On globally solving the maximum weighted clique problem *

NGUYEN CANH Nam

Department of Applied Mathematics
School of Applied Mathematics and Informatics
Hanoi University of Science and Technology
nam.nguyencanh@hust.edu.vn

Abstract

In this paper, we consider a combinatorial optimization problem, the Maximum Weighted Clique Problem (MWCP), a NP-hard problem. The considered problem is first formulated in the form of binary constrained quadratic program and then reformulated as a Difference Convex (DC) program. A global optimal solution is found by applying DC Algorithm (DCA) in combining with Branch and Bound scheme with SDP relaxation technique. Computational experiments are reported for problem instances provided by Macambria and de Souza [4] and other instances generated following the scheme of H.Späth [24].

Keywords: DC Programming, DCA, Quadratic Programming, Weighted Cliques, SemiDefinite Programming

1 Introduction

The Maximum Weighted Clique Problem (MWCP) can be seen as the generalization of Boolean quadratic programming and maximal clique problems with a cardinality constraint. So it is a NP-hard problem. In the work of Hunting et al. [17] several applications are given, for example to facility location and dispersion problem. This problem also arises as a subproblem in a column generation approach to graph partionning (see [19]).

Exact Branch and Cut algorithms for MWCP based on linear programming have been proposed by Dijkhuizen and Faigle [3], Park et al. [20], Mehrotra[18]. These studies deal with graphs with up to 30 nodes whereas - as formulated by Macambira and de Souza [4] - "the real challenge is to solve problems with $n \geq 40$ nodes". Different cutting strategies for a branch and cut algorithm are examined in the work of Macambira and Souza and they managed to solve problems instances with up to 48 nodes. Hunting et al. also provided several new classes of facet defining inequalities for the clique polytope.

*This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 101.01-2013.19

They used some of the inequality classes in Lagrangian relax-and-cut algorithm to solve the same problem instances as in [4]. In recent work, M.Sorensen [19] also introduced four new classes of inequalities and used the Branch and Cut algorithm for the MWCP in which the same problem instances in [4] were tested. Several heuristic methods have been also provided in [1].

In the present paper, we propose a continuous approach based on DC programming and DCA for MWCP. DC programming and DCA (DC Algorithms) were introduced by Pham Dinh Tao in their preliminary form in 1985. They have been extensively developed since 1994 by Le Thi Hoai An and Pham Dinh Tao and become now classic and more and more popular (see e.g. [8, 15, 21, 22]). DCA has been successfully applied to many large-scale (smooth or nonsmooth) nonconvex programs in various domains of applied sciences for which it provided quite often a global solution and proved to be more robust and efficient than standard methods.

A so-called DC program is that of minimizing a DC function over a convex set. According to the theory of DC programming and via the well-known result concerning exact penalty [11, 16], we can easily formulate the MWCP as DC programs. We then suggest using DC programming approach and DCA to solve the problem. A global algorithm is introduced when we combine DCA in a Branch and Bound scheme and speed up the algorithm with the SDP relaxation techniques.

The paper is organized as follows. A short introduction, followed by problem statement and different formulations in Section 2. Section 3 is devoted to DCA and SDP relaxation in a Branch and Bound scheme. Section 4 reports the computational results. The paper will be completed by some conclusions.

2 Problem statement

Let $K_n = (V, E)$ be the complete undirected graph with $V = \{1, 2, \dots, n\}$. To every node $i \in V$ a weight d_i is assigned and weights c_{ij} are associated to the edges in E. The Maximum Weighted Clique Problem (more precisely, the maximum weighted b-clique problem - MWCP$_b$) is to find, among all complete subgraphs with at most b nodes (for some integer $b \in \{1, 2, \dots, n\}$), a subgraph (clique) for which the sum of the weights of all the nodes and edges in the subgraph is maximum.

2.1 Binary Constrained Quadratic Programming Formulation

Since a given edge weight is included in the sum only if the associated *pair* of nodes is in the subgraph, a natural, nonlinear formulation of this problem is :

$$\max \quad \sum_{i=1}^{n} d_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} c_{ij} x_i x_j \tag{1}$$

subject to

$$\sum_{i=1}^{n} x_i \leq b$$

$$x_i \in \{0,1\} \quad \forall i = 1,2,\ldots,n,$$

where x_i equals 1 if node i is in the subclique; otherwise x_i equals 0.

