

OBLICZENIA RÓWNOLEGŁE W ZAGADNIENIACH POLA TEMPERATURY W NANOUKŁADACH ELEKTRONICZNYCH

Anna GRYKO-NIKITIN

Streszczenie: W artykule omówiono możliwość zastosowania obliczeń równoległych w zagadnieniach optymalizacji pola temperatury w nanoukładach elektronicznych. Na wstępie podano koncepcję autorskiego modelu systemu wspomagającego badanie zagadnień pola temperatury wskazując jednocześnie miejsce dla obliczeń równoległych. Na potrzeby pracy podano definicję obliczeń równoległych oraz omówiono korzyści wynikające ze stosowania tego podejścia. Ze względu na złożony charakter tematu rozważania ograniczono do poszukiwania optymalnego rozkładu elementów elektronicznych w obrębie nanoukładu elektronicznego co zdaniem Autorki stanowi punkt wyjścia w modelowaniu pola temperatury współczesnych układów elektronicznych. W tym celu podano szczegółową charakterystykę jednego z modułów opracowanego systemu: Modułu Wstępnej Alokacji Elementów Elektronicznych (Generator), w którym zaimplementowano równoległy algorytm genetyczny.

Słowa kluczowe: obliczenia równoległe, algorytmy genetyczne, optymalizacja, alokacja zasobów.

1. Wstęp

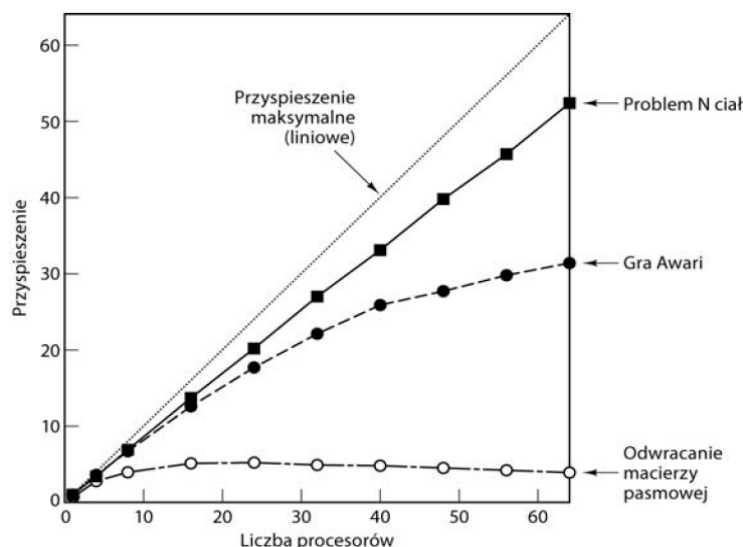
W literaturze przedmiotu można spotkać wiele definicji obliczeń równoległych. Na potrzeby pracy przyjęto, że z obliczeniami równoległymi mamy do czynienia, gdy na wielu procesorach tej samej lub wielu maszynach programy i procesy wykonują się jednocześnie [1].

Wśród najważniejszych zalet obliczeń równoległych podaje się m.in.:

- (i) przyśpieszenie obliczeń,
- (ii) stworzenie możliwości rozwiązywania zadań o dużej wymiarowości,
- (iii) umożliwienie wykorzystania pełnej mocy procesorów wielordzeniowych.

Stopień przyśpieszenia obliczeń w dużej mierze zależy od specyfiki samego programu. Wynika to z faktu, że każdy program równoległy posiada w swym ciele fragmenty (frakcje) ściśle sekwencyjne, których nie da się wykonać inaczej niż sekwencyjnie [2] np.: inicjowanie zmiennych.

W konsekwencji, niektóre programy mogą być odporne na zwiększanie liczby procesorów (przypadek odwracania macierzy pasmowej), dla innych zaś obserwujemy wzrost przyśpieszenia tylko do pewnej granicy (tzw. problem N ciał), po czym zwiększanie liczby procesorów nie wpływa znacząco na przyśpieszenie obliczeń (rys. 1) [3].



