



**INSTYTUT BADAŃ SYSTEMOWYCH  
POLSKIEJ AKADEMII NAUK**

**ANALIZA SYSTEMOWA W FINANSACH  
I ZARZĄDZANIU**

Wybrane problemy  
Tom 11

Pod redakcją  
Jerzego HOŁUBCA

Warszawa 2009



**INSTYTUT BADAŃ SYSTEMOWYCH  
POLSKIEJ AKADEMII NAUK**

**ANALIZA SYSTEMOWA W FINANSACH  
I ZARZĄDZANIU**

Wybrane problemy  
Tom 11

Pod redakcją  
Jerzego HOŁUBCA

Warszawa 2009

Wykaz opiniodawców artykułów zamieszczonych  
w niniejszym tomie:

prof. dr hab. inż. Jerzy HOŁUBIEC  
dr inż. Lech KRUŚ  
doc. dr hab. inż. Wiesław KRAJEWSKI  
doc. dr hab. Jacek MALINOWSKI  
dr inż. Edward MICHALEWSKI  
prof. dr Adam SKOREK  
dr hab. Ryszard SMARZEWSKI  
prof. dr hab. inż. Andrzej STRASZAK  
dr Dominik ŚLĘZAK  
prof. dr hab. inż. Stanisław WALUKIEWICZ  
doc. dr hab. Sławomir ZADROŻNY

© Instytut Badań Systemowych PAN  
Warszawa 2009

**ISBN 9788389475220**

Druk: Zakład Poligraficzny Jerzy Kosiński, Warszawa

# ZASTOSOWANIE METODY SYMULOWANEGO WYŻARZANIA DO OPTYMALIZACJI PRZEPUSTOWOŚCI SIECI KOMPUTEROWEJ

**Robert Dekowski**

*Studia Doktoranckie IBS PAN*

*W artykule zostało przedstawione zastosowanie zmodyfikowanej metody symulowanego wyżarzania do optymalizacji przepustowości sieci teleinformatycznej. Metoda ta wykorzystuje algorytm Nicolasa Metropolisa zmodyfikowany poprzez wprowadzenie parametru „przepustowość routera”. Na podstawie zmian wartości tego parametru zmienne decyzyjne są losowane nie z pełnego zbioru wartości dopuszczalnych, lecz z odpowiednio określonego ich podzbioru. W opracowaniu tym zostały także przedstawione autorskie algorytmy służące do budowy tablicy routingu z udziałem tras o nierównym koszcie oraz wyznaczania przepustowości środowiska sieciowego. Dodatkowo opracowanie zawiera opisy wybranych parametrów badawczych wraz ze scenariuszami przykładowych badań.*

*Reasumując, artykuł ten przedstawia koncepcję scentralizowano-rozproszoną metody optymalizacji przepustowości sieci teleinformatycznej z wykorzystaniem algorytmu symulowanego wyżarzania.*

**Słowa kluczowe:** *Symulowane wyżarzanie, optymalizacja przepustowości sieci teleinformatycznej, algorytm Metropolisa, równoważenie obciążenia z udziałem tras o nierównym koszcie, wyznaczanie przepustowości środowiska sieciowego, algorytm budowy tablicy routingu.*

## 1. Ogólny opis metody symulowanego wyżarzania

Metoda symulowanego wyżarzania zawdzięcza swoją nazwę zjawisku fizycznemu, które polega na stopniowym schładzaniu ciała będącego początkowo w postaci płynnej, aż do jego zestalenia. Dobrym przykładem jest np. wytapianie żelaza. W tym miejscu warto wspomnieć, że inspiracją do powstania takiej metody był proces technologiczny stosowany w metalurgii i hutnictwie. Będąc w stanie płynnym cząstki metalu mają możliwość dość swobodnego poruszania się, natomiast wraz ze spadkiem temperatury ich energia (kinetyczna), a więc zdolności przemieszczania, zmniejszają się. W momencie zestalenia powstaje tzw. sieć krystaliczna, a ruch cząstek niemalże ustaje – ciało wchodzi w stan równowagi termo-

dynamicznej. W stygnącej cieczy poruszające się cząstki metalu szukają „swojego miejsca”, czyli dążą do stworzenia takiej struktury, w której energia wewnętrzna całego ciała jest najmniejsza. To, jaką strukturę wewnętrzną (w jakim stopniu uporządkowaną) zbudują cząstki zależy od temperatury początkowej i końcowej, a także od szybkości oraz sposobu schładzania. Główna idea omawianej metody polega na zastosowaniu rozwiązań występujących w procesach technologicznych (wyżarzanie) do rozwiązania innych problemów, w tym także optymalizacyjnych. Dobór właściwego miejsca dla cząstek w całej strukturze może być traktowany jako swoista optymalizacja.

## 2. Szczegółowy opis metody badawczej

Powstaje zatem pytanie jak dostosować metodę symulowanego wyżarzania do dziedziny problemów optymalizacyjnych. Można tego dokonać poprzez następujące analogie: minimum globalne – stan o minimalnej energii, iteracja algorytmu (przekształcenie lokalne) – ruch cząstki, funkcja celu (oceny) – energia układu, otrzymane rozwiązanie – stan układu termodynamicznego. Dodatkowo wprowadza się zamiast stałej Boltzmana oraz rzeczywistej temperatury odpowiedni parametr służący dostrojeniu algorytmu do określonego zadania. W tym miejscu warto wspomnieć, jaka jest rola tego parametru (zwanego dalej temperaturą) w procesie wyżarzania. Dobierając odpowiednią jego wartość (większą od zera) ustala się zakres tolerancji dla zaakceptowania gorszego rozwiązania, w sensie wartości przyjętej funkcji oceny, powstałego w kolejnej iteracji algorytmu. To właśnie ta tolerancja umożliwia algorytmowi opuszczenie ekstremum lokalnego w celu wyszukania ekstremum globalnego – co niewątpliwie jest jego istotną zaletą. Kontynuując rozważania nad zastosowaniem algorytmu do zadań optymalizacyjnych warto omówić kwestie takie jak: warunki startowe oraz schematy schładzania. Jeżeli chodzi o kwestię pierwszą to powszechnie uważa się, że algorytm ten należy stosować w powiązaniu z tzw. wielostartem, gdyż działa on jak „wzmacniacz prawdopodobieństwa” wylosowania dobrego rozwiązania. Wspomniany wielostart należy rozumieć jako wielokrotne uruchamianie algorytmu z różnymi wartościami początkowymi. W odniesieniu do schematu schładzania, najczęściej stosuje się jeden z trzech następujących:

