



POLSKA AKADEMIA NAUK
Instytut Badań Systemowych

ANALIZA SYSTEMÓW PRZESTRZENNYCH

WYBRANE ZAGADNIENIA

Redakcja

Jan W. Owsieński

Warszawa 2010



ANALIZA SYSTEMÓW PRZESTRZENNYCH

WYBRANE ZAGADNIENIA

Polska Akademia Nauk • Instytut Badań Systemowych
Seria: BADANIA SYSTEMOWE
tom 67

Redaktor naukowy:
Prof. dr hab. inż. Jakub Gutenbaum

Warszawa 2010

Rada Redakcyjna serii: BADANIA SYSTEMOWE

Prof. dr hab. inż. Olgierd Hryniewicz – przewodniczący

Prof. dr hab. inż. Jakub Gutenbaum – redaktor naczelny

Prof. dr hab. inż. Janusz Kacprzyk

Prof. dr hab. inż. Tadeusz Kaczorek

Prof. dr hab. inż. Roman Kulikowski

Doc. dr hab. inż. Marek Libura

Prof. dr hab. inż. Krzysztof Malinowski

Prof. dr hab. inż. Zbigniew Nahorski

Dr hab. inż. Marek Niezgódka, prof. UW

Prof. dr hab. inż. Roman Słowiński

Doc. dr hab. inż. Jan Studziński

Prof. dr hab. inż. Stanisław Walukiewicz

Prof. dr hab. inż. Andrzej Weryński

Doc. dr hab. inż. Antoni Żochowski



**INSTYTUT BADAŃ SYSTEMOWYCH
POLSKIEJ AKADEMII NAUK**

**ANALIZA SYSTEMÓW
PRZESTRZENNYCH**

WYBRANE ZAGADNIENIA

**Redakcja
Jan W. Owsieński**

Warszawa 2010

Copyright © by Instytut Badań Systemowych PAN
Warszawa 2010

Autorzy:

Jan W. Owsiniński, redaktor

Instytut Badań Systemowych PAN

Pracownia Zastosowań Metod Badań Systemowych

Tel. (48 22) 3810 213

e-mail: Jan.Owsinski@ibspan.waw.pl

Jan Gadomski

Jerzy W. Hołubiec

Barbara Maźbic-Kulma

Michał Milczewski

Jan W. Owsiniński

Grażyna Petriczek

Aneta M. Pielak

Henryk Potrzebowski

Krzysztof Sęp

Eugeniusz Sobczak

Jarosław Stańczak

Recenzenci:

Prof. dr hab. inż. Jacek Mercik

Prof. dr hab. Tadeusz Trzaskalik

Opinie, wyrażone przez autorów w pracach, zawartych w niniejszym tomie, nie są oficjalnymi opiniami Instytutu Badań Systemowych PAN

ISBN 9788389475251

ISSN 0208-8029

Redakcja i opracowanie techniczne: Jan W. Owsiniński i Aneta M. Pielak

V. Metody teorii grafów i heurystyki kombinatoryczne

Barbara Maźbic-Kulma i Krzysztof Sęp

V.1. Wstęp

Rozmaitość zagadnień, których modele matematyczne są problemami optymalizacji kombinatorycznej lub kombinatorycznymi problemami decyzyjnymi, doprowadziła do poszukiwania algorytmów, służących do ich dokładnego lub przybliżonego rozwiązywania (Cormen, Leiserson, Rivest, 1997; Cowen, 1998; Corte, Vygen, 2000; Jukna, 2001; Lund, Yannakakis, 1994; Motvani, 1992; Vazirani, 1997; Williamson, 1999). Zazwyczaj zadania decyzyjne są problemami sformułowanymi w postaci zapytań, na które można udzielić odpowiedzi *tak* lub *nie*, czyli podjąć decyzję. W problemach kombinatorycznych, zwłaszcza w ich praktycznym zastosowaniu, istotna jest kwestia efektywnego ich rozwiązania, przede wszystkim z punktu widzenia czasu obliczeń. Najbardziej zasadnicze pytanie, jakie się nasuwa to, czy dla rozważanego problemu istnieje jakikolwiek rozwiązujący go algorytm. W zastosowaniach praktycznych jest to szczególnie istotne, gdyż nie wystarczy nam stwierdzenie, że jakiś problem jest obliczeniowo trudny, lecz musimy jeszcze zaproponować sposób rozwiązania tego zadania, uwzględniając ograniczenia czasowe, jak również żadaną dokładność.

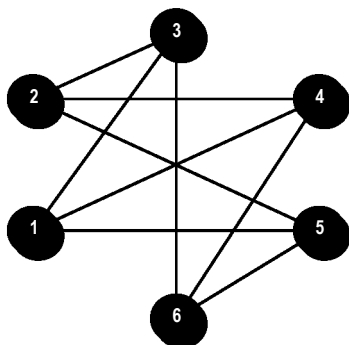
W niniejszym rozdziale przedstawimy zatem algorytmy wspomagające podejmowanie decyzji w konkretnych praktycznych zagadnieniach. W ostatnich latach zmiany na rynku przewozowym zaostrzyły walkę konkurencyjną pomiędzy przewoźnikami i przyczyniły się do rozwoju systemów transportowych. Stąd też przedstawione w końcowej części rozdziału zastosowania zaprezentowanych algorytmów dotyczą głównie zagadnień transportowych.

V.2. Podstawowe pojęcia

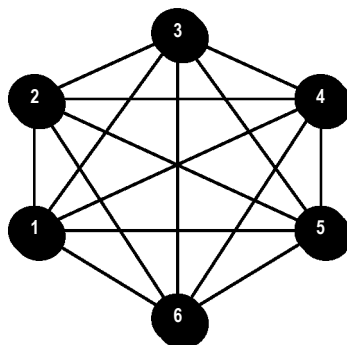
Dla dalszych rozważań wprowadźmy teraz pewne podstawowe pojęcia teorii grafów (Berge, 1989; Wilson, 1996).

V.2.1. Graf

Graf jest parą $G = (V, E)$, gdzie V jest niepustym zbiorem *wierzchołków*, a E jest zbiorem *krawędzi*. Każda krawędź jest parą wierzchołków (v_1, v_2) takich, że $v_1 \neq v_2$.



Rys. V.1 Przykładowy graf



Rys. V.2 Graf pełny

Źródło: opracowanie własne

Dwa wierzchołki grafu $G=(V, E)$ są **incydentne** jeżeli dla $v_1, v_2 \in V$ mamy $\{v_1, v_2\} \in E$.

