Evolutionary Approach to Solve Hub-and-Spoke Problem Using *α–cliques*

Barbara Mażbic-Kulma[1], Henryk Potrzebowski[2], Jarosław Stańczak[3], Krzysztof Sęp[4]

[1]kulma@ibspan.waw.pl, [2]potrzeb@ibspan.waw.pl, [3]stanczak@ibspan.waw.pl, [4]sep@ibspan.waw.pl
[1,2,3,4]IBS PAN ul. Newelska 6 01-447 Warszawa

Abstract: The theory of transportation systems deals with models of phenomena connected with movement of goods and persons. The model of the transportation system should simulate a real system, but should also be a tool that enables to solve given transportation tasks. In order to describe transportation system (rail, bus or air), as a routine a connection graph would be used. Vertices of the graph can be train stations, bus stops or in case of air transport – airports. The edges of the graph show direct connections between vertices. It can be noticed that such a graph can have many vertices as well as many edges. Its direct application can be difficult and computational problems can occur while one would try to organize or optimize such a transportation system. Therefore, a method of aggregation of such a graph was introduced, using the hub-and-spoke structured graph of connections. This structure enables to concentrate and order the transport of goods/persons among vertices. To obtain the hub-and-spoke structure an evolutionary algorithm (EA) was applied. EA divides the connection graph into *α-cliques* (a generalization of the notion of a clique, which groups into sub-graphs highly connected vertices) and then in each *α-clique* a vertex with a maximum degree in this sub-graph and a maximal number of connections among other selected hubs is chosen. The *α-clique* with chosen vertex constitutes a "hub" with point-to-point connections - "spokes". This method enables reducing the number of analyzed vertices as well as arcs of the graph. Examples visualizing functioning of the described algorithms are presented later in this paper.

1. Introduction

Transportation system is defined by three essential components [1, 5, 6, 8, 9]:

- work task – necessity to relocate objects (cargo or (and) persons),
- composition– type and number of elements describing the equipment and crew systems,
- organization– methods of system's elements reaction during task realization.

Freight system is a transport system for the relocation of objects. The tasks of the freight system are determined by the system customers' needs and they are described by the following components:

- type and number objects to freight
- route for relocation of objects

• time of delivery

There are many transport systems classifications, based on, for instance :

• kind of transported objects (cargo, persons),
• quantity of transported objects ,
• route of transported objects,
• transport means (railway, aircraft, vessel...).

The theory of transportation systems does not directly investigate physical phenomena connected with this domain, but its aim is to model and test models of transport systems. The model of transportation system should be accurate enough to replace the real system during the process of a solving particular problem. Mathematical description of transportation system is usually based on the notion of connection graph. The vertices of this graph are railway stations, bus stations or airports, depending on the means of transport considered. Edges of this graph determine the presence of connections among vertices. As it can be easily noticed, a connection graph may have a big number of vertices and/or a big number of edges and the form of this graph has a big influence on transport organization. Thus, this article is devoted to the task of optimization of air transportation using the hub-and-spoke structured graph of air connections.

The hub-and-spoke structure of a connection graph enables to concentrate flows of transported persons/goods among vertices. Fig. 1 presents initial structure of air connections among main cities before concentration. There were many point-to-point connections. The flights were rather infrequent and expensive. There were difficulties with flight synchronization on the interchange airports, mainly due to fact that connections were serviced by more than one airway. An adequate choice of several transit nodes and local connections could improve the transport system, reducing carrier costs and increasing service efficiency. After the concentration process (Fig. 2), the graph of connections turned into a hub-and-spoke structure and each airway controlled one hub airport with several short point-to-point connections (spokes). The graph presented below in Fig. 1 may represent a structure of an existing traffic system, where the set of vertices corresponds to the set of traffic nodes and the set of edges correspond to set the of traffic connections. The hub-and-spoke structure presented in Fig. 2 reduces the complexity of the management problem.