When the nodes are not weighted, the problem (1) is called the Maximum Edge-Weighted Clique Problem, the case studied in [4].

Problem (1) belongs to the class of Binary Constrained Quadratic Programming problem (BCQP). More precisely, the above problem is one typical case of Quadratic 0-1 Knapsack Problem.

$$\min \quad f(x) = \frac{1}{2}x^T Qx + q^T x \tag{2}$$

subject to

$$a^T x \leq b$$
$$x \in \{0,1\}^n,$$

where Q is a symmetric matrix order n, a is a vector in $\mathbb{R}^n$, and b is a real number. Our algorithm propose in the follows can apply generally to Problem (2) .

Remark 1 *We can always suppose that Q is positive definite, i.e., the function f is convex, by using the binary property of x. Precisely,*

$$\begin{aligned} f(x) = \frac{1}{2}x^T Qx + q^T x &= \frac{1}{2}(x^T Qx - \rho\|x\|^2) + \frac{1}{2}\rho\langle e_n, x\rangle + q^T x \\ &= \frac{1}{2}x^T (Q - \rho I_n)x + \frac{1}{2}\rho\langle e_n, x\rangle + q^T x, \end{aligned}$$

where I_n is unit matrix order n, $e_n = (1,1,\ldots,1)^T \in \mathbb{R}$. So by choosing $\rho \leq \lambda_1(Q)$ ($\lambda_1(\bullet)$ is defined as the smallest eigen value of matrix $\bullet$) we obtain an equivalent convex function.

2.2 DC Program Formulation

A general DC program is as follows

$$\alpha = \inf\{f(x) := g(x) - h(x)\}$$

where g and h are lower semi-continuous proper convex functions on $\mathbb{R}^n$.

The vector space of DC functions is quite large to contain almost real life objective functions and is closed under all the operations usually considered in optimization. Thanks to a well-known result called exact penalty in [11] (Theorem 1) and its extension in [16] Problem (2) is equivalent to the following

$$\min \quad \varphi(x) = \frac{1}{2}x^T Qx + q^T x + \tau\sum_{i=1}^{n}\min\{x_i, 1 - x_i\} \tag{3}$$

subject to

$$a^T x \leq b,$$

$$x \in [0,1]^n,$$

where τ is a real number and greater than a $\tau_0 \geq 0$.

Problem (3) is a DC program since the objective function can be expressed as a DC function. Indeed, by the convexity of $f(x)$ and the concavity of $p(x) = \sum_{i=1}^{n} \min\{x_i, 1-x_i\}$, we have

$$\varphi(x) = f(x) + \tau p(x) = f(x) - (-\tau p(x)).$$

3 A branch and bound algorithm

The binary property of Problem (2) makes it suitable for a Branch and Bound algorithm. However an efficient algorithm is obtained when we integrate DCA and SDP relaxation.

3.1 DCA for solving Problem (3)

Based on local optimality conditions and duality in DC programming, the DCA consists in the construction of two sequences $\{x^k\}$ and $\{y^k\}$. These two sequences $\{x^k\}$ and $\{y^k\}$ are determined in the way that x^{k+1} (resp. y^k) is a solution to the convex program (P_k) (resp. (D_k)) defined by

$$\inf\{g(x) - h(x^k) - \langle x - x^k, y^k\rangle : x \in \mathbb{R}^n\}, \quad (P_k)$$

$$\inf\{h^*(y) - g^*(y^{k-1}) - \langle y - y^{k-1}, x^k\rangle : y \in \mathbb{R}^n\} \quad (D_k).$$

DCA performs so a double linearization with the help of the subgradients of h and g and the DCA then yields the next scheme:

$$y^k \in \partial h(x^k); \quad x^{k+1} \in \partial g^*(y^k).$$

Applying DCA to Problem (3) lead to calculate ∂h and $\partial^* g$ where $h(x) = -\tau p(x)$ and $g(x) = f(x)$.

