ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: INFORMATYKA z. 30 Nr kol. 1315

Aleksander CZARNOŻYŃSKI

PROBLEM WZAJEMNEJ BLOKADY W SIECI POŁĄCZEŃ MIĘDZYPROCESOROWYCH

Streszczenie. W artykule przedstawiono problem zapobiegania zjawisku wzajemnej blokady, które może mieć miejsce podczas transmisji danych w sieciach połączeń systemów wieloprocesorowych, metody zapobiegania wzajemnej blokadzie proponowane przez autora dla tzw. sieci AMP (ang. A Minimum Path configurations) oraz oryginalne algorytmy konstrukcji i optymalizacji wolnych od wzajemnej blokady funkcji wyboru drogi w sieci AMP.

THE PROBLEM OF DEADLOCK-FREE MESSAGE ROUTING IN MULTIPROCESSOR INTERCONNECTION NETWORK

Summary. In the article the problem of deadlock-free message routing in multiprocessor interconnection networks is discussed. A solution involving original algorithms is proposed for the construction of deadlock-free routing functions for a class of interconnection networks known as AMP (A Minimum Path configurations).

LE PROBLEME DE ROUTAGE SANS INTERBLOCKAGE DANS UN RESEAU DES SYSTEMES MULTIPROCESSEURS

Résumé. L'article discute le problème de routage sans interblockage dans des rèseaux des systèmes multiprocesseurs. On présente les méthodes de prévention de l'interblockage proposées par l'auteur pour le réseau type AMP et les algorithmes de construction et l'optimisation des fonctions de choix de routage dans ces réseaux.

Niniejszy tekst został opracowany w ramach Projektu Badawczego KBN nr 8 T11C 041 08.

1. Wstęp

Dokonujący się w ostatnich kilkunastu latach postęp technologiczny utorował drogę ku szerszemu rozpowszechnieniu komputerów równoległych o architekturze wieloprocesorowej (ang. multiprocessor systems). Odmianą takich architektur są tzw. systemy wieloprocesorowe z przesyłaniem wiadomości (ang. message-passing multiprocessors), do których należą między innymi znane systemy transputerów T800. Systemy takie składają się z wielu jednostek przetwarzających PE (ang. Processing Elements), z których każda zawiera własny procesor i moduł pamięci operacyjnej. Poszczególne PE komunikują się za pomocą sieci połączeń międzyprocesorowych.

W praktyce najczęściej stosowane są sieci połączeń o topologiach niekompletnych grafów. Wówczas pomiędzy PE, które nie są bezpośrednio połączone, dane transmitowane są poprzez szereg węzłów pośrednich. Jeżeli drogi wiadomości nie zostały odpowiednio wybrane, w sieci połączeń wystąpić może zjawisko wzajemnej blokady [4], którego efektem jest całkowite wstrzymanie transmisji danych.

Wybór drogi w sieci połączeń jest zadaniem algorytmów komunikacji (ang. routing algorithms). Dotychczas zaproponowano wiele rozwiązań tzw. wolnych od wzajemnej blokady (ang. deadlock-free) algorytmów komunikacji dla sieci połączeń o typowych, regularnych topologiach (np. [1, 4, 5, 7, 8]). Problemem rozważanym w niniejszym artykule jest kwestia konstrukcji takich algorytmów dla sieci połączeń o nieregularnych topologiach, a przede wszystkim tzw. sieci AMP (ang. A Minimum Path configurations) [2].

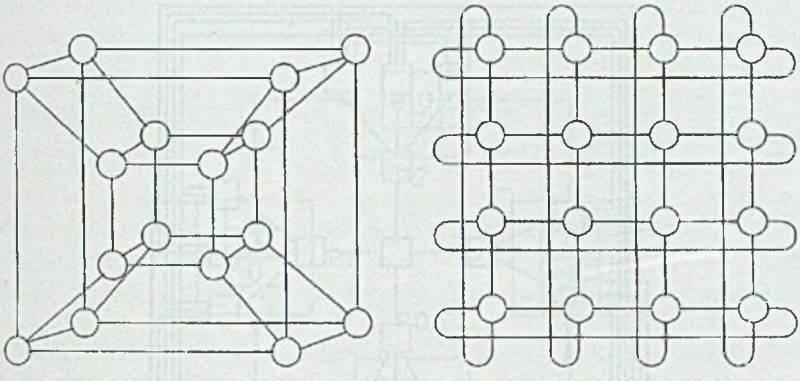
W porównaniu do typowych sieci połączeń sieci AMP cechują się korzystnymi wartościami parametrów mających wpływ na czas trwania obliczeń aplikacji oraz przyspieszenie (ang. speed-up). Rodzą się więc następujące pytania – jak zapobiegać wzajemnej blokadzie, aby zachowane zostały korzystne (w porównaniu do typowych sieci połączeń) cechy sieci AMP oraz jak skonstruować odpowiednie algorytmy komunikacji.

2. Sieci połączeń i sieci AMP

Mianem sieci połączeń międzyprocesorowych określać będziemy spójny, nieskierowany graf $I = (P, L)$, którego zbiór wierzchołków P reprezentuje procesory, zaś zbiór krawędzi $L \subseteq \{(p_i, p_j) \mid p_i, p_j \in P,\ p_i \neq p_j\}$ – dwukierunkowe *łącza komunikacyjne* (ang. link). Każde łącze składa się z dwóch autonomicznych, jednokierunkowych *kanałów komunikacyjnych* (ang. channel), posiadających własne kolejki (o ograniczonej długości) buforów danych.

