 3] = 66,$	$N[0, 2, 0, 4, 1] = 90,$
$N[1, 1, 0, 3, 2] = 7,$	$N[0, 1, 2, 3, 0] = 3,$	$N[1, 2, 0, 4, 1] = 43,$
$N[0, 1, 0, 3, 3] = 15,$	$N[1, 1, 2, 3, 0] = 7,$	$N[0, 2, 1, 4, 0] = 21,$
$N[1, 1, 0, 3, 3] = 71,$	$N[0, 1, 2, 3, 1] = 46,$	$N[1, 2, 1, 4, 0] = 65,$
$N[0, 1, 1, 3, 0] = 9,$	$N[1, 1, 2, 3, 1] = 12,$	$N[0, 2, 1, 4, 1] = 11,$
$N[1, 1, 1, 3, 0] = 3,$	$N[0, 1, 2, 3, 2] = 27,$	$N[1, 2, 1, 4, 1] = 40.$
$N[0, 1, 1, 3, 1] = 11,$	$N[1, 1, 2, 3, 2] = 83,$	

3. JEDNOKRYTERIALNA ALOKACJA OBIEKTÓW OPISANYCH SIECIAMI BEZ WŁASNOŚCI MARKOWA.

Rozpatrzony będzie problem alokacji 9 różnych urządzeń technologicznych X_i dla $i=1...9$ (np. obrabiarek) w 9 różnych punktach hali produkcyjnej P_j dla $j=1...9$ w taki sposób aby droga jaką pokonuje obrabiany przedmiot była najkrótsza. Każde urządzenie X_i może zostać rozmieszczone w dowolnym punkcie P_j , co daje następujące alokacje:

- X_1 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_2 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_3 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_4 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_5 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_6 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_7 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_8 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$,
- X_9 na $\{P1, P2, P3, P4, P5, P6, P7, P8, P9\}$.

Przejścia między punktami stanowią tzw. drogę, która może obrazować wartości różnych kryteriów, może przedstawiać np. koszty wytworzenia, koszty przebrojenia, koszty transportu, czas przebiegu czynności technologicznych lub transportowych itd. W rozpatrywanym przykładzie przyjęto, iż droga obrazuje koszty wytworzenia tak, więc znajdując najkrótszą możliwą drogę znajdziemy najniższy koszt wytworzenia.



Rys. 7. Model sieci 9x9 i możliwa lokalizacja urządzeń na hali

Przedstawiony problem można opisać za pomocą sieci bez własności Markowa, w której liczba możliwych kombinacji dróg, jakie może odbyć przemieszczany przedmiot wynosi 9^9 czyli 387 420 489. Schemat możliwych rozmieszczeń urządzeń na hali i odległości pomiędzy nimi, oraz zdefiniowanie dodatkowych dodanych krawędzi $E=2268$ (powodujących utratę przez sieć własności Markowa) zaznaczonych kolorowymi łukami obrazuje rysunek 7. Dla sieci nałożono ograniczenia, które opisują brak możliwości realizacji dowolnych alokacji obiektów a tym samym realizacji dowolnych dróg przepływu przedmiotu. W przykładzie ograniczenia obejmują brak możliwości umieszczania różnych urządzeń X_i w tym samym miejscu P_j . Dla tak sformułowanego zadania zastosowano algorytm przeszukiwania systematycznego każdej kombinacji oraz algorytm ewolucyjny [1, 2]. Algorytm przeszukiwania systematycznego znalazł 41 możliwych alokacji urządzeń a więc jednocześnie możliwych dróg przepływu przedmiotu dla minimum kryterium równego 252.6 [j]. Czas wygenerowania rozwiązania wyniósł 20.39 [h]. Poszukiwanie rozwiązania optymalnego algorytmem ewolucyjnym dawało praktycznie w każdym przebiegu jedno z 41 rozwiązań optymalnych, zależnych wyłącznie od jego parametrów startowych. W każdym przypadku czas obliczeń wynosił od kilku do kilkudziesięciu sekund. Wydruk wyników przykładowego przebiegu algorytmu ewolucyjnego przedstawiono w Tabeli 1.

