PERGAMON

Computers & Operations Research 27 (2000) 559–569

computers & operations research

www.elsevier.com/locate/orms

Heuristics for finding k-clubs in an undirected graph

Jean-Marie Bourjolly[a,b], Gilbert Laporte[b,c,*], Gilles Pesant[b,d]

[a]*Department of Decision Sciences and MIS, Concordia University, 1455 de Maisonneuve boulevard West, Montréal, Canada H3G 1M8*

[b]*Centre de recherche sur les transports, Université de Montréal, Case postale 6128, Succursale "Centre-ville", Montréal, Canada H3C 3J7*

[c]*GERAD and École des Hautes Études Commerciales, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7*

[d]*Département de génie électrique et de génie informatique, École Polytechnique de Montréal, P.O. Box 6079, Succursale "Centre-ville", Montréal, Canada H3C 3J7*

Received 1 August 1998; received in revised form 1 February 1999

Abstract

In a graph G, a *k-club* is a vertex set inducing a subgraph of diameter k. These structures play an important role in several applications arising in social and behavioral sciences. We derive some properties of k-clubs and we propose three heuristics for determining a largest k-club in a graph. Comparative computational results confirm the speed and efficiency of these heuristics.

Scope and purpose

Social and behavioral scientists frequently use network analysis to study linkages between groups, individuals or abstract entities. Applications are encountered in psychology, marketing, organization theory, anthropology, economics, sociology, etc. A common problem is to identify dense structures in a graph. This article is about the determination of one such type of structure called *k-club*. Three efficient heuristics for the determination of k-clubs are developed and compared.

Keywords: k-clubs; k-cliques; Heuristics

1. Introduction

Social and behavioral scientists frequently use network analysis to study linkages between groups, individuals or abstract entities. Early examples are found in the work of Gestalt

* Corresponding author. Centre de recherche sur les transports, Université de Montréal, Case postale 6128, Succursale "Centre-ville", Montréal, Canada H3C 3J7. Tel.: + 1514-343-6143; fax: + 1514-343-7121.

E-mail address: gilbert@crt.umontreal.ca (G. Laporte)

0305-0548/00/$ - see front matter
PII: S0305-0548(99)00047-7

psychologists who have for long studied the structure patterns associated with thought and perception [1], or in the writings of Moreno [2] who has investigated the relationships between psychological well-being and structures referred to as social configurations. Baker [3] has used social networks to study the type of ties existing between members of a population. Snyder and Kick [4] present a network-based study of world-system/dependency theories of differential growth among nations and provide an analysis based on ten groups of nations. Social networks have been used to study community power by Breiger [5], the connectivity of social fabric by Doreian [6], patterns of prevalence of interpersonal agreements in a population [7], market-areas and market-shares [8]. Mintz and Schwartz [9] have studied data on interlocking directorates to test three theories of corporate organization, using social networks, while Roy [10] has analyzed interindustry interlocking directorates between the years 1886–1905. In an interesting article, Bougon et al. [11] apply social network theory to study cognition relationships among the players of the Utrecht jazz orchestra. Several other applications in marketing, organization theory, and sociology have appeared in the literature. For further readings on this subject, refer to the article of Everett [12] and to the books of Burt [13], Hage and Harary [14], Wellman and Berkowitz [15], Scott [16] and Wasserman and Faust [17].

Some of the main concepts used in network analysis are borrowed from graph theory. The most widely known is that of a *clique* which is defined as a set of vertices all directly linked to each other by an edge. Several contexts, however, call for the use of looser structures in which vertices are not required to be directly connected, but may be linked through a chain of at most k edges, as often happens in a network of acquaintances. In other words, what is of interest is the identification of *dense structures* in a graph, i.e., subsets of vertices with a high density of interconnections. Several definitions can be used to operationalize this concept. Social network packages such as UCINET now include routines for the identification of dense structures in a graph [18]. These are often based on the enumerative algorithm of Bron and Kerbosch [19], but full enumerations are limited to relatively small graph sizes. While several algorithms have been developed to enumerate dense structures possessing some characteristics, little effort has been put on optimization techniques aimed at determining largest dense structures. There are contexts where this is relevant. For example, in the study of interlocking corporate directorates, one may wish to identify the largest group of highly interconnected people controlling the most important American companies. Several of the real-life problems considered in the above references are too large to be solved optimally and can only be tackled by means of a heuristic. The contribution of this work is to propose efficient heuristics for the identification of a family of dense structures called k-clubs.