Since

$$-p(x) = \sum_{i=1}^{n} \max\{-x_i, x_i - 1\} = \sum_{i=1}^{n} \max\{(-e^i)^T x, (e^i)^T x - 1\}$$

where $\{e^i\}_{i=1,\ldots,n}$ is the standard basis of $\mathbb{R}^n$, we obtain that

$$\partial h(x) = u : u_i = -\tau \text{ if } x_i < 0.5, \tau \text{ if } x_i > 0.5 \text{ and in } [-\tau, \tau] \text{ otherwise }, i = 1, 2, \ldots, n. \tag{4}$$

In the calculation of $\partial^* g(y)$, precisely a subgradient of g^* at y, we need to solve a quadratic convex problem

$$\min\{f(x) - \langle y, x\rangle : a^T x \leq b, x \in [0,1]^n\}. \tag{5}$$

We describe below DCA applying to Problem (3).

Algorithm 1 DCA for solving Problem (3)

1. Let $k = 1$ and ϵ is a very small positive number. Choose x^k as an initial point (not necessary feasible).
2. Calculate $y^k \in \partial h(x^k)$ by using (4).
3. Calculate $x^{k+1} \in \partial g^*(y^k)$ by solving the quadratic convex program (5).
4. If
$$\|x^{k+1} - x^k\| \leq \epsilon \text{ or } \|\varphi(x^{k+1}) - \varphi(x^k)\| \leq \epsilon$$
then STOP and x^{k+1} is the obtained solution.

   Otherwise, let $k \leftarrow k + 1$ and return to 2.

For a complete study of DC programming and DCA the readers are referred to [8, 15, 21, 22] and references therein. The solution of a nonconvex program by DCA must be composed of two stages: the search for an *appropriate* DC expression and that for a *good* initial point.

3.2 SDP Relaxation

The standard SDP relaxation techniques are known to provide tighter bounds than standard convex relaxations but more effort is required to compute them [6, 2].

SDP relaxation is quite suitable when apply to binary quadratic programs (see [6, 2]). A relaxation is obtained by relaxing the binary condition of variables and then rewrite the linear constraints that we call the lifting process. The formulation of MWCP allows us to do the lifting in all four standard ways. They are : the *diagonal representation*, the *square representation*, the *extended square representation* and the one introduced by Lovasz and Schrijiver - we call the LS representation. The bound obtained by SDP relaxation is tighter and tighter if we go from diagonal representation to the LS representation [6, 2].

In our algorithm, we just do the *extended square representation* in relaxation because it gives a lower bound which is tight enough. The lifting introduced by Lovasz and Schrijver could give tighter lower bound but it takes much more time to carry out. Further details can be found in [2, 6].

Finally, the relaxation for Problem (2) was used as follows

$$\min \quad \frac{1}{2}(Q + 2\,\mathrm{Diag}(q)) \bullet X \tag{6}$$

$$\left\langle \begin{pmatrix} b \\ -a \end{pmatrix} \begin{pmatrix} 0 \\ a \end{pmatrix}^T, \bar{X} \right\rangle \geq 0$$

$$X - \mathrm{diag}(X)\,\mathrm{diag}^T(X) \succeq 0,$$

where Diag($\bullet$) is the diagonal matrix formed by a vector $\bullet$ and diag($\bullet$) is the vector obtained by taking the diagonal of a matrix $\bullet$.

Remark 2 *At the solution X^* of (6), we have the property that* $\text{diag}(X^*) \in [0,1]^n$*, see [6, 2]. We then take* $\text{diag}(X^*)$ *as the initial point for DCA.*

Remark 3
At the first launch of DCA, we want to find a good feasible point for DCA, once again the DCA applied to the concave programming problem developed in [12]:

$$0 = \min \Big\{ \sum_{i=1}^{n} \min \{x_i, 1 - x_i\} : a^T x \leq b, x \in [0,1]^n \Big\}. \tag{7}$$

For the next launches, we apply DCA directly as DCA can give a feasible solution of problem although it begins with a non feasible one.

When do we restart DCA?
During the B&B process we can (and should) restart DCA to find a better upper bound. There are many strategies to restart DCA. In our algorithm, we restart DCA when the value of DC functions at the optimal solution of relaxed problem is "closed enough" to the current upper bound.

