Barbara MAŻBIC-KULMA*, Jan W. OWSIŃSKI*, Krzysztof SĘP*

APPLICATION OF SELECTED METHODS OF GRAPH THEORY AND COMBINATORIAL HEURISTICS TO MINIMISING THE NUMBER OF TRANSITS NODES IN AN AIR NETWORK

Abstract: In the paper we present the notion of *α-clique* and some of its properties. Covering with *α-cliques* is a preprocessing method for an air network, described as a graph, in which vertices correspond to airports and edges correspond to air connections. Using the *α-clique* cover we obtain a hypergraph, in which we find the minimum transversal. The set of vertices thus obtained is the sought-for set of transits nodes, called hubs. Using the *α-clique* concept instead of proper cliques we can obtain the solution to the graph covering problem easier.

Keywords: hub and spoke, graph, decision support systems, clique.

1. Introduction

The hub and spoke problem is one of the most important issues in air traffic optimisation. Most of the market is serviced by the former national carriers. These lines make use of the concept based on the hub-and-spoke structure. Owing to hubs they can offer more frequent flights. They gain the capacity of servicing long-distance flights, which shall remain outside the capability of the "low cost" carriers. Hubs support concentration of traffic flows, and thereby an increase of the load factor in the aircraft.

The hub-and-spoke structure of a connection graph enables concentration of flows of transported persons/goods among vertices. Figure 1 presents an example of the initial structure of air connections among main cities before concentration. There are multiple point-to-point connections. The flights are rather infrequent and expensive. There are difficulties with flight synchronisation at the interchange airports, mainly due to the fact that connections are serviced by more than one carrier. An adequate choice of several transit nodes and local connections could improve the transport system, reducing carrier costs and increasing service efficiency. After

* Systems Research Insitute, Polish Academy of Sciences, Warsaw, Poland

concentration (Fig. 2), the graph of connections turns into a hub-and-spoke structure and each carrier controls one hub airport with several short point-to-point connections (spokes). The hub-and-spoke structure presented in Figure 2 reduces the complexity of the management problem.

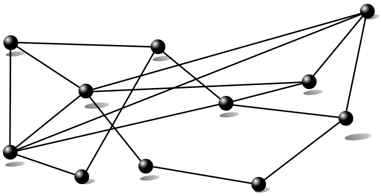

Fig. 1. *Input structure* (Cormen *et al.* 2004)

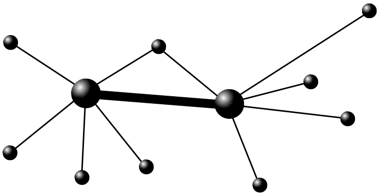

Fig. 2. *Corresponding "hub-and-spoke" structure* (Horner *et al.* 2001)

Thus, given some starting point structure, represented by a graph, in which the majority of edges represent "point-to-point", connections, our problem is to obtain the hubs in such a graph, with the remaining airports corresponding to spokes (Fig. 3).

The literature contains numerous variants of the hub location problem. It is common to represent it in the form of integer programming (Horner *et al.*2001, Min *et al.* 2004, O'Kelly 1987, O'Kelly *et al.* 2002). In Mażbic-Kulma *et al.* (2005) the authors hereof showed an algorithm based on the methods of graph theory and networks.

The algorithm proposed was composed of two parts:

– obtaining a separable cover of graph with cliques,
– obtaining a transversal in the hypergraph, a minimum one, in the sense of inclusion.

In this paper we describe a modified method based on the following algorithms:

– obtaining a separable cover of graph with cliques,
– obtaining the maximum α*-clique*,
– obtaining a transversal in the hypergraph, a minimum one in the sense of inclusion.

Thus we shall successively present the notion of α*-clique* and its definite properties (Potrzebowski *et al.* 2006a, 2006b, 2007). Finding the covering of the respective graph with cliques or α*-cliques* is the method of initial processing of the air network (Mażbic-Kulma *et al.* 2005). Owing to the application of the inseparable covering with α*-cliques* of the graph, we obtain a hypergraph, in which we look for the minimum vertex base, that is – a transversal in the hypergraph (Mażbic-Kulma *et al.* 2006, 2007a, 2007b, 2007c). The set of vertices thus obtained is the sought-for set of transit nodes, i.e. hubs (Mażbic-Kulma *et al.* 2005, O'Kelly *et al.* 2002). Owing to the application of the notion of α*-clique* instead of clique, we can more easily obtain the set cover of the graph of connections.

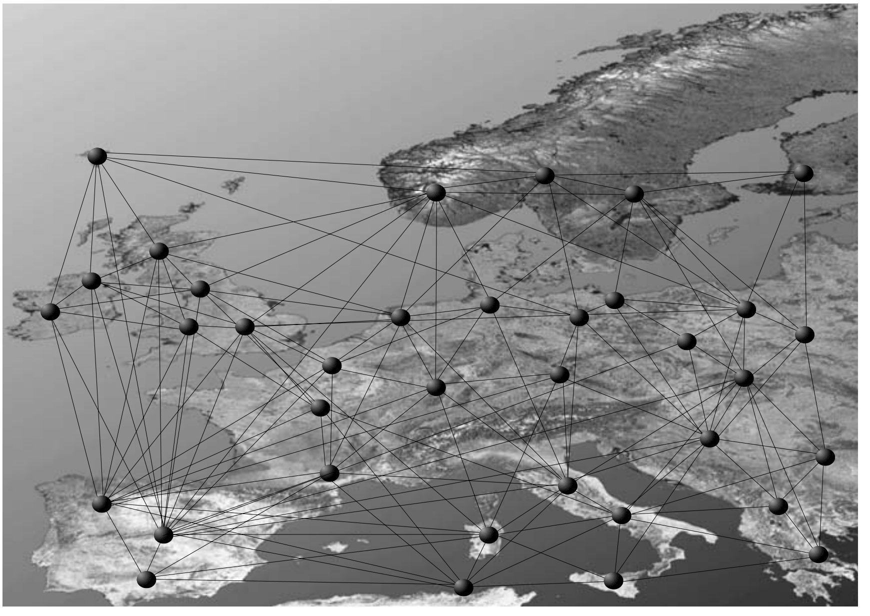

Fig. 3. *An example of air network*

2. Basic notions

Following (Berge 1989) and (Wilson 1996), we shall first introduce some basic notions of graph theory, illustrated in Figures 4 through 10:

- A ***graph*** is a pair $G = (V, E)$, where V is a non-empty set of vertices and E is a set of edges. Each edge is a pair of vertices $\{v_1, v_2\}$ with $v1 \neq v2$.
- Two vertices $v_i, v_j \in V$ in a graph $G = (V, E)$ are called ***incident*** if $v_i, v_j \in E$.
- A ***sub-graph*** of a graph $G = (V, E)$ is a graph $G' = (V', E')$, where $V' \subseteq V$ and $E' \subseteq E$, such that for all $e \in E$ and $e = v_1, v_2$ if $v_1, v_2 \in V'$ then $e \in E'$.
- A ***path*** in a graph $G = (V, E)$ from vertex s to vertex t is a sequence of vertices $v_1, \ldots, v_n$ such that $s, v_1 \in E, \{v_i, v_{i+1}\} \ni E$ for $i = 1, 2, \ldots, n-1, \{v_n, t\} \in E$.
- A graph $G = (V, E)$ is a ***connected graph***, if for each pair of vertices there is a path between them.
- A ***clique*** (a *complete sub-graph*) $Q = (V_q, E_q)$ in graph $G = (V, E)$ is a graph such that $V_q \subseteq V$ and $E_q \subseteq E$ and each pair of vertices $v_1, v_2 \in V_q$ fulfils the condition $\{v_1, v_2\} \in E_q$.
- A ***maximum clique*** $Q_M = (V_q, E_q)$ in a graph $G = (V, E)$ is a clique for which there exists no vertex $v \in V$ and $v \notin V_q$ such that $Q' = (V', E')$ is a clique, where $V' = V \cup \{v\}$ and $E' \subseteq E$ and each pair $v_1, v_2 \in V'$ of vertices fulfils the condition $v_1, v_2 \in E'$.

– The ***degree*** of a vertex is the number of edges to which this vertex belongs. Thus, for instance, in the graph in Figure 8, vertex 2 has degree 3 and vertex 4 has degree 5.

– A ***hypergraph*** is an ordered pair $H = (X, F)$, where X is a non-empty, finite set of vertices, and F is a non-empty family of different subsets of the set X fulfilling the condition:

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

We call F the set of edges of the hypergraph.

Note that an edge of a hypergraph can contain any number of vertices (even one), which makes a difference between hypergraphs and graphs. In fact, the hypergraph can be viewed as a direct generalisation of the graph.

– A ***transversal*** Tr of a hypergraph $H = (X, F)$ is a subset of the set of vertices, $Tr \subset X$, such that for each edge f of the hypergraph, there exists a vertex $x \in Tr$ such that $x \in f$.

– A transversal T_{mi} is called minimum in the sense of inclusion if no subset of T_{mi} is a transversal.

– A transversal is called minimum in the sense of cardinality if there exists no transversal with fewer elements.

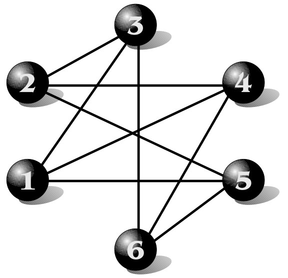

Fig. 4. *An example of a graph*

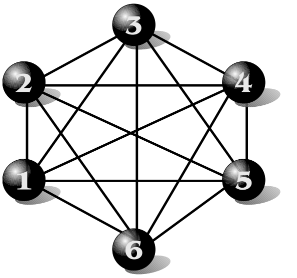

Fig. 5. *Fully connected graph*

In (Potrzebowski *et al.* 2006) the notion of *α-clique* was introduced: Let $A = (V_\alpha, E_\alpha)$ be a sub-graph of a graph $G = (V, E)$, where $\alpha(0, 1], V_\alpha \subseteq V, E_\alpha \subseteq E, k = Card(V_\alpha)$, k_i be the number of vertices $v_j \in V_\alpha$ such that $\{v_i, v_j\} \in E_\alpha$:

1. For $k = 1$ the sub-graph A of graph G is an *α-clique*.
2. For $k > 1$ the sub-graph A of graph G is an *α-clique* if for every vertex $v_j \in V_\alpha$ the condition

$$\alpha \leqslant \frac{k_i}{k}$$

is fulfilled.

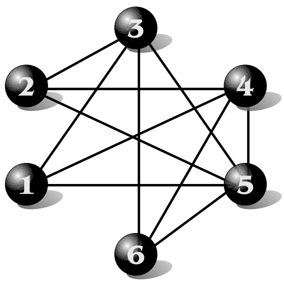

Fig. 6. *An example of a graph*

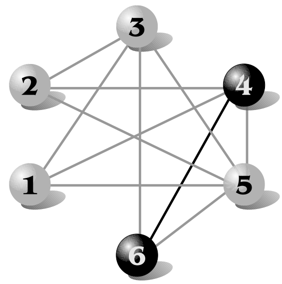

Fig. 7. *A clique in the graph from Figure 6*

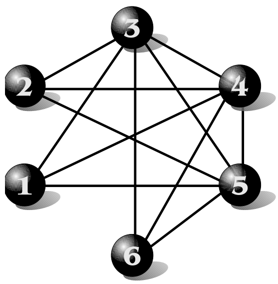

Fig. 8. *An example of a graph*

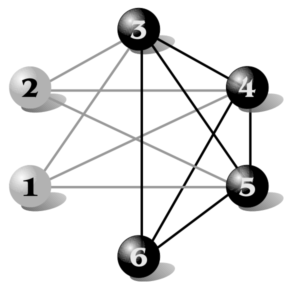

Fig. 9. *A fully connected sub-graph in the graph from Figure 8*

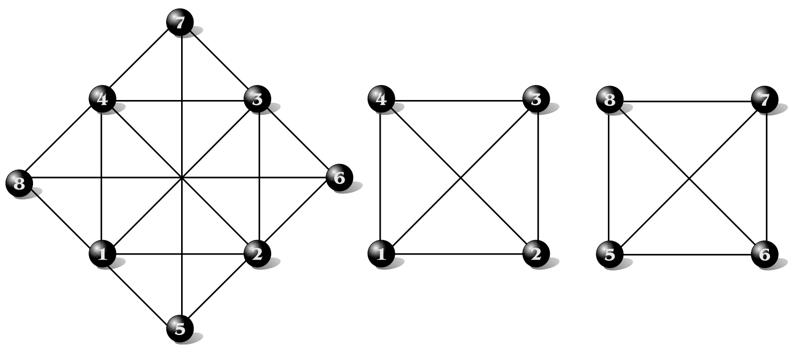

Fig. 10. *An example of a not connected graph for* $\alpha = \frac{1}{2}$

Take an arbitrary vertex v_i belonging to an α-*clique* $A = (V_\alpha, E_\alpha)$. Let $k = Card(V_\alpha)$, ki be the number of vertices $v_j \in V$? fulfilling the condition $\{v_i, v_j\} \in E_\alpha$. For $\alpha > \frac{1}{2}$ we have $\frac{k_i}{k} \geqslant \alpha > \frac{1}{2}$, that is $\frac{k_i}{k} > \frac{1}{2}$; hence $k_i > \frac{1}{2}k$. It can easily be proved with the set theoretical argument that then for any two vertices, the sets of vertices incident with each of them have a non-empty intersection, so the α-*clique* with $\alpha > \frac{1}{2}$ constitutes a connected graph (Potrzebowski *et al.* 2006). For $\alpha \leqslant \frac{1}{2}$ the obtained sub-graph may not fulfil the condition of connectedness – an example of such a situation is shown in Figure 10.

2.1. Algorithms for finding *maximum clique* and maximum α-*clique*

The Exact Algorithm

The Exact Algorithm (Kułaga *et al.* 2005), (Mażbic-Kulma *et al.* 2005), (Sapiecha *et al.* 2004) is a backtracking algorithm with selection of variables in the lexicographical order.

The Exact Algorithm uses DFS (Depth-first search) strategy. Owing to the use of the backtracking method The Exact Algorithm avoids multiplication of the same sets. For a hypergraph with n vertices the full backtracking tree has $2^n - 1$ nodes. For a hypergraph $H = (X, F)$ The Backtracking Algorithm works as follows:

Algorithm 1: *The Exact – Backtracking algorithm*

```
Tr := X;
V := X;
procedure BT(V)
begin
  for a ∈ V selected in lexicographical order do
    if V\{a} is not a transversal of hypergraph H then
      begin
        if Card(V){a} < Card(Tr) then
          Tr := V{a};
        BT(V{a});
      end

  return Tr;
end
```

An Approximating Greedy Algorithm

Maximum clique

Let be:

$G(V, E)$ – a graph with nonempty set of vertices, V, and set of edges, E

Q – subset of the set of vertices V (a clique)

$deg(v)$ – degree of a vertex v

$maxdeg(G)$ – vertex with the biggest degree in graph G

$remove(G, v)$ – removal from graph G of vertex v and all edges containing vertex v.

$Y(Q)$ – subset of the set of vertices V such that all vertices from Y are incident to each other.

Z – graph with the set of vertices Y, set of edges is a subset of the set V, such that each edge is a pair of vertices (v_1, v_2), with $v1 \neq v2$ and $v1, v2 \in Y$

Cl – set of graphs $\{G_1(V_1, E_1), G_2(V_2, E_2), \ldots, G_m(V_m, E_m)\}$ such that $V_i \cap V_j = \emptyset$ for all $i \neq j,\ i, j = 1, 2, \ldots, m$, and $V_1 \cup V_2 \cup \cdots \cup V_m = V$

***Algorithm* 2**: *Maximum clique*

```
procedure max_clique(G)
input
V;    E;
output
Q;
begin
    Q := ∅;
    Q = Q ∪ maxdeg(G);
    while Y(Q) ≠ ∅ do
        begin
            Q = Q ∪ maxdeg(Z); remove(G, maxdeg(Z));
        end
end
```

In solving the maximum *α-clique* problem, we can use The Maximum Clique Algorithm with the set $Y(Q)$ being such a subset of the set of vertices V that if we add any vertex from $Y(Q)$ to the set Q, then the set Q with the corresponding edges is an *α-clique*.

For obtaining the division of graph $G(V, E)$, we propose the following algorithm:

***Algorithm* 3**: *Clustering*

```
procedure cluster(G)
input
V;    E;
output
Cl;
begin
    Cl :=;
    while V ≠ ∅ do
        begin
            Cl = max_clique(G); //Cl{V_i, E_i}
            V = V\V_i;
            E = E\E_i;
        end
end
```

where Cl is a set of *α-cliques*, in particular, for $\alpha = 1$, Cl is a set of cliques.

The algorithm presented above provides one solution that is not subject to later modification. It is a greedy algorithm, yielding an approximation of the optimal solution.

The maximum clique problem is an NPH problem, meaning that we do not know exact algorithms of polynomial complexity solving this problem (Chvatal 1979, Corment *et al.* 2004, Cowen 1998, Feige 1998, Kumlander 2007, Lovasz 1975, Protasi 2001).

The maximum clique problem is a particular case of the maximum *α-clique* problem; hence the maximum *α-clique* problem is also an NPH problem. It can be solved in an exact manner by generating all the non-empty subsets of the set of vertices using the backtracking tree. In this case computational complexity is at $O(2^n)$ (Chvatal 1979, Hansen *et al.* 2004, Johnson 1974, Lovasz 1975). In the case of a problem of a clique in a graph, if the set of vertices is a clique, then each nonempty subset of this set is a clique too. In the *α-clique* problem for $\alpha < 1$, there exist cases in which for a set of vertices being an *α-clique* there exist its nonempty subsets not being *α-cliques*. This constitutes an additional problem in constructing an approximating greedy algorithm.

In a general case, verification whether a graph is an *α-clique* for a given α has a polynomial computational complexity (if we dispose of a table of vertices and their degrees, the complexity is linear). To verify whether a graph is an *α-clique*, it suffices to find all vertices with the minimum degree. If these vertices fulfil the *α-clique* criterion, the graph is an *α-clique*; otherwise, the graph is not an *α-clique*. For the incidence matrix of a graph, from which we have separated a subgraph, if we need to verify whether this subgraph is an *α-clique*, we have to additionally calculate the degrees of all the vertices of this subgraph. In the worst case this could be $frack^2 - k2$, where k is the number of vertices in the graph considered. For problems with exponential computational complexity the exact algorithms are practically useless, and so it is advisable to use approximation algorithms (Chvatal 1979, Cowen 1998, Feige 1998, Hansen *et al.* 2004, Hochbaum 1997, Jukna 2001, Kloks *et al.* 1998, Kumlander. 2007).

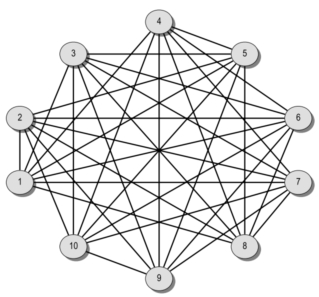

Fig. 11. *An example of α-clique for $\alpha = 0.8$*

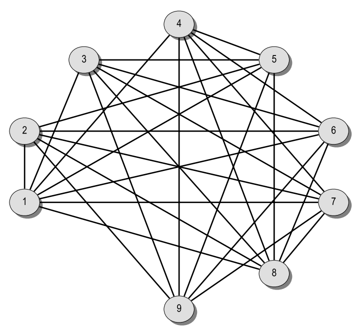

Fig. 12. *A subgraph of the graph from Figure 11 which is not an α-clique for $\alpha = 0.8$*

2.2. Algorithms for finding the minimum transversal in hypergraphs

A minimum transversal of a hypergraph can be found by generating all nonempty subsets of the set of vertices, so we also here propose to use the lexicographical backtracking algorithm as an exact algorithm (Kułaga *et al.* 2005, Mażbic-Kulma *et al.* 2005, Sapiecha *et al.* 2004).

Selected approximation algorithms

The *Lovász-Johnson-Chvatal* algorithm (Chvatal 1979, Hansen et al. 2004, Johnson 1974, Kułaga *et al.* 2005, Lovasz 1975, Sapiecha *et al.* 2004) is a greedy algorithm – at each step it chooses a vertex that covers the largest number of edges.

Using the Lovász-Johnson-Chvatal algorithm we assume that the set of vertices covering the hyper-edges should contain these vertices which cover the largest number of edges. Let $deg(x)$ be the degree of a vertex x and $F(x)$ be the set of edges to which the vertex x belongs.

The *Lovász-Johnson-Chvatal* algorithm for the hypergraph $H = (X, F)$ works as follows:

Algorithm 4: *The Lovász-Johnson-Chvatal algorithm*

```
procedure LV(H)
begin
    Tr := ∅;
    V := X;
    E := F;
    while E ≠ ∅ do
        begin
            select vertex v : ∀z ∈ V : deg(z) ⩽ deg(v)
            V := V\v;
            E := E\ F(v);
            Tr := Tr ∪ v;
        end
end
```

The *Lovász-Johnson-Chvatal* algorithm finds a transversal of a hypergraph; for some hypergraphs it may be a minimum transversal in the sense of cardinality, but there exist hypergraphs for which the *Lovász-Johnson-Chvatal* algorithm finds a transversal which is not minimum in the sense of inclusion; see Figure 13 (Sapiecha *et al.* 2004).

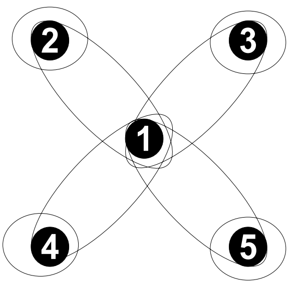

Fig. 13. *An example of a hypergraph for which the Lovász-Johnson-Chvatal algorithm finds a transversal which is not minimum in the sense of inclusion*

The MSBT Algorithm

The MSBT algorithm seeks the vertices with the lowest degrees and removes them from the set of vertices. If after the removal of a vertex the reduct is not a transversal, then the removed vertex should be added to the transversal under construction, and the edges incident with this vertex are eliminated from the hypergraph – they are deemed to be covered.

If without the removed vertex the reduct is still a transversal, then we have to look in it for all the vertices with the following property: in the reduct there is at least one edge covered by exactly that very vertex. We remove these vertices and all the edges covered by them from the reduct and we add these vertices to the transversal. This procedure is repeated until all edges are removed, i.e., until all edges are covered.

We carry out the $MSBT$ algorithm as follows:

***Algorithm** 5*: *MSBT Algorithm*

```
procedure MSBT(H)
begin
    T_r := ∅; V := X; Q := X; E := F;
    for each edge covered by exactly one vertex v
            T_r := T_r ∪ {v};        E := E\ F(v);
            V := V\{v};
    while V ≠ ∅ and E ≠ ∅ do
        begin
            k := m(V);
            V := V\k;
            if V is not a transversal of hypergraph (Q, E) then
                begin
                    Tr := Tr ∪ k;
                    E := E\ F(x);
                    V := V\ Z(V, E);
                end
            else
                begin
                    for each edge covered by exactly one vertex v
                    Tr := Trv;
                    E := E\ F(v);
                    V := V\v;
                end
            Q := V;
        end

end
```

The MSBT algorithm ensures finding a minimum transversal in a hypergraph in the sense of inclusion, and for an arbitrary tree it finds a minimum transversal in the sense of cardinality (the smallest transversal), (Mażbic-Kulma *et al.* 2006, 2007a, 2007b, 2007c).

Let $H = (X, F)$ be a hypergraph for which a minimum transversal is sought; X – the set of its vertices, F – the set of its edges. Let $m(x)$ be a vertex in X incident with the minimum number of edges; if there are several such vertices, we pick any one of them. Let $Z(X, F)$ be the set of vertices in the reduct of the hypergraph $H(X, F)$; the elements of $Z(X, F)$ do not belong to any edge. Let $F(x)$ be the set of all edges incident with the vertex x.

3. The method for obtaining the "hub & spoke" network

For obtaining the "hub & spoke" topology, we propose the following method.
Let Cl be the set of α-*cliques* and $G(V, E)$ be the graph of connections.

- We obtain the maximum separated cover of graph G by *maximum cliques* using *cluster* procedure.
- We expand each maximum clique to an α-*clique* by adding vertices from the outside of it.
- We obtain hypergraph $H(V, Cl)$;
- We obtain a transversal tr_1 using the LJC algorithm.
- We obtain a transversal tr_2 using the $MSBT$ algorithm.
- We obtain a transversal tr_3 using the $MSBT$ algorithm for the hypergraph (tr_1, Cl).
- The sought for transversal, tr, is this of the transversals tr_2 and tr_3 whose cardinality is smaller.
- Each vertex v_i of transversal tr is connected with the vertices belonging to the edges covered by vertex v_i.