Graf $G(V, E)$ nazywamy **pełnym** jeżeli każde dwa wierzchołki $v_i, v_j \in V, i \neq j$, są ze sobą incydentne.

Podgraf grafu $G = (V, E)$ jest grafem $G' = (V', E')$, gdzie $V' \subseteq V$ i $E' \subseteq E$ takim, że dla każdego $e \in E$ i $e = \{v_1, v_2\}$ jeżeli $v_1, v_2 \in V'$ to $e \in E'$.

Drogą (ścieżką) w grafie $G = (V, E)$ z wierzchołka s do wierzchołka t nazywamy ciąg wierzchołków $\{v_1, \dots, v_n\}$ taki, że: $\{s, v_1\} \in E, \{v_i, v_{i+1}\} \in E$ dla $n=1, 2, \dots, n-1, \{v_n, t\} \in E$.

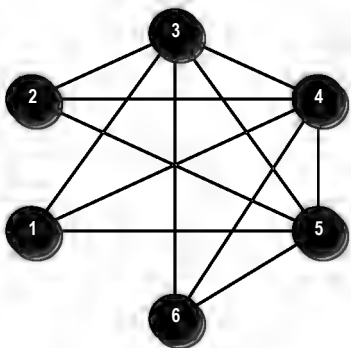
Graf $G = (V, E)$ nazywamy **grafem spójnym**, jeżeli dla każdych dwóch różnych wierzchołków istnieje droga łącząca te wierzchołki.

Kliką (podgrafem pełnym) $Q=(V_q, E_q)$ w grafie $G = (V, E)$ jest graf taki, że $V_q \subseteq V$ i $E_q \subseteq E$ oraz każda para wierzchołków $v_1, v_2 \in V_q$ spełnia warunek $\{v_1, v_2\} \in E_q$.

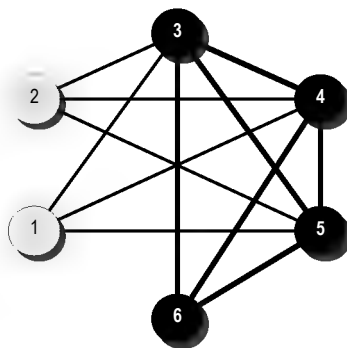
Maksymalną kliką nazywamy klikę $Q_M=(V_q, E_q)$ w grafie $G = (V, E)$ taką, że nie istnieje wierzchołek $v \in V$ i $v \notin V_q$ taki, że $Q'=(V', E')$ jest kliką, gdzie $V'=V \cup \{v\}$ i $E' \subseteq E$ i gdzie każda para $v_1, v_2 \in V'$ wierzchołków spełnia warunek $\{v_1, v_2\} \in E'$.

Struktura **hub-and-spoke** (O'Kelly, 1987; O'Kelly i Bryan, 2002) to graf $H_s=(V_h \cup V_s, E)$ w którym niepusty podzbiór V_h wraz z odpowiednimi

krawędziami wyznacza graf pełny. Każdy wierzchołek z podzbioru V_s ma stopień 1 i jest połączony z dokładnie jednym wierzchołkiem ze zbioru V_h .

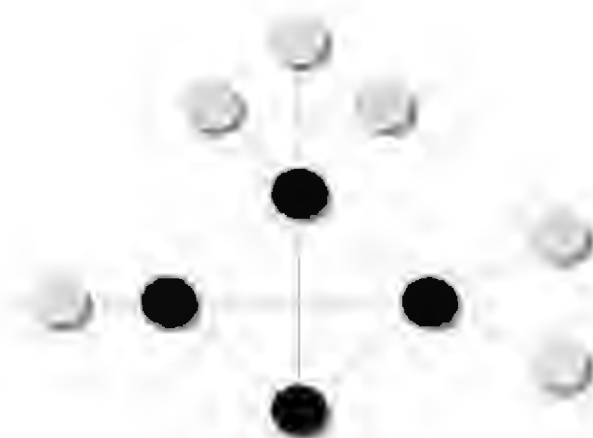


Rys. V.3 Przykładowy graf



Rys. V.4 Podgraf pełny grafu z Rys. V.3

Źródło: opracowanie własne



Rys. V.5. Przykład struktury typu *hub-and-spoke*

Źródło: opracowanie własne

W celu uzyskania struktury typu *hub-and-spoke* proponujemy następujące podejście:

Niech $G(V, E)$ będzie rozważanym grafem.

1. Znajdujemy pokrycie niezależnymi, równomiernymi *klikami* $A_1(V_1, E_1), \dots, A_m(V_m, E_m)$ spełniającymi warunki: $V_1 \cup \dots \cup V_m = V$ i $V_1 \cap \dots \cap V_m = \emptyset$, gdzie m jest liczbą *klik*.
2. W każdej *klice* A_i wybieramy dokładnie jeden wierzchołek χ_i o najwyższym stopniu w grafie $G'(\bigcap V_i \cup \{\chi_i\}, E \setminus E_i)$ – każdy wierzchołek χ_i jest nowym hubem.
3. Łączymy każdy hub ze wszystkimi wierzchołkami z *kliki* do której ten hub należy.
4. Łączymy huby każdy z każdym.

Strukturę typu **hub-and-spoke** uzyskujemy poprzez wybranie z każdej *kliki* węzła o najwyższym stopniu w danej *klice*.

V.2.2. Hipergraf

Hipergrafem nazywamy parę uporządkowaną $H=(X, F)$, gdzie X jest niepustym, skończonym zbiorem wierzchołków, a F jest niepustą rodziną różnych podzbiorów zbioru X spełniających warunek:

$$\bigcup_{f \in F} f = X.$$

F nazywamy zbiorem hiperkrawędzi hipergrafu.

Stopniem wierzchołka hipergrafu nazywamy liczbę hiperkrawędzi, do których ten wierzchołek należy.

Gwiazdą wierzchołka x nazywamy zbiór wszystkich hiperkrawędzi, do których ten wierzchołek należy.

Bazą wierzchołkową lub **Transwersalą** Tr hipergrafu $H=(X, F)$, nazywamy podzbiór zbioru wierzchołków $Tr \subset X$ taki, że dla każdej hiperkrawędzi f istnieje wierzchołek $x \in Tr$ taki, że $x \in f$.

Dowolny zbiór wierzchołków T_{min} takich, że jeżeli T_{min} jest transwersalą, a żaden podzbiór T_{min} nie jest transwersalą nazywamy **transwersalą minimalną ze względu na inkluzję** lub **transwersalą minimalną**.

Dowolny zbiór wierzchołków T_{opt} takich, że jeżeli T_{opt} jest transwersalą, taką, że każda transwersala ma moc nie mniejszą niż T_{opt} nazywamy **transwersalą minimalną ze względu na moc** lub **transwersalą najmniejszą**.

Problem *najmniejszego zbioru testów* polega na znalezieniu dla danej funkcji n -argumentowej takiego podzbioru zbioru argumentów, aby funkcja była dobrze określona dla całej przeciwdziedziny.

Twierdzenie V.1. Problem *najmniejszego zbioru testów* jest *NP* trudny (Błażewicz, 1988).

Twierdzenie V.2. Problem *najmniejszego zbioru testów* można, przy pomocy algorytmu o złożoności wielomianowej, sprowadzić do problemu *najmniejszej transwersali* w hipergrafie (Błażewicz, 1988).

Z Twierdzeń V.4 i V.5 wynika, że problem *najmniejszej transwersali* w hipergrafie jest *NP* trudny.

Obecnie znane są tylko algorytmy o złożoności co najmniej wykładniczej rozwiązujące problem *najmniejszej transwersali* w hipergrafie (Eiter i Gottlob, 1995; Slavik, 1996).

V.3. Algorytmy wyznaczania najmniejszej transwersali w hipergrafach

Omówimy trzy algorytmy znajdowania minimalnej transwersali: algorytm dokładny – backtrackingowy, oraz dwa algorytmy aproksymacyjne: algorytm *Lovasa-Johnsona-Chvatala* (Johnson, 1974; Lovasz, 1975; Chvatal, 1979) i algorytm *MSBT*.

V.3.1. Algorytm dokładny

Opiszemy teraz algorytm dokładny, znajdujący minimalną transwersalę w zadanym hipergrafie, i przedstawimy jego średnią złożoność obliczeniową.

Zasada działania algorytmu dokładnego

Algorytm dokładny jest algorytmem z *nawrotami* (*backtrackingowym*), w którym zastosowano wybór zmiennych według porządku leksykograficznego.

W algorytmie dokładnym zastosowano strategię pełnego przeszukiwania w

głąb. Dzięki metodzie z *nawrotami* tworzenia drzewa przeszukiwania unikamy wielokrotnego tworzenia takich samych węzłów, czyli tych samych podzbiorów.

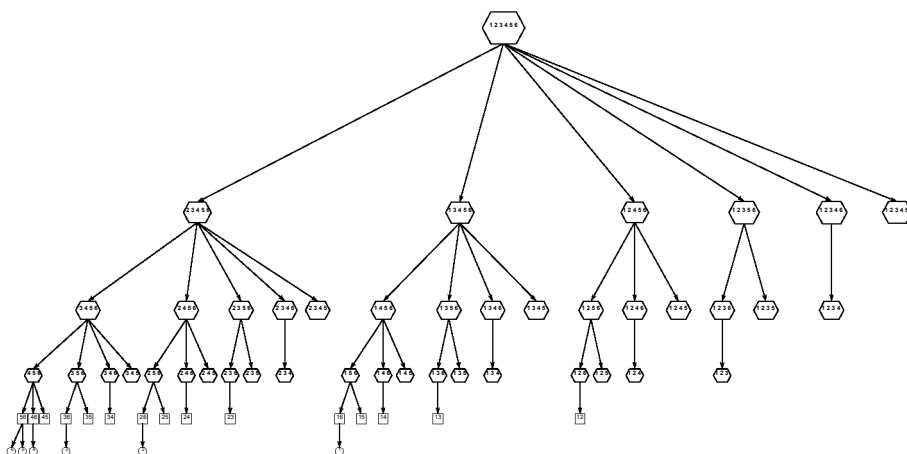
Pełne drzewo backtrackingowe dla n -wierzchołkowego hipergrafu będzie miało $2^n - 1$ węzłów. Algorytm z *nawrotami* dla hipergrafu $H=(X, F)$ działa następująco (Sapiecha, 1993; Sapiecha i Sapiecha, 1994):

```

Tr:=X;
V:=X;
procedure BT(V)
begin
  for a∈V wybrany w porządku leksykograficznym do
    if V\{a} jest transversalą hipergrafu H then
      begin
        if Card(V)\{a}<Card(Tr) then
          Tr:=V\{a};
          BT(V\{a});
        end;
      return Tr;
    end;
end;

```

Rys. V.6 pokazuje przykładowe drzewo backtrackingowe dla sześciowierzchołkowego hipergrafu.



Rys. V.6. Drzewo backtrackingowe dla $n=6$
 Źródło: opracowanie własne

Aby przedstawić zasadę działania algorytmu z *nawrotami* (*back-*

trackingowego) rozważmy hipergraf o czterech wierzchołkach: 1, 2, 3, 4 i trzech hiperkrawędziach: {1, 2}, {2, 3, 4} i {3}.

Macierz incydencji tego hipergrafu będzie miała postać (Tabela V.1):

wierzchołek	hiperkrawędź		
	1 2	3	2 3 4
1	×		
2	×		×
3		×	×
4			×

Tabela V.1. Przykładowa macierz incydencji hipergrafu. Znak × oznacza, że wierzchołek należy do hiperkrawędzi.



Rys. V.7. Hipergraf przedstawiony w Tabeli V.1

Źródło: opracowanie własne

W przypadku hipergrafów zbudowanych z większej liczby wierzchołków i większej liczby hiperkrawędzi macierz incydencji byłaby, ze względu na rozmiary, nieczytelna, na przykład dla 10 wierzchołków liczba wszystkich możliwych hiperkrawędzi wynosi $10^2 - 1 = 1023$ co nawet dla 1/10 liczby hiperkrawędzi powodowałoby, że rozmiary takiej macierzy byłyby znaczne, dlatego dla większych hipergrafów proponujemy inny zapis:

f_1 : 1, 2

f_2 : 3

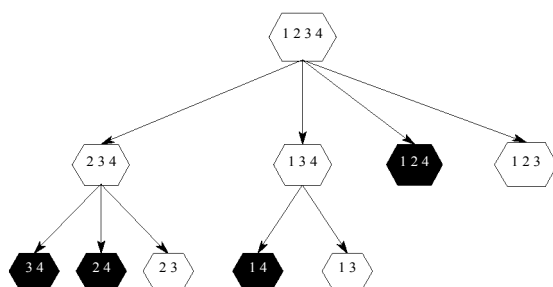
f_3 : 2, 3, 4

gdzie f_i – jest i -tą hiperkrawędzią, a liczby po dwukropku oznaczają numery wierzchołków należących do hiperkrawędzi. Zapis taki jest bardziej zwięzły.

Zadanie polega na znalezieniu najmniejszego zbioru wierzchołków, w którym dla każdej hiperkrawędzi będzie istniał przynajmniej jeden należący do

tej hiperkrawędzi wierzchołek. Takim zbiorem co do incydencji z hiperkrawędziami jest na pewno zbiór $\{1, 2, 3, 4\}$, czyli jest on transwersalą danego hipergrafu. Pojawia się jednak pytanie czy istnieje podzbiór tego zbioru, który również jest transwersalą. Mamy do rozpatrzenia cztery podzbiory trzejelementowe: $\{2, 3, 4\}$, $\{1, 3, 4\}$, $\{1, 2, 4\}$, $\{1, 2, 3\}$. Kontynuując wybieranie podzbiorów powinniśmy uzyskać ich $2^4 = 16$. Odrzucając zbiór pusty mamy 15 możliwości.

Na Rys. V.8 pokazujemy drzewo backtrackingowe utworzone dla danego wyżej hipergrafu przy permutacji wierzchołków $IF=(1\ 2\ 3\ 4)$.



Rys. V.8

Źródło: opracowanie własne

Czarnym kolorem zaznaczono liście, które nie są już transwersalami. Uzyskaliśmy trzy transwersale będące liśćmi drzewa backtrackingowego: $\{2, 3\}$, $\{1, 3\}$ i $\{1, 2, 3\}$. Najmniejsze transwersale to, zatem: $\{2, 3\}$ i $\{1, 3\}$. Transwersala $\{1, 2, 3\}$ nie tylko nie jest najmniejszą transwersalą, ale także nie jest minimalną transwersalą, gdyż transwersale $\{2, 3\}$ i $\{1, 3\}$ są podzbiorem transwersali $\{1, 2, 3\}$. Pojawia się tu dodatkowy ciekawy problem wygenerowania wszystkich minimalnych transwersal hipergrafu.

Problem znalezienia najmniejszej transwersali jest problemem *NP*-trudnym. Złożoność pesymistyczna problemu dla dowolnego n -wierzchołkowego hipergrafu jest rzędu 2^n . Złożoność pesymistyczna algorytmu z *nawrotami* (*backtrackingowego*) wynika z faktu, że aby znaleźć najmniejszą transwersalę w hipergrafie, trzeba wygenerować wszystkie transwersale, liczba wszystkich transwersal w hipergrafie w pesymistycznym wypadku jest rzędu liczby niepustych podzbiorów zbioru wierzchołków, tj. $2^n - 1$.

Złożoność optymistyczna algorytmu dokładnego jest rzędu $O(n)$. Występuje ona wtedy, gdy cały zbiór wierzchołków jest minimalną transwersalą, zatem

do sprawdzenia jest tylko n podzbiorów zbioru wierzchołków.

V.3.2. Algorytm aproksymacyjny Lovásza – Johnsona – Chvatala

Algorytm *Lovásza–Johnsona–Chvatala* jest algorytmem zachłannym - w każdym kroku wybiera wierzchołek pokrywający jak najwięcej hiperkrawędzi. Stosując algorytm *Lovásza–Johnsona–Chvatala* zakładamy, że w zbiorze wierzchołków pokrywających hiperkrawędzie powinny znajdować się te wierzchołki, które pokrywają najwięcej hiperkrawędzi.

Zasada działania algorytmu

Niech funkcja $deg(x)$ oznacza *stopień* wierzchołka x , a $F(x)$ będzie *gwiazdą* wierzchołka x . Algorytm aproksymacyjny *Lovásza–Johnsona–Chvatala* dla hipergrafu $H=(X, F)$ działa w następujący sposób:

Algoritm 1

```

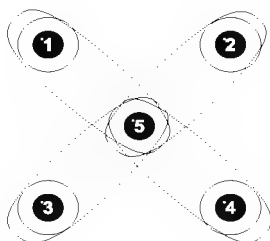
procedure LV (H)
begin
  Tr := ∅;
  V := X;
  E := F;
  while (E ≠ ∅) do
    begin
      wybierz wierzchołek v:  $\forall z \in V : deg(z) \leq deg(v)$ 
      V := V \ {v};
      E := E \ F(v);
      Tr := Tr ∪ {v};
    end;
  end;

```

Rozważmy hipergraf, którego macierz incydencji przedstawia Tabela V.2.

wierzchołek	hiperkrawędź							
	1	2	3	4	1 5	2 5	3 5	4 5
1	x				x			
2		x				x		
3			x				x	
4				x				x
5					x	x	x	x

Tabela V.2.



Rys. V.9. Hipergraf przedstawiony w Tabeli V.2.
Źródło: opracowanie własne

Zgodnie z algorytmem *Lovásza–Johnsona–Chvatala* powinniśmy wybrać wierzchołek 5. Macierz incydencji hipergrafu, po usunięciu wierzchołka 5 i incydentnych z nim krawędzi: $\{1,5\}$, $\{2,5\}$, $\{3,5\}$, $\{4,5\}$ pokazuje Tabela V.3.

wierzchołek	hiperkrawędź			
	1	2	3	4
1	×			
2		×		
3			×	
4				×

Tabela V.3.

Każdy z pozostałych wierzchołków należy tylko do jednej z niepokrytych hiperkrawędzi, zatem dla tak zredukowanego hipergrafu jest tylko jedna transwersala: $Tr = \{1, 2, 3, 4\}$. Uwzględniając usunięty na początku wierzchołek 5 mamy transwersalę dla rozważanego hipergrafu: $Tr = \{1, 2, 3, 4, 5\}$, jednak tak uzyskana transwersala nie jest nie tylko transwersalą najmniejszą, ale nawet transwersalą minimalną.

W wyniku zastosowania algorytmu zachłannego *Lovásza–Johnsona–Chvatala* uzyskujemy transwersalę, której podzbiór właściwy również może być transwersalą.

Pozostaje jeszcze oszacowanie dokładności uzyskanej aproksymacji. Poniżej podajemy oszacowanie jakości aproksymacji uzyskanej w wyniku zastosowania algorytmu zachłannego *Lovásza–Johnsona–Chvatala*.

Twierdzenie V.3. Jeżeli n jest mocą zbioru wierzchołków, t_{opt} jest mocą najmniejszego zbioru wierzchołków pokrywających hipergraf, a t_a jest mocą zbioru wierzchołków uzyskanym w wyniku przedstawionej aproksymacji, to spełniona jest następująca nierówność (Johnson, 1974; Lovasz, 1975; Chvatal, 1979):

$$t_{opt} \leq t_a \leq \cdot t_{opt} (1 + \lg_2 n).$$

Algorytm aproksymacyjny pozwala znaleźć transwersalę zbliżoną do rozmiaru najmniejszej transwersali, w specyficznym hipergrafie może on znaleźć najmniejszą transwersalę, lecz istnieją hipergrafy, dla których algorytm aproksymacyjny *Lovásza–Johnsona–Chvatala* znajdzie transwersalę, która nie będzie nawet transwersalą minimalną ze względu na inkluzję.

V.3.3. Algorytm aproksymacyjny MSBT

Algorytm backtrackingowy, będący algorytmem dokładnym, ma złożoność wykładniczą, algorytm zachłanny *Lovásza–Johnsona–Chvatala* ma złożoność wielomianową, ale znaleziona transwersala może nie być nawet minimalną transwersalą. Zaprezentujemy algorytm o złożoności wielomianowej, który będzie zawsze znajdował minimalną, ze względu na inkluzję, transwersalę.

Zasada działania algorytmu MSBT

Niech $H=(X, F)$, będzie hipergrafem, dla którego szukamy minimalnej transwersali, gdzie X jest zbiorem wierzchołków, a F zbiorem hiperkrawędzi. Niech $m(X)$ będzie wierzchołkiem ze zbioru X , który należy do najmniejszej liczby hiperkrawędzi. Jeżeli jest kilka takich wierzchołków, to wybrany zostanie dowolny z nich. Niech $Z(X, F)$ będzie zbiorem wierzchołków w redukcje hipergrafu $H=(X, F)$, nie należących do żadnej hiperkrawędzi. Niech $F(x)$ będzie zbiorem wszystkich hiperkrawędzi do których należy wierzchołek x .

Algoritmy:

```

procedure MSBT(H)
begin
  Tr:=∅
  V:=X;
  Q:=X;
  E:=F;

```

dla każdej hiperkrawędzi pokrytej przez dokładnie jeden wierzchołek v

```

  Tr:=Tr∪{v};
  E:=E\F(v);
  V:=V\{v};

```

```

while  $V \neq \emptyset$  and  $E \neq \emptyset$  do
  begin
     $k := m(V)$ ;
     $V := V \setminus \{k\}$ ;
    if  $V$  nie jest transwersalą hipergrafu  $(Q, E)$  do
      begin
         $Tr := Tr \cup \{k\}$ ;
         $E := E \setminus F(x)$ ;
         $V := V \setminus Z(V, E)$ ;
      end;
    else
      begin
        dla każdej hiperkrawędzi pokrytej przez
        dokładnie jeden wierzchołek  $v$ 
           $Tr := Tr \cup \{v\}$ ;
           $E := E \setminus F(v)$ ;
           $V := V \setminus \{v\}$ ;
        end;
       $Q := V$ ;
    end;
  end;
end;

```

Istnieją hipergrafy, dla których algorytm *LJC* znajdzie mniejszą transwersalę niż algorytm *MSBT*, na przykład:

- $x_0: f_1, f_3, f_5, f_7, f_{12}, f_{17}, f_{18}, f_{20}$
- $x_1: f_1, f_3, f_4, f_8, f_{10}, f_{12}, f_{13}, f_{14}, f_{16}, f_{18}, f_{19}, f_{20}$
- $x_2: f_1, f_4, f_5, f_6, f_7, f_8, f_{11}, f_{14}, f_{16}, f_{19}$
- $x_3: f_0, f_3, f_7, f_9, f_{10}, f_{12}, f_{13}, f_{14}, f_{15}, f_{18}, f_{19}, f_{20}$
- $x_4: f_3, f_4, f_9, f_{10}, f_{14}, f_{16}, f_{17}$
- $x_5: f_5, f_6, f_8, f_9, f_{10}, f_{11}, f_{12}, f_{16}, f_{17}, f_{19}, f_{20}$
- $x_6: f_4, f_7, f_9, f_{11}, f_{15}, f_{18}, f_{19}$
- $x_7: f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{19}, f_{20}$
- $x_8: f_0, f_2, f_5, f_8, f_9, f_{10}, f_{13}, f_{15}, f_{16}, f_{18}$
- $x_9: f_2, f_6, f_8, f_{11}, f_{13}, f_{15}, f_{16}, f_{17}, f_{18}, f_{20}$

Algorytm	Moc transwersali	Transwersala			
MSBT	4	1	2	5	8
LJC	3	2	3	9	

Istnieją też hipergrafy, dla których algorytm *MSBT* znajdzie mniejszą

transwersalę niż algorytm *LJC*, na przykład:

- $x_0: f_1, f_4, f_6, f_8, f_9, f_{10}, f_{11}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{18}, f_{19}, f_{20}$
 $x_1: f_0, f_3, f_{10}, f_{11}, f_{13}, f_{15}, f_{17}, f_{18}, f_{19}, f_{20}$
 $x_2: f_1, f_2, f_7, f_{12}, f_{13}, f_{14}, f_{18}, f_{19}$
 $x_3: f_6, f_{11}, f_{12}, f_{13}, f_{14}, f_{15}, f_{16}, f_{17}, f_{19}, f_{20}$
 $x_4: f_2, f_4, f_5, f_6, f_8, f_9, f_{11}, f_{16}, f_{17}, f_{18}, f_{19}$
 $x_5: f_0, f_4, f_9, f_{10}, f_{12}, f_{16}, f_{17}, f_{18}, f_{20}$
 $x_6: f_2, f_3, f_4, f_8, f_{12}, f_{13}, f_{14}, f_{18}, f_{20}$
 $x_7: f_1, f_3, f_7, f_8, f_{13}, f_{14}, f_{16}$
 $x_8: f_0, f_6, f_7, f_9, f_{10}, f_{15}, f_{17}, f_{18}$
 $x_9: f_3, f_5, f_6, f_7, f_9, f_{11}, f_{15}, f_{16}$

Algorytm	Moc transwersali	Transwersala			
MSBT	3	1	2	4	
LJC	4	0	6	8	9

Oznaczmy przez:

$\{x_n\}$ - ciąg n -wierzchołków hipergrafu uporządkowany rosnąco ze względu na liczbę pokrytych hiperkrawędzi.