Rys. 1. Przyspieszenie szybkości obliczeń trzech różnych typów programów wskutek zwiększania liczby procesorów

Do grupy zadań o dużej skali złożoności należą problemy przestrzennej alokacji zasobów. Pod pojęciem przestrzennej alokacji zasobów należy rozumieć wszelkie zagadnienia, w których poszukujemy optymalnego, ze względu na przyjęte kryteria, rozmieszczenia określonej liczby zasobów. Znamienne instance tego problemu to m.in.: kwestia rozmieszczenia czujników skanujących, alokacja stacji bazowych telefonii komórkowej, czy też określenie lokalizacji hurtowni, bądź oddziałów straży pożarnej [4].

W artykule zaprezentowano problem optymalnego rozmieszczenia elementów elektronicznych w obrębie nanoukładu elektronicznego jako przykład przestrzennej alokacji zasobów. Poruszony problem jest istotny ze względu na wytwarzanie przez układy coraz większej ilości ciepła, co spowodowane jest wzrostem liczby elementów umieszczanych na układach. Postęp technologiczny polegający na miniaturyzacji pozwala na umieszczeniu coraz większej liczby elementów elektronicznych w obrębie jednego układu przy jednoczesnym zmniejszaniu rozmiarów samego układu. Na przestrzeni lat obserwujemy wzrost gęstości strat energii zamienianej na ciepło w ciągu jednej sekundy z 0.5 W/cm^2 do ponad 100 W/cm^2 . Wzrost strumienia ciepła bezpośrednio przekłada się na niezawodność układów, stąd też konieczność prowadzenia symulacji termicznych [5].

Gęstość upakowania elementów elektronicznych w obrębie jednego układu osiągnęła obecnie skalę utrudniającą symulację pracy układu elektronicznego zarówno ze względu na moce obliczeniowe jak i złożoność czasową, stąd koncepcja wykorzystania w tym celu obliczeń równoległych.

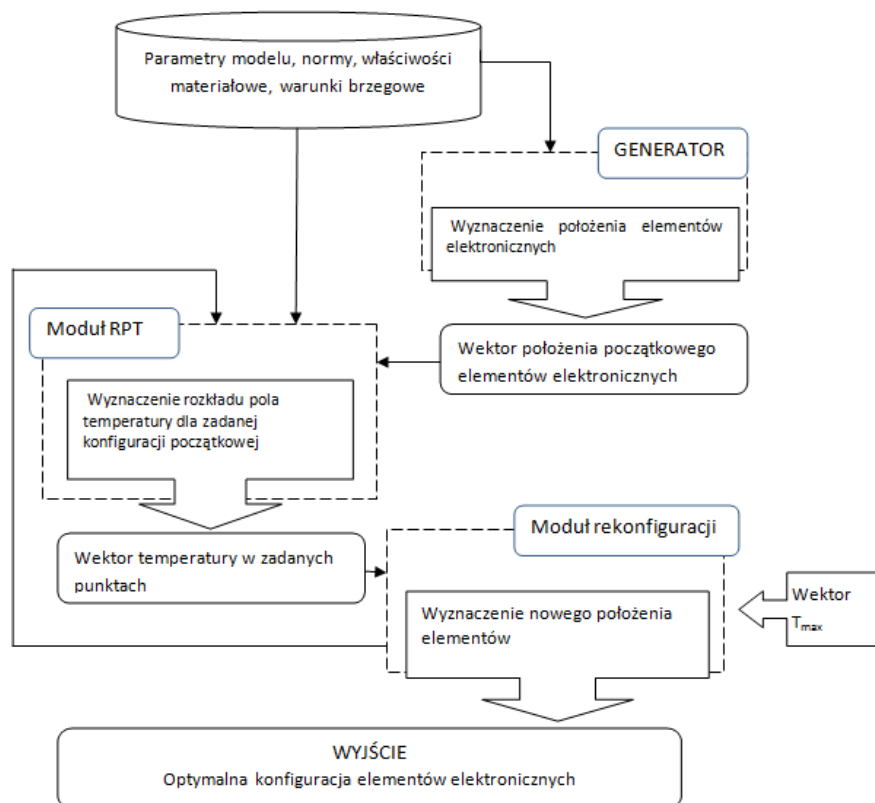
2. Ogólna koncepcja systemu

Koncepcja systemu wspomagającego badanie zależności termicznych uwzględnia budowę modułową. Przyjęcie takiego podejścia pozwoliło włączyć do opracowanego modelu istniejące programy wykorzystywane w projektowaniu elektrotermicznym (np. ANSYS, NISA itp.). Na podstawie przeglądu literatury oraz badań własnych

opracowano system w skład, którego wchodzi następujące moduły [6] [7]:

- (iv) Moduł wstępnej alokacji - Generatorem;
- (v) Moduł obliczania rozkładu pola temperatury – RPT;
- (vi) Moduł rekonfiguracji - MR.