- logarytmiczny (Boltzmana):  $T(k) = T_p * 1/\log(k)$ ;
- liniowy (Cauchy’ego):  $T(k) = T_p * 1/k$ ;
- geometryczny  $T(k) = T_p * a^k$ , gdzie  $0 < a < 1$ ;

gdzie  $T_p$  – temperatura początkowa,  $T(k)$  – temperatura w iteracji k-tej, „a” – podstawa funkcji wykładniczej.

Schemat logarytmiczny (w odróżnieniu od pozostałych) przy pewnych naturalnych założeniach gwarantuje znalezienie optimum globalnego. Średni czas potrzebny na jego wyszukanie jest porównywalny z rozmiarem przestrzeni rozwiązań, zaś rozmiar ten zależy od tego do rozwiązania jakiego problemu metoda została użyta. Natomiast badania empiryczne sugerują, że największą przydatność praktyczną ma schemat geometryczny (najszybszy). Omawiając metodę symulowanego wyżarzania warto jeszcze wspomnieć o warunkach zatrzymania algorytmu. Warunków tych może być kilka i mogą one występować łącznie lub rozdzielnie. Pierwszym jest osiągnięcie wcześniej założonej temperatury końcowej. Drugim przekroczenie limitu czasu pracy algorytmu lub założonej liczby iteracji. Kolejnym warunkiem jest osiągnięcie stanu równowagi termicznej dla bieżącej temperatury – jest to warunek uznany za mało konstruktywny. Poniżej został zamieszczony ogólny schemat dla omawianego algorytmu.

BEGIN

$T := T(0);$

$k := 0;$

$x_{\text{BEST}} := x_0;$

$\phi_{\text{BEST}} := \phi(x_0);$

REPEAT

$x := \text{losowy element z } S;$

wyznacz  $\phi(x);$

IF  $(\phi(x); \leq \phi_{\text{BEST}})$  THEN

$\phi_{\text{BEST}} := \phi(x);$

$x_{\text{BEST}} := x;$

ELSE

IF  $\text{RM}(\phi(x), \phi_{\text{BEST}}, T)$  TRUE THEN

$\phi_{\text{BEST}} := \phi(x);$

$x_{\text{BEST}} := x;$

$k := k+1;$

$T := T(k);$

UNTIL  $(T > T_{\text{MIN}} \text{ OR } k < k_{\text{MAX}})$  TRUE;

return  $x$

END

Gdzie  $RM(\phi(x), \phi_{BEST}, T)$  jest to tzw. reguła akceptacji Metropolisia [93][98][99] (dodatkowo opisana w podrozdziale 4.2) zapisana jako:

$RM(\phi(x), \phi_{BEST}, T)$

IF random  $\leq p(\phi(x), \phi_{BEST}, T)$  THEN RM := TRUE

ELSE RM := FALSE

Gdzie  $p(\phi(x), \phi_{BEST}, T) = \exp\left(-\frac{|\phi_{BEST} - \phi(x)|}{T}\right)$ , random – wartość losowa o rozkładzie jednostajnym z przedziału  $\langle 0, 1 \rangle$ .

Ponadto w algorytmie występują następujące oznaczenia:

T – temperatura w bieżącej iteracji algorytmu;

T(0) – temperatura początkowa (w zerowej iteracji);

k – numer iteracji algorytmu;

k<sub>MAX</sub> – maksymalna liczba iteracji algorytmu;

T(k) – temperatura w k-tej iteracji algorytmu;

T<sub>MIN</sub> – temperatura minimalna po osiągnięciu której algorytm koczy działanie;

„x” – zmienna decyzyjna ze zbioru wartości dopuszczalnych „S”;

„x<sub>0</sub>” – wartość początkowa zmiennej decyzyjnej należącej do zbioru S;

„x<sub>BEST</sub>” - zmienna decyzyjna ze zbioru wartości dopuszczalnych „S”, dla której wartość funkcji energii przyjmuje wartość najmniejszą;

$\phi(x)$  – wartość funkcji energii dla zmiennej decyzyjnej „x”;

$\phi_{BEST}$  – najmniejsza wartość funkcji energii.

Więcej informacji na temat samej metody symulowanego wyżarzania, a także zastosowania jej do zadań optymalizacji znajduje się w [7] oraz [8].

### 3. Zmienne decyzyjne

W proponowanym rozwiązaniu tablice routingu poszczególnych routerów powinny na stałe zawierać wpisy odnośnie wszystkich tras optymalnych do każdej sieci składowej środowiska – obszaru OSPF. Dodatkowo w sytuacji, kiedy pojawiają się przeciążone interfejsy, do tablicy tej powinny zostać dodane inne (o gorszej metryce) trasy prowadzące do tych sieci. W takiej sytuacji dla każdej z sieci w tablicy routingu występuje trasa optymalna i trasy pozostałe (o ile istnieją), gdzie

## R. Dekowski – Zastosowanie metody symulowanego wyżarzania do optymalizacji...

każda z tras prowadzi przez inny interfejs wyjściowy (posiada inny adres routera następnego skoku). Trasa alternatywna prowadząca do danej sieci zostaje dodana do tablicy routingu gdy spełnia warunek konieczny tj. przedstawiony wzorem nr 1.