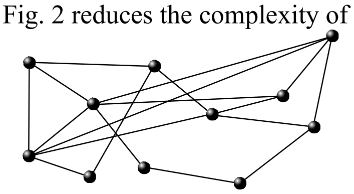

Figure 1. Input structure [3, 7]

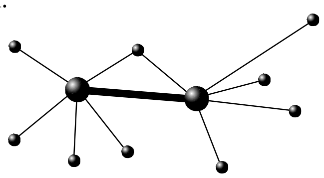

Figure 2. Relevant "hub-and-spoke" structure [3, 7]

The set of hubs should constitute a fully connected sub-graph (a clique).

The advantages of such transport structure are following:

• more frequent connections among points
• lower average times of journeys
• lower costs of transport
• lower number of required transport means to assess all connections

2. Basic Concepts

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 $v_1 \neq v_2$ [15].

Two vertices in graph $G = (V, E)$ are called ***incident*** if for v_i, $v_j \in V$ there is v_i, $v_j \in E$.

A ***sub-graph*** of 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 $e \in E'$ [15].

A ***path*** in graph $G = (V, E)$ from vertex s to vertex t is a sequence of vertices $\{v_1, \ldots , v_n\}$ such that $\{s, v_1\} \in E$, $\{v_i, v_{i+1}\} \in E$ for n=1, 2, ..., n-1, $\{v_n, t\} \in E$ [15].

A ***degree*** of vertex is the number of edges to which this vertex belongs [15].

Graph $G = (V, E)$ is a ***connected graph***, if for each pair of vertices there is a path between them [15].

Graph $G = (V, E)$ is a ***fully connected graph***, if for each pair of vertices there is an edge $e \in E$ between them [15].

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 $v_1, v_2 \in V_q$ fulfills the condition $\{v_1, v_2\} \in E_q$.[4].

A *maximum clique* $Q_M =(V_q, E_q)$ in 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 fulfills the condition $\{v_1, v_2\} \in E'$ [4, 5, 12].

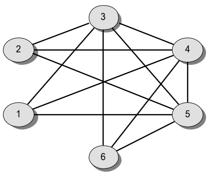

Figure 3. An example of a graph

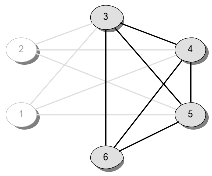

Figure 4. The maximum clique in the graph from Fig. 3

An ***α-clique***

Let $A=(V', E')$ be a sub-graph of graph $G = (V, E)$, $V' \subseteq V$, $E' \subseteq E$, $k=Card(V')$, k_i be a number of vertices $v_j \in V'$ that $\{v_i, v_j\} \in E'$ [10]

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 all vertices $v_i \in V'$ fulfill the condition $\alpha \leqslant \frac{k_i+1}{k}$, where $\alpha \in (0, 1]$.

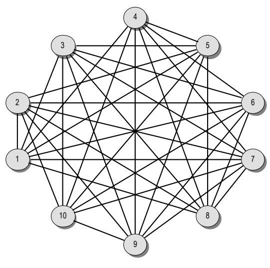

Figure 5. An example of *α-clique* for $\alpha = 0.8$

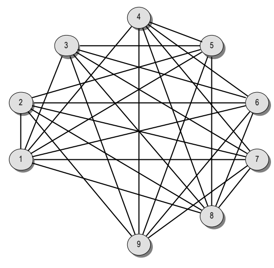

Figure 6. A sub-graph of graph from Fig. 5 which is not *α-clique* for $\alpha = 0.8$

As it can be seen in Figs. 5 and 6, a sub-graph of an α-*clique* (α=0.8) is not an α-*clique* (α =0.8), thus the property of being α-*clique* with some value of parameter α may not be preserved by the sub-graphs of an α-*clique.*

Let α-*clique* A=(V', E') be a graph with $\alpha > \frac{1}{2}$, thus, for all of vertices v_i belong to α-*clique* $k_i + 1 > \frac{1}{2} k$.

From the set theory, 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.

If $\alpha = \frac{1}{2}$, the obtained sub-graph may be disconnected - an example of such a situation is shown in Fig. 7.