Zależności pomiędzy modułami przedstawia rysunek 2.



Rys. 2. Schemat systemu wspomagającego zagadnienia pola temperatury w nanoukładach elektronicznych

Do wyznaczenia rozkładu pola temperatury w układzie elektronicznym dla zadanej początkowej konfiguracji elementów elektronicznych przy ustalonych: geometrii układu, właściwościach materiałowych i warunkach brzegowych, rekomenduje się wykorzystanie modułu FEAP wchodzącego w skład komercyjnego pakietu obliczeniowego - NISA.

Programy tego typu składają się zasadniczo z trzech głównych części, tj.:

- preprocesora,
- procesora,
- postprocesora.

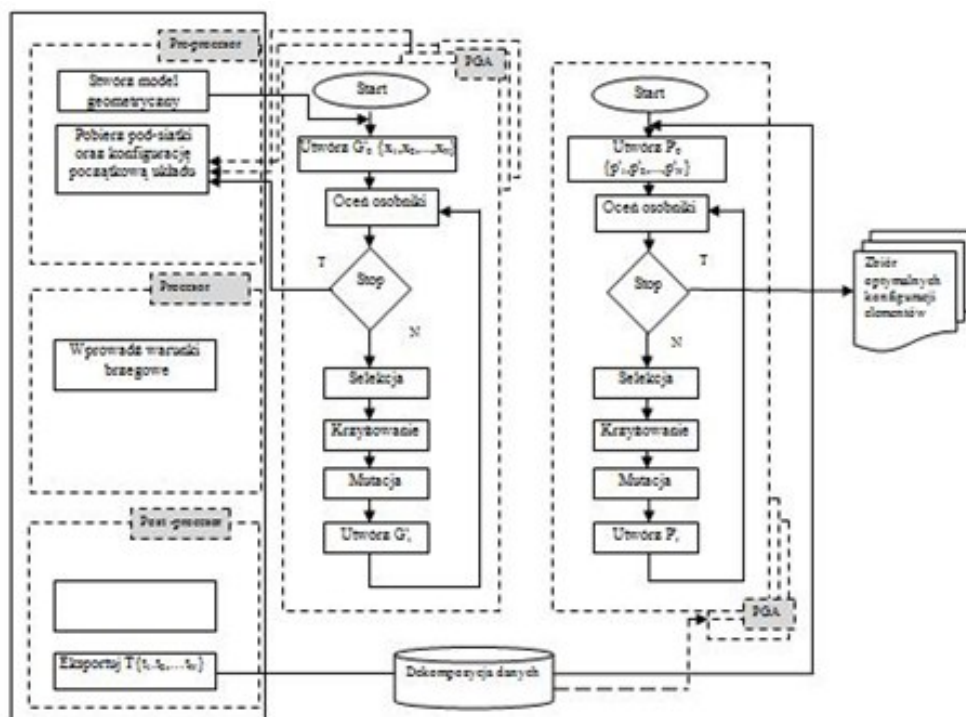
W preprocesorze użytkownik tworzy geometrię konstrukcji oraz generuje siatkę elementów. W części procesora użytkownik definiuje typ zagadnienia oraz określa rodzaj analizy i jej szczegółowe opcje. Dodatkowo użytkownik ma możliwość i określenia warunków brzegowych i/lub zdefiniowania obciążeń. Postprocesor umożliwia prezentację

otrzymanych wyników w formie graficznej i/lub tekstowej. Tabela 1 zawiera dane wejściowe i wyniki generowane przez ten moduł.

Tab. 1. Dane wejścia/wyjścia dla modułu RPT

<p>Dane wejściowe:</p> <ul style="list-style-type: none"> • geometria układu, • wstępne rozmieszczenie elementów elektronicznych otrzymane z Modułu Generator, • właściwości materiałowe, • warunki brzegowe.
<p>Dane wyjściowe:</p> <ul style="list-style-type: none"> • rozkład temperatury

Wektor temperatury uzyskany na wyjściu modułu RPT stanowi wejście dla Modułu Rekonfiguracji (rys. 3).