$$c(r) \leq c_{opt} * l_{pi} * \beta \quad (1)$$

gdzie:

$c(r)$  – koszt analizowanej trasy;

$c_{opt}$  – koszt trasy optymalnej;

$l_{pi}$  – liczba przeciążonych interfejsów, przez które prowadzą trasy do sieci docelowej;

$l_{pi} \geq 1$  – mechanizm równoważenia obciążenia z udziałem tras typu NK działa tylko wówczas gdy przynajmniej jeden interfejs jest przeciążony. Podczas wyznaczania wartości współczynników beta za pomocą symulowanego wyżarzania współczynniki  $l_{pi}$  powinny zostać ustawione na swoje wartości maksymalne. Działanie takie ma na celu symulację maksymalnego obciążenia sieci przed rozpoczęciem optymalizacji jej przepustowości. Natomiast  $\beta$  jest parametrem służącym do dostrojenia proponowanego rozwiązania do istniejącej topologii oraz charakterystyki ruchu sieciowego. Każdy router z danego obszaru posiada jeden przypisany do niego parametr beta. Wartości tych parametrów generowane są centralnie przez stację zarządzającą i co pewien czas dystrybuowane do wszystkich zarządzanych routerów. Sposób generowania wspomnianych parametrów jest tak dobrany żeby globalnie maksymalizować przepustowość całego obszaru (z uwzględnieniem spodziewanej charakterystyki ruchu sieciowego w danym interwale czasu) i stanowi główny przedmiot badań poruszonych w tym artykule. Wyznaczanie wartości parametrów beta jest realizowane z zastosowaniem metody symulowanego wyżarzania. Dodatkowo,  $\beta$  zależy od maksymalnego kosztu trasy ( $c_{max}(r)$ ) z danego routera do dowolnej sieci docelowej w obrębie obszaru OSPF i przyjmuje wartości z przedziału  $0 < \beta \leq \beta_{max}$  gdzie  $\beta_{max}$  jest wyrażone wzorem (2).

$$\beta_{max} = \frac{c_{max}(r)}{(n_i - 1)c_{opt}} \quad (2)$$

przy czym  $n_i$  jest maksymalną liczbą interfejsów, przez które prowadzą trasy (optymalna i alternatywne) do sieci docelowej. Schemat blokowy algorytmu budowy tablicy routingu został przedstawiony na rys. 1. W algorytmie tym występuje funkcja o nazwie "Dijkstra", której argumentami są baza topologii sieci (a dokładniej jej kopia) dla danego obszaru OSPF oraz adres IP sieci do której szukana jest



najkrótsza trasa. Funkcja ta zwraca koszt osiągnięcia danej sieci oraz adres IP routera następnego skoku. Zwracane wartości są zapisywane w tablicy [koszt, trasa] przy czym jej rozmiar jest zmienny. Zmienność ta jest spowodowana faktem, że funkcja „Dijkstra” wyznacza wszystkie ścieżki o najniższym koszcie do danej sieci – liczba tych ścieżek może być inna dla każdej z sieci. Danymi wejściowymi dla algorytmu są:

- wartość zmiennej  $l_{pi}$  - liczba przeciążonych interfejsów;
- wartość parametru  $X$  – maksymalna liczba tras, dla których przeprowadzane jest równoważenie obciążenia – parametr definiowany przez administratora routera.

Dodatkowo w algorytmie tym zmienna  $i$  jest używana do kontroli liczby tras dodawanych do tablicy routingu. W chwili rozpoczęcia pracy jej wartość jest ustalana jako zero.

#### 4. Algorytmu wyznaczania przepustowości środowiska sieciowego

Dla potrzeb algorytmu przyjęto następujące oznaczenia.

$T_{ij}(n)$  – trasa komunikacji (ciąg łączy pomiędzy routerami i sieciami) między sieciami o indeksach

„ $i$ ” i „ $j$ ”;  $n$  jest numerem trasy, na której odbywa się komunikacja,

$P(l)$  – nominalna przepustowość łącza  $l$ , mierzona w [Mb/s].

$LT(l)$  – liczba tras które prowadzą przez łącze  $l$  (dla wszystkich komunikujących się par sieci).

$EP(l)$  – efektywna przepustowość łącza  $l$ , liczona jako iloraz  $EP(l) = \frac{P(l)}{LT(l)}$ , mie-

rzona w [Mb/s]. Dodatkowo można wprowadzić współczynnik wykorzystania (obciążenia) łącza przez daną trasę, co także wpłynie na jej efektywną przepustowość. Wprowadzenie takiego współczynnika daje pewną elastyczność przy dzieleniu dostępnego pasma na poszczególne trasy. Natomiast w niniejszym opracowaniu, (dla uproszczenia) zakłada się jednakowe wykorzystanie dostępnego pasma (nominalnej przepustowości łącza) przez wszystkie trasy z niego korzystające.

$EP_{min}(T_{ij}(n))$  – efektywna przepustowość „najwolniejszego” łącza na trasie  $T_{ij}(n)$  (mierzona w Mb/s).