Λ - zbiór wierzchołków spełniających warunek

$$\forall x \in \Lambda \exists f \in F : f \cap X = \{x\}.$$

Zbiór Λ może na początku być zbiorem pustym i, przy odpowiednim przygotowaniu, powinien nim być. W pierwszym kroku z $\{x_n\}$ usuwamy wierzchołek x_1 , jeżeli $x_1 \in \Lambda$, to z hipergrafu usuwamy także wszystkie hiperkrawędzie f , dla których spełniony jest warunek $x_i \in f$, a wierzchołek x_1 dodajemy do zbioru Tr , który na początku jest zbiorem pustym.

Zatem w nowo powstałym hipergrafie nie ma już hiperkrawędzi pokrytych przez wierzchołek x_1 , oraz Tr wraz z pokrytymi hiperkrawędziami tworzy hipergraf dla którego jest on minimalną transwersalą. Ponieważ w każdym następnym kroku tworzymy nowy ciąg wierzchołków $\{x_i\}$ z wierzchołków które pozostały w zredukowanym hipergrafie, uporządkowany rosnąco ze

względu na c_k , oraz powtarzamy krok pierwszy, tylko dla coraz bardziej zredukowanego hipergrafu, uzyskujemy zbiór Tr spełniający warunek :

$$\forall x \in Tr \exists f \in F : f \cap Tr = \{x\}.$$

Zatem tak uzyskany zbiór Tr jest minimalną, ze względu na inkluzję, transwersalą hipergrafu.

Rozważmy działanie algorytmu *MSBT* dla drzewa:

Zauważmy, że

- każda krawędź jest zbiorem dwóch wierzchołków
- liść ma stopień 1.

Działanie algorytmu *MSBT* dla drzewa można opisać rekurencyjnie:

```
if zbiór krawędzi nie jest zbiorem pustym do  
  begin  
    usuń liść  
    dodaj do transwersali wierzchołek,  
    który należy do tej samej krawędzi  
    co usunięty liść  
    usuń z drzewa wierzchołek, który należy  
    do tej samej krawędzi co usunięty liść  
    oraz wszystkie krawędzie pokryte przez  
    ten wierzchołek  
    usuń wszystkie wierzchołki  
    o stopniu równym 0  
    wywołaj algorytm MSBT dla zredukowanego  
    drzewa i powiększonej transwersali.  
  end;
```

Dzięki takiemu działaniu wybierany jest co drugi poziom drzewa licząc od dołu co daje nam minimalne pokrycie.

V.3.4. Algorytm złożony

Algorytm dokładny jest algorytmem o wykładniczej złożoności i jako taki może mieć zastosowanie tylko w hipergrafach o niewielkiej liczbie wierzchołków. Algorytmy *LJC*, *SBT* oraz *RSBT* mają wilelomianową złożoność obliczeniową i jako takie mogą być zastosowane do większych hipergrafów. W celu wykorzystania tych algorytmów proponujemy następujące podejście:

Niech $H=(X, F)$ będzie rozważanym rozważany hipergrafem.

Tr_1 := transwersala znaleziona algorytmem LJC
 Tr_2 := transwersala znaleziona algorytmem MSBT
 Tr_3 := transwersala znaleziona w hipergrafie $H' = (Tr_1, F)$ algorytmem MSBT
 Tr := $\min(Tr_2, Tr_3)$

Dzięki takiemu zastosowaniu obydwu algorytmów zachłanych uzyskujemy transwersalę spełniającą warunek oszacowania jakości aproksymacji algorytmu *Lovásza–Johnsona–Chvatala*, będącą jednocześnie transwersalą minimalną.

V.4. Zastosowania

V.4.1. Zadanie optymalizacji w logistyce

W ostatnich latach nastąpił burzliwy rozrost sieci centrów logistycznych na wszystkich kontynentach, w postaci różnego rodzaju sieci hurtowni specyficznych towarów, dealerów wyrobów poszczególnych firm, supermarketów artykułów konsumpcyjnych itp. Spowodowało to powstanie wielu przedsiębiorstw transportowych. Przedsiębiorstwa te są bądź jedną ze składowych centrum logistycznego, bądź świadczą swoje usługi niezależnie. Istotnym elementem takiego przedsiębiorstwa transportowego jest liczba rodzajów posiadanego taboru. Optymalizacja tej liczby ma ogromny wpływ na kondycję finansową przedsiębiorstwa (Maźbic-Kulma, Sęp, Stańczak, 2006; Eiter, Gottlob, 1995).

Załóżmy, zatem, że rozpatrywane przez nas przedsiębiorstwo miałyby świadczyć usługi w zakresie przewozów materiałów budowlanych i opałowych, surowców, gruzu, węgla, żwiru, piachu itp. Środkami transportu będą samochody ciężarowe jednego typu wraz z wymiennymi naczepami lub przyczepami. Jak wiadomo, nie wszystkie wspomniane materiały mogą być wożone razem, niektóre z nich wymagają specjalizowanego sprzętu, takiego jak np. betoniarka. Zakładając szeroki wachlarz tych przewozów firma w swoim biznes planie musi określić minimalną liczbę rodzajów naczep lub przyczep. Poniżej przedstawimy metodę wyznaczania liczby rodzajów naczep lub przyczep przy pomocy algorytmu znajdowania minimalnej transwersalni w hipergrafie (Maźbic-Kulma, Sęp, 2006a,b, 2007; Eiter, Gottlob, 1995).

Przyjmijmy, zatem następujące oznaczenia:

n – liczba towarów które firma zamierza przewozić

f_i - i -ty towar który firma zamierza przewozić ($i=1,2,\dots,n$),

$F = \{f_1, f_2, \dots, f_n\}$ – zbiór towarów, które firma zamierza przewozić

m – liczba rodzajów naczep, które firma zamierza zakupić

x_j - j -ta naczepa ($j=1,2,\dots,m$)

$X = \{X_1, X_2, \dots, X_m\}$ – zbiór rodzajów naczep.

