

100/2005

**Raport Badawczy**  
**Research Report**

**RB/55/2005**

**Procedura szybkiego  
kojarzenia obiektów  
w klastery**

**H. Potrzebowski**

**Instytut Badań Systemowych**  
**Polska Akademia Nauk**

**Systems Research Institute**  
**Polish Academy of Sciences**



# **POLSKA AKADEMIA NAUK**

## **Instytut Badań Systemowych**

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 8373578

fax: (+48) (22) 8372772

Kierownik Pracowni zgłaszający pracę:  
Dr inż. Jan Owsiański

Warszawa 2005

## Procedura szybkiego kojarzenia obiektów w klasterzy

Henryk Potrzebowski

*Streszczenie.* Przedstawiono szczególny model klasteringu, jako zadanie na grafie skierowanym. Omówiono implementację heurystycznego algorytmu, sposób jego pracy oraz uzyskane wyniki.

### 1. Geneza problemu

Klastering, jako problem literaturowy ma wiele zastosowań, w takich dziedzinach, jak wydobywanie wiedzy, statystyczna analiza danych, kompresja i grupowanie danych, itp.

*Porter* znany ekonomista-badacz zagadnień konkurencji i kooperacji w gospodarce wskazuje na ogromną wagę umiejętności tworzenia powiązań przemysłowych w warunkach rozwijającej się konkurencji i globalizacji. Warto tu zacytować za nim, że ważne jest np. posiadanie wysokiej jakości krajowych dostawców, najlepiej w pobliżu głównego przemysłu, co pozwala na bardziej dynamiczne interakcje, wymianę wiedzy i doświadczeń, wyprzedzanie trendów światowych w branży. Wspólnota działań w sferze wytwarzania może skutkować wspólnym rozwojem technologicznym, lepiej zorganizowaną produkcją, dystrybucją, marketingiem czy usługami. Obecność międzynarodowo konkurencyjnego przemysłu powiązanego z przemysłem krajowym stwarza możliwość przepływu informacji i wymiany technologii. Obecność powiązanego przemysłu zwiększa też prawdopodobieństwo, że nowe szanse w przemyśle zostaną dostrzeżone, pojawiają się nowe firmy w danej branży, a wraz z nimi - nowe podejście do zagadnień konkurowania. Czasem firmy w powiązaniu przemysłowym z danym krajem podejmują wspólne działania i formują formalne strategiczne aliance. Powiązania między komplementarnymi działaniami i przemysłami kreują silną przewagę wiodącej na rynku firmy. Znajomość tych zagadnień organizacyjnych, umiejętność ich modelowania (np. w postaci grafu wzajemnych powiązań) może pomóc w kreowaniu strategii rozwoju gospodarki.

T. R. Browning w swoim przeglądowym artykule wskazuje na tablicę decyzyjną DSM (Design Structure Matrix), jako na model reprezentacji i analizy związanych z sobą struktur w postaci kwadratowej lub prostokątnej tablicy danych. Taka tablica jest wykorzystywana w proponowanym modelu. Reprezentuje ona siły powiązań wewnętrznych i wzajemnych oddziaływań zewnętrznych pomiędzy stworzonymi iteracyjnie klasternymi (skupieniami obiektów). W ogólności czytelna postać takiej reprezentacji sugeruje wiele metod kombinatorycznych podporządkowanych celom organizacyjnym czy konstrukcyjno-projektowym.

## 2. Model

Formalnie rzecz biorąc, *klastering* jest to problem rozbicia danego skończonego zbioru obiektów na rozłączne podzbiory – klastry, w taki sposób, że obiekty znajdujące się wewnątrz klastrow są tak sobie bliskie, jak to tylko możliwe, podczas gdy obiekty znajdujące się na zewnątrz klastrow, różnią się na tyle, na ile to tylko możliwe. Innymi słowy, powtarzając za [Browning 2001], *problem klaseringu* polega na znalezieniu takiego rozbicia obiektów, przy którym maksymalizowana jest suma interakcji wewnętrznych między obiektami, podczas gdy minimalizowana jest suma interakcji zewnętrznych. Naturalnie, w konkretnym przypadku, kryterium klaseringu może przyjmować bardziej odpowiednią postać.