4 Numerical Results

This section summarizes the numerical results obtained by applying our algorithm to two sets of test problems. The first set of test problems has been provided by Macambria and de Souza [4] and has also been used in [17, 19]. We generate the second set of test problems. The weights were randomly generated according to the scheme described in [24], which is also used by Macambria and de Souza in their tests. Precisely,
• $1 \leq c_{ij} \leq 1000r^k (1 \leq c_{ij} \leq 1000)$ for positive weights and
• $-1000r^k \leq c_{ij} \leq 1000r^k (-1000 \leq c_{ij} \leq 1000)$ for positive and negative weights,
where $k \in \{1, \ldots, 5\}$ and $r \in (0, 1]$. The value of b is chosen as $\lfloor n/2 \rfloor$. In this set of test problems, we increase the size of graph up to 100.

The algorithm has been implemented in a C program using CPLEX 11.0 as LP solver and the callable library function for solving SDP [5]. All computational results have been obtained by implementing our program under Window XP on a 2GH PC portable. In all instances, we find an exact global solution.

Table 1 and Table 2 show the computational results for the problem instances provided by Macambria and de Souza, table 1 for positive edge-weights and 2 for mixed edge-weights. For the larger instances, the numerical results are provided in Table 3.

The following notations are used in the Tables

- Val. Opt. : Optimal value,
- Val. DCA : Value given by DCA in the first run,

- Iter : Number of iterations,
- #DCA : Number of times of restarting DCA.

A direct comparison in the computation times is quoted in [17, 4, 19]. The comparison is relative because different computer systems have been used. However we can see that our approach is very efficient even when we find the exact optimal global solution.

With larger dimension generated, up to 80 nodes, our algorithm still works well. We tested the case of 100 nodes but a strongl remark can be seen that, with the SDP relaxation, the cost to accomplish is expensive.

We can also see that, DCA at the first launch, gives a solution which is very closed to the global optimal one.

Conclusions

We have proposed a new approach to solve the Maximum Weighted Clique Problem. The numerical experiments show that our approach is efficient. The generated instances with larger dimension can be solved in the acceptable time. This result is obtained from the efficiency of DCA and the tight bounds given by the SDP relaxation.

Acknowledgements. I would send a big thank to Dr. Macambira for providing his and Professor Cid de Souza's test problems for me.

References

[1] Bahram Alidaee, Fred Glover, Gary Kochenberger and Haibo Wang: Solving the maximum edge weight clique problem via unconstrained quadratic programming. European Journal of Operational Research **181**, 592–597 (2007)

[2] Christoph Helmberg : SemiDefinite Programming for Combinatorial Optimization. ZIB-Report 00-34, Octorber(2000)

[3] Dijkhuizen, G., Faigle, U. : A cutting-plane approach to the edge-weighted maximal clique problem. European Journal of Operational Research **69**, 121–130 (1993)

[4] Elder Magalhães Macambira and Cid Carvalho de Souza : The edge-weighted clique problem : Valid inequalities, facets and polyhedral computations. European Journal of Operational Research **123**, 346–371 (2000)

[5] Fujisawa, K., Kojima, M., Nakata, K. and Yamashita, M. : SDPA - User's Manual Version 6.00. Research reports on Mathematical and Computing Sciences, Tokyo Institue of Technology (2002)

[6] Henry Wolkowicsz, Romesh Saigal and Lieven Vandenberghe : Hanbook of Semidefinite programming; Theory, Algorithms, and Applications. Kluwer Academic Publisher (2000)

[7] Hiriart Urruty, J. B. and Lemaré chal, C. : Convex Analysis and Minimization Algorithms. Springer Verlag berlin Heidelberg (1993)