Ponadto zakładamy, że dla każdej naczepy x_j dany jest podzbiór $F_j \subseteq F$ towarów, które dana naczepa może przewozić.

$$\bigcup_{j=1}^m F_j = F.$$

Dla dalszych rozważań możemy przyjąć, że x_j są wierzchołkami hipergrafu. Zbiór X będzie więc zbiorem wierzchołków hipergrafu. Przypisany każdemu wierzchołkowi x_j podzbiór F_j będzie zbiorem hiperkrawędzi, czyli

$$\bigcup_{F_j \in F} F_j = X.$$

Przyjmując powyższe oznaczenia nasze zadanie można sformułować jako:

Dla danego hipergrafu (X, F) wyznaczyć minimalną transwersalę. Wyznaczona transwersala będzie stanowiła odpowiedź na pytanie, które rodzaje naczep powinno zakupić przedsiębiorstwo transportowe.

Zakładamy, że dana jest lista $n=270$ towarów, które firma zamierza przewozić. Rozpoczynając działalność gospodarczą, firma zmierza zakupić samochody ciężarowe jednego typu wraz z wymiennymi naczepami lub przyczepami. Firma ma do wyboru $m=30$ typów naczep lub przyczep, którymi mogłaby przewozić oferowane do przewozu materiały. Niektóre materiały mogą być wożone tą samą naczepą lub przyczepą. Przyjmijmy zatem, że dana jest macierz $A = [a_{ij}]_{n \times m}$, gdzie $a_{ij} = 1$, gdy towar i można wozić przyczepą (naczepą) j , zaś $a_{ij} = 0$ w przeciwnym przypadku.

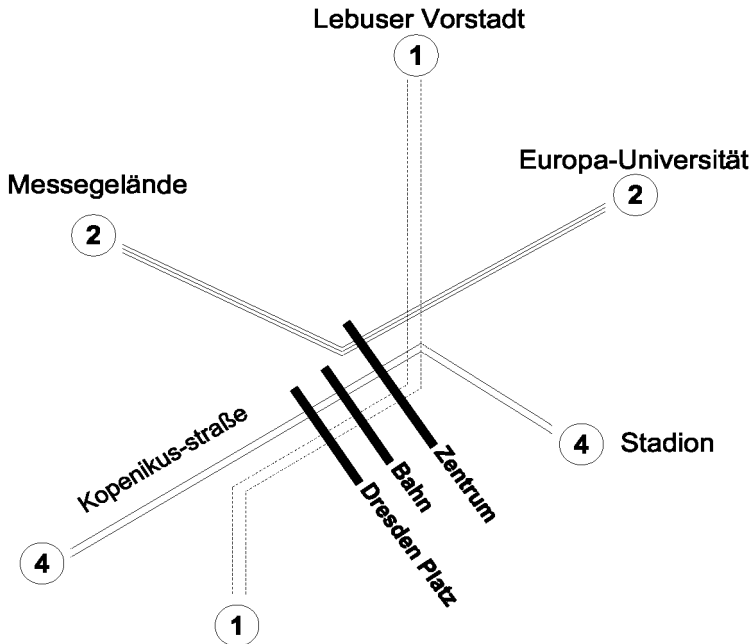
Omówione tutaj algorytmy dały następujące wyniki:

Algorytm	Moc transw.	Wybrane rodzaje naczep											
		1	6	9	11	13	14	16	17	20	22	24	26
MSBT	12	1	6	9	11	13	14	16	17	20	22	24	26
LJC-MSBT	11	6	9	13	14	16	20	22	23	25	26	28	

LJC-MSBT to transwersala znaleziona przy pomocy algorytmu *MSBT* w redukcje uzyskanym przy pomocy algorytmu *LJC*.

V.4.2. Sieć transportowa – struktura „hub-and-spoke”

Zdefiniowana poprzednio struktura **hub-and-spoke (HS)** może być zastosowana w wielu dziedzinach. Używając tej struktury w systemach transportowych uzyskujemy dużą koncentrację ruchu pomiędzy węzłami oraz synchronizację połączeń, zwiększając jakość usług. Powoduje to również obniżenie kosztów przewozu pasażerów. I tak np. struktura **hub-and-spoke** została wykorzystana w transporcie miejskim we Frankfurcie nad Odrą. Wydzielono tam cztery węzły przesiadkowe (huby). Wprowadzenie **hub-and-spoke** spowodowało znaczne zwiększenie częstotliwości ruchu. Funkcjonalność tego systemu jest szczególnie widoczna w godzinach pozaszczytowych, gdy oferta przewozowa jest już z powodów ekonomicznych ograniczona.



Rys. V.10. Struktura komunikacji miejskiej typu **hub-and-spoke** we Frankfurcie nad Odrą

Źródło: opracowanie własne na podstawie <http://zielonagora...>

Innym przykładem zastosowania struktury typu **hub & spoke** jest sieć połączeń lotniczych.



Rys. V.11. Przykładowa sieć połączeń lotniczych typu *hub & spoke*

Źródło: opracowanie własne

Linie korzystające z koncepcji opartej na węzłach tranzytowych (hub-and-spoke) mogą oferować loty z większą częstotliwością. Uzyskują one możliwość obsługi lotów długodystansowych, a także umożliwiają koncentrację potoku, co pozwala uzyskać większe zapełnienie (load factor) miejsc siedzących w samolocie.

Jak już wspomnieliśmy na wstępie, przedstawione powyżej algorytmy oraz przykłady ich zastosowań nie wyczerpują wszystkich aspektów problematyki optymalizacji kombinatorycznej. Skoncentrowaliśmy się na tych metodach, które były przedmiotem naszych badań i zostały opublikowane w pracach autorów, wymienionych w bibliografii. Mamy nadzieję, że będą one pomocne tym wszystkim czytelnikom, którzy pragną się zaznajomić z tą tematyką, bądź rozwiązać własne zadanie.