W obrębie kanału komunikacyjnego mogą (ewentualnie) zostać zdefiniowane również tzw. *kanały wirtualne* (ang. virtual channels), zdolne do niezależnej transmisji danych i dysponujące własnymi kolejkami buforów. Funkcję przełączników (ang. switch) na drodze wiadomości w sieci pełnią *procesy komunikacyjne*. Przykłady typowych *regularnych* topologii sieci przedstawiono na rys. 1.

Topologia sieci połączeń dobierana jest tak, aby uzyskać jak największe wartości przyspieszenia obliczeń. Zazwyczaj korzystny wpływ na wartości przyspieszenia ma przesy-



Rys. 1. Przykłady typowych sieci połączeń
Fig. 1. Examples of typical interconnection networks

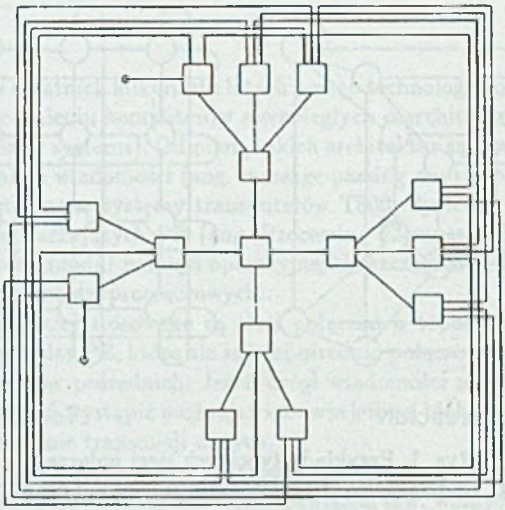
lanie danych w sieci po możliwie najkrótszych drogach. Długość dróg wiadomości zależy zaś od takich parametrów sieci, jak średnia odległość pomiędzy wierzchołkami d_{ave} oraz średnica d_m. Wpływ tych parametrów na przyspieszenie obliczeń jest tym większy, im w większym stopniu daną aplikację cechuje intensywna wymiana danych o charakterze globalnym [10].

Z przedstawionych powyżej powodów szczególną uwagę zwrócono na zaproponowane w [2] sieci AMP (patrz rys. 2), które charakteryzują się bardzo niskimi wartościami parametrów d_m i d_{ave}. Topologie AMP są *nieregularne*. Sieci te są konstruowane za pomocą algorytmów, działających na zasadzie *przeszukiwania heurystycznego* (problem konstrukcji optymalnych AMP jest $\mathcal{NP}$-trudny [2]). Przy ich konstrukcji przyjęto założenie, że stopień wierzchołków k jest *stały* (tj. $k_{AMP} = 4$ ze względu na zastosowania w systemach transputerów), z wyjątkiem dwóch wierzchołków stopnia $k-1$, których dwa wolne łącza służą do komunikacji z otoczeniem. W [2] pokazano, że wartości parametrów d_{ave} i d_m sieci AMP są *znacznie niższe* w porównaniu do tradycyjnych sieci połączeń o stopniu wierzchołków $k \leq k_{AMP}$.

3. Wzajemna blokada w sieciach połączeń i przegląd metod jej zapobiegania

3.1. Zjawisko wzajemnej blokady

W celu transmisji wiadomości w sieci połączeń, procesy komunikacyjne korzystają z kanałów komunikacyjnych i buforów danych. W przypadku, gdy grupa procesów komunikacyjnych wejdzie w stan wzajemnego, permanentnego oczekiwania na transmisję



Rys. 2. Sieć AMP o rozmiarze $p = 15$
Fig. 2. AMP network of $p = 15$ processors

danych przez kanały komunikacyjne, których kolejki są pełne, w sieci połączeń ma miejsce *wzajemna blokada procesów komunikacyjnych* [4].

Przykład wzajemnej blokady w sieci połączeń przedstawiono na rys. 3. Wiadomości transmitowane są w trybie przechowaj-i-wyślij (ang. store-and-forward). Każdy kolejny wierzchołek na drodze dowolnej wiadomości w sieci nie jest wierzchołkiem końcowym (na tej drodze), zaś kolejki kanałów komunikacyjnych są pełne. W rezultacie żadna z wiadomości nie może zostać odebrana w kolejnym wierzchołku sieci.

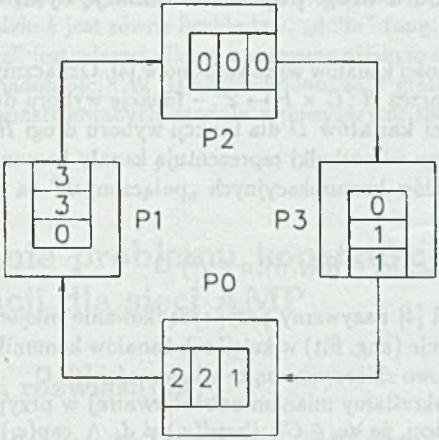
3.2. Klasyfikacja metod ochrony przed wzajemną blokadą

Znane z literatury metody ochrony przed skutkami wzajemnej blokady można ogólnie podzielić na dwie grupy:

- metody prewencyjne, polegające na zapobieganiu wzajemnej blokadzie,
- metody polegające na wykrywaniu i likwidacji (już zaistniałej) wzajemnej blokady.

W praktyce konstrukcja metod ochrony przed wzajemną blokadą zależy od specyfiki systemu komputerowego oraz związanych z tym wymagań (np. w zakresie złożoności obliczeniowej, czasów reakcji itp.).