1. Przy zadanych parametrach wyznaczyć (wszystkie spełniające te parametry) trasy routingu.  $\forall (i, j) \in K, \forall n \in N(i, j)$ ;

$$T_{ij}(n) = \text{wyznacz\_trasy}(i, j)$$

w następstwie powyższych operacji otrzymujemy  $T_{ij}(n) = \{l_a, l_b, \dots, l_z\}$

2. Dla każdego z łączy obliczyć w ilu trasach routingu ono występuje.

$$\text{dla } \forall l \in T_{ij}(n); LT(l) = \sum_{(i,j) \in K} \sum_{1 \leq n \leq N(i,j)} I_T(l) \text{ gdzie } I_T(l) = \begin{cases} 1 & l \in T \\ 0 & l \notin T \end{cases}$$

3. Dla każdej z tras obliczyć efektywną przepustowość łączy wchodzących w jej skład.

$$\forall l \in T_{ij}(n), \forall (i, j) \in K, \forall n \in N(i, j); EP(l) = \frac{P(l)}{LT(l)}$$

4. Dla każdej z tras wyznaczyć przepustowość „najwolniejszego” łącza wchodzącego w jej skład (najwęższego gardła)

$$\forall l \in T_{ij}(n), \forall (i, j) \in K, \forall n \in N(i, j); EP_{\min}(T_{ij}(n)) = \min(EP(l_a), EP(l_b), \dots, EP(l_z)),$$

gdzie  $T_{ij}(n) = (l_a, l_b, \dots, l_z)$

5. Obliczyć przepustowość komunikacji Sieci A z Siecią B jako sumę najmniejszych efektywnych przepustowości łączy wchodzących w skład wszystkich tras, którymi przebiega komunikacja.

$$\forall (i, j) \in K; P_{ij} = \sum_{1 \leq n \leq N(i,j)} EP_{\min}(T_{ij}(n))$$

6. Obliczyć przepustowość środowiska sieciowego jako sumę przepustowości wszystkich komunikujących się z sobą par sieci.

$$PSS = \sum_{(i,j) \in K} P_{ij}$$

Kiedy sieci dowolnej z komunikujących się par zakończą wymianę informacji, wtedy warunki panujące w analizowanym środowisku ulegną zmianie i część łączy zostanie zwolniona lub odciążona. W takiej sytuacji należy ponownie uruchomić cały proces wyznaczania przepustowości, lecz już z innymi (zaktualizowanymi) warunkami początkowymi.

Powyższy werbalny opis algorytmu przekłada się na zapis w formie następującego pseudokodu:

**procedura** WyznaczPSS( $T_{ij}(n)$ ,  $P(l)$ )

```
1 forall  $l \in T_{ij}(n)$  do
    if  $l \in T_{ij}(n)$  then
         $LT(l) \leftarrow LT(l) + 1$ ;
    end if
end forall
forall  $l \in T_{ij}(n)$  do
     $EP(l) \leftarrow P(l) / LT(l)$ ;
end forall
forall  $T_{ij}(n)$  do
     $EP_{\min}(T_{ij}(n)) \leftarrow \infty$ ;
    forall  $l \in T_{ij}(n)$  do
        if  $EP(l) < EP_{\min}(T_{ij}(n))$ ;
             $EP_{\min}(T_{ij}(n)) \leftarrow EP(l)$ ;
        end if
    end forall
end forall
forall  $T_{ij}(n)$  do
     $P_{ij} \leftarrow 0$ ;
    forall  $n \in N$  do
         $P_{ij} \leftarrow P_{ij} + EP_{\min}(T_{ij}(n))$ ;
    end forall
     $PSS \leftarrow PSS + P_{ij}$ ;
end forall
end WyznaczPSS.
```

Jak już wcześniej wspomniano, algorytm ten należy traktować jako „narzędzie” służące do mierzenia jakości (wydajności) działania głównego algorytmu służącego do optymalizacji przepływu ruchu sieciowego w obszarze OSPF.

Dodatkowo można wyliczyć czasy komunikacji dla poszczególnych par sieci.

$$t_{k(ij)} = \frac{\text{ilosc\_wymienianych\_danych}}{P_{ij}} \text{ [s].}$$

Wzór na przepustowość środowiska sieciowego (PSS) przyjmuje postać (3) lub tożsamą (4)

$$PSS = \sum_{(i,j) \in K} \sum_{1 \leq n \leq N(i,j)} \left[ \min \left( \frac{P(l_a)}{LT(l_a)}, \frac{P(l_b)}{LT(l_b)}, \dots, \frac{P(l_z)}{LT(l_z)} \right) \right] \quad (3)$$

lub

$$PSS = \sum_{(i,j) \in K} \sum_{1 \leq n \leq N(i,j)} [EP_{\min}(T_{ij}(n))] \quad (4)$$

Funkcjonał określony wzorem (3) lub (4) jest funkcją celu zagadnienia optymalizacji. W dalszej części artykułu rozważane będzie zagadnienie maksymalizacji tego funkcyjonału za pomocą metody symulowanego wyżarzania. Dodatkowo warto wspomnieć, że omawiany funkcyjonał jest to tzw. (ang. Convex Piecewise Linear Function), co można przetłumaczyć jako funkcja wypukła, kawałkami liniowa, czyli składająca się w obrębie swojej dziedziny z połączonych pod różnym kątem prostych (funkcji liniowych). W miejscu tym należy zaznaczyć, że istnieją inne powszechnie znane algorytmy wyznaczania w grafach maksymalnej przepustowości (przepływu) pomiędzy źródłem „s”, a miejscem przeznaczenia „d”. Przykładami mogą być algorytmy Forda-Fulkersona, Edmonsa-Karpa lub Dinica [9][10][11][12]. Niestety wszystkie te algorytmy działają przy założeniu, że ruch odbywa się tylko między dwoma węzłami sieci, co w rzeczywistych sieciach komputerowych występuje niezwykle rzadko. Do zagadnienia wyznaczania maksymalnej przepustowości (przepływu) w danym grafie, przy założeniu jednoczesnej komunikacji wielu par sieci ze sobą, można także podejść w inny sposób. Mianowicie, za pomocą jednego ze wzmiankowanych algorytmów należy wyznaczyć maksymalną przepustowość komunikacji dla pierwszej pary sieci, a następnie, uwzględniając już obciążenia łączy powtórzyć obliczenia dla wszystkich pozostałych par. Uwzględnienie obciążonych łączy polegałoby na tym, że przed każdym kolejnym uruchomieniem algorytmu modyfikowane byłyby dostępne ich przepustowości. Podejście takie posiada jednak pewną zasadniczą wadę, polegającą na tym, że pary sieci, dla których obliczenia zostały wcześniej wykonywane stają się „faworyzowane – posiadają większy dostęp do pasma sieciowego” w stosunku do pozostałych. Faworyzacja taka nie odzwierciedla rzeczywistego działania protoko-

łów IP/ICMP, w wyniku czego omawiana metoda posiada wątpliwe znaczenie praktyczne.