Rozważany problem klaseringu sformułowany dla grafu skierowanego  $G = (V, A)$ , gdzie  $V$  – zbiór wierzchołków reprezentujących obiekty,  $A$  - zbiór łuków reprezentujących ilościowo powiązania pomiędzy obiektami typu „od – do”. Omawiana procedura klaseringu łączy iteracyjnie wierzchołki grafu  $G$  w klastry (skupienia). Ważna dla procedury tablica decyzyjna jest tablicą kwadratową, której elementy na głównej przekątnej reprezentują siłę wiązań wewnętrznych tj. liczby łuków zamknięte wewnątrz klastrow, a elementy pozostałe – siły wiązań zewnętrznych tj. liczby łuków łączących pary różnych klastrow. Na starcie tablica decyzyjna ma rozmiar  $n \times n$ , gdzie  $n$  – liczba wierzchołków grafu  $G$  i jest macierzą incydencji tego grafu.

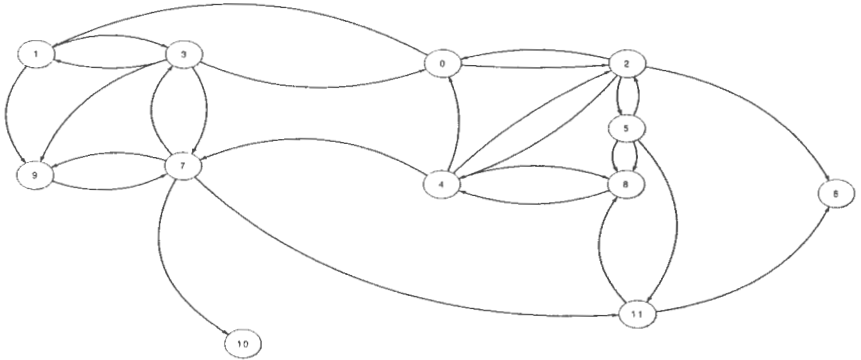
W omawianej procedurze wykorzystano tradycyjne pojęcie klik i klik maksymalnej. Wprowadzono i wykorzystano również pojęcie  $(\alpha k)$ -klik definiowanej jako spójny podgraf grafu  $G$ , dla którego liczba łuków wewnętrznych jest nie mniejsza niż  $\alpha k(k-1)$ , gdzie parametr  $\alpha$  jest wielkością arbitralną,  $0 < \alpha \leq 1$ . W przeprowadzonych obliczeniach dla parametru  $\alpha$  przyjmowano wartości 0.4, 0.5, 0.6.

## Dane

$n$  - liczba klasyfikowanych obiektów

$D = [d_{ij}]$  - tablica powiązań pomiędzy obiektami,  $i, j = 0, \dots, n-1$ .

Niżej pokazano charakterystyczny przykładowy graf powiązań dla  $n = 12$  obiektów o wiązaniach jedno- i dwukierunkowych. Jak łatwo zauważyć, są to wiązania niesymetryczne, bez pętli. Pokazany graf jest spójny, ale nie ma to znaczenia dla omawianej metody.



Rys. 1. Graf powiązań pomiędzy obiektami.

Dla grafu z rysunku 1 tablica powiązań (jednocześnie postać numeryczna zadania) ma postać następującą:

|    |   |   |   |   |   |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0  | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1  | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1  | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1  | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0  | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0  | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0  | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |

## Problem

Niech  $P$  będzie rozbiem zbioru  $I$  wszystkich  $n$  obiektów, a  $m$  - liczbą klastrow w  $P$ , taką że

$$P = \{X_i, i = 1, \dots, m\}, \quad \bigcup_i X_i = I \text{ and } X_i \cap X_j = \Phi \quad \forall i \neq j.$$

Definiujemy funkcje reprezentujące siły powiązań wewnętrznych i zewnętrznych jako:

$$f(P) = \sum_{i=1}^m \sum_{k,l \in X_i, k \neq l} d_{kl} \quad - \text{funkcja siły powiązań wewnętrznych}$$

oraz

$$g(P) = \sum_{i=1}^m \sum_{j>i}^m \sum_{k \in X_i, l \in X_j} (d_{kl} + d_{lj}) \quad - \text{funkcja siły powiązań zewnętrznych}$$

Zauważmy, że dla rozpatrywanego grafu suma  $f(P)+g(P)$  jest stała i niezależna od sposobu powiązania w klaster, czyli od rozbicia  $P$ . Stąd mamy problem w postaci następującej:

Należy wyznaczyć rozbicia  $P$  i odpowiadające mu  $m$ , dla których pomiędzy  $m$  i wskaźnikiem  $f(P)$  zachodzi określony kompromis, co zapiszemy symbolicznie jako

$$(f(P), m) = \text{minimum.} \quad (P)$$

Takie sformułowanie sugeruje szukanie rozwiązań typu PARETO optymalnych, chyba, że wartość  $m$  można będzie ustalić.