Figure 7. An example of a not connected graph for $\alpha = \frac{1}{2}$

A *Hub-and-spoke* structure is a graph H_s=($G_h \cup G_s$, E) where the subset G_h determines a fully connected graph with the relevant subset of set E and each vertex of subset G_s has degree 1 and connects with exactly one vertex from subset G_h .

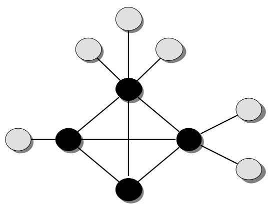

Figure 8. An example of hub-and-spoke structure

The hub-and-spoke structure defined above can be applied in many domains. Transportation systems, using this structure give a higher concentration of traffic flow among nodes (graph vertices) and synchronization of connections, thus increasing the coefficient of efficiency of services delivered by individual carriers. There is yet another application of the hub-and-spoke structure, in the so-called business (industry) clusters. These are geographic concentration of interconnected businesses, suppliers and associated institutions in a particular field. Clusters are introduced to increase the productivity and competitiveness of such institutions. Cluster concept was introduced by Michael Porter in The Competitive Advantage of Nations (1990), but the presented report deals only with the first domain of the hub-and-spoke structure application. To obtain hub-and-spoke topology from undirected graph we propose the following approach (Algorithm 1):

Let $G(V, E)$ be a considered graph.

1.	Find even independent cover by *α-cliques* $A_1(V_1, E_1), \ldots, A_m(V_m, E_m)$ fulfilling conditions: $V_1 \cup \ldots \cup V_m = V$ and $V_1 \cap \ldots \cap V_m = \varnothing$, where m is the number of *α-cliques*
2.	In each *α-clique* A_i choose exactly one vertex χ_i with maximum degree in graph $G'(V \backslash V_i \cup \{\chi_i\}, E \backslash E_i)$ - each χ_i is a new hub
3.	Connect every hub with all vertices from *α-clique* which this hub belongs to
4.	Connect every hub to each other

Algorithm 1. The method of obtaining the hub-and-spoke structure of a connection graph.

In this study a specialized evolutionary algorithm is used to obtain both *α-cliques* [11, 12] and a set of hubs, selected from *α-cliques* in the manner that maximizes the number of connections among the hub and its *α-clique* and the number of connections among hubs.

3. The Evolutionary Method to Find *α-cliques* and Hubs

Standard evolutionary algorithm (EA) works in the manner shown in Algorithm 2, but this simple scheme requires many problem specific improvements to work efficiently. The adjustment of the genetic algorithm to the solved problem requires a proper encoding of solutions, an invention of specialized genetic operators for the problem, an accepted data structure and a fitness function to be optimized by the algorithm.

1.	Random initialization of the population of solutions.
2.	Reproduction and modification of solutions using genetic operators.
3.	Valuation of the obtained solutions.
4.	Selection of individuals for the next generation.
5.	If a stop condition is not satisfied, go to 2.

Algorithm 2. The standard evolutionary algorithm.

3.1 Individual Representation

Whole information about the problem is stored in an array of data that describes all data connections. This array can be binary (a matrix of incidence of undirected graph: 0 – no connection, 1– presence of connection) or non-negative (undirected graph) real-valued and in this case the stored value denotes the strength of the connection.

Members of the population (Fig. 9) contain their own solutions of the problem as a dynamic table of derived *α-cliques* (their number may change during computations). Each element of this table (*α-clique*) has a list of nodes attached to this *α-clique* and an element chosen as a hub for this *α-clique.* Each node is considered only once in one solution (population member), thus *α-cliques* are separate. Besides a member of the population contains several more data items including: a vector of real numbers, which describe its knowledge about genetic operators and the number of the operator chosen to modify the solution in the current iteration. More details abut genetic operators and the method of their evaluation will be given later in this paper.