## **5. Propozycja udoskonalenia algorytmu symulowanego wyżarzania**

Do przeprowadzenia optymalizacji przepustowości środowiska sieciowego można zastosować algorytm Metropolis. Ze względu na zasługi i wkład w rozwój nauki zasadnym jest krótkie przybliżenie osoby Nicolasa Metropolis (1915-1999). Był on członkiem zespołu badawczego Projektu Manhattan, a także współtwórcą komputerów MANIAC (1952) i MANIAC II (1957) oraz jednym z autorów metody Monte Carlo (wraz z S. Ulamem i J. Von Neumannem). Algorytm Metropolis zaliczony został do 10 czołowych algorytmów, które wywarły „największy wpływ na rozwój i praktykę nauki i techniki w XX wieku” (wg Computing Science and Engineering). Ponieważ algorytm ten nie jest rozwiązaniem dedykowanym do optymalizacji przepustowości sieci komputerowych, dlatego należy wprowadzić do niego pewne udoskonalenie. Polega ono na dodaniu do zbioru parametrów *przepustowości routera*. Jeżeli po wykonaniu iteracji algorytmu przepustowość ta wzrasta, to nowa wartość parametru beta nie jest losowana z całego zbioru wartości dopuszczalnych, lecz z podzbioru <wartość obecna, wartość maksymalna>. Analogicznie, gdy przepustowość routera spada, wartość beta jest losowana z przedziału <zero, wartość obecna>. Działanie takie ma na celu zapobieżenie zbyt szybkiemu wzrostowi wartości parametru beta, co w konsekwencji prowadzi do kierowania pakietów na zbyt wiele tras jednocześnie. Natomiast użycie zbyt wielu tras może skutkować spadkiem przepustowości środowiska sieciowego – co stosunkowo łatwo wykazać. Wracając do przepustowości routera należy wyjaśnić, że jest ona wyliczana w każdej iteracji algorytmu jako suma efektywnych przepustowości wszystkich tras przez ten router prowadzących. W omawianym rozwiązaniu maksymalizacja przepustowości całego obszaru OSPF odbywa się poprzez optymalizację przepustowości poszczególnych routerów wchodzących w jego skład, przy użyciu algorytmu, który w pewnych warunkach jest w stanie wyznaczyć ekstremum globalne.

Algorytm użyty do optymalizacji przepustowości składa się z następujących kroków.

1. Ustaw wszystkie bety na wartość zero.
2. Oblicz PSS.
3. Oblicz przepustowości routerów.
4. Ustaw nowe przepustowości routerów.
5. Dopóki nie jest spełniony warunek stopu wykonuj.

6. Wygeneruj nowe wartości parametrów beta.
7. Oblicz nowe wartości PSS.
8. Jeżeli nowe PSS nie gorsze niż PSS wykonaj
  - 8.1 Ustaw bety = nowe bety.
  - 8.2. Ustaw PSS = nowe PSS.
9. W przeciwnym wypadku do 8 wykonaj.
  - 9.1 Oblicz prawdopodobieństwo akceptacji (wartość funkcji energii).
  - 9.2 Jeżeli prawdopodobieństwo akceptacji większe niż wartość losowa wykonaj.
    - 9.2.1 Ustaw bety = nowe bety.
    - 9.2.2 Ustaw PSS = nowe PSS.
10. W przeciwnym przypadku do 9.2 idź do 11
11. Oblicz nowe przepustowości routerów.
12. Wyznacz routery ze wzrostem przepustowości.
13. Ustaw przepustowości routerów = nowe przepustowości routerów
14. Aktualizuj warunek stopu.
15. Sprawdź warunek stopu.
16. Jeżeli warunek stopu nie spełniony idź do 5, w przeciwnym przypadku Koniec.

Stosując wcześniej już przyjęte oznaczenia, a także poniżej przedstawione:

PSS\_new – nowo wyznaczona przepustowość środowiska sieciowego (w kolejnej iteracji).

Bety\_new – wektor zawierający nowo wyznaczone wartości parametrów beta (w kolejnej iteracji).

PR – wektor zawierający wartości przepustowości routerów.

PR\_new – wektor zawierający nowo wyznaczone wartości przepustowości routerów (w kolejnej iteracji).

E – wartość funkcji energii rozumiana jako  $\exp\left(-\frac{|PSS\_new - PSS|}{T}\right)$ ,

gdzie T jest wartością temperatury i zarazem parametrem stopu.

## R. Dekowski – Zastosowanie metody symulowanego wyżarzania do optymalizacji...

$T_{max}$  i  $T_{min}$  – temperatura początkowa i końcowa.

los – wartość losowa o rozkładzie jednostajnym z przedziału  $\langle 0, 1 \rangle$ .

pseudo kod algorytmu użytego do optymalizacji przepustowości wygląda następująco:

BEGIN

1.  $T \leftarrow T_{max}$ ;

2.  $Bety \leftarrow 0$ ;

3. Oblicz PSS;

4. Oblicz PR;

5.  $PR_{new} \leftarrow PR$ ;

6. REPEAT

7.     Wygeneruj  $Bety_{new}$ ;

8.     Oblicz  $PSS_{new}$ .

9.     IF ( $PSS_{new} \geq PSS$ )

10.          $Bety \leftarrow Bety_{new}$ ;

11.          $PSS \leftarrow PSS_{new}$ ;

12.     Else

13.         Oblicz E;

14.         IF ( $E > los$ )

15.              $Bety \leftarrow Bety_{new}$ ;

16.              $PSS \leftarrow PSS_{new}$ ;

17.     Oblicz  $PR_{new}$ ;

18.     Porównaj PR z  $PR_{new}$  // wyznaczenie routerów ze wzrostem przepustowości.

19.      $PR \leftarrow Pr_{new}$ ;

20.     Zmniejsz T;

21. UNTIL ( $T > T_{min}$ )

RETURN Bety

END

Dodatkowego wyjaśnienia może wymagać funkcja *Wygeneruj  $Bety_{new}$*  generująca nowe wartości parametrów beta. Są one generowane indywidualnie dla każdego routera, zależnie od tego, czy w ostatniej iteracji przepustowość danego routera uległa zmniejszeniu czy nie. Jeżeli przepustowość nie zmniejszyła się to



nowa wartość parametru beta jest generowana losowo z przedziału domkniętego  $\langle 0, \text{wartość\_obecna} \rangle$ . W przeciwnym przypadku nowa wartość parametru beta dla jest generowana losowo z przedziału domkniętego  $\langle \text{wartość\_obecna}, \text{wartość\_maksymalna} \rangle$ .

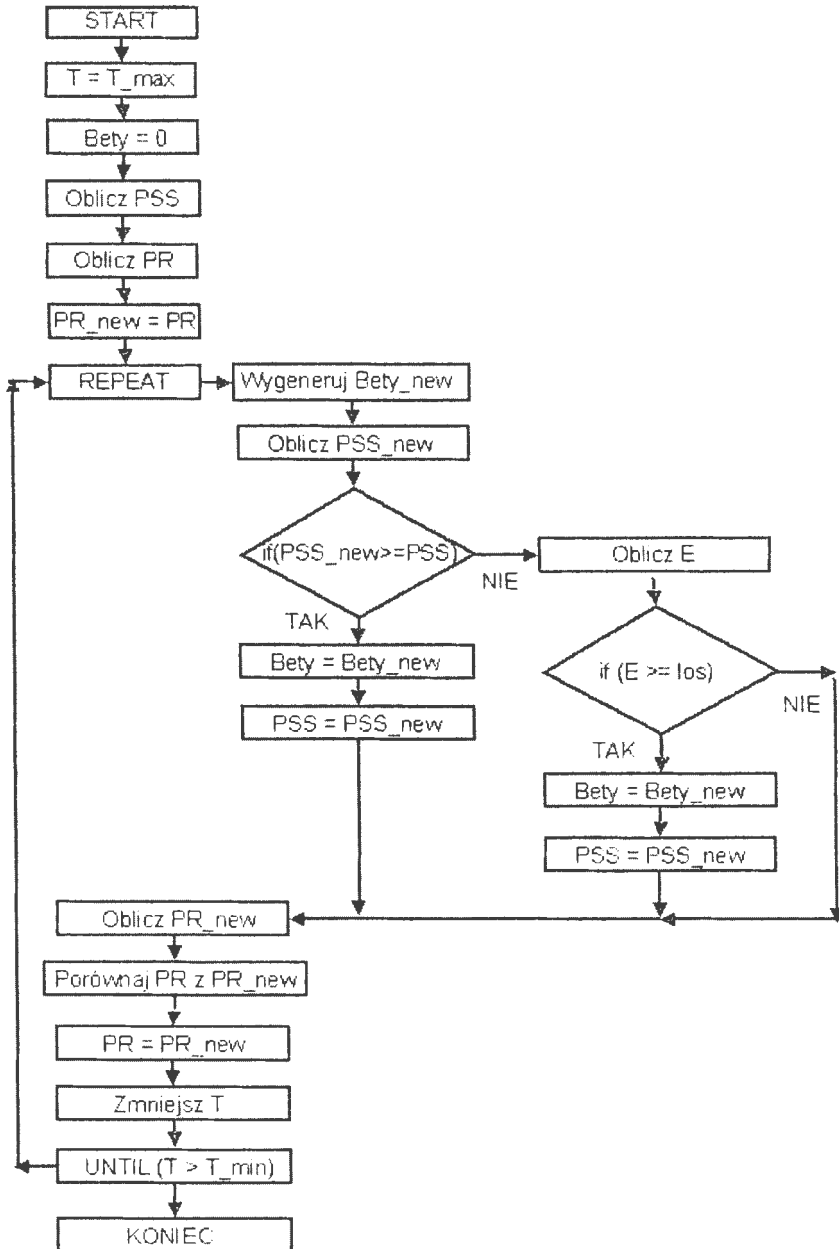
Przy zachowaniu tych samych oznaczeń, schemat blokowy powyższego algorytmu przedstawiono na następnej stronie - rys. 2.

## 6. Zastosowanie symulowanego wyżarzania do optymalizacji przepustowości środowiska sieciowego

Przed zastosowaniem metody symulowanego wyżarzania do optymalizacji przepustowości środowiska sieciowego należy odpowiedzieć na kilka pytań związanych ogólnie z samą metodą, a także z konkretnym jej zastosowaniem do wzmiankowanej optymalizacji. Odnosząc się do ogólnych zagadnień dotyczących metody symulowanego wyżarzania należy poruszyć cztery kwestie takie jak:

- warunki startowe dla algorytmu;
- warunek (warunki) zatrzymania;
- schemat schładzania;
- tempo schładzania.

W ogólnym przypadku metoda symulowanego wyżarzania jest metodą tzw. wielostartu, czyli należy ją powtarzać dla różnych warunków początkowych – w konsekwencji otrzymuje się różne wyniki końcowe. W konkretnym zastosowaniu (np. do optymalizacji przepustowości), nie wszystkie parametry powinny zmieniać swoje wartości początkowe przy kolejnych uruchomieniach algorytmu. Dokładniej mówiąc zmianie powinna (między innymi) ulegać temperatura początkowa, natomiast startowe wartości parametrów beta powinny być zawsze takie same i równe zero. Dla uzasadnienia zauważmy, że wartości współczynników beta równe zero odpowiadają sytuacji w której wyłączone jest równoważenie obciążenia NK. Obliczona przepustowość sieci bez włączonego równoważenia NK stanowi punkt startowy optymalizacji oraz umożliwia porównanie efektywności otrzymanego rezultatu końcowego z wartością początkową. Inne parametry, których wartości powinny być zmieniane przy kolejnych uruchomieniach algorytmu to temperatura końcowa, czas pracy algorytmu, tempo oraz schemat schładzania.