### 3. Szybki algorytm kojarzenia

Dla danego  $P$  niech  $P(k,l)$  będzie rozbiem otrzymanym z  $P$  po połączeniu w jeden klaster jego dwóch klasterów składowych  $X_k$  i  $X_l$ . W wyniku tej operacji  $f$  wzrośnie o wielkość

$$\delta(k,l) = f(P(k,l)) - f(P).$$

Na początek startujemy z rozbicia, którego klastry zawierają pojedyncze obiekty, czyli z  $P = \{X_i = \{i\}, i = 1, \dots, n\}$  oraz  $m = n$ . Budując kolejne wiązania postępujemy w sposób następujący:

1. jeżeli  $m = 1$  stop.
2. znajdź klastry kandydujące do połączenia wg reguły maksymalnego przyrostu siły powiązań wewnętrznych,
 
$$(k,l) = \arg \max \{ \delta(i,j), 0 \leq i, j \leq m \}$$
3. aktualizuj  $P$  łącząc  $X_k$  z  $X_l$ , idź do kroku 1.

Odpowiedni program opracowano w języku C++, dwa ważne fragmenty kodu programu zamieszczono w dodatku.

W pracy z różnymi przykładami okazało się, że ten prosty algorytm trzeba było nieco zmodyfikować, tzn. dodać kilka prostych ograniczeń, kryterium maksymalizacji przyrostu  $\delta(k,l)$  zamienić na kryterium maksymalizacji uśrednionego przyrostu przypadającego na jeden element połączonego klastra. Tak powstały metody A, B i C.

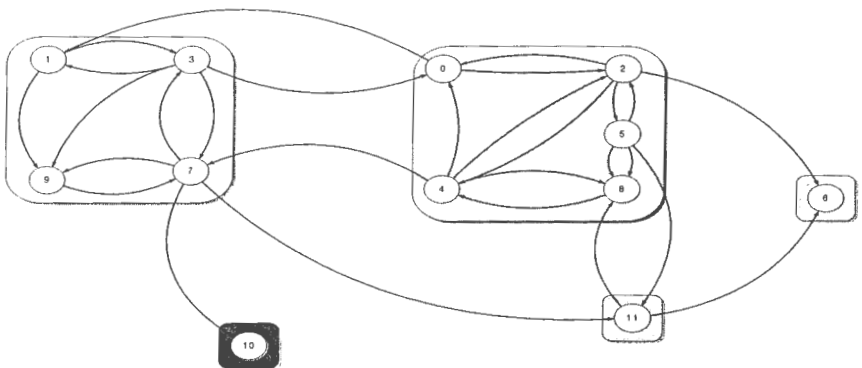
Metoda A – zakłada najpierw łączenie obiektów w kliki maksymalne, następnie iteracyjne dołączanie do istniejących klastrów (na starcie - klik) innych obiektów i klastrów, przy ograniczeniu, że klaster nie przekracza rozmiaru  $q$ , gdzie  $q \leq n$ .

Metoda B – najpierw połącz obiekty w kliki maksymalne, a następnie iteracyjnie łącz kliki w nowe ( $\alpha k$ ) - kliki, aż do wyczerpania możliwości.

Metoda C – startując z klastrów jednoelementowych łącz iteracyjnie istniejące kliki w nowe ( $\alpha k$ ) - kliki, do wyczerpania możliwości.

#### 4. Wyniki

Niżej na rysunku 2 pokazano kompromisowe rozwiązanie zadania klaseringu dla grafu pokazanego na rys. 1.



Rys. 2. Uzyskane metodą A rozwiązanie dla grafu pokazanego na rysunku 1

Niżej dołączono wyniki kontrolne z rozwiązania zadania testowego T12man.txt, rozwiązanego metodą A. Pokazane na rysunku rozwiązanie uzyskano w 7-ej iteracji tej metody. Podobne wyniki zaobserwowane w przypadku metod B i C, które zakończyły pracę w mniejszej liczbie iteracji. Podobne obliczenia przeprowadzono dla zadań losowanych, o większej wymiarowości, gdzie obserwowano bardziej zróżnicowane wyniki (z małą przewagą dla metody C), a obserwowany błąd dla funkcji  $f$  nie przekraczał 10%.

Przeczytane zadanie: T12man.txt

[( 0)( 1)( 2)( 3)( 4)( 5)( 6)( 7)( 8)( 9)( 10)( 11)]

Tablica D rozmiaru 12, Fwew = 0, Fzew = 28

|   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

#### Metoda A

[( 0, 2)( 1)( 3)( 4)( 5)( 6)( 7)( 8)( 9)( 10)( 11)]

Tablica D rozmiaru 11, Fwew = 2, Fzew = 26

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

[( 0, 2)( 1, 3)( 4)( 5)( 6)( 7)( 8)( 9)( 10)( 11)]

Tablica D rozmiaru 10, Fwew = 4, Fzew = 24

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 2 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

[( 0, 2)( 1, 3)( 4, 8)( 5)( 6)( 7)( 9)( 10)( 11)]

Tablica D rozmiaru 9, Fwew = 6, Fzew = 22

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 2 | 0 | 0 | 0 | 1 | 2 | 0 | 0 |
| 2 | 0 | 2 | 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |



[( 0, 2)( 1, 3)( 4, 8)( 5)( 6)( 7, 9)( 10)( 11)]

Tablica D rozmiaru 8, Fwew = 8, Fzew = 20

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 2 | 0 | 0 | 0 | 3 | 0 | 0 |
| 2 | 0 | 2 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 2 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |

[( 0, 2)( 1, 3, 7, 9)( 4, 8)( 5)( 6)( 10)( 11)]

Tablica D rozmiaru 7, Fwew = 12, Fzew = 16

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| 1 | 8 | 0 | 0 | 0 | 1 | 1 |
| 2 | 1 | 2 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 |

[( 0, 2, 4, 8)( 1, 3, 7, 9)( 5)( 6)( 10)( 11)]

Tablica D rozmiaru 6, Fwew = 15, Fzew = 13

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 7 | 3 | 2 | 0 | 0 | 0 |
| 1 | 8 | 0 | 0 | 1 | 1 |
| 2 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 | 0 |

[( 0, 2, 4, 8, 5)( 1, 3, 7, 9)( 6)( 10)( 11)] (pokazane na rysunku)

Tablica D rozmiaru 5, Fwew = 19, Fzew = 9

|    |   |   |   |   |
|----|---|---|---|---|
| 11 | 3 | 0 | 0 | 1 |
| 1  | 8 | 0 | 1 | 1 |
| 0  | 0 | 0 | 0 | 0 |
| 0  | 0 | 0 | 0 | 0 |
| 1  | 0 | 1 | 0 | 0 |

[( 0, 2, 4, 8, 5)( 1, 3, 7, 9)( 6, 11)( 10)]

Tablica D rozmiaru 4, Fwew = 20, Fzew = 8

|    |   |   |   |
|----|---|---|---|
| 11 | 3 | 1 | 0 |
| 1  | 8 | 1 | 1 |
| 1  | 0 | 1 | 0 |
| 0  | 0 | 0 | 0 |

[( 0, 2, 4, 8, 5, 6, 11)( 1, 3, 7, 9)( 10)]

Tablica D rozmiaru 3, Fwew = 22, Fzew = 6

|    |   |   |
|----|---|---|
| 14 | 3 | 0 |
| 2  | 8 | 1 |
| 0  | 0 | 0 |

[( 0, 2, 4, 8, 5, 6, 11)( 1, 3, 7, 9, 10)]

Tablica D rozmiaru 2, Fwew = 23, Fzew = 5

|    |   |
|----|---|
| 14 | 3 |
| 2  | 9 |

## **5. Wnioski**

Opracowana procedura rozwiązywania zadań klaseringu dla grafu skierowanego jest bez procedurą szybką, pomocną w interaktywnym trybie pracy.

Procedura ta może dostarczyć rozwiązania startowe dla iteracyjnych algorytmów: ewolucyjnego, genetycznego i innych.

## **6. Literatura**

- Browning T. R.: "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions", IEEE Transactions on Engineering Management, Vol. 48, No 3, 2001.
- Bagirov A. M.: A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, EJofOR 170(2006) 578-595

## 7. Dodatki

### Program główny

```
/*
    Heurystyczna procedura
    dla binarnego zadania klasteringu nad grafem skierowanym
    (nonsymmetric graph cluster 0,1 task procedure)
wywołanie:
    >FnsCluster01prog.exe nazwa_zadania.txt metoda
gdzie
    FnsCluster01prog.exe - program wykonywalny
    nazwa_zadania.txt    - nazwa rozwiązywanego zadania
    metoda(opcje A, B, C) - parametr opcjonalny(opcja A - domylna)
*/
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstdlib>

using namespace std;

#include "ListV1.h"
#include "ListV1.cpp"
#include "TabList.h"
#include "TabList.cpp"
#include "TabC2.h"
#include "TabC2.cpp"
#include "FnsCluster01.h"
#include "FnsCluster01.cpp"

const double alpha = 0.6;
const double upper = 0.6;

int main(int argc, char *argv[])
{
    if(argc == 1)
        blad(argv[0], "Blad!, brakuje nazwy zadania.");

    TabC2 A; // tworzymy obiekt klasy TakC2, bo tam jest metoda czytania
    zadania
    A.CzytajZadanie(argv[1]);
    FnsCluster FC(A.sizeC2(),A);
    FC.drukuj(); // wydruk przeczytanego zadania

    char z =(argc==3)?argv[2][0]:'A';
    cout<<"Metoda "<<z<<endl<<endl;

    w_i_k wk;
    switch (z)
    {
        case 'A': wk = FC.wybor();
                while(wk.k-wk.w) // proste kojarzenie w kliki
                {
                    FC.redukcja(wk);
                    FC.drukuj();
                    wk = FC.wybor();
                }
                wk = FC.wybor(int(upper*FC.LE()));
    }
}
```

```

while(wk.k-wk.w) // rozszerzanie do granicy upper
{
    FC.redukcja(wk);
    FC.drukuj();
    wk = FC.wybor(int(upper*FC.LE()));
}
break;
case 'B': wk = FC.wybor();
while(wk.k-wk.w) // proste kojarzenie w kliki
{
    FC.redukcja(wk);
    FC.drukuj();
    wk = FC.wybor();
}
wk = FC.wybor(alpha);
while(wk.k-wk.w) // rozszerzanie do granic alpha-kliki
{
    FC.redukcja(wk);
    FC.drukuj();
    wk = FC.wybor(alpha);
}
break;
case 'C': wk = FC.wybor(alpha);
while(wk.k-wk.w) // rozszerzanie do granic alpha-kliki
{
    FC.redukcja(wk);
    FC.drukuj();
    wk = FC.wybor(alpha);
}
break;
}
return 0;
}

```

## Procedury wyboru

```

w_i_k FnsCluster::wybor() // kojarzenie w k-klikę, k=2,3,...
{
    int Tw, Tk, Dwk;
    w_i_k wk;
    for(unsigned w=0;w<m;++w)
    {
        Tw = T[w].LSize();
        for(unsigned k=w+1;k<m;++k)
        {
            Dwk = D[w][w]+D[w][k]+D[k][w]+D[k][k];
            Tk = T[k].LSize();
            if(Dwk == (Tw+Tk)*(Tw+Tk-1))
            {
                wk.w = w; wk.k = k; return wk;
            }
        }
    }
    return wk;
}

w_i_k FnsCluster::wybor(int q) //dopełnienie co najwyżej do q elementów
{
    int tw, twk, delta;

```

```

double D_, D_max = .0;
w_i_k wk;
for(unsigned w=0;w<m;++w)
{
    tw = T[w].LSize();
    for(unsigned k=w+1;k<m;++k)
    {
        twk = tw+T[k].LSize();
        delta = D[w][k]+D[k][w];
        if((twk>q)||!delta) continue;
        if((D_ = double(delta)/twk) > D_max )
        {
            wk.w = w; wk.k = k; D_max = D_;
        }
    }
}
return wk;
}

w_i_k FnsCluster::wybor(double alpha) // rozszerzenie do alpha-kliki
{
    int tw, twk, delta;
    double D_, D_max = .0;
    w_i_k wk;
    for(unsigned w=0;w<m;++w)
    {
        tw = T[w].LSize();
        for(unsigned k=w+1;k<m;++k)
        {
            delta = D[w][k]+D[k][w];
            twk = tw + T[k].LSize();
            if(!delta ||
                (delta+D[w][w]+D[k][k] < alpha*twk*(twk-1))) continue;
            if((D_ = double(delta)/twk) > D_max )
            {
                wk.w = w; wk.k = k; D_max = D_;
            }
        }
    }
    return wk;
}

void FnsCluster::redukcja(w_i_k wk)
{
    redukcjC2(wk.w, wk.k);
    redukcjT1(wk.w, wk.k);
    if(m-R)
    {
        cout << "Blad po etapie redukcji : " << endl;
        exit(-1);
    }
    lC = R;
}

void FnsCluster::drukuj()
{
    cout<<TabL1(*this)<<endl;
    (TabC2(*this)).display();
    cout.clear();
}

```