The remainder of this paper is organized as follows. In Section 2, we introduce the notion of k-club and identify some of its properties. In Section 3, we propose three heuristics for the identification of k-clubs in an undirected graph. This is followed by computational results in Section 4 and by the conclusion in Section 5.

2. Definitions and properties

The problems studied in this article are defined on an undirected graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the vertex set and the edge set E is a subset of $\{(i, j)\colon i, j \in V, i < j\}$. Given a subset W of V, the subgraph H of G induced by W is defined as $H = (W, F)$, where

$F = \{(i, j) \in E: i, j \in W\}$. A *chain of length p* is a sequence of p different edges in which every intermediate edge shares a vertex with its predecessor and the other vertex with its successor. A *cycle* is a chain whose extremities coincide. The *diameter* of a subgraph H of G is the length of the longest chain among all shortest chains in H between any two of its vertices. Our purpose is to identify large vertex sets inducing subgraphs of diameter at most k in G, called *k-clubs*. This concept is related to that of a k-clique. A *k-clique* is a subset of V in which every vertex pair is connected by a chain of G of length at most k. Note that a k-clique may not be fully interconnected, as is required in some applications, in the following sense: if C_k is a k-clique, all the chains of length at most k between two vertices of C_k may contain some vertices not included in C_k. This has led some researchers to define "self-contained" structures, such as k-clubs. Therefore, every pair of vertices of the k-club D_k is linked by a chain of length at most k, totally included in D_k. In keeping with modern usage, (see, e.g., Johnson and Trick [20]), our definitions depart from those used by certain authors in social sciences (e.g., Alba [21,22] and Mokken [23]) in that we do not require that k-cliques and k-clubs be maximal with respect to inclusion. We thus make a distinction between a k-clique and a maximal k-clique, and between a k-club and a maximal k-club. While all k-clubs are k-cliques, the converse is not true, as shown in the graph depicted in Fig. 1, taken from Alba [21] and Scott [16]. The set $S = \{1, 2, 3, 4, 5\}$ is a 2-clique but not a 2-club since the only two-edge chain connecting vertices 4 and 5 passes through vertex 6 which does not belong to S. The set $S' = \{1, 2, 3, 4, 5, 6\}$ is both a 3-clique and a 3-club.

In the case of a k-clique, the notion of maximality is easily checked, but this is not the case for k-clubs. In the example depicted in Fig. 2 and adapted from Mokken [23], the vertex set $S = \{1, 2, 3, 4\}$ is a 2-club. It is not maximal since $\{1, 2, 3, 4, 5, 6\}$ is also a 2-club. However, $\{1, 2, 3, 4, 5\}$, $\{1, 2, 3, 4, 6\}$, and $\{1, 2, 3, 4, 7\}$ are not 2-clubs.

We will refer to the k-clique (resp. k-club) problems as the problems consisting of determining a largest k-clique (resp. k-club) in an undirected graph. It is common to omit k when it is equal to 1. It is well known that the clique problem is NP-hard [24]. Exact algorithms and heuristics for this problem are described or quoted in Johnson and Trick [20].

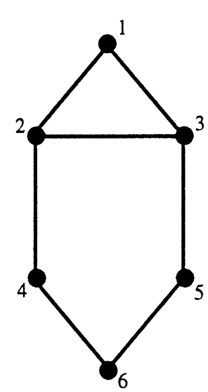

Fig. 1. Example showing a 2-clique that is not a 2-club.

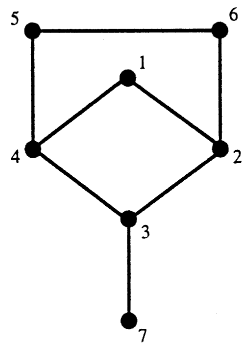

Fig. 2. Example illustrating a maximality check.

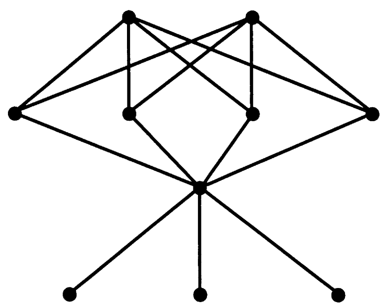

Fig. 3. A $K_{2,4,1,3}$ graph.

The k-clique problem is easily transformed into a clique problem on $G^k = (V, E^k)$, where $(i, j) \in E^k$ $(i < j)$ if and only if there exists a chain of length at most k between i and j in G (see, e.g., Scott [16]). The k-club problem is NP-hard as can be shown by polynomially transforming an instance of the clique problem into a k-club instance. The following properties of the k-club problem can be stated without proof. They will be used to develop heuristics or to compute upper bounds on the optimal value z_k^*, the size of the largest k-club.

Property 1. *$z_{k'}^* \geqslant z_k^*$ whenever $k' \geqslant k$.*

Property 2. *Let u_k^* be the value of an optimal solution to the k-clique problem, then*

$$u_1^* \leqslant z_k^* \leqslant u_k^*.$$

Property 3. *If G contains a cycle of length $\ell \leqslant 2k + 1$, then $z_k^* \geqslant \ell$.*

Now denote by $K_{r_1, \ldots, r_k}$ a k-layered subgraph of G in which r_i is the number of vertices in layer i, and all pairs of vertices from two successive layers are linked by edges (see Fig. 3).

Property 4. *If $k \geqslant 2$ and $K_{r_1,\ldots,r_{k+1}}$ is a subgraph of G, then*

$$z_k^* \geqslant \sum_{h=1}^{k+1} r_h.$$

Property 5. *If $K_{r,s}$ is a subgraph of G, then $z_k^* \geqslant r + s$ for $k \geqslant 2$.*

The star graph $K_{r,1}$ will play a special role in one of our heuristics.

3. Heuristics

We describe three heuristics for the k-club problem.

Heuristic 1: CONSTELLATION. This heuristic stems from Property 5 which states that the $K_{r,1}$ star graph provides a 2-club. This idea is generalized to find k-clubs ($k \geqslant 3$). Step 1 identifies a t-club for $t = 2$, and each application of Step 2 increases t by 1 until k is reached.

Step 1 (*First star*): Set $t := 2$. Obtain a first 2-club by determining the vertex of maximum degree in G and by using it as the center of a star graph of maximum size. The 2-club is the vertex set W of this star graph.

Step 2 (*Next star*): If $t = k$ or $W = V$, stop. Otherwise, set $t := t + 1$. Determine the vertex of W having the largest set S of neighbour vertices in $V \backslash W$. Set $W := W \cup S$. Repeat this step.

The complexity of CONSTELLATION is determined as follows. Assume an adjacency list representation for G which requires $\theta(|E|)$ space, and a bit-vector representation of W which requires $\theta(|V|)$ space. Step 1 can be executed in $O(|V| + |E|)$ time and each of the $k - 2$ applications of Step 2 requires $O(|V| + |E|)$ time. Therefore, the overall complexity is $O(k(|V| + |E|))$.

To illustrate, consider the subgraph of $G = (V, E)$ shown in Fig. 4. The initial set W determined in Step 1 consists of vertex 1 and of its six neighbours. In Step 2, the procedure will select vertex 2 having four neighbours in $V \backslash W$, as opposed to vertex 3 which has only three neighbours in $V \backslash W$, even though it has more neighbours in V than does vertex 2.

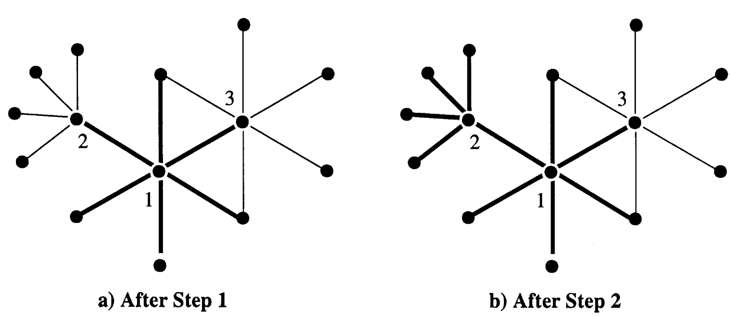

Fig. 4. Construction of the first two stars in CONSTELLATION.

The following two heuristics make use of an auxiliary procedure for checking whether the vertex set of a graph is a k-club. It does so by efficiently maintaining shortest chain lengths between vertex pairs as the graph is modified, i.e., by solving a dynamic version of the all-pairs shortest path problem. Clearly, if no chain length exceeds k, then a k-club has been identified. Both heuristics successively remove vertices from V until it becomes a k-club. In Heuristic 2, V is initially taken as the vertex set of the whole graph G and subsequently updated. In Heuristic 3, the search concentrates on an initial largest k-clique, a good candidate to contain a largest k-club.

Heuristic 2. (DROP):

Step 1: (*Data structure initialization*): Compute shortest chain lengths between all vertex pairs.

Step 2: (*Termination check*): Compute for each vertex i of V the number q_i of vertices of V whose shortest chain to i has length at least $k+1$. If $q_i = 0$ for every vertex i, stop: V is a k-club.

Step 3: (*Vertex removal*): Let W be the set of vertices for which q_i is maximized. Determine a vertex $i^* \in W$ of least degree in V. Remove i^* and its incident edges from the graph.

Step 4: (*Data structure update*): Update shortest chain lengths. (Two vertices belonging to different connected components are said to be linked by an infinite length chain.) Go to Step 2.

The dynamic algorithm which we implemented for the all-pairs shortest path problem has an asymptotic worst-case time complexity of $O(|V|^3)$ per edge deletion, which may not be better than that of some static algorithms but which has a definite impact on actual running times since it does not start from scratch every time. Other dynamic algorithms exist which have a better time complexity, but they tend to be harder to implement. (For example, Ronert [25] describes a dynamic algorithm to solve the all-pairs shortest path problem $O(|V||E| + |V|^2\log|V|)$ time per edge deletion.) Step 1 of DROP is executed in $O(|V|^2|E|)$ time and $O(|V|^2)$ space. The complexity of Steps 2 and 3 is dominated by that of Step 4 which requires $O(|V|^3)$ time per edge deletion. Since at most $|E|$ edges are deleted, the overall time complexity of DROP is in $O(|V|^3|E|)$.

To illustrate, consider applying DROP to the graph of Fig. 5. For $k = 2$, Step 2 computes $q_1 = q_2 = 1$, $q_3 = q_4 = q_5 = 0$, and $q_6 = 2$. Therefore, $W = \{6\}$ in Step 3 and vertex 6 is removed from the graph. Applying Step 2 again yields a 2-club since $q_i = 0$ for all i.

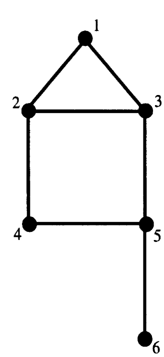

Fig. 5. Illustration of heuristic DROP.

Heuristic 3: k-CLIQUE & DROP

Step 1 (k-*clique*): Determine a largest k-clique in G as explained in Section 2. Remove from the graph all vertices not belonging to the k-clique and all their incident edges. Let V be the resulting vertex set.

Step 2 (DROP): Call DROP.

The complexity of this algorithm is dominated by Step 1 which is potentially exponential.

A fourth heuristic valid for $k = 2$, called ADD, was also developed and tested. It consists of first identifying a largest star in G and gradually adding cliques that are connected to it. Unfortunately, it was consistently dominated by one or the other of the three heuristics just described and was therefore abandoned.

4. Computational results

The three heuristics were coded in C. They were run on a Sun Sparc 10 (55 MHz), and tested on $\hat{p}$-generated graphs obtained using the algorithm of Gendreau et al. [26], and controlled by two density parameters a and b ($0 \leqslant a \leqslant b \leqslant 1$). It generalizes the classical uniform random graph generator. Its description is as follows:

begin
    for $i := 1$ to n **do** $\hat{p}[i] := uniform(a, b)$;
    for $i := 1$ to $n - 1$ **do**
        for $j := i + 1$ to n **do**
        generate edge (i, j) with probability $(\hat{p}[i] + \hat{p}[j])/2$;
end

The expected edge density of such graphs is equal to $(a + b)/2$, and the vertex degree variance increases with $b - a$. To help assess the quality of the heuristics, we have computed for each graph an upper bound u_k^* on the size z_k^* of a largest k-club. As per Property 2, we use for u_k^* the value of the size of a largest k-clique, determined by using an exact algorithm for the clique problem. All our tests were performed on 100-vertex graphs. We have generated ten different instances for $k = 2$, 3, 4 and several combinations of a and b. We only considered values of a and b which yielded meaningful instances: when the graph density becomes too high, all vertices belong to a maximum k-club and the corresponding problem is therefore trivial. Similarly, values of k larger than 4 tend to produce uninteresting instances for the graph size considered. Tables 1 and 2 contain average computational results over the 10 instances of each category. Results corresponding to the best heuristic are given in bold characters.

Our results indicate that in terms of solution quality, DROP is the best of the three heuristics for higher densities. When the density becomes very high, this heuristic consistently identifies optimal or near-optimal solutions. For lower densities, the best heuristic is CONSTELLATION when $k = 2$, and k-CLIQUE & DROP when $k = 3$ or 4. The good performance of CONSTELLATION on low density graphs when $k = 2$ is not surprising since the heuristic is initialized with the largest star subgraph which corresponds to a 2-club. The idea of starting the DROP process from

Table 1
Size of k-clubs for three different heuristics

k	Density	$[a, b]$	Largest k-clique	k-club heuristics		
				CONSTELLATION	DROP	k-CLIQUE & DROP
2	0.05	[0.05, 0.05]	12.7	**12.7**	8.3	**12.7**
		[0.00, 0.10]	12.4	**12.3**	8.3	**12.3**
	0.1	[0.10, 0.10]	20.8	**19.2**	10.8	13.9
		[0.05, 0.15]	21.7	**19.8**	11.8	15.6
		[0.00, 0.20]	24.0	**21.4**	14.4	14.9
	0.15	[0.15, 0.15]	44.2	**25.3**	20.5	18.6
		[0.10, 0.20]	44.6	**25.5**	23.5	17.9
		[0.05, 0.25]	46.0	**26.9**	26.7	22.9
	0.2	[0.20, 0.20]	75.0	31.7	**54.3**	47.1
		[0.15, 0.25]	75.7	31.8	**62.5**	54.5
		[0.10, 0.30]	74.4	33.3	**63.9**	58.6
	0.25	[0.25, 0.25]	94.8	36.9	**94.2**	93.4
		[0.20, 0.30]	94.5	38.0	**93.5**	92.9
		[0.15, 0.35]	93.6	39.3	**92.8**	92.0
	0.3	[0.30, 0.30]	99.5	42.8	**99.4**	**99.4**
		[0.25, 0.35]	99.7	44.4	**99.7**	99.6
		[0.20, 0.40]	99.3	45.4	**99.3**	**99.3**
3	0.015	[0.015, 0.015]	9.5	9.1	**9.5**	**9.5**
	0.02	[0.020, 0.020]	11.3	10.7	**11.2**	**11.2**
	0.025	[0.025, 0.025]	13.4	12.3	12.7	**13.2**
		[0.000, 0.050]	14.9	12.8	12.7	**14.5**
	0.05	[0.050, 0.050]	33.2	20.0	**23.2**	22.2
		[0.025, 0.075]	33.1	19.7	**23.9**	22.4
		[0.000, 0.100]	35.7	20.1	**23.8**	22.6
	0.075	[0.075, 0.075]	72.2	26.4	**58.8**	56.4
		[0.050, 0.100]	70.6	25.9	**55.4**	54.0
		[0.025, 0.125]	69.8	27.4	**58.1**	57.1
	0.1	[0.100, 0.100]	95.0	32.6	**94.0**	93.7
		[0.075, 0.125]	95.3	31.5	**95.2**	94.8
		[0.050, 0.150]	93.8	32.7	**93.5**	93.1
	0.125	[0.125, 0.125]	99.6	36.7	**99.6**	**99.6**
		[0.100, 0.150]	99.5	36.3	**99.5**	**99.5**
		[0.075, 0.175]	99.6	37.6	**99.6**	**99.6**
	0.15	[0.150, 0.150]	100.0	42.3	**100.0**	**100.0**
		[0.125, 0.175]	100.0	41.7	**100.0**	**100.0**
		[0.100, 0.200]	99.9	42.2	**99.9**	**99.9**
4	0.015	[0.015, 0.015]	14.5	11.4	**14.4**	**14.4**
	0.02	[0.020, 0.020]	19.0	13.2	17.5	**18.3**
	0.025	[0.025, 0.025]	25.1	16.6	21.5	**24.0**
		[0.000, 0.050]	27.5	16.9	24.3	**25.5**

Continued opposite

Table 1 (*continued*)

k	Density	$[a, b]$	Largest k-clique	k-club heuristics		
				CONSTELLATION	DROP	k-CLIQUE & DROP
	0.05	[0.050, 0.050]	79.5	27.0	**74.8**	74.4
		[0.025, 0.075]	77.2	26.5	**74.1**	72.2
		[0.000, 0.100]	76.9	27.1	**73.8**	73.2
	0.075	[0.075, 0.075]	99.0	36.0	**99.0**	**99.0**
		[0.050, 0.100]	99.2	35.0	**99.2**	**99.2**
		[0.025, 0.125]	99.3	37.5	**99.3**	**99.3**

Table 2
CPU time in seconds on a Sun Sparc 10 (55 MHz)

k	Density	Largest k-clique	k-club heuristics		
			CONSTELLATION	DROP	k-CLIQUE & DROP
2	0.05	0.93	0.00	6.82	4.68
	0.1	2.61	0.00	6.52	6.14
	0.15	81.52	0.00	6.35	85.51
	0.2	13.63	0.00	5.58	18.71
	0.25	0.17	0.00	4.54	5.70
	0.3	0.15	0.00	4.28	5.71
3	0.015	0.99	0.00	4.53	4.68
	0.02	1.09	0.00	5.50	4.94
	0.025	1.23	0.00	6.23	5.17
	0.05	2.06	0.00	6.96	6.18
	0.075	2.17	0.00	6.09	7.35
	0.1	0.48	0.00	4.44	6.12
	0.125	0.32	0.00	4.22	5.93
	0.15	0.24	0.00	4.17	5.79
4	0.015	2.36	0.00	4.52	6.10
	0.02	2.86	0.00	5.90	7.25
	0.025	3.28	0.00	6.37	7.70
	0.05	2.28	0.00	5.78	8.27
	0.075	0.91	0.00	4.42	6.96

a maximum k-clique seems to work well, as expected, on low density graphs. When the density becomes high, almost all vertices belong to a k-club and the starting point of the DROP heuristic then becomes less critical. The k-club produced by the best of our three heuristics typically has a size close to that of the upper bound given by the largest k-clique size. When there are large discrepancies (see, e.g., $k = 2$ and density $= 0.15$ or 0.2), it is not clear whether this can be attributed to the heuristics or to the poor quality of the upper bound. In terms of computation times, CONSTELLATION is extremely quick, but the times for DROP are also relatively modest. The computation time of k-CLIQUE & DROP includes the time required to determine a largest k-clique and is therefore sometimes rather high.

5. Conclusion

We have described three heuristics for determining a maximum k-club in an undirected graph. This problem has several applications in social network analysis. Our algorithms are efficient and quick. They should therefore constitute useful tools in several areas of social sciences. Natural extensions to our work include the determination of k-clubs that contain a prespecified set of vertices, the determination of a smallest family of k-clubs covering the entire graph and the identification of several large, but not necessarily maximum, k-clubs.

Acknowledgements

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grants OGP0039359, OGP0039682, and OGP0005052. This support is gratefully acknowledged. Thanks are also due to Martin G. Everett and Ann Langley for useful discussions and for suggesting some references, and to two anonymous referees for their valuable comments.

References

[1] Köhler W. The mentality of apes. London: Routledge & Kegan Paul, 1927.
[2] Moreno JL. Who shall survive? Beacon House, New York: Foundations of Sociometry, Group Psychotherapy and Sociodrama, 1953.
[3] Baker WE. Market networks and corporate behavior. American Journal of Mathematical Sociology 1986;12:191–223.
[4] Snyder D, Kick EL. Structural position in the world system and economic growth, 1955–1970: a multiple-network analysis of transactional interactions. American Journal of Sociology 1979;84:1096–126.
[5] Breiger RL. Toward an operational theory of community elite structure. Quality and Quantity 1979;13:21–47.
[6] Doreian P. On the connectivity of social networks. Journal of Mathematical Sociology 1974;3:245–58.
[7] Friedkin NE. A formal theory of social power. Journal of Mathematical Sociology 1986;12:103–26.
[8] Berkowitz SD. Markets and market-areas: some preliminary formulations. In: Wellman B, Berkowitz SD, editors. Social structures: A network approach. Cambridge: Cambridge University Press, 1988:261–303.
[9] Mintz B, Schwartz M. Interlocking directorates and interest group formation. American Sociological Review 1981;46:851–69.
[10] Roy W. The unfolding of the interlocking directorate structure of the United States. American Sociological Review 1983;48:248–57.
[11] Bougon M, Weik K, Binkhorst D. Cognition in organizations: an analysis of the Utrecht jazz orchestra. Administrative Science Quarterly 1977;22:606–39.
[12] Everett MG. A graph theoretic blocking procedure for social networks. Social Networks 1982;4:147–67.
[13] Burt RS. Structural moles: the social structure of competition. Cambridge: Harvard University Press, 1992.
[14] Hage P, Harary F. Structural models in anthropology. Cambridge: Cambridge University Press, 1983.

[15] Wellman B, Berkowitz SD, editors. Social structures: a network approach. Cambridge: Cambridge University Press, 1988.
[16] Scott J. Social network analysis. London: Sage, 1994.
[17] Wasserman S, Faust K. Social network analysis: methods and applications. Cambridge: Cambridge University Press, 1994.
[18] UCINET, Release 3.0, Mathematical Social Science Group, Social School of Social Sciences, University of California at Irvine.
[19] Bron C, Kerbosch J. Finding all cliques of an undirected graph. Communications of the ACM 1973;16:575–77.
[20] Johnson DS, Trick MA. Cliques, coloring and satisfiability, Series in Discrete Mathematics and Theoretical Computer Science. Providence, RI: American Mathematical Society, 1996.
[21] Alba RD. A graph-theoretic definition of a sociometric clique. Journal of Mathematical Sociology 1973;3:113–26.
[22] Alba RD. Taking stock of network analysis: a decade's result. Research in the Sociology of Organizations 1982;1:39–74.
[23] Mokken RJ. Cliques, clubs and clans. Quality and Quantity 1979: 13: 161–73.
[24] Karp RM. Reducibility among combinatorial problems. In: Miller RF, Thatcher JW, editors. Complexity of computer computations. New York: Plenum Press 1972:85–103.
[25] Ronert H. A dynamization of the all pairs least cost path problem. Proceedings of STACS'85, Lecture Notes in Computer Science, vol. 182, Berlin: Springer, 1985:279–86.
[26] Gendreau M, Soriano P, Salvail L. Solving the maximum clique problem using a tabu search approach. Annals of Operations Research 1993;41:385–403.

Jean-Marie Bourjolly is an Associate Professor at Concordia University in Montreal. His research interests include combinatorial optimization and telecommunication problems.

Gilbert Laporte is a Professor at the École des Hautes Études Commerciales de Montréal and member of the Centre for Research on Transportation (CRT) and of the GERAD. His research interests include combinatorial optimization, vehicle routing, location and scheduling.

Gilles Pesant is an Assistant Professor in the Department of Electrical and Computer Engineering at the École Polytechnique de Montréal. He is also a member of the CRT. His research interests mainly focus on constraint programming and its applications in the areas of transportation, telecommunications, personnel scheduling, and geometric modeling.