We obtained the graph of connections. Figure 14 shows an example of the ***hub*** & ***spoke*** air connection network.

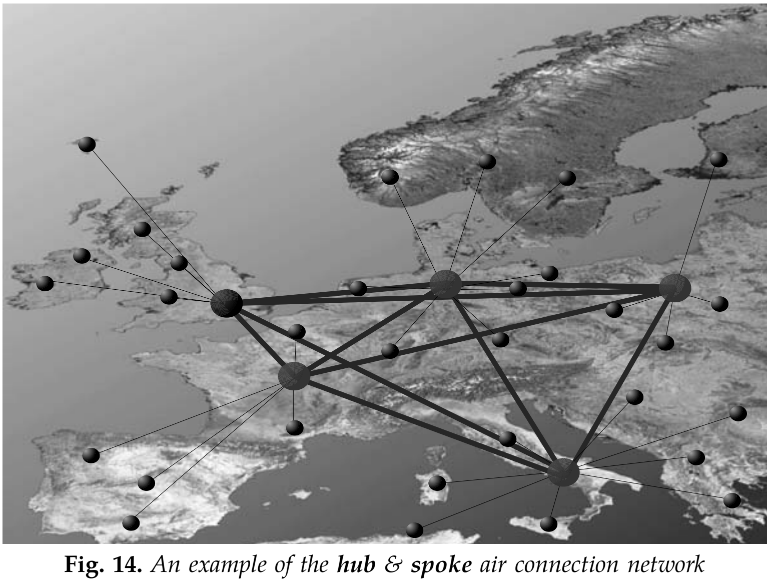

Fig. 14. *An example of the **hub** & **spoke** air connection network*

When we use the above method, we are certain that the cardinality of the transversal is not larger than the number of maximum cliques. By expanding cliques to *α-cliques*, we could obtain the smallest cover, i.e., the smallest number of hubs.

4. Conclusions

As already indicated in the introduction, selection of hub vertices is one of the most important problems in air traffic optimisation. The airlines using the concept based on hub-and-spoke (or transit nodes) can offer more frequent flights. They gain the possibility of servicing long-distance flights and concentrating traffic flows, which leads to higher load factor of the flights executed.

References

Berge C. 1989. *Hypergraphs, Combinatorics of Finite Sets*. North-Holland.

Chvatal V. 1979. *A greedy heuristic for the set-covering problem*. Mathematics and Operations Research, 4.

Cormen Th.H., Leiserson Ch.E., Rivest R.L., Stein C. 2004. *Wprowadzenie do algorytmów*. WNT, Warszawa.

Cowen L. 1998. *Approximation Algorithms*. Johns Hopkins University.

Feige U. 1998. *A Threshold of ln(n) for Approximating Set Cover*. Journal of the ACM.

Hansen P., Mladenovic N., Urosevic D. 2000. *Variable neighborhood search for the maximum clique*. The Fourth International Colloquium on Graphs and Optimisation (GO-IV), Lòeche-les-Bains, Suisse (20.08.2000) 2004, Vol. 145, No. 1, pp. 117–125.

Horner M., O'Kelly M.E. 2001. *Embedding economy of scale concepts for hub network design.* Journal of Transport Geography, 9 (4), pp. 255–265.

Hochbaum D.S. (Ed.) 1997. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company.

Johnson D.S. 1974. *Approximation Algorithms for Combinatorial Problems*. Journal of Computer and System Sciences, 9.

Jukna S. 2001. *Extremal Combinatorics,* Springer-Verlag, Berlin Heidelberg.

Kloks T., Kratsch D. 1998. *Listing all minimal separators of a graph.* SIAM J. COMPUT., Vol. 27, No. 3, pp. 605–613.

Korte B., Vygen J. 2000. *Combinatorial Optimization, Theory and Algorithms*. Springer.

Kułaga P., Sapiecha P., Sęp K.2005. *Approximation Algorithm for the Argument Reduction Problem*. Advances in soft computing. Computer recognition systems CORES'05, Springer-Verlag, Berlin.

Kumlander D. 2007. *An Approach for the Maximum Clique Finding Problem Test Tool Software Engineering.* Software Engineering, SE 2007, Innsbruck, Austria.

Lovasz L. 1975. *On the ratio of optimal integral and fractional covers*. Discrete Mathematics, 13.