Tab. 1. Wyniki uzyskane podczas przebiegu algorytmu ewolucyjnego i przeszukiwania systematycznego dla tego samego rozwiązania

Przeszukiwanie systematyczne (wybrane rozwiązanie)	KRYTERIUM: 252.60 [j] Rozwiązanie: OBIEKT: X1 X2 X3 X4 X5 X6 X7 X8 X9 PUNKT : P6 P3 P8 P4 P1 P7 P5 P2 P9 Czas obliczeń: 1223.15[min.](20.39 h)
Parametry algorytmu ewolucyjnego	Tournament Selection Multipoint Crossover Uniform Mutation Population size = 50 Number of generations = 20 Probability of crossover = 0.60 Probability of mutation = 0.01 Penalty parameter = 1.000000e+004 Seed = 0
Wygenerowane wyniki	Last improvement generation = 6 Best fitness value = 252.600000 F = 252.600000 X[0] = 5.000000 X[1] = 2.000000 X[2] = 7.000000 X[3] = 3.000000 X[4] = 0.000000 X[5] = 6.000000 X[6] = 4.000000 X[7] = 1.000000 X[8] = 8.000000 Time = 14.7 [s] *indeks i w X[i] oraz uzyskane wyniki należy zwiększyć o 1

4. DWUKRYTERIALNA ALOKACJA OBIEKTÓW OPISANYCH SIECIAMI POSIADAJĄCYMI WŁASNOŚĆ MARKOWA

Alokacja obiektów została opisana siecią posiadającą własność Markowa z dwoma kryteriami optymalizacji $K=2$ i o ośmiu etapach $L=8$. Ilość stanów na poszczególnych etapach: $I[0]=3$, $I[1]=4$, $I[2]=3$, $I[3]=5$, $I[4]=3$, $I[5]=3$, $I[6]=4$, $I[7]=3$. Opis połączeń w sieci i wartości łuków sieci zawiera macierz M , która wygląda następująco:

$$\begin{array}{cccc}
 M[0,0,i,j] & M[0,1,i,j] & M[0,2,i,j] & M[0,3,i,j] \\
 \begin{bmatrix} 41 & 67 & 134 & 100 \\ 169 & 124 & 78 & 158 \\ 162 & 64 & 105 & 145 \end{bmatrix} & \begin{bmatrix} 81 & 27 & 161 \\ 91 & 195 & 142 \\ 27 & 36 & 191 \\ 4 & 102 & 153 \end{bmatrix} & \begin{bmatrix} 92 & 182 & 21 & 116 & 118 \\ 95 & 47 & 126 & 171 & 138 \\ 69 & 112 & 67 & 99 & 35 \end{bmatrix} & \begin{bmatrix} 94 & 103 & 11 \\ 122 & 133 & 73 \\ 64 & 141 & 111 \\ 53 & 68 & 147 \\ 44 & 62 & 157 \end{bmatrix}
 \end{array}$$

$$\begin{array}{cccc}
 M[0,4,i,j] & M[0,5,i,j] & M[0,6,i,j] & M[1,0,i,j] \\
 \begin{bmatrix} 37 & 59 & 123 \\ 141 & 129 & 178 \\ 116 & 35 & 190 \end{bmatrix} & \begin{bmatrix} 42 & 88 & 106 & 40 \\ 142 & 64 & 48 & 46 \\ 5 & 90 & 129 & 170 \end{bmatrix} & \begin{bmatrix} 150 & 6 & 101 \\ 193 & 148 & 29 \\ 23 & 84 & 154 \\ 156 & 40 & 166 \end{bmatrix} & \begin{bmatrix} 176 & 131 & 108 & 144 \\ 39 & 26 & 123 & 137 \\ 138 & 118 & 82 & 129 \end{bmatrix}
 \end{array}$$

$$\begin{array}{cccc}
 M[1,1,i,j] & M[1,2,i,j] & M[1,3,i,j] & M[1,4,i,j] \\
 \begin{bmatrix} 141 & 33 & 115 \\ 39 & 58 & 104 \\ 130 & 177 & 106 \\ 73 & 186 & 21 \end{bmatrix} & \begin{bmatrix} 145 & 124 & 72 & 70 & 29 \\ 177 & 173 & 97 & 112 & 186 \\ 90 & 161 & 36 & 155 & 167 \end{bmatrix} & \begin{bmatrix} 55 & 174 & 31 \\ 52 & 150 & 150 \\ 141 & 124 & 166 \\ 30 & 107 & 191 \\ 7 & 137 & 57 \end{bmatrix} & \begin{bmatrix} 87 & 153 & 183 \\ 145 & 109 & 9 \\ 158 & 21 & 188 \end{bmatrix}
 \end{array}$$

$$\begin{array}{cc}
 M[1,5,i,j] & M[1,6,i,j] \\
 \begin{bmatrix} 22 & 146 & 106 & 30 \\ 13 & 168 & 100 & 191 \\ 162 & 55 & 10 & 159 \end{bmatrix} & \begin{bmatrix} 24 & 137 & 148 \\ 83 & 195 & 41 \\ 2 & 150 & 91 \\ 36 & 174 & 20 \end{bmatrix}
 \end{array}$$