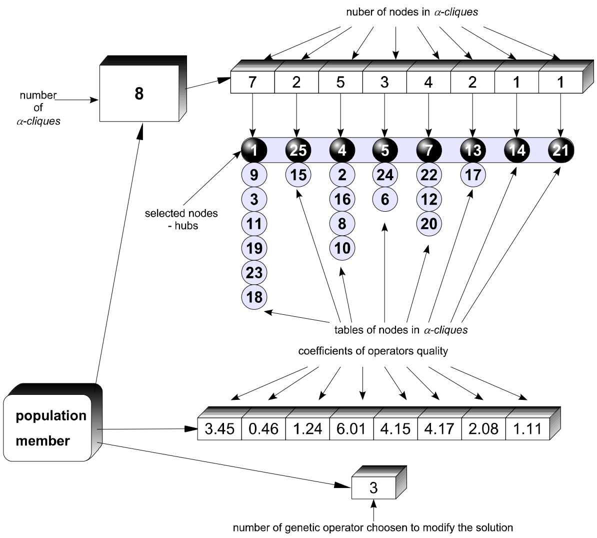

Figure 9. Structure of the population member.

3.2 Fitness Function

The problem specific quality function is closely connected with the fitness function, which evaluates the members of the population. In the problem solved several quality functions may be considered, depending on input data (binary, integer or real) or what set of *α-cliques* one wants to obtain (equal size or maximal size etc.). The fitness function does not have to possess any punishment part for *α-clique* constraint violation, because forbidden solutions are not produced during the process of population initialization or by the genetic operators. Thus all population members contain only valid *α-cliques* with their local values of α not less than the global value imposed on the problem solved. For computer simulations we used the following fitness function:

$$\max Q=\frac{1}{n}\sum_{i=1}^{n}\left(k_i-\left|\frac{k}{n}-k_i\right|+\frac{l_i}{k_i}-1+\frac{h_i}{n}-n\right) \tag{1}$$

where:

n – number of *α-cliques* in the solution evaluated, k_i – number of nodes in the i^{th} *α-clique*, k – number of nodes in the whole graph, l_i – number of connections between the hub from i^{th} *α-clique* and other nodes in this *α-clique*, h_i – number of connections between hub i and other hubs.

The fitness function (1) promotes *α-cliques* of size almost equal to average number of nodes in *α -cliques*, minimizing the number of obtained *α -cliques*, maximizing the number of connections among hubs and the number of connections between each selected hub and its *α-clique*.

3.3 Specialized Operators

The described data structure requires specialized genetic operators, which modify the population of solutions. Each operator is designed in a manner preserving the property of being *α-clique* for the modified parts of solutions. If the modified solution violates the

limitation of being α-*clique*, the operation is cancelled and no modification of solution is performed. With this method it is more difficult for the evolutionary algorithm to find satisfying solutions, due to the possible bigger problems with local maximums than method with penalty function, but it gives the certainty that computed solutions are always feasible.
The designed genetic operators are:

1. mutation – an exchange of randomly chosen nodes in different α-*cliques*;
2. movement of a randomly chosen node to a different α-*clique*;
3. "intelligent" movement – performed only if this modification gives a better value of the fitness function;
4. concatenation – attempt to concatenate (mainly small) α-*cliques*;
5. also multiple versions of operators are applied.

Additionally, each operator modifies elements selected as hubs for all α-*cliques*, using a simple mutation method.

3.4 Evolutionary Algorithm Used to Solve the Problem

Use of specialized genetic operators requires having a method of selecting and executing them in all iterations of the algorithm. In the approach used [14] it is assumed that an operator that generates good results should have bigger probability and more frequently effect the population. But it is very likely that the operator, that is proper for one individual, gives worse effects for another, for instance because of its location in the domain of possible solutions. Thus, every individual may have its own preferences. Every individual has a vector of floating point numbers, besides the encoded solution. Each number corresponds to one genetic operation. It is a measure of quality of the genetic operator (a quality factor). The higher the factor, the higher the probability of using the operator. The ranking of quality factors becomes a basis for computing the probabilities of appearance and execution of genetic operators. Simple normalization of the vector of quality coefficients turns it into a vector of operator execution probabilities. This set of probabilities is also a basis of experience of every individual and according to it, an operator is chosen in each epoch of the algorithm. Due to the experience gathered one can maximize chances of its offspring to survive.