Table 1: Numerical results - Positive Weight

ID	Val. Opt.	Val. DCA	DCA - B&B - SDP			M&de S	HF&K	M.S
			Iter	#DCA	time(s)	times(s)		
P40.1	109346	**108851**	24	2	**5.92**	4267.23	1677	103
P40.2	82451	**82280**	28	2	**6.75**	784.85	946	46
P40.3	68759	62870	108	2	**22.16**	3015.97	2115	131
P40.4	60782	**60238**	68	2	**14.92**	3125.80	2915	114
P40.5	60513	**58380**	25	2	**6.13**	621.12	917	51
P42.1	120299	**98119**	134	2	**32.33**	8894.25	3845	201
P42.2	87810	**86232**	108	2	**26.38**	8732.11	4203	215
P42.3	76554	70743	104	2	**25.28**	4736.14	3947	163
P42.4	69482	**68444**	80	2	**19.84**	1245.75	1381	102
P42.5	67383	**62663**	38	2	**10.13**	752.83	1245	52
P44.1	136525	**135225**	23	2	**7.70**	3902.93	1912	133
P44.2	98186	**97382**	79	2	**23.11**	13599.25	3672	284
P44.3	84675	**78524**	42	2	**12.84**	2149.72	1332	169
P44.4	75274	**74691**	27	2	**8.67**	3553.14	1403	214
P44.5	69540	**67592**	40	2	**12.48**	1508.55	4113	110
P45.1	138694	**136264**	162	3	**48.38**	22275.82	(N/A)	245
P45.2	98321	**76444**	223	2	**62.89**	26268.18	(N/A)	369
P45.3	82743	**80260**	367	2	**97.33**	18208.38	(N/A)	312
P45.4	77500	**74753**	173	2	**50.02**	6685.92	(N/A)	225
P45.5	69563	**68833**	116	3	**33.91**	8712.95	(N/A)	291
P46.1	142985	**141988**	126	2	**40.13**	35398.46	5590	415
P46.2	108243	**97194**	164	2	**52.08**	18581.20	7481	439
P46.3	94859	**94563**	89	3	**29.70**	2971.74	2421	232
P46.4	78747	72628	152	2	**47.66**	12944.28	4222	419
P46.5	72431	**72164**	137	2	**43.38**	12413.82	4434	370
P48.1	163397	**157846**	90	2	**34.44**	39312.96	4959	386
P48.2	115471	**108658**	423	2	**139.48**	60796.54	27479	1135
P48.3	96666	92049	552	2	**179.84**	45622.65	11166	517
P48.4	88728	74985	73	2	**28.22**	6147.11	2122	272
P48.5	82117	**81374**	56	2	**22.03**	3186.07	2093	262

Table 2: Numerical results - Mixed Weight

ID	Val. Opt.	Val. DCA	DCA - B&B - SDP			M&de S	HF&K	M.S
			Iter	#DCA	time(s)	times(s)		
M40.1	70348	63284	139	2	**27.55**	12481.00	14079	223
M40.2	45404	**42897**	37	3	**8.28**	2219.68	1857	68
M40.3	34091	30594	58	2	**12.31**	1298.51	1129	64
M40.4	27758	26151	133	3	**25.75**	4759.35	2548	84
M40.5	27967	26397	45	2	**9.70**	477.35	2785	44
M42.1	81633	**78975**	175	2	**39.16**	18754.11	14841	269
M42.2	46828	37554	169	2	**37.41**	5569.29	4159	193
M42.3	36689	**30740**	105	2	**24.39**	1119.90	2130	84
M42.4	35987	**32710**	46	2	**11.64**	66.63	534	58
M42.5	35460	34595	24	2	**6.38**	707.50	1800	59
M44.1	90620	76524	286	2	**70.38**	20388.73	27445	347
M44.2	56960	**56168**	128	2	**34.03**	4201.43	3329	195
M44.3	40697	**36849**	54	2	**15.30**	1277.06	2142	151
M44.4	32601	32101	121	3	**31.11**	14388.82	4208	169
M44.5	29407	25441	72	2	**19.56**	2633.66	1502	129
M45.1	102295	**101374**	40	2	**12.84**	16111.23	(N/A)	252
M45.2	55103	**52014**	125	2	**35.52**	11021.61	(N/A)	353
M45.3	43914	39981	19	2	**6.33**	637.28	(N/A)	84
M45.4	33990	29530	83	2	**23.95**	7549.38	(N/A)	140
M45.5	30974	26730	245	2	**63.66**	9397.09	(N/A)	237
M46.1	**99550**	**99950**	97	1	**30.50**	19276.87	10348	383
M46.2	58361	**56025**	96	2	**29.44**	5988.41	4579	358
M46.3	43915	**41867**	219	3	**62.59**	7323.71	5418	242
M46.4	32968	**29519**	396	4	**105.63**	20632.69	10185	344
M46.5	31000	**30321**	83	2	**25.77**	1693.50	2350	144
M48.1	113478	84079	174	3	**59.02**	63603.96	55917	800
M48.2	61768	**55988**	169	2	**61.84**	33527.03	36963	840
M48.3	45941	**45941**	108	2	**37.55**	6625.02	3277	290
M48.4	36903	32758	58	2	**20.72**	2781.26	2257	206
M48.5	31351	**29484**	373	2	**115.06**	24048.16	4505	307

Table 3: Numerical results - Generated Data

Problem	Val. Opt.	Val. DCA	DCA - B&B - SDP		
			Iter	#DCA	time(s)
50.1-Pos	87105	**86709**	178	2	72.77
50.1-Mix	60576	**59949**	113	2	46.06
60.1-Pos	179187	**156364**	124	2	94.91
60.1-Mix	21541	**20224**	220	2	150.86
70.1-Pos	103782	**103079**	84	2	108.03
70.1-Mix	289118	**285282**	437	2	467.61
80.1-Pos	177403	**176179**	371	2	702.63
80.1-Mix	23346	**21542**	509	2	869.19
100.1-Mix	112412	101560	1826	2	**6364.83**

[8] Le Thi Hoai An and Pham Dinh Tao : Solving a class of linearly constrained indefinite quadratic problems by DC algorithms. Journal of Global Optimization **11**, No 3, 253–285 (1997)

[9] Le Thi Hoai An, Pham Dinh Tao : DC (Difference of convex functions) optimization algorithm (DCA) for globally minimizing nonconvex quadratic forms on euclidean balls and spheres. Operations Research Letters **19**, 207–216 (1996)

[10] Le Thi Hoai An, Pham Dinh Tao : Numerical solution for optimization over the efficient set by DC optimization algorithms. Operations Research Letters **19**, 117–128 (1996)

[11] Le Thi Hoai An, Pham Dinh Tao and Le Dung Muu : Exact Penalty in DC Programming. Vietnam Journal of Mathematics **27**, No 2, 169–178 (1999)

[12] Le Thi Hoai An, Pham Dinh Tao : A continuous Approach for Globally Solving Linearly Constrained Quadratic Zero - One Programming Problems. Optimization **50**, 93–120 (2001)

[13] Le Thi Hoai An and Pham Dinh Tao : DC Programming: Theory, Algorithms and Applications. The State of the Art. A. (ed.) The First International Workshop on Global Constrained Optimization and Constraint Satisfaction (Cocos' 02), Valbonne-Sophia Antipolis, France, October 2-4, 28 pages, (2002)

[14] Le Thi Hoai An and Pham Dinh Tao : Large Scale Molecular Optimization from distances matrices by a DC optimization approach. SIAM Journal of Optimization **14**, No 1, 77–116 (2003)

[15] Le Thi Hoai An and Pham Dinh Tao : The DC (difference of convex functions) Programming and DCA revisited with DC models of real world nonconvex optimization problems. Annals of Operations Research **133**, 23–46 (2005)

[16] Le Thi Hoai An, Pham Dinh Tao and Huynh Van Ngai : Exact penalty and error bounds in DC programming. Journal of Global Optimization, volume 52(3): pp. 509-535, 2012.

[17] Marcel Hunting, Ulrich Faigle and Walter Kern : A Lagrangian relaxation approach to the edge-weighted clique problem. European Journal of Operational Research **131**, 119–131 (2001)

[18] Mehrotra, M.A : Cardinality constrained Boolean quadratic polytope. Discrete Applied Mathematics **79**, 137–154 (1997)

[19] Michael M.Sorensen : New facets and a branch-and-cut algorithm for the weighted clique problem. European Journal of Operational Research **154**, 57–70 (2004)

[20] Park, K., Lee, K., Park, S. : An extended formulation approach to the edge-weighted maximal clique problem. European Journal of Operational Research **95**, 671–682 (1996)

[21] Pham Dinh Tao and Le Thi Hoai An : Convex analysis approach to DC programming: Theory, Algorithms and Applications. Acta Mathematica Vietnamica, dedicated to Professor Hoang Tuy on the occasion of his 70th birthday **22**, No1, 289–355 (1997)

[22] Pham Dinh Tao and Le Thi Hoai An : DC optimization algorithms for solving the trust region subproblem. SIAM J. Optimization **8**, 476–505 (1998)

[23] Rockafellar, R.T. : Convex Analysis. Princeton, Princeton University (1970)

[24] Späth, H. : Heuristically determining cliques of given cardinality and with minimal cost within weighted complete graphs. Zeitschrift für Operations Research **29**, 125–131 (1985)