Rys. 3. Elementy składowe Modułu Rekonfiguracji

Moduł rekonfiguracji ma za zadanie na podstawie danych wejściowych i przy użyciu równoległego algorytmu genetycznego wyznaczyć optymalne rozmieszczenie elementów

elektronicznych. Tabela 2 zawarto zestawienie danych wejściowych oraz wyniki generowane przez ten moduł.

Tab. 2. Dane wejścia/wyjścia dla Modułu Rekonfiguracji

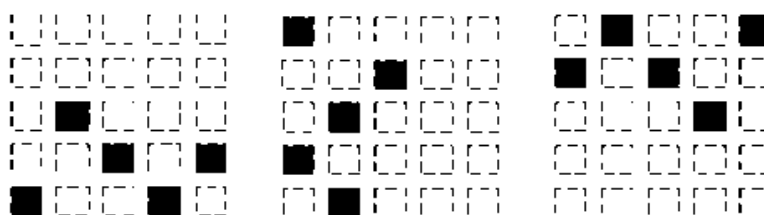
<p>Dane wejściowe:</p> <ul style="list-style-type: none"> (i) regularna siatka przestrzenna; (ii) N-liczba elementów grzewczych (iii) wektor temperatury uzyskany z programu NISA (iv) wektor temperatury dopuszczalnej T_{\max} (v) parametry algorytmu optymalizującego położenie elementów elektronicznych <p>Dane wyjście:</p> <ul style="list-style-type: none"> (vi) optymalna konfigurację elementów elektronicznych
--

Punktem wyjścia w przedstawionej koncepcji modelowania pola temperatury w nanoukładach elektronicznych jest optymalizacja położenia początkowego elementów elektronicznych układu. Według autorki etap ten ma skrócić czas potrzebny na uzyskanie optymalnej ze względu na wydzielaną moc konfiguracji elementów elektronicznych.

3. Opis problemu

Przedstawione rozważania dotyczą sytuacji, w której dziedziną określoności układu jest pewien obszar Ω w przestrzeni R^2 . Elementy elektroniczne mogą być rozmieszczone w zadanym obszarze w regularnych węzłach siatki przestrzennej. Liczbę możliwych konfiguracji elementów elektronicznych określa się ze wzoru na kombinację bez powtórzeń.

Zakładając układ elektroniczny składający się z 5 elementów umieszczonych na kwadratowej płytce o rozmiarze 10 cm x10 cm i przyjmując węzeł co 0.5 cm uzyskamy jednorodną siatkę składającą się z 25 węzłów. Tym sposobem otrzymujemy ponad 53000 różnych konfiguracji do przetestowania. Na rysunku 4 przedstawiono przykładowe konfiguracje [8].



Rys. 4. Przykładowe rozmieszczenie elementów w węzłach siatki przestrzennej

Zadanie polega na znalezieniu optymalnej ze względu na przyjęte kryterium konfiguracji elementów elektronicznych. Do rozwiązania tak postawionego problemu wykorzystano równoległy algorytm genetyczny.

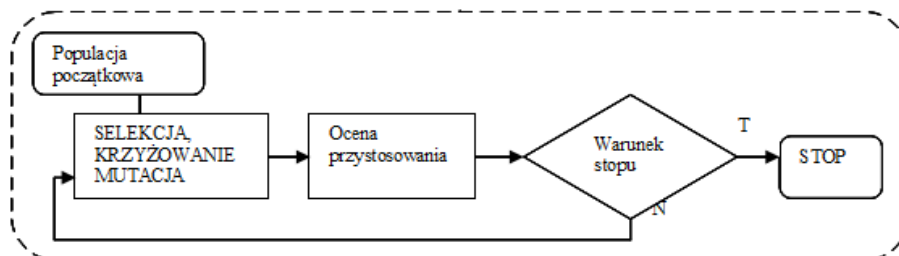
3.1. Równoległe algorytmy genetyczne

Algorytmy genetyczne są narzędziem szeroko stosowanym w zagadnieniach optymalizacyjnych. Wykorzystując inherentną efektywność przetwarzania równoległego algorytmów genetycznych uzyskuje się możliwość wykonania obliczeń dla zadań o dużej wymiarowości przy jednoczesnym przyspieszeniu obliczeń.

Algorytmy genetyczne należą do klasy algorytmów stochastycznych, polegających na losowym przeszukiwaniu przestrzeni możliwych rozwiązań [8]. W klasycznym algorytmie genetycznym operuje się grupą chromosomów (osobników) zwaną populacją, która stanowi zakodowaną (zazwyczaj przy użyciu kodu binarnego) postać propozycji rozwiązań. Jakość osobnika opisana jest wartością liczbową, zwaną przystosowaniem (funkcja przystosowania). Wyselekcjonowane osobniki, podlegają w poszczególnych pokoleniach przemianom (krzyżowaniu, mutacji) doprowadzając ostatecznie do otrzymania najlepszego chromosomu (optymalnego rozwiązania).

Zadaniem algorytmu genetycznego jest ciągła poprawa średniej wartości funkcji dopasowania całej populacji w iteracjach, co prowadzi do otrzymania optymalnego rozwiązania.

Ogólny schemat działania algorytmu genetycznego przedstawia rysunek 5 [8].



Rys. 5. Algorytm genetyczny - schemat działania

Osobniki z bieżącej populacji wybierane są do nowego pokolenia względem wartości funkcji dopasowania. Do najpopularniejszych metod selekcji należy wybór losowy z powtórzeniami. W metodzie tej prawdopodobieństwo znalezienia się w populacji potomnej danego osobnika jest proporcjonalne do wskaźnika przystosowania ciągów kodowych [8]. Po selekcji następuje krzyżowanie osobników polegające na zamianie odpowiadających sobie odcinków chromosomów. W wyniku działania operatora na dwóch osobnikach pochodzących z puli rodzicielskiej otrzymujemy dwa chromosomy potomne, które wejdą do następnego pokolenia.

Kolejną operacją wykonywaną na losowo wybranych osobnikach jest zmiana pojedynczych genów zwana - mutacją.

Po wykonaniu selekcji, krzyżowania i mutacji osobniki z populacji są ponownie oceniane. Proces jest powtarzany dopóki nie osiągnie się warunku stopu (np. wcześniej określona liczba generacji). Rozwiązaniem jest osobnik z ostatniej populacji, który

charakteryzuje się najwyższą funkcją dopasowania.

Dla omawianego w artykule problemu pojedynczemu osobnikowi (chromosom) będzie odpowiadała konfiguracja elementów. W związku z tym k -tą populację chromosomów można opisać następująco:

$$P^k = \{X^k_1, X^k_2, X^k_3, \dots, X^k_i, \dots, X^k_M\} \quad i=1,2,\dots, M, k=1,2,\dots, q \quad (1)$$

gdzie:

- (vii) X^k_i - chromosom reprezentujący potencjalne rozwiązanie (i -tą konfigurację systemu składającego się z N elementów);
- (viii) M – liczba chromosomów w populacji;
- (ix) k - ta generacja.

Kolejnym krokiem w pracy z algorytmami genetycznymi jest stworzenie populacji startowej. W prostym algorytmie genetycznym populacja startowa jest losowana. W omawianym problemie elementy elektroniczne wydzielając ciepło oddziałują na elementy sąsiednie prowadząc do podgrzania całej struktury. Aby zmniejszyć wpływ temperatury poszczególnych elementów elektronicznych na sąsiednie elementy, tworząc populację startową wprowadzono ograniczenie na zachowanie dopuszczalnych odległości między elementami. Generowanie populacji startowej odbywa się zgodnie z algorytmem opisanym w tabeli 3.

Tab. 3. Algorytm generowania populacji startowej

KROK 1

Dopóki nie ustalono wymaganej liczby chromosomów w populacji startowej powtarzaj

KROK 2

Dopóki nie ustalono wymaganej liczby genów w chromosomie powtarzaj

wylosuj dostępną pozycję należącą do siatki przestrzennej

wstaw wylosowaną pozycję do i -tego chromosomu na pozycję j -tą

dla wylosowanego węzła generuj sąsiedztwo wg. von Neumanna zdefiniowane następująco:

$$N^v_{(x_0, y_0)} = \{(x, y) : |x - x_0| + |y - y_0| \leq r\} \quad (2)$$

i ustaw wszystkich sąsiadów w siatce przestrzennej jako węzły niedostępne

KROK 3

Zapamiętaj populację startową

Ze względu na złożoność zagadnienia badaną przestrzeń poszukiwań podzielono na mniejsze podprzestrzenie, realizując tym sposobem model obliczeń równoległych wykorzystujący równoległość danych. W tym celu wykorzystano schemat komunikacji master - slave.

W podejściu tym wyróżnia się jeden procesor jako główny (ang. master), a reszta procesorów tak zwanych podwładnych (ang. slaves) – odpowiada za wykonanie obliczeń. Procesor główny jako nadzorca posiada całą wiedzę o stanie algorytmu i kontroluje kolejność wykonywania obliczeń przez poszczególne procesory tak jak to pokazano w tabeli 4.

Tab. 4. Schemat komunikacji master – slave

<pre> Pobierz identyfikator IF (jeśli jesteś procesorem głównym) { //Wykonuj pracę przydzieloną procesorowi głównemu Krok 1. Podziel badaną przestrzeń na rozłączne podzbiory $S = \bigcup_{i=1}^P S_i \quad \text{oraz} \quad S_i \cap S_j = \emptyset \quad \text{dla } i \neq j$ Krok 2. Wyślij dane do procesorów podwładnych Krok 3. Przydziel zadania poszczególnym procesorom podwładnym Krok 4. Zbierz wyniki Krok 5. Zaprezentuj wyniki } ELSE { //Wykonuj pracę przydzieloną procesorowi podwładnemu Krok 1. Odbierz dane od procesora głównego Krok 2. Wykonaj powierzone zadania Krok 3. Zwróć wyniki do procesora głównego } </pre>

Pierwsze zadanie powierzone poszczególnym procesorom polega na przygotowaniu siatki przestrzennej. Zdecydowano, że poszczególne procesory na podstawie informacji o początku i końcu przydzielonego fragmentu obszaru będą konstruowały siatkę przestrzenną według następującego algorytmu:

Procesor 0

Przyjmując jako pierwszy węzeł początek układu (0,0). Następny węzeł otrzymamy gdy $x \rightarrow x + \Delta x$ i $y = 0$ pozostaje niezmienny.

Następny węzeł ma

$x \rightarrow x + \Delta x$, $y = 0$, i tak dalej aż Δx_i zostaną wyczerpane.
 Następny, drugi poziomy rząd rozpoczyna się od
 $x = 0$, $y \rightarrow y + \Delta y$ i x rośnie dopóki nie wyczerpią się Δx_i .
 Powtarzamy proces aż do osiągnięcia ostatniego węzła $(n_x + 1)(n_y + 1)$ tzn., kiedy Δx_i i Δy_i są wyczerpane jednocześnie.

Tabela 5 zawiera implementację algorytmu generowania siatki przestrzennej.

Tab. 5. Generowanie siatki przestrzennej - fragment kodu.

```

for (long int j = 0; j <=cells; j++)
{
pom_x=0.0;
for(long int i = 0; i <=cells; i++)
{
dyn_tab[j][i].x=pom_x;
dyn_tab[j][i].y=pom_y;
pom_x=pom_x+krok;
}
pom_y=pom_y+krok;
}
  
```

Kolejnym zadaniem wykonywanym na poszczególnych procesorach jest realizowanie wyspowego algorytmu genetycznego. W modelu tym ewoluuje jednocześnie kilka populacji zwanych wyspami. Każdy algorytm działa według schematu zamieszczonego w punkcie 3.1 z wykorzystaniem następujących danych wejściowych:

- regularna siatka przestrzenna;
- liczba elementów grzewczych
- promień do zdefiniowania sąsiedztwa von Neumanna
- parametry algorytmu genetycznego
 - funkcja przystosowania,
 - rozmiar populacji,
 - warunek stopu,
 - liczba genów w chromosomie,
 - metoda selekcji,
 - prawdopodobieństwo krzyżowania,
 - prawdopodobieństwo mutacji.

Rezultatem działania wyspowego algorytmu genetycznego jest wstępne położenie poszczególnych elementów elektronicznych.

Dla omawianego zadania, aby zmniejszyć wpływ termiczny elementów względem siebie należy te elementy rozproszyć w obrębie układu. Wówczas przykładowa funkcja przystosowania może zostać wyrażona średnią liczbą sąsiadów. Dąży się do znalezienia układu elementów (chromosom) o najmniejszej średniej liczbie sąsiadów. Tabela 6 zawiera kod realizujący metodę wyznaczania wartości funkcji przystosowania.

Tab. 6. Kod wyznaczający wartość funkcji przystosowania

```

//metoda wyznaczająca dopasowanie
void Wyznacz(){
int i,j, licz, l_j;
licz=0; l_j=0;
for(i=1;i<(len-1);i++) {
if (tab[i][0]==1) { l_j++;
licz=licz+(tab[i-1][0]+tab[i+1][0]+tab[i][1]); }
if (tab[i][len-1]==1) { l_j++;
licz=licz+(tab[i-1][len-1]+tab[i+1][len-1]+tab[en-2][1]); }
if (tab[0][i]==1){ l_j++;
licz=licz+(tab[0][i+1]+tab[0][i-1]+tab[1][i]); }
if (tab[len-1][i]==1){ l_j++;
licz=licz+(tab[len-1][i+1]+tab[len-1][i-1]+tab[en-2][i]); }
for(j=1;j<(len-1);j++){
if (tab[i][j]==1){ l_j++; licz=licz+licz_s(i,j);}; }
};
if (tab[0][0]==1) { l_j++;
licz=licz+(tab[0][1]+tab[1][0]);}
if (tab[len-1][0]==1) { l_j++;
licz=licz+(tab[en-2][0]+tab[en-1][1]);}
if (tab[0][len-1]==1) { l_j++;
licz=licz+(tab[0][len-2]+tab[1][len-1]);}
if (tab[en-1][len-1]==1) {
l_j++;
licz=licz+(tab[en-1][len-2]+tab[en-2][len-1]);}
fit=licz/(float)l_j;
}
int licz_s(int a, int b){
return tab[a+1][b]+tab[a-1][b]+tab[a][b+1]+tab[a][b-1];
};

```

4. Wnioski

W pracy zawarto teoretyczne rozważania dotyczące zagadnień pola temperatury w nanoukładach elektronicznych. Zaprezentowano model koncepcyjny w skład, którego weszły dwa autorskie moduły oraz moduł komercyjny do symulacji elektrotermicznych. Przedstawione podejście oparto na założeniu, w którym za istotne uznaje się optymalizację rozmieszczenia początkowego elementów elektronicznych. Do rozwiązania problemu optymalizacyjnego wykorzystano równoległe algorytmy genetyczne. Przedstawiono prosty do realizacji schemat komunikacji master-slave oraz pokazano pracę poszczególnych procesorów uwzględniając ich rolę w obliczeniach równoległych.

Literatura

1. Karbowski A., Niewiadomska-Szynkiewicz E., Obliczenia równoległe i rozproszone, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 2001.
2. Marksteiner P. High-performance computing – an overview, *Computer Physics Communications* 97, 1996, 16-35.
3. Tanenbaum A. S., Strukturalna organizacja systemów komputerowych, Wydanie V, Helion, 2006.
4. Zak S., Łukasik S., Zastosowanie równoległych algorytmów genetycznych do rozwiązywania problemów przestrzennej alokacji zasobów, *Krakowska Konferencja Młodych Uczonych*, 21-23 września 2006, w: "Materiały konferencyjne Krakowskiej Konferencji Młodych Uczonych", Wydawnictwo Academica, Krakow, 2006, 171-179.
5. Dias Jr. T., Milanez L.F., Optimal location of heat sources on a vertical wall with natural convection through genetic algorithms *International Journal of Heat and Mass Transfer*, Volume 49, Issues 13–14, July 2006, 2090-2096.
6. Liu G.R., Zhou J.-J., Wang J.G, 2002, Coefficients identification in electronic system cooling simulation through genetic algorithm, *Computers and Structures* 80, 23-30.
7. Liu, F.-B., 2008, A modified genetic algorithm for solving the inverse heat transfer problem of estimating plan heat source, *International Journal of Heat and Mass Transfer* 51, 3745-3752.
8. Goldberg D. E., Algorytmy genetyczne i ich zastosowanie, Wydawnictwo-Naukowo Techniczne, Warszawa 1995.

Mgr inż. Anna Gryko-Nikitin
Katedra Informatyki Gospodarczej i Logistyki
Wydział Zarządzania Politechnika Białostocka
15-351 Białystok, ul. Wiejska 45 A
tel./fax: (0-85) 7468989
e-mail: a.gryko@pb.edu.pl