The method of computing quality factors is based on reinforcement learning [2] (one of algorithms used in machine learning). An individual is treated as an agent, whose role is to select and call one of the evolutionary operators. When the selected i^{th} operator is applied, it can be regarded as an agent action a_i leading to a new state s_i, which, in this case, is a new solution. Agent (genetic operator) receives reward or penalty depending on the quality of the new state (solution). The aim of the agent is to perform the actions, which give the highest long term discounted cumulative reward V^*:

$$V^{\pi}=E_{\pi}\sum_{k=0}^{\infty}\gamma^{k}r_{t+k+1} \tag{2}$$

$$V^{*}=\max_{\pi}(V^{\pi}) \tag{3}$$

The following formula can be derived from (2) and (3) and is used for evaluation purposes:

$$V(s_{t+1})=V(s_t)+\alpha r_{t+1}+\gamma V^{*}(s_{t+1})-V(s_t) \tag{4}$$

where:

Π - represents the strategy of the agent, V^{π} - represents discounted cumulative reward obtained using strategy Π, E - represents expected value, k - represents consecutive time steps, t - represents current time, $V(s_t)$ - is a quality factor or discounted cumulative reward, $V^{*}(s_{t+1})$

- estimated value of the best quality factor (in our experiments we take the value attained by the best operator), α - is a learning factor, γ - is a discount factor, r_{t+1} - represents the reward for the best action, which is equal to the improvement of the quality of a solution after execution of the evolutionary operator.

In the presented experiments the values of α and γ were set to 0.1 and 0.2 respectively

4. Obtained Results

Unfortunately, we did not have a real traffic data, thus we used as a testing example from BHOSLIB: Benchmarks with Hidden Optimum Solutions for Graph Problems (Maximum Clique, Maximum Independent Set, Minimum Vertex Cover and Vertex Coloring) – Hiding Exact Solutions in Random Graphs [16]. The chosen problem was a graph with 450 vertices and 83 198 edges with the maximum clique size number equal 30 (frb30-15-clq.tar.gz). The size of the problem is relatively big, but it is similar its complexity to problems encountered during planning of connections among bigger European cities.

The first step to obtain the hub-and-spoke structure depends on the accepted value of parameter α. This step was made using the EA method and the results were as follows:

Table 1. Comparation of obtained results

α		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0.75		3	150				138	139						2						
	num		3				2	1						3						
0.80		9	50				41	42	43	44	45	46	47	8						
	num		9				1	1	1	1	2	2	1	9						
0.85		15	24	29	30	34	22	25	26	27	28	31	32	13	14					
	num		1	2	10	2	1	2	2	7	1	1	1	8	7					
0.90		21	21	22			19	20	21					17	18	19	20			
	num		12	9			11	8	2					2	5	9	4			
0.95		29	14	15			15	14						23	24	25	26	27		
	num		15	16			15	14						3	4	4	11	7		
0.99		29	14	15			15	14						22	23	24	25	26	27	28
	num		15	16			15	14						1	1	5	4	10	5	3
1.00		29	14	15			15	14						24	25	26	27	28		
	num		15	16			15	14						5	12	4	6	2		

where:

column 1 number of α-*cliques*

column 2 - 5 cardinality of particular α-*cliques* and number (num) of α-*cliques* with this cardinality

column 6 - 12 degree of hubs in their α-*cliques* and number (num) of hubs with this degree

column 13 - 19 number of hubs connected with a particular hub and number (num) of such hubs

Using different values of α we obtained several different solutions. It is difficult to foresee a priori, taking in account only the value of α, which would be the best, but after conducted computations it is quite easy to choose the solution with the most acceptable parameters. The most important factor that influences the decision is the obtained number of hubs. Probably for the problem of connecting bigger European cities the best solutions are the ones for α=0.8 or α=0.85 with 9 or 15 hubs. Of course we can also try to find different values of hub number by accepting the values of α between the tested values. The process of genetic computations lasts about 5-10 minutes, depending on the used operating system, machine and value of α, thus it is possible to compute solutions using several values of α, before accepting one.