Dla systemów wieloprocesorowych zaproponowano niewiele metod polegających na wykrywaniu i likwidacji wzajemnej blokady. Można wśród nich wymienić metody polegające na zastosowaniu heurystyki w celu detekcji blokady oraz chwilowym usuwaniu z sieci zablokowanych wiadomości [9]. W grupie metod znanych z literatury i stosowanych w praktyce dominują metody prewencyjne, polegające na zapobieganiu blokadzie poprzez odpowiednią konstrukcję *funkcji wyboru drogi* (ang. routing function) [1, 4, 5, 7, 8].



Rys. 3. Przykład wzajemnej blokady w sieci połączeń
Fig. 3. Example of deadlock in interconnection network

3.3. Metody zapobiegania blokadzie polegające na odpowiedniej konstrukcji funkcji wyboru drogi

W celu transmisji wiadomości niezbędne jest określenie jej drogi w sieci. Zadanie to realizują algorytmy komunikacji definiowane zazwyczaj za pomocą funkcji wyboru drogi, która wyznacza w dowolnym wierzchołku sieci, kolejny kanał komunikacyjny na dalszej drodze wiadomości lub zbiór alternatywnych kanałów komunikacyjnych. W tym drugim przypadku wiadomość może podążać do wierzchołka docelowego jedną z kilku alternatywnych dróg. Mianem *wolnych od wzajemnej blokady* (ang. deadlock-free routing function) określane są funkcje wyboru drogi, które skonstruowano w taki sposób, że wzajemna blokada sieci połączeń nie jest możliwa.

Wśród znanych z literatury metod konstrukcji wolnych od wzajemnej blokady funkcji wyboru drogi można wyróżnić następujące grupy:

1. Metodę polegającą na konstrukcji funkcji dekomponowalnych, posiadających wolne od wzajemnej blokady podfunkcje [5].

2. Metody polegające na konstrukcji tzw. *acyklicznych sieci wirtualnych* [7, 8],

3. Metody działające na zasadzie eliminacji cykli w tzw. *grafach zależności kanałów* (ang. channel dependency graphs) [4, 1].

Zastosowanie pierwszej z metod wymaga implementacji skomplikowanego algorytmu szeregowania. Z kolei koncepcja „acyklicznych sieci wirtualnych" dotyczy sieci regularnych, a więc sieci o cechach odmiennych niż AMP. Z tych powodów uwagę skoncentrowano na metodach polegających na eliminacji cykli.

3.3.1. Warunek wystarczający konstrukcji wolnych od wzajemnej blokady funkcji wyboru drogi poprzez eliminację cykli w grafach zależności kanałów

Pojęcie grafu zależności kanałów wprowadzono w [4]. Oznaczmy przez C zbiór kanałów komunikacyjnych, zaś przez $R : C \times P \mapsto 2^C$ – funkcję wyboru drogi.

Grafem zależności kanałów D dla funkcji wyboru drogi R nazywamy *skierowany* graf $D = \langle C, E \rangle$, którego wierzchołki reprezentują kanały komunikacyjne, zaś krawędzie reprezentują pary kanałów komunikacyjnych „połączonych" za pomocą funkcji wyboru drogi R:

$$E = \{(c_i, c_j) | c_j \in R(c_i, n), n \in P\} \quad \square \tag{1}$$

Konfiguracją sieci [4] nazywamy przyporządkowanie miejscom zajmowanym przez wiadomości lub ich porcje (ang. flit) w kolejkach kanałów komunikacyjnych wierzchołków sieci, do których docelowo skierowane są te wiadomości [4]. □

Konfigurację sieci określamy mianem **zablokowanej** w przypadku, gdy istnieje taki podzbiór kanałów C' sieci, że $\forall c_i \in C'$, $(\text{head}(c_i) \neq d_i \ \wedge \ \text{cap}(c_i) \neq 0)$ zachodzi:

$$c_j = S(R(c_i, \text{head}(c_i))) \Rightarrow (c_j \in C' \ \wedge \ \text{size}(c_j) = \text{cap}(c_j)) \tag{2}$$

gdzie: $S : 2^C \mapsto C$ oznacza funkcję, zgodnie z którą dokonywany jest wybór jednego ze zbioru alternatywnych kanałów, zwracanych przez funkcję R, $\text{head}(c_i) = n_d$ oznacza, że pierwsza wiadomość lub porcja wiadomości w kolejce kanału $c_i \in C$ jest docelowo skierowana do n_d, przez $\text{size}(c_i)$ oznaczono długość kolejki kanału $c_i \in C$, przez $\text{cap}(c_i)$ oznaczono maksymalną długość kolejki, zaś d_i jest wierzchołkiem końcowym kanału c_i. □

W przypadku konfiguracji zablokowanej istnieje taki zbiór wiadomości w sieci, że dla dowolnej wiadomości (lub jej porcji) z tego zbioru następny wierzchołek na jej drodze nie jest wierzchołkiem docelowym, a ponadto żadna wiadomość z tego zbioru nie może być transmitowana dalej, ponieważ kolejka następnego kanału na jej drodze jest pełna - wystąpiła wzajemna blokada.

Twierdzenie 3.1 [4]: Jeżeli w grafie zależności kanałów D dla funkcji wyboru drogi R nie występują cykle, to w wyniku zastosowania funkcji R nie utworzy się zablokowana konfiguracja sieci. □

Powyższe twierdzenie stanowi **warunek wystarczający** konstrukcji wolnych od wzajemnej blokady funkcji wyboru drogi. Przedstawione definicje i twierdzenie wzorowane są na [4], gdzie wprowadzono pojęcia grafu zależności kanałów, konfiguracji oraz konfiguracji zablokowanej.

3.3.2. Przykład metody – zliczanie dolin

Na podstawie twierdzenia 3.1 w literaturze zaproponowano kilka metod konstrukcji funkcji wyboru drogi [4, 1]. Do grupy metod działających na tej zasadzie należy między innymi metoda *zliczania dolin* (ang. valley counting) [1].

Zgodnie z tą metodą w obrębie każdego kanału komunikacyjnego $c_{i,j}$ sieci połączeń definiowanych jest l klas kanałów wirtualnych $c^0_{i,j}, c^1_{i,j}, \ldots, c^{l-1}_{i,j}$, zaś wierzchołki sieci połączeń numerowane są kolejnymi, nieujemnymi liczbami całkowitymi. Bezpośrednio po

wprowadzeniu do sieci wiadomość transmitowana jest wyłącznie kanałami c^0. Podczas transmisji pomiędzy kolejnymi wierzchołkami sieci wiadomość przesyłana jest kanałami wirtualnymi klasy c^k, gdzie k jest równe liczbie tzw. „dolin" (ang. valleys) na jej dotychczasowej drodze („dolina" jest wierzchołkiem o numerze niższym od swego poprzednika i następnika na drodze wiadomości). W [1] udowodniono, że w grafach zależności kanałów funkcji wyboru drogi, konstruowanych zgodnie z powyższymi zasadami, nie występują cykle.

4. Rozwiązanie problemu konstrukcji algorytmów komunikacji dla sieci AMP

4.1. Koncepcja rozwiązania

Koncepcję rozwiązania proponowaną dla sieci AMP można sprowadzić do dwóch punktów:

1. **Automatycznej** konstrukcji i optymalizacji wolnych od wzajemnej blokady funkcji wyboru drogi dla sieci AMP (tj. za pomocą opracowanych w tym celu algorytmów).

2. Opracowania **alternatywnych** rozwiązań dla dwóch przypadków: sieci z kanałami wirtualnymi oraz sieci, w których nie zdefiniowano kanałów wirtualnych.

Aby zachować korzystne cechy sieci AMP, w celu konstrukcji funkcji wyboru drogi zastosowano następujące kryteria oceny rozwiązania:

1. W przypadku gdy definiowane są kanały wirtualne i transmisja danych ma miejsce po *najkrótszych* drogach – *liczbę klas kanałów wirtualnych*,

2. W przypadku gdy w sieci nie są definiowane kanały wirtualne – *wartość maksymalną* d_m *i średnią* d_{ave} *długości dróg wiadomości, wyznaczonych przy założeniu, że transmisja pomiędzy każdą parą procesorów jest jednakowo prawdopodobna.*

4.2. Metody proponowane w celu zapobiegania blokadzie

Kierując się powyższymi kryteriami, zaproponowano dwie alternatywne metody zapobiegania blokadzie, działające na zasadzie eliminacji cykli w grafach zależności kanałów:

1. W przypadku, gdy w sieci definiowane są kanały wirtualne – znaną z literatury metodę **zliczania dolin**.

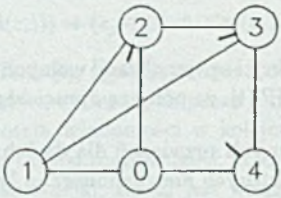
2. W przypadku, gdy nie zdefiniowano kanałów wirtualnych – oryginalną metodę **sieci z barierami**, która jest modyfikacją znanej z literatury [11] metody opracowanej dla sieci komputerowych.

Dla obydwu metod opracowano odrębne algorytmy konstrukcji funkcji wyboru drogi, działające na zasadzie przeszukiwania grafów. W celu optymalizacji funkcji wyboru drogi zastosowano ogólne metody optymalizacji kombinatorycznej: *algorytmy genetyczne* oraz metodę „wielowątkowego" przeszukiwania heurystycznego. W dalszym ciągu opisano zasadnicze elementy proponowanych rozwiązań.

4.3. Metoda sieci z barierami

Tak zwane *bariery* wprowadzane są w sieci połączeń w wyniku etykietowania jej wierzchołków kolejnymi, nieujemnymi liczbami całkowitymi.

Siecią połączeń z barierami nazywamy parę (I, f), gdzie $I = (P, L)$ jest siecią połączeń międzyprocesorowych, zaś $f : P \mapsto [0, p-1]$ jest bijekcją, przyporządkowującą każdemu wierzchołkowi sieci połączeń I, liczbę całkowitą z zakresu $[0, p-1]$, $p = |P|$. Mówimy, że w dowolnym wierzchołku $i \in P$ sieci, pomiędzy parą incydentnych krawędzi (j, i) oraz (i, k) występuje bariera, gdy $f(i) > f(j) \wedge f(i) > f(k)$. □



Rys. 4. Przykład sieci z barierami
Fig. 4. Example of barrier network

Aby wykazać, że funkcje wyboru drogi dopuszczające transmisję wiadomości wyłącznie po drogach bez barier są wolne od wzajemnej blokady, wystarczy udowodnić, że w grafach zależności kanałów dla funkcji wyboru drogi (konstruowanych zgodnie z tą zasadą) nie występują cykle.

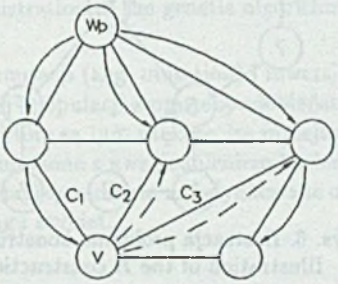
4.4. Algorytmy konstrukcji funkcji wyboru drogi

4.4.1. Algorytm dla metody zliczania dolin

W celu konstrukcji funkcji wyboru drogi zaproponowano algorytm składający się z dwóch etapów. W pierwszym etapie konstruowana jest funkcja pomocnicza $R^* : P \times P \mapsto 2^{C^*}$, gdzie przez C^* oznaczono zbiór fizycznych kanałów komunikacyjnych. Funkcja R^* wyznacza wszystkie najkrótsze drogi pomiędzy każdą parą wierzchołków sieci. W drugim etapie, z pomocą R^*, konstruowana jest wolna od wzajemnej blokady funkcja wyboru drogi $R : C \times P \mapsto 2^C$, gdzie C oznacza zbiór kanałów wirtualnych.

Algorytm konstrukcji R^* działa na zasadzie p-krotnego przeszukiwania „wszerz" sieci połączeń, począwszy od różnych wierzchołków początkowych w_p. W rezultacie pojedynczego przeszukiwania dla każdego wierzchołka $w \neq w_p$ określany jest zbiór wartości funkcji

$R^*(v, w_p)$, czyli pojedyncza „warstwa" funkcji R^*. W trakcie przeszukiwania, oprócz kanałów odpowiadających gałęziom drzewa rozpinającego (np. c_1 na rys. 5), rozważane są również kanały odpowiadające krawędziom zwrotnym (tj. krawędziom incydentnym do wierzchołków już przeszukanych, np. c_2, c_3 na rys. 5). Jeżeli długość drogi zawierającej krawędź zwrotną równa jest długości drogi należącej do drzewa rozpinającego, to kanał odpowiadający krawędzi zwrotnej włączany jest do zbioru wartości funkcji $R^*(v, w_p)$.



Rys. 5. Ilustracja działania algorytmu konstrukcji R^*
Fig. 5. Illustration of the R^* construction algorithm

W drugim etapie, w celu konstrukcji funkcji R, przeglądane są wszystkie drogi określone przez R^*. W trakcie przeglądania rozpoznawane są doliny oraz definiowane klasy kanałów. Oznaczmy przez $\alpha(i)$ numer przypisany dowolnemu wierzchołkowi i sieci, zaś przez ϵ_w oraz δ_w odpowiednio źródło i ujście wiadomości w dowolnym wierzchołku w. Wolną od wzajemnej blokady funkcję wyboru drogi R można wówczas zdefiniować następująco:

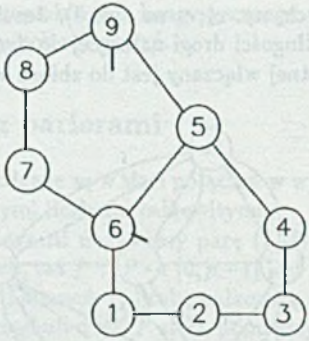
$$\forall_{n_d, i,j \in P}: c_{i,j} \in R^*(\epsilon_i, n_d) \Rightarrow c^0_{i,j} \in R(\epsilon_i, n_d),$$
$$\forall_{i,j \in P}: R^*(c_{i,j}, j) = \delta_j \Rightarrow \forall_{0 \le k < l}: R(c^k_{i,j}, j) = \delta_j,$$
$$\forall_{n_d, i,j,m \in P}: c_{j,m} \in R^*(c_{i,j}, n_d) \wedge \alpha(j) < \alpha(i) \wedge \alpha(j) < \alpha(l) \Rightarrow \forall_{0 \le k < l-1}: c^{k+1}_{j,m} \in R(c^k_{i,j}, n_d),$$
$$\forall_{n_d, i,j,m \in P}: c_{j,m} \in R^*(c_{i,j}, n_d) \wedge (\alpha(j) > \alpha(i) \vee \alpha(j) > \alpha(l)) \Rightarrow \forall_{0 \le k < l}: c^k_{j,m} \in R(c^k_{i,j}, n_d).$$

4.4.2. Algorytm dla sieci z barierami

Algorytm konstrukcji funkcji wyboru drogi, proponowany w przypadku metody sieci z barierami, działa na zasadzie „zmodyfikowanego" przeszukiwania grafów wszerz. Zastosowanie „zwykłego" przeszukiwania wszerz nie jest celowe, ponieważ ze względu na rozmieszczone w sieci bariery, wartość funkcji wyboru drogi R w dowolnym wierzchołku w zależy od tego, którym kanałem wiadomość „przybyła" do w.

Problem ten ilustruje rys. 6. Rozpoczynając przeszukiwanie wszerz od wierzchołka 9 i uwzględniając bariery rozmieszczone w sieci, wierzchołek 6 zostanie zawsze odwiedzony najpierw na skutek przeszukiwania prowadzonego z wierzchołka 5. W rezultacie, jeżeli uwzględniony zostanie fakt, że w wierzchołku 6 występuje bariera (pomiędzy krawędziami (5,6) i (6,1)), przeszukanie wierzchołka 6 nie spowoduje odwiedzenia wierzchołka 1. Wierzchołek 1 zostanie odwiedzony w wyniku przeszukiwania prowadzonego po

drodze $(9,5,4,3,2,1)$, pomimo że pomiędzy 9 i 1 istnieje krótsza niż $(9,5,4,3,2,1)$ droga bez barier: $(9,8,7,6,1)$.



Rys. 6. Ilustracja problemu konstrukcji R
Fig. 6. Illustration of the R construction problem

W celu konstrukcji funkcji wyboru drogi sieć połączeń przeszukiwana jest p razy, począwszy od różnych wierzchołków początkowych $w_p \in P$. W trakcie każdego przeszukiwania konstruowana jest pojedyncza warstwa funkcji wyboru drogi R, która wyznacza drogi z każdego wierzchołka $w \neq w_p$ do wierzchołka w_p. Zgodnie z proponowaną metodą zmodyfikowanego przeszukiwania wszerz dowolny wierzchołek $w \neq w_p$ jest przeszukiwany kilka razy, jednak nie więcej niż wynosi jego stopień. Przez analogię do „zwykłego" przeszukiwania wszerz w można uznać za „przeszukany", gdy wyczerpane zostały możliwości jego odwiedzenia i przeszukania ze wszystkich wierzchołków incydentnych.

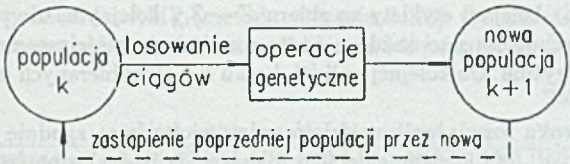
4.5. Algorytmy optymalizacji funkcji wyboru drogi

Od sposobu poetykietowania wierzchołków sieci zależy rozmieszczenie barier, a więc wartości d_m i d_{ave} lub (w przypadku metody „zliczania dolin") liczba klas kanałów wirtualnych przypadających na pojedynczy kanał fizyczny. W celu etykietowania sieci zaproponowano dwa algorytmy oparte na idei algorytmów genetycznych oraz tzw. „przeszukiwaniu wielowątkowym".

4.5.1. Algorytmy genetyczne

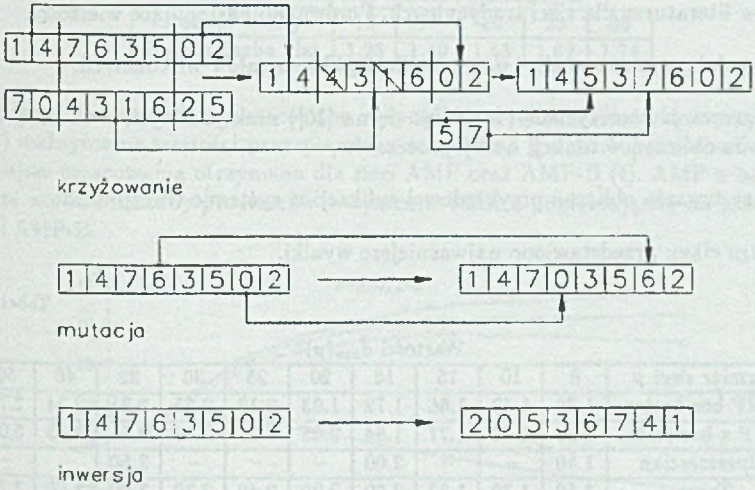
Działanie algorytmów genetycznych GA (ang. Genetic Algorithm) [6] polega na *jednoczesnym* przetwarzaniu populacji (ang. population) ciągów reprezentujących zakodowane rozwiązania problemu w celu maksymalizacji tzw. *funkcji dopasowania* (ang. fitness function). Zastosowanie idei algorytmów genetycznych wymagało określenia reprezentacji rozwiązań oraz zdefiniowania funkcji dopasowania.

W proponowanym algorytmie zastosowano reprezentację rozwiązań w formie ciągów liczb całkowitych z zakresu $\langle 0, p-1 \rangle$, określających etykietowanie wierzchołków sieci. Początkowa populacja ciągów generowana jest w sposób przypadkowy. Kolejne populacje tworzone są za pomocą „operacji genetycznych": reprodukcji (ang. reproduction),



Rys. 7. Ilustracja idei algorytmów genetycznych
Fig. 7. Illustration of the genetic algorithm idea

krzyżowania (ang. crossover), mutacji (ang. mutation) i inwersji (ang. inversion). W tym celu ciągi są losowane ze „starej" populacji z prawdopodobieństwem proporcjonalnym do wartości funkcji dopasowania, które są tym większe, im mniejsza jest liczba klas kanałów wirtualnych lub d_{ave} sieci (wyznaczone z uwzględnieniem barier). Ze względu na zastosowaną reprezentację rozwiązań zaproponowano zmodyfikowane operacje genetyczne, które „zachowują permutację" w ciągu etykiet.



Rys. 8. Zmodyfikowane operacje genetyczne
Fig. 8. Modified genetic operations

4.5.2. Przeszukiwanie wielowątkowe

W przypadku *przeszukiwania wielowątkowego*, w przeciwieństwie do typowych strategii zachłannych, kolejne rozwiązania częściowe wywodzone są nie z pojedynczego rozwiązania częściowego, ale z całej „puli" najlepszych rozwiązań częściowych.

Przez Z oznaczmy zbiór wszystkich etykiet, zaś przez $Z_{i,k}$ zbiór etykiet i-tego częściowego rozwiązania z puli n najlepszych rozwiązań w k-tym kroku algorytmu. Zgodnie z proponowaną metodą, w kroku $k+1$, rozwiązania częściowe otrzymywane są w wyniku

prób przydzielenia kolejnej etykiety ze zbioru $Z - Z_{i,k}$ kolejnym, nie poetykietowanym dotychczas, wierzchołkom sieci każdego i-tego rozwiązania częściowego z puli rozwiązań k-tego kroku algorytmu. Do kolejnej puli (tj. kroku $k+1$) wybieranych jest n najlepszych rozwiązań.

W każdym kroku rozwiązania częściowe wartościowane są zgodnie z kryterium minimalnej wartości d_{ave} (z uzględnieniem rozmieszczenia barier) albo średniej liczby klas kanałów wirtualnych, przypadających (po poetykietowaniu części wierzchołków) na pojedynczy kanał fizyczny. Przeszukiwanie kończy się wybraniem najlepszego rozwiązania z puli rozwiązań finalnych.

5. Otrzymane wyniki oraz praktyczne przykłady

Aby ocenić skuteczność proponowanych metod i algorytmów porównano funkcje wyboru drogi wygenerowane dla sieci AMP o różnych rozmiarach z rozwiązaniami proponowanymi w literaturze dla sieci tradycyjnych. Porównano następujące wielkości:

- wartości parametrów d_m i d_{ave} oraz liczbę klas kanałów wirtualnych,
- oszacowania teoretycznie (wzorując się na [10]) maksymalnych wartości przyspieszenia obliczeń w funkcji liczby procesorów,
- czasy trwania obliczeń przykładowej aplikacji w systemie transputerów.

W dalszym ciągu przedstawiono najważniejsze wyniki.

Tabela 1

Wartości $d_{ave}(p)$

rozmiar sieci p	8	10	15	16	20	25	30	32	40	50
AMP bez barier	1.28	1.42	1.66	1.72	1.93	2.12	2.25	2.31	2.54	2.70
AMP z barierami	1.28	1.42	1.77	1.84	2.05	2.27	2.45	2.55	2.83	3.02
Hipersześcian	1.50	–	–	2.00	–	–	–	2.50	–	–
Torus	1.50	1.70	1.87	2.00	2.20	2.40	2.70	3.00	3.50	3.70

W tabeli 1 zestawiono wartości $d_{ave}(p)$ AMP, AMP z barierami oraz dwóch tradycyjnych sieci połączeń o najmniejszych wartościach d_{ave}. Pomimo wprowadzenia barier wartości d_{ave} sieci AMP pozostały mniejsze niż torusa (bez zastosowania jakiejkolwiek metody zapobiegania blokadzie) oraz hipersześcianu o liczbie procesorów $p \leq 16$ (wolny od wzajemnej blokady algorytm e-cube [4], który nie przyczynia się do wzrostu d_{ave}). Należy zaznaczyć, że porównanie AMP i hipersześcianu dla liczby procesorów większej niż kilkanaście nie jest miarodajne, bowiem stopień wierzchołków hipersześcianu jest wówczas wyższy niż $k_{AMP} = 4$.

W tabeli 2 zestawiono wartości niezbędnej liczby klas kanałów wirtualnych, przypadających na pojedynczy kanał fizyczny torusa i hipersześcianu (patrz [1]), oraz jej górne

oszacowania otrzymane dla AMP (metoda zliczania dolin). Oszacowanie otrzymane dla AMP jest nie gorsze, niż wartości znane dla torusa i hipersześcianu.

Tabela 2

Liczba klas kanałów wirtualnych

rozmiar sieci p	8	10	15	16	20	25	30	32	40	50	64
AMP	2	2	2	2	2	2	2	2	3	3	3
Hipersześcian	2	–	–	3	–	–	–	3	–	–	4
Torus	2	3	3	3	3	3	3	3	3	3	4

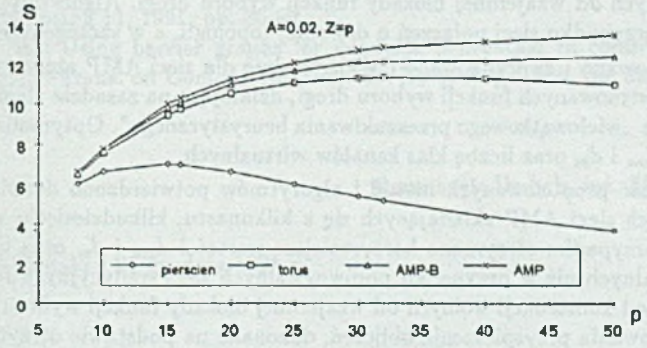
Liczbę klas kanałów wirtualnych można zmniejszyć stosując proponowane metody optymalizacji. W tabeli 3 zestawiono otrzymane doświadczalnie dla AMP wartości średniej liczby klas kanałów wirtualnych, przypadających na pojedynczy kanał fizyczny. Wartości te są niższe niż ich górne oszacowanie.

Tabela 3

Średnia liczba klas kanałów wirtualnych

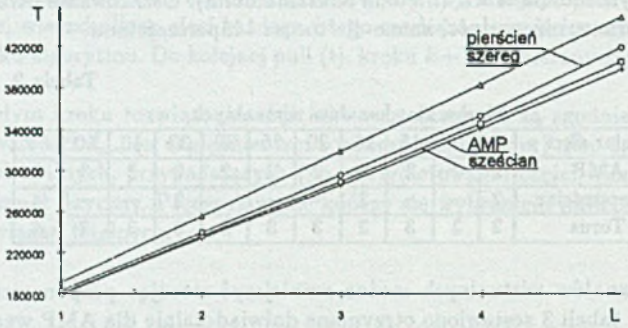
rozmiar sieci p	10	15	20	25	30
średnia liczba klas	1.26	1.40	1.53	1.62	1.74

Na rys. 9 przedstawiono oszacowane w sposób teoretyczny (metodą zaproponowaną w [10]) maksymalne wartości przyspieszenia w systemie $p \in [2..50]$ procesorów. Najkorzystniejsze oszacowania otrzymano dla sieci AMP oraz AMP-B (tj. AMP z barierami). Wraz ze wzrostem liczby procesorów w systemie różnice pogłębiają się na korzyść sieci AMP i AMP-B.



Rys. 9. Maksymalna wartość przyspieszenia w funkcji liczby procesorów p
Fig. 9. Maximum speed-up as a function of p

Na wykresie 10 przedstawiono czasy wykonania równoległej implementacji algorytmu genetycznego w *modelu wyspowym* (ang. island model), zmierzone w systemie transputerów T800, składającym się z $p = 8$ procesorów. Najkrótsze czasy wykonania programów otrzymano w przypadku, gdy zastosowano sieci AMP z barierami.



Rys. 10. T [takt] w funkcji liczby migrujących ciągów
Fig. 10. T [tick] as a function of the migrating strings number

6. Wnioski i uwagi

Zaproponowano i przebadano metodę *sieci z barierami* – zapobiegania wzajemnej blokadzie w sieci połączeń międzyprocesorowych – oryginalną modyfikację metody opracowanej dla sieci komputerowych [11]. Metoda ta może być zastosowana w przypadku sieci połączeń o dowolnych topologiach, a w szczególności AMP.

Dla metod: zliczania dolin i sieci z barierami opracowano oryginalne algorytmy konstrukcji wolnych od wzajemnej blokady funkcji wyboru drogi. Algorytmy te można zastosować w przypadku sieci połączeń o dowolnej topologii, a w szczególności sieci AMP.

Zaproponowano i zweryfikowano doświadczalnie dla sieci AMP algorytmy optymalizacji tak konstruowanych funkcji wyboru drogi, działające na zasadzie algorytmów genetycznych oraz „wielowątkowego przeszukiwania heurystycznego". Optymalizacji poddano parametry d_{ave} i d_m oraz liczbę klas kanałów wirtualnych.

Skuteczność proponowanych metod i algorytmów potwierdzono doświadczalnie dla przykładowych sieci AMP składających się z kilkunastu, kilkudziesięciu wierzchołków. W każdym przypadku otrzymano korzystniejsze wartości d_{ave} i d_m oraz liczby klas kanałów wirtualnych niż w przypadku porównywalnych sieci tradycyjnych i znanych z literatury metod konstrukcji wolnych od wzajemnej blokady funkcji wyboru drogi. Teoretyczne oszacowania przyspieszenia obliczeń, dokonane na podstawie otrzymanych wyników, wskazują, że zastosowanie sieci AMP oraz proponowanych algorytmów komunikacji pozwala osiągnąć wartości przyspieszenia **większe o kilka do kilkunastu procent** od osiąganych przy zastosowaniu sieci tradycyjnych i znanych z literatury algorytmów komunikacji.

Teoretyczne oszacowania przyspieszenia potwierdzono w praktyce za pomocą przykładowej aplikacji w systemie $p = 8$ transputerów T800, porównując czasy wykonania programu otrzymane w przypadku sieci AMP z barierami oraz tradycyjnych sieci połączeń i znanych z literatury metod zapobiegania blokadzie.

LITERATURA

[1] Adamo J., Alhafez N.: Minimal adaptive and deadlock-free routing for multiprocessors, LIP Report, No 91-25, 1991.

[2] Chalmers A.G., Gregory S.: Constructing minimum path configurations for multiprocessor systems, Parallel Computing 91, 1993, pp. 343-355.

[3] Czarnożyński A.: Metoda zapobiegania wzajemnej blokadzie w sieciach AMP, Archiwum Informatyki Teoretycznej i Stosowanej, w przygotowaniu.

[4] Dally W.J., Seitz C.L.: *Deadlock-free message routing in multiprocessor interconnection networks*, IEEE Trans. on Comp., Vol. C-36, No 5, May 1987, pp. 547-553.

[5] Douato J.: On the design of deadlock-free adaptive routing algorithms for multicomputers: theoretical aspects, Proc. 2nd European Conference on Distributed Memory Computing , April 1991, pp. 234-243.

[6] Holland J.: Adaptation in natural and artificial systems, Ann Arbor: University of Michigan Press, 1975.

[7] Jesshope H.: High performance communication architectures, in: Parallel and Distributed Processing, K. Boyanow ed., Elsevier Science Publishers B.V. (North Holland), 1991, pp. 7-25.

[8] Linder D.H., Harden J.C.: An adaptive and fault tolerant wormhole routing strategy for k-ary n-cubes, IEEE Trans. on Comp., Vol. 40, No 1, Jan. 1991, pp. 2-12.

[9] Reeves D.S., Gehringer E.F., Chandiramani A.: Adaptive routing and deadlock recovery: a simulation study, Proc. 3rd Conf. on Hypercube Concurrent Computers and Applications, Monterey, CA, March 1989.

[10] Scherson I.D., Corbett P.F.: Communication overhead and the exspected speedup of multidimensional mesh-connected parallel processors, Journal of Parallel and Distributed Computing 11, 1991, pp. 86-96.

[11] Wimmer W.: Using barrier graphs for deadlock prevention in communication networks, IEEE Trans. on Comm., Vol. Com-32, No 8, Aug. 1984, pp. 897-901.

Recenzent: Dr hab. inż. Zbigniew Czech

Wpłynęło do Redakcji 21 grudnia 1995 r.

Abstract

Many deadlock-free routing algorithms have been developed so far for regular interconnection networks of multiprocessors. In the article the problem of deadlock-free message routing in irregular networks is discussed. The solution and original algorithms of deadlock-free routing functions construction and optimisation are proposed for a class of irregular networks called AMP (A Minimal Path systems). For deadlock prevention

there are proposed two alternative methods: *barrier networks* (a modification of the barrier graphs method developed for computer networks [11]) and *valley counting* method [1]. Algorithms for construction of routing functions are based on graph searching. Optimisation is based on the idea of genetic algorithms and modified hill-climbing. The final results show that it is possible to obtain routing functions that could preserve the benefits of using AMP configurations as the processor interconnection strategy.