Mażbic-Kulma B., Sęp K. 2005. *Problem wyboru węzłów tranzytowych w sieci lotniczej* (*The problem of selection of the transit nodes in an air network*; in Polish). Materiały konferencji "Systemy logistyczne. Teoria i praktyka", Warszawa, Politechnika Warszawska, pp. 341–348.

Mażbic-Kulma B., Sęp K. 2006. *Baza wierzchołkowa w hipergrafie jako metoda przydzielania zadań* (*Transversal of a hypergraph as a metod of assigning tasks*; in Polish). Kacprzyk J., Budziński R., Badania operacyjne i systemowe 2006. Metody i techniki, Akademicka Oficyna Wydawnicza EXIT, pp. 283–290.

Mażbic-Kulma B., Sęp K. 2007a. *Minimalizacja liczby węzłów w sieci logistycznej* (*Minimisation of the number of nodes in a logistic network*; in Polish). [in:] Bukowski L.A., Feliks J., Karkula M., Lichota A. (Eds), Rocznik 2007, Seria: Wybrane Zagadnienia Logistyki Stosowanej, Komitet Transportu PAN, Kraków, pp. 59–65.

Mażbic-Kulma B., Sęp K. 2007b. *Some approximation algorithms for minimum vertex cover in a hypergraph*. [in:] Kurzyński M., Puchała E., Woźniak M., Żołnierek A. (Eds), Computer Recognition Systems 2, Series: Advances in Soft Computing, Springer-Verlag, Berlin-Heidelberg, pp. 250–257.

Mażbic-Kulma B., Sęp K. 2007c. *Zastosowanie algorytmu SBT jako metody minimalizacji taboru w przedsiębiorstwie transportowym* (*Application of the SBT algorithm as a fleet minimisation method for a transport enterprise*; in Polish). [in:] Dąbrowa-Bajon M., Krawczyk G., Ambroziak T., Grochowski L., Lozia Z., Manerowski J. (Eds), Transport w systemach logistycznych, Seria: Prace Naukowe – Transport, Politechnika Warszawska, Warszawa, pp. 115–125.

Min H., Gou Z. 2004. *The location of hub-seaports in the global supply chain network using a co-operative competition strategy.* Integrated Supply Management, Vol. 1 No. 1, pp. 51–63.

O'Kelly M.E 1987. *A quadratic integer program for the location of interacting hub facilities.* European Journal of Operational Reasearch, 32, pp. 392–404.

O'Kelly M.E., Bryan D. 2002. *Interfacility interaction in models of hubs and spoke networks.* Journal of Regional Science, 42 (1), pp. 145–165.

Potrzebowski H., Stańczak J., Sęp K. 2006a. *Evolutionary Algorithm to Find Graph Covering Subsets Using α-Cliques.* [in:] Arabas J. (Ed.), Evolutionary Computation and Global Optimization. Prace naukowe Politechniki Warszawskiej, pp. 351–358.

Potrzebowski H., Stańczak J., Sęp K. 2006b. *Heurystyczne i ewolucyjne metody znajdowania pokrycia grafu, korzystajace z pojęcia alfa-kliki i innych ograniczeń* (*Heuristic and evolutionary methods of finding graph covering, using the notion of alpha-clique and other constraints*; in Polish.) [in:] Badania operacyjne i systemowe 2006. Metody i techniki, Akademicka Oficyna Wydawnicza EXIT, Warszawa, pp. 387–395.

Potrzebowski H., Stańczak J., Sęp K. 2007. *Separable decomposition of graph using alpha-cliques*. [in:] Kurzyński M., Puchała E., Woźniak M., Żołnierek A. (Eds), Computer Recognition Systems 2, Series: Advances in Soft Computing, Springer-Verlag, Berlin-Heidelberg, pp. 386–393.

Protasi M. 2001. *Reactive local search for the maximum clique problem.* Algoritmica Vol. 29, No. 4, pp. 610–637.

Sapiecha P., Selvaraj H., Stańczak J., Sęp K., Łuba T. 2004. *A hybrid approach to a classification*. Intelligent information processing and web mining, Springer-Verlag, Berlin.

Wilson R.J. 1996. *Introduction to graph theory.* Addison Wesley Longman.