Rys 2. Schemat blokowy proponowanego algorytmu symulowanego wyżarzania

Dokładne proponowane wartości wymienionych parametrów oraz sposoby ich modyfikacji zostały opisane w rozdziale 7. Przechodząc do warunków zatrzymania algorytmu, na wstępie należy stwierdzić, że może być to jeden warunek lub wiele.

Dodatkowo warto wspomnieć, że specyfika zagadnienia, do którego ma być zastosowane symulowane wyżarzanie nie wpływa na to jakie te warunki należy przyjąć. Dokładny opis takich warunków został przedstawiony w rozdziale 7. W tym miejscu warto wspomnieć o niektórych z nich. Precyzyjniej mówiąc warunki zatrzymania algorytmu można podzielić przynajmniej na trzy grupy. Odpowiadające następującym kryteriom: przekroczenie przez czas działania algorytmu pewnej wartości progowej, osiągnięcie określonej temperatury, brak wystarczającej poprawy rozwiązania w określonym przedziale czasu lub po określonej liczbie iteracji. Można też sformułować inne kryteria, np. osiągnięcie pożądanej wartości funkcji energii (funkcji celu) lub jakiegokolwiek rozwiązania hybrydowe powstałe na skutek superpozycji trzech przedstawionych kryteriów.

Przechodząc do schematów schładzania, podobnie jak z warunkami zatrzymania algorytmu, nie wydaje się koniecznym wprowadzanie specyficznych zmian. Zmian które miałyby na celu dostosowanie algorytmu do wspomnianej wcześniej optymalizacji. Tematem do osobnych rozważań może być zastosowanie innego schematu schładzania niż trzy wcześniej wymienione na stronie nr 2. Schematu np. polegającego na liniowym spadku temperatury względem czasu lub iteracji algorytmu. Schemat taki niekoniecznie odpowiada temu, co można zaobserwować w naturze. Wykres zmian temperatury w czasie dla cieczy stygnącej w warunkach normalnych ma kształt wykładniczy (z ujemnym wykładnikiem potęgi) nie zaś liniowy. Patrząc z drugiej jednak strony, istnieją w przemyśle rozwiązania technologiczne wykorzystujące schładzanie z liniowym spadkiem temperatury w czasie np. przy tzw. epitaksji z fazy ciekłej (ang. Liquid Phase Epitaxy - LPE), czyli technice wzrostu warstw epitaksjalnych. Przechodząc do tempa schładzania podobnie jak w przypadkach poprzednich nie wydaje się koniecznym wprowadzanie jakichś szczególnych zmian dostosowujących algorytm do optymalizacji przepustowości sieci. Warto natomiast wspomnieć, że zbyt szybkie schłodzenie może prowadzić do niepożądanych efektów polegających na niewłaściwej budowie struktury krystalicznej (cząstki „nie zdążą” znaleźć dla siebie odpowiedniego miejsca). Jako przykład można potraktować przemysłowy proces wytopu szkła. W procesie tym płynne szkło jest schładzane dość szybko, a cząstki wchodzące w jego skład nie tworzą struktury krystalicznej. Wydawać by się mogło, że w warunkach normalnych szkło jest ciałem stałym, jednak z fizycznego punktu widzenia jest to jedynie przechło-

dzona ciecz. Analogicznie jak w kwestiach poruszanych poprzednio (warunki początkowe i zatrzymania algorytmu oraz schemat schładzania), tempo schładzania może być parametrem badawczym, który dokładniej został opisany w rozdziale 7.

Reasumując, główne udoskonalenie mające na celu dostosowanie metody symulowanego wyżarzania do specyfiki zagadnienia optymalizacji przepustowości sieci komputerowej, polega na dodaniu parametru „przepustowość routera”. Porównanie jego wartości w kolejnych iteracjach bezpośrednio przekłada się na to, z jakiego przedziału (w bieżącej iteracji) jest losowana wartość parametru beta (dla tego routera). Natomiast w klasycznej postaci wartości wspomnianych parametrów byłyby losowane z pełnego zbioru rozwiązań dopuszczalnych. Inną, choć już mniej istotną modyfikacją, jest wprowadzenie parametru  $\beta_{max}$  (indywidualnie dla każdego routera) określającego górną granicę przedziału z jakiego wartość parametru beta jest losowana (w kolejnych iteracjach algorytmu). Informacje na temat sposobu wyznaczania parametru  $\beta_{max}$ , a także wzór służący do jego wyznaczenia znajduje się w rozdziale 3. Natomiast więcej informacji na temat zastosowania symulowanego wyżarzania do zadań optymalizacji w odniesieniu do sieci komputerowych znajduje się np. w [1] [2] [3] [4] [5] [6].

## **7. Proponowane scenariusze badań**

Przy przeprowadzaniu badań modyfikacji powinny podlegać następujące parametry pracy algorytmu:

- schemat schładzania;
- temperatura początkowa;
- liczba iteracji algorytmu bez zmiany temperatury;
- warunek zatrzymania algorytmu.

Dodatkowo jest zalecane, aby podczas każdego z badań modyfikacji ulegała wartość tylko jednego z powyżej wymienionych parametrów. Natomiast same modyfikacje powinny być przeprowadzane w taki sposób, żeby każda możliwa kombinacja przyjętych wartości i parametrów pracy algorytmu została sprawdzona. W szczególności przyjmuje się, że:

1. Wykorzystane zostaną trzy schematy schładzania: logarytmiczny, liniowy i geometryczny. Dla schematu geometrycznego parametr alfa (podstawa potęgi) będzie miał dwie wartości: 0.95 i 0.99.