Celem porównania metod dla sieci wygenerowano wszystkie możliwe drogi (19 440 kombinacji) obliczono wartości funkcji $F1$ i $F2$ oraz wyznaczono zbiór Pareto. Następnie dokonano optymalizacji tej sieci algorytmem ewolucyjnym i otrzymano zbiór Pareto 14 elementowy, który pokrywa się z wyliczonym metodą przeszukiwania systematycznego, przy czym czas generowania zbioru Pareto algorytmem ewolucyjnym sięgał kilku procent

czasu potrzebnego na systematyczne przeszukanie. Wybrane rozwiązania ze zbioru Pareto ustawione wg rosnącej wartości pierwszego kryterium przedstawiono w tabeli 2.

Tab. 2. Wybrane rozwiązania ze zbioru Pareto

PARETO OPTIMAL SOLUTION NO.1				
VALUE OF 1 OBJECTIVE FUNCTION		= 2.740000e+002		
VALUE OF 2 OBJECTIVE FUNCTION		= 6.760000e+002		
VECTOR OF DECISION VARIABLES				
0.000e+000	3.000e+000	0.000e+000	2.000e+000	0.000e+000
0.000e+000	0.000e+000	1.000e+000		
PARETO OPTIMAL SOLUTION NO.7				
VALUE OF 1 OBJECTIVE FUNCTION		= 4.020000e+002		
VALUE OF 2 OBJECTIVE FUNCTION		= 4.390000e+002		
VECTOR OF DECISION VARIABLES				
2.000e+000	1.000e+000	0.000e+000	4.000e+000	0.000e+000
0.000e+000	0.000e+000	1.000e+000		
PARETO OPTIMAL SOLUTION NO.14				
VALUE OF 1 OBJECTIVE FUNCTION		= 8.170000e+002		
VALUE OF 2 OBJECTIVE FUNCTION		= 2.090000e+002		
VECTOR OF DECISION VARIABLES				
1.000e+000	1.000e+000	0.000e+000	4.000e+000	2.000e+000
1.000e+000	0.000e+000	0.000e+000		

5. PODSUMOWANIE

Przeprowadzone obliczenia na wybranych przykładach pozwalają stwierdzić, iż modelowanie problemu alokacji obiektów metodą sieciową daje możliwość jego rozwiązania jako problemu optymalizacyjnego z jednym lub wieloma kryteriami. Optymalizacja sprowadza się tu do wyboru najlepszej drogi przejścia w sieci. Tradycyjne metody przeszukiwania systematycznego lub metody programowania dynamicznego w wielu, szczególnie bardziej złożonych przypadkach zawodzą ze względu na bardzo długi czas lub niską dokładność obliczeń. Uzyskane wyniki obliczeń z zastosowaniem algorytmów ewolucyjnych dają natomiast podstawę do stwierdzenia ich dużej przydatności w rozwiązywaniu tego typu problemów, zarówno jedno jak i wielokryterialnej alokacji obiektów, przy czym zalety algorytmów ewolucyjnych są szczególnie widoczne dla bardziej złożonych problemów. Dla problemów o małych rozmiarach sieci niezasadne jest stosowanie tego narzędzia, gdyż podobne wyniki otrzymamy szybciej innymi metodami.

6. BIBLIOGRAFIA

- [1] Osyczka A.: *Evolutionary Algorithms for Single and Multicriteria Design Optimization*. Springer – Verlag Physica, Berlin Heidelberg 2001.
- [2] Osyczka A., Krenich S.: *Evolutionary Algorithms for Global Optimization*, Chapter in: J. Pinter (Ed.) *Global Optimization - Selected Case Studies*, Kluwer Academic Publishers, Dordrecht/Boston/London, 2007.
- [3] Osyczka A., Krenich S., Habel J.: *Algorytmy genetyczne w optymalizacji konstrukcji*. Projekt Badawczy PB-1429/T07/98/14.