V.5. Literatura

- Berge C. (1989) *Hypergraphs. Combinatorics of Finite Sets*. North-Holland.
- Błażewicz J. (1989) *Złożoność obliczeniowa problemów kombinatorycznych*. WNT.
- Chvatal V. (1979) A greedy heuristic for the set-covering problem. *Mathematics and Operations Research* 4.
- Cormen T., Leiserson C., Rivest R. (1997) *Wprowadzenie do algorytmów*. WNT, Warszawa.
- Cowen L. (1998) *Approximation Algorithms*. Hopkins University.
- Eiter T., Gottlob G. (1995) Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal of Computing*, 24, 1278-1304.
- Fechner I. (2006) Centra logistyczne jako czynnik wzrostu efektywności łańcuchów dostaw. *Najlepsze praktyki w logistyce*. LOGISTICS 2006. <http://zielonagora.blogspot.com/2008/02/znaczenie-wzw-przesiadkowych-w.html>
- Johnson D. S. (1974) Approximation Algorithms for Combinatorial Problems. *Journal of Computer and System Sciences* 9.
- Jukna S. (2001) *Extremal Combinatorics with Applications in Computer Science*. Springer.
- Korte B., Vygen J. (2000) *Combinatorial Optimization, Theory and Algorithms*. Springer.
- Lovasz L. (1975) On the ratio of optimal integral and fractional covers. *Discrete Mathematics* 13.
- Lund C., Yannakakis M. (1994) On the Hardness of Approximating Minimization Problems. *Journal of the ACM*.
- Mażbic-Kulma B., Piasecki S. (2004) Geneza i organizacja sieci centrów logistycznych. *Wybrane zagadnienia logistyki stosowanej*. Kraków.
- Mażbic-Kulma B., Owiński J., Sęp K. (2008) Application of selected methods of graph theory and combinatorial heuristics to minimising the number of transit nodes in an air network. *Total Logistic Management*, Kraków.
- Mażbic-Kulma B., Sęp K. (2005) Problem wyboru węzłów tranzytowych w sieci lotniczej. W: *Materiały konferencji: "Systemy logistyczne. Teoria i praktyka"*, Warszawa, 14 września 2005, Politechnika Warszawska, 341-348.
- Mażbic-Kulma B., Sęp K. (2006a) Algorytm znajdowania minimalnej transwersali w grafie jako narzędzie wspomagające planowanie działalności przedsiębiorstwa transportowego. *Logistyka*, 6, 1-7.

- Maźbic-Kulma B., Sęp K. (2006b) Baza wierzchołkowa w hipergrafie jako metoda przydzielania zadań. W: Kacprzyk J., Budziński R. (red.): *Badania operacyjne i systemowe 2006. Metody i techniki*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 283-190.
- Maźbic-Kulma B., Sęp K. (2007a) Minimalizacja liczby węzłów w sieci logistycznej. W: Bukowski L.A., Feliks J., Karkula M., Lichota A. (Red.): *ROCNIAK 2007*. Komitet Transportu PAN, Kraków, 59-65. Seria: *Wybrane Zagadnienia Logistyki Stosowanej*.
- Maźbic-Kulma B., Sęp K. (2007b) Some approximation algorithms for minimum vertex cover in a hypergraph. W: Kurzyński M., Puchała E., Woźniak M., Żołnierek A. (Eds.): *Computer Recognition Systems 2*. Springer-Verlag, Berlin-Heidelberg, 250-257. Seria: *Advances in Soft Computing*.
- Maźbic-Kulma B., Sęp K. (2007c) Zastosowanie algorytmu SBT jako metody minimalizacji taboru w przedsiębiorstwie transportowym. W: Dąbrowa-Bajon M. i inni (red.): *Transport w systemach logistycznych*. Politechnika Warszawska, Warszawa, 115-125. Seria: *Prace Naukowe - Transport*.
- Maźbic-Kulma B., Sęp K., Stańczak J. (2006) Ewolucyjna metoda optymalizacji rozmieszczenia punktów recyklingu pojazdów. W: A. Bukowski L.A. (red.): *Wybrane zagadnienia logistyki stosowanej*. TEXT, Kraków, 318-325.
- Motvani R. (1992) *Lecture Notes on Approximation Algorithms*. Stanford University.
- O'Kelly M.E (1987) A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research* 32, 392-404.
- O'Kelly M.E. and Bryan D. (2002) Interfacility interaction in models of hubs and spoke networks. *Journal of Regional Science*, 42 (1), 145-165.
- Sapiecha P. (1993) Approximation algorithm for a certain class of NP-hard problems. *Found. Comput. Decision Sci*.
- Sapiecha P., Sapiecha E. (1994) An algorithm that is capable of solving the minimal reduct problem fast, on average. *RSSC'1994*, San Jose, CA.
- Sapiecha P., Selvaraj H., Stańczak J., Sęp K., Łuba T. (2004) A hybrid approach to a classification. W: Kłopotek M.A., Wierzchoń S.T., Trojanowski K. (Eds.): *Intelligent information processing and web mining*. Springer Verlag, Berlin, 99-108. Seria: *Advances in Soft Computing*.
- Sęp K., Kułaga P., Sapiecha P. (2005) Approximation algorithm for the

- argument reduction problem. W: Kurzyński M., Puchała E., Woźniak M., Żołnierek A. (Eds.): *Computer Recognition Systems CORES'05*. Springer-Verlag, Berlin, Heidelberg, New York, 243-248. Seria: *Advances in Soft Computing*.
- Slavik P. (1996) *A Tight Analysis of the Greedy Algorithm for Set Cover*. STOC'96. State University of New York at Buffalo, Dept. of Mathematics.
- Stańczak J., Potrzebowski H., Maźbic-Kulma B., Sęp K. (2008) Evolutionary approach to solve hub-and-spoke problem using alpha-cliques *Evolutionary Computation and Global Optimization*. Oficyna Wydawnicza Politechniki Warszawskiej.
- Vazirani V. (1997) *Approximation Algorithms*. Lecture Notes, Georgia Institute of Technology.
- Williamson D. (1999) Lecture Notes on Approximation Algorithms. *IBM Research Report* RC 21409 02 17.
- Wilson R.J. (1996) *Introduction to Graph Theory*. Addison Wesley Longman.

Barbara Mażbic-Kulma, Krzysztof Sęp

Książka poświęcona jest opisowi zastosowań metod sformalizowanych do wybranych zagadnień społeczno-gospodarczych i administracyjnych o charakterze przestrzennym. Rozpatrywane są zagadnienia regionalizacji i typologii przestrzennej, logistyki i organizacji transportu, zrównoważonego rozwoju, czy jakości stron internetowych samorządów w zestawieniu z położeniem odpowiednich jednostek.

ISSN 0208-8029
ISBN 9788389475251

Instytut Badań Systemowych PAN

W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy prosimy o kontakt z Instytutem Badań Systemowych PAN
ul. Newelska 6, 01-447 Warszawa
tel. (22) 3810 277; e-mail: bibliote@ibspan.waw.pl