2. Temperatura początkowa będzie miała jedną z trzech wartości: 100, 1000, 1000000 stopni.
3. Liczba iteracji algorytmu bez zmiany temperatury będzie wynosiła 1 oraz 10.
4. Warunkiem zatrzymania algorytmu będzie: osiągnięcie temperatury końcowej bliskiej zeru (dokładniej mniejszej niż jeden stopień), przekroczenie czasu działania algorytmu podczas 30 minut, brak zmian wartości funkcji energii w czasie 5 minut.

Przyjmując, że badaniu będą podlegały:

- trzy schematy schładzania (w tym jeden z dwoma wartościami parametru alfa) [4];
- trzy temperatury początkowe [3];
- dwie wartości liczby iteracji bez zmiany temperatury [2];
- trzy warunki zatrzymania algorytmu [3];

w wyniku powyżej przyjętych założeń powstają  $[4*3*2*3]$  72 różne scenariusze badań.

Przykładowy scenariusz badania może wyglądać następująco:

Schemat schładzania – geometryczny;

Temperatura początkowa – 1000 stopni;

Liczba iteracji algorytmu bez zmiany temperatury – 10;

Warunek zatrzymania algorytmu – brak zmian wartości funkcji energii w czasie 5 minut;

### **Zakończenie**

Mając na uwadze fakt, że sieci komputerowe stają się coraz bardziej powszechnym narzędziem pracy czy rozrywki, zapotrzebowanie na szybko i niezawodną wymianę informacji między nimi wzrasta i jeszcze długo będzie wzrastać. Jedną z możliwości zaspokojenia tego zapotrzebowania jest optymalizowanie przepustowości sieci teleinformatycznych.

Możliwości przeprowadzenia takiej optymalizacji jest wiele. Warto zauważyć, że ze względu na złożoność problemu podejścia polegające na pełnym przeglądzie możliwych rozwiązań niestety, ale najczęściej są nieefektywne czasowo, tzn. panujące warunki szybciej ulegają zmianie niż algorytm jest w stanie wyznaczyć rozwiązanie.

W takiej sytuacji z pomocą przychodzą algorytmy z obszaru sztucznej inteligencji. Koncentrując się na algorytmach heurystycznych, takich jak np.: „tabu search”, genetycznych, mrówkowych, czy symulowanego wyżarzania otrzymujemy możliwość wyznaczenia rozwiązania suboptymalnego lecz wystarczająco zadowalającego lub nawet optymalnego i to w czasie rzeczywistym. Nie umniejszając wartości jakic posiadają inne metody z dziedziny heurystycznych, osobiście uważam metodę symulowanego wyżarzania jako bardzo ciekawą oraz elastyczną – dającą się zastosować w wielu obszarach nauki i techniki.

Podobnie jak wiele innych rozwiązań z obszaru sztucznej inteligencji, symulowane wyżarzanie jest wzorowane na zachowaniach natury. W konsekwencji, poznawanie, stosowanie i udoskonalanie wspomnianych metod może prowadzić do lepszego poznania i zrozumienia otaczającego nas świata. Podsumowując, uważam, że zastosowanie metody symulowanego wyżarzania do optymalizacji przepustowości sieci komputerowej jest ciekawym i przyszłościowym podejściem do tego zagadnienia, a także efektywnym i posiadającym duże możliwości dalszego udoskonalania.

### **Słowniczek pojęć**

*wyżarzanie* - operacja obróbki cieplnej polegającą na nagraniu przedmiotu z metalu lub stopu metali do określonej temperatury, wygrzaniu w tej temperaturze i schłodzeniu z szybkością pozwalającą na utrzymanie struktury w określonym stopniu zbliżonej do stanu równowagi. [NEP].

### **Literatura**

- [1]. S. Pierre, M. Hyppolite, J. Bourjolly, O. Dioume (1995): Topological Design of Computer Communication Networks Using Simulated Annealing. *Engineering Applications of Artificial Intelligence*, Vol. 8, Number 1, February 1995.
- [2]. Alekasndar Tsenov (2006): Simulated Annealing and Genetic Algorithm in Telecommunications Network Planning, *International Journal of Computational Intelligence* 2,4, 2006.

- [3]. S. Routray, A. M. Sherry, B.V.R. Reddy (2007): A Simulated Annealing Based Approach for ATM Network Optimization, *Engineering, Computing and Architecture*, Vol. 1, Issue 2, 2007.
- [4]. T.M. Mahmoud: A Genetic and Simulated Annealing Based Algorithms for Solving the Flow Assignment Problem in computer Networks, *International Journal of Electronics, Circuits and Systems*, vol. 1, number 2.
- [5]. E. Costamagna, A. Fanni, G. Giacinto (1995): A Simulated annealing Algorithm for the Optimization of Communication Networks, *VLSI Systems*, October 1995.
- [6]. M.M. Atiqullah, S.S. Rao (1993): Reliability Optimization of Computer Networks using Simulated Annealing, *Microelectronics Reliability*, vol. 33, November 1993.
- [7]. E. Aarts & J. Korst (1989): *Simulated Annealing and Boltzman Machines*, Willey & Sons Ltd.
- [8]. S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi (1993): Optimization by Simulated Annealing. *Science*, 220, 1993.
- [9]. T.H. Cormen, C.E. Leiserson, R.L. Rivest (2004): *Wprowadzenie do algorytmów*. WNT, Warszawa.
- [10]. Ross, Wright (1999): *Matematyka dyskretna*. WNT, Warszawa.
- [11]. N. Wirth: *Algorytmy+Struktury Danych = Programy*. WNT, Warszawa.
- [12]. L. Banachowski, K. Diks, W. Rytter: *Algorytmy i struktury danych*. WNT, Warszawa.





ISBN 9788389475220