Presented results provide a distinctly simplified structure of connections, for example in the case of α=0.80, we obtained output structure with 486 connections (edges) in place of the input structure with 83 198 connections. The lower number of connections implies faster and more frequent connections with lower costs for carriers and their clients.

5. Conclusions

It is well known for problems with large-scale complexity that there are no effective algorithms to solve them, but the possibility of constructing the approximate ones is still open. Generally, the problem of finding the hub-and-spoke structure finding considered in the paper can probably be solved using some greedy-type method, but at the moment we developed the evolutionary method that efficiently solves that problem.

The results of the series of conducted experiments are rather optimistic, the parameter α introduced into the traditional notion of a clique gives rise a flexible tool that enables solving of the hub-and-spoke structure finding problem.

Bibliography

[1] Ambroziak T. (2000) *O pewnych aspektach modelowania systemów transportowych,* (eng.: *Some aspects of transportation system modelling*) Prace Naukowe Transport z.44 OW PW Warszawa.

[2] Cichosz P. (2000) *Systemy uczące się* (in Polish), (eng.:*Learning systems*) WNT, Warszawa.

[3] Coyle J.J., Bardi E.J, Novack R.A. (1994) *Transportation*, Fourth Edition, New York: West Publishing Company, pp. 402.

[4] Hansen P., Mladenovic N., Urosevic D.(2004) *Variable neighborhood search for the maximum clique,* The Fourth International Colloquium on Graphs and Optimization (GO-IV)GO-IV International Colloquium on Graphs and Optimization No4, Lòeche-les-Bains , SUISSE (20/08/2000) 2004, vol. 145, no 1 (28 ref.), pp. 117-125.

[5] Jacyna M. (2001) *Modelowanie wielokryterialne w zastosowaniu do oceny systemów transportowych,* (eng.: *Multi-criteria modelling applied to transportation systems valuation*) Prace Naukowe Transport, z.47 OW PW, Warszawa.

[6] O'Kelly M.E (1987) *A quadratic integer program for the location of interacting hub facilities,* European Journal of Operational Research 32, pp. 392-404.

[7] Kulma-Mażbic B., Sęp K. (2005) *Problem wyboru węzłów tranzytowych w sieci lotniczej,* (eng.: *The problem of selection transit nodes in an airline network*) Logistic Systems Theory And Practice OW PW, pp. 341-348.

[8] Leszczyński J. (1994) *Modelowanie systemów i procesów transportowych,* (eng.: *Modelling of transportation precesses and systems*) OW PW, Warszawa.

[9] Piasecki S. (1973) *Optymalizacja systemów przewozowych,* (eng.: *Optimization of freight systems*) WKiŁ Warszawa.

[10] 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, in: Advances in soft computing Springer-Verlag, Berlin-Heidelberg, pp. 386-393.

[11] Potrzebowski H., Stańczak J., Sęp K. (2006) *Evolutionary Algorithm to Find Graph Covering Subsets Using α-Cliques,* Evolutionary Computation and Global Optimization, Prace naukowe PW, Warszawa, pp. 351-358.

[12] Potrzebowski H., Stańczak J., Sęp K. (2006) *Heurystyczne i ewolucyjne metody znajdowania pokrycia grafu, korzystajace z pojęcia alfa-kliki i innych ograniczeń* (eng.: *Heuristic and evolutionary methods for solving graph vertex cover problem using alpha-cliques and other constraints*) (in Polish), Badania operacyjne i systemowe 2006. Metody i techniki, Akademicka Oficyna Wydawnicza EXIT, Warszawa.

[13] Protasi M. (2001) *Reactive local search for the maximum clique problem,* Algoritmica vol. 29, no 4. pp. 610-637.

[14] Stańczak J. (2003) *Biologically inspired methods for control of evolutionary algorithms*, Control and Cybernetics, 32(2), pp. 411-433.

[15] Wilson R.J. (1996) *Introduction to graph theory* Addison Wesley Longman.

[16] http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm