Journal of Global Optimization **21:** 265–288, 2001.
© 2001 *Kluwer Academic Publishers. Printed in the Netherlands.*

On solving the maximum clique problem

ANTONINA KUZNETSOVA and ALEXANDER STREKALOVSKY
Institute of System Dynamics and Control Theory SB of RAS, Lermontov Str., 134, Irkutsk-33, 664033, Russia
E-mail: kuznet@icc.ru and strekal@icc.ru

Abstract. The Maximum Clique Problem (MCP) is regarded here as the maximization of an indefinite quadratic form over the canonical simplex. For solving MCP an algorithm based upon Global Optimality Conditions (GOC) is applied. Furthermore, each step of the algorithm is analytically investigated and tested. The computational results for the proposed algorithm are compared with other Global Search approaches.

Key words: d.c. maximization, global optimality conditions, local search, linearized problem, global search algorithm

1. Introduction

In this paper we develop an approach proposed earlier for continuous nonconvex problems (Strekalovsky 1993, 1997, 2000; Strekalovsky and Tsevendorj, 1998; Kuznetsova et al., 1999) and apply it for solving the well-known combinatorial problem, the Maximum Clique Problem (MCP).

We follow here the continuous formulation of MCP due to Motzkin and Strauss (1965), which was regularized by Bomze (1997). This allows us to present MCP as an indefinite quadratic maximization problem over the canonical simplex and then to apply the so-called Global Optimality Conditions (GOC) for d.c. maximization problem (Strekalovsky 1993, 1997, 2000; Strekalovsky and Tsevendorj, 1998; Kuznetsova et al., 1999) .

In order to take into account the structure of the problem we propose an analytical investigation of the steps of Global Search Algorithm (GSA). The results of computational experiments for the proposed approach show its competitive ability.

The rest of the paper is organized as follows.

After the statement of the problem and recalling GOC in Section 3 we present GSA for general d.c. maximization problem. In Section 4 we study the Linearized Problem which is one of the corner-stones of our approach.

In Section 5 we investigate three types of local search and choose the most suitable for our needs.

After developing the modulus of GSA we move to a preliminary testing of the derived global search algorithm called $\mathfrak{R}$-strategy.

Finally in Section 7 the analysis of preceding computational experiments and additional analytical investigation enabled us to derive an almost discrete version of $\Re$-algorithm, the so-called $\Re D$-algorithm.

The latter version allows to solve DIMACS test-examples of MCP of cardinality up to 800 within a reasonable CPU-time. Moreover in Sanchis' examples, $\Re D$-algorithm found cliques of size 1.5–2 times as large as other known methods.

The authors are indebted to anonymous referees for very helpful remarks and suggestions as well as for pertinant corrections in English.

2. Problem's statement and GOC.

Let $G = G(V, E)$ be a simple undirected graph with vertex set $V = \{1, ..., n\}$ and set of edges E. In addition assume that the graph and its complement have no isolated vertices.

A subset C of V is called a clique if every pair of vertices in C is joined by an edge. MCP is the problem of finding a clique C of the maximum cardinality. Due to Motzkin and Straus (1965), MCP can be stated as the following indefinite quadratic programming problem:

$$\left.\begin{aligned} F_G(x) = \sum_{(i,j)\in E} x_i x_j = \frac{1}{2}\langle x, A_G x\rangle \uparrow \max, \\ x \in S = \{x = (x_1, ..., x_n)^T : \sum_1^n x_i = 1,\ x_i \geqslant 0,\ i = 1, .., n\}, \end{aligned}\right\} \tag{2.1}$$

where A_G is the adjacency matrix of G.

It is well-known that G has a maximum clique C of cardinality $k = (1-2\alpha)^{-1}$, where $\alpha = \max(F_G, S)$. This maximum can be attained by setting $x_i = 1/k$, where $i \in C$, and $x_i = 0$ if $i \notin C$.

Nevertheless, a global solution to (2.1) is not directly related to a maximum clique (Horst et al., 1995). For instance, Pelillo and Jagota (1995), gave characterizations of some 'spurious' or 'infeasible' solutions. Therefore, Bomze (1995) gave a regularization of Problem (2.1) replacing the goal function in (2.1) by the quadratic function with the matrix $A = A_G + \frac{1}{2}I_n$, where I_n is the identity matrix. Then a global solution x^* to the following regularized problem:

$$F(x) = \frac{1}{2}\langle x, Ax\rangle \uparrow \max, \quad x \in S, \tag{P}$$

allows to define the corresponding maximum clique as follows:

$$C = \{i \in V : x_i^* > 0\}.$$

Consider a more general (than (P)) d.c. maximization problem:

$$F(x) = f(x) - g(x) \uparrow \max, \quad x \in D, \tag{2.2}$$

where f, g are convex functions over a convex set $D \subset R^n$.

THEOREM 1. *(Strekalovsky, 1997, 2000). Let $z \in D$ be a global solution to* (2.2) *($z \in Sol(2.2)$), and $\zeta = F(z) = f(z) - g(z)$. Then for every pair $(y, \beta) \in R^n \times R$ belonging to the boundary of the epigraph of the function $(f(\cdot) - \zeta)$, i.e.*

$$f(y) - \beta = \zeta, \tag{2.3}$$

and such that

$$y \in D, \ g(y) \leqslant \beta \leqslant \sup(g, D), \tag{2.4}$$

the following variational inequality holds

$$g(x) - \beta \geqslant \langle \nabla f(y), x - y \rangle \quad \forall x \in D. \tag{2.5}$$

If in addition the following assumption takes place

$$\exists v \in D : F(v) < F(z), \tag{H}$$

then conditions (2.3)–(2.5) *become sufficient for $z \in D$ to be a global solution to* (2.2).

Proof. Consider only the necessity's proof. If the conditions (2.3)–(2.5) fail, i.e. there exists a triplet (y, β, u), such that

$$\beta - f(y) = F(z), \ u \in D,$$

$$g(u) < \beta + \langle \nabla f(y), u - y \rangle$$

then due to convexity of $f(\cdot)$ we have

$$g(u) < \beta - f(y) + f(u) = f(u) - F(z),$$

whence $F(u) < F(z)$. The latter contradicts the fact, that z is a solution of (2.2).

The proof of sufficiency can be found in (Strekalovsky, 1997, 2000). □

Note, that GOC is related to the classical extremum theory (Strekalovsky, 1997, 2000). Furthermore, (2.5) suggests to consider a family of convex Linearized Problems:

$$g(x) - \langle \nabla f(y), x \rangle \downarrow \min \ x \in D; \tag{PL}$$

depending on the parameters (y, β) verifying (2.3)–(2.4).

Besides, GOC possesses the so-called Algorithmic Property (AP). The latter means that, if (2.3)–(2.5) are violated then there exists a procedure for constructing a better feasible point (see the proof above).

Exploiting this AP, one gets a Global Search Algorithm (GSA) for (2.2) described in Strekalovsky, (2000) and called '$\mathfrak{R}$-strategy'. Below we present a version of $\mathfrak{R}$-strategy appropriate to Problem (P), and in the following sections we consider the most important parts of the Algorithm.

3. D.C. decomposition of Problem (P) and Global Search Algorithm

It is well-known that an indefinite matrix A can be represented as a difference of two positive definite matrices:

$$A = A_1 - A_2. \tag{3.1}$$

As a consequence, Problem (P) turns out to be a particular case of d.c. maximization problem (2.2), where

$$f(x) = \frac{1}{2}\langle x, A_1 x\rangle, \quad g(x) = \frac{1}{2}\langle x, A_2 x\rangle. \tag{3.2}$$

One of well-known ways to present the matrix A in the form (3.1) is as follows

$$A = (A + \mu I_n) - \mu I_n,$$

where $\mu > 0$ is rather large.

We propose another decomposition of A more appropriate for the structure of the matrix $A = A_G + \frac{1}{2}I_n$. Let $d_i = \sum_{j=1}^{n} a_{ij}^G$ be the degree of the vertex i w.r.t. the graph G. Then

$$\begin{aligned} F(x) &= \frac{1}{2}\langle x, Ax\rangle = \frac{1}{4}\sum_{i=1}^{n} x_i^2 + \frac{1}{2}\sum_{(i,j)\in E} x_i x_j \\ &= \frac{1}{4}\left(\sum_{i=1}^{n} x_i^2 + \sum_{(i,j)\in E}(x_i + x_j)^2\right) - \frac{1}{2}\sum_{i=1}^{n} d_i x_i^2, \end{aligned}$$

where

$$1 \leqslant d_i \leqslant n-2, \; i = 1, ..., n, \tag{3.3}$$

provided that there is no isolated vertex.

So, we obtained a d.c. decomposition of the goal function $F(\cdot)$ where both functions

$$f(x) = \frac{1}{4}\left(\sum_{i=1}^{n} x_i^2 + \sum_{(i,j)\in E}(x_i + x_j)^2\right), \tag{3.4}$$

$$g(x) = \frac{1}{2}\sum_{i=1}^{n} d_i x_i^2 \tag{3.5}$$

are strongly convex.

It is clear that the decomposition $F(x) = f(x) - g(x)$ corresponds to the decomposition (3.1) of the matrix A, where

$$A_2 = \mathrm{diag}\{d_1, ...d_n\}, \quad A_1 = A + A_2. \tag{3.6}$$

Now let us describe a Global Search Algorithm exploiting AP of GOC (see also [12]-[15]). Denote $\beta_- := \inf(g, S_n)$, $\beta_+ := \sup(g, S_n)$. Let a point $x^0 \in S$ and a number sequence $\{\varepsilon_k\}$, $\varepsilon_k > 0$, $k = 0, 1, 2, \dots$, $\varepsilon_k \downarrow 0$ $(k \to \infty)$ are given.

$\Re$-strategy.

```
Step 0.  Set $k := 0$, $x^k := x^0$.
Step 1.  Starting from $x^k \in D$ obtain $z^k \in D$ by one of local search methods for
         Problem $(P)$, so that $z^k$ is an $\varepsilon_k$-stationary point in $(P)$. Set $\zeta_k := F(z^k)$.
Step 2.  Choose some $\beta \in [\beta_-, \beta_+]$. In particular, one can begin with $\beta_0 = g(z^k)$.
Step 3.  Construct an approximation
             $A_k(\beta) = \{y^1, ..., y^{N_k} \ / \ f(y^i) = \beta + \zeta_k, \ i = \overline{1, N_k}, \ N_k = N_k(\beta)\}$.
Step 4.  Introduce the set $I_k = I_k(\beta) = \{i \in \{1, ..., N_k\} \ / \ g(y^i) \leqslant \beta\}$.
Step 5.  For every $i \in I_k$ solve the Linearized Problem
             $g(x) - \langle \nabla f(y^i), x \rangle \downarrow \min, \ x \in S.$        $(PL_i)$
         Let $u^i$ be an $\varepsilon_k$-solution of $(PL_i)$.
Step 6.  For $i \in I_k$ starting at the point $u^i$ find a stationary point $v^i \in S$ by a local
         search method.
Step 7.  For every $i \in I_k$ find a point $w^i$, $f(w^i) = \beta + \zeta_k$, such that
             $\langle \nabla f(w^i), v^i - w^i \rangle + \varepsilon_k \geqslant \sup_y \{\langle \nabla f(y), v^i - y \rangle \ / \ f(y) = \beta + \zeta_k\}.$
Step 8.  Set
             $\eta_k(\beta) = \langle \nabla f(w^j), v^j - w^j \rangle + \beta - g(v^j) =$        (3.7)
             $= \max_{i \in I} \{\langle \nabla f(w^i), v^i - w^i \rangle + \beta - g(v^i)\}.$        (3.8)
Step 9.  If $\eta_k(\beta) > 0$, set $x^{k+1} := v^j$ and loop to Step 1.
Step 10. If $\eta_k(\beta) \leqslant 0$, set $\beta := \beta + \triangle\beta$ and go to Step 3.
Step 11. Stop, if $\eta_k(\beta) \leqslant 0$ for all $\beta \in [\beta_-, \beta_+]$ (i.e maximization of $\eta_k(\beta)$ over
         $[\beta_-, \beta_+]$ is finished) and if $\varepsilon_k \leqslant \delta$, where $\delta$ is a given tolerance.  □
```

REMARK 1. It can be seen that Step 6 is new comparing with the versions proposed in Strekalovsky (1993, 2000), Strekalovsky and Tsevendorj (1979) and Kuznetsova et al., (1999).

4. Linearized Problem's Solving

Consider Problem (PL_i) where $f(y^i) = \beta + \zeta$, $\beta \in [\beta_-, \beta_+]$, $\zeta = F(z)$, where $i \in \{1, \dots, N_k\}$ and $g(\cdot)$ is defined in (3.5). Set $y = y^i$ and denote

$$r = (r_1, ... r_n)^T, \quad r = \nabla f(y).$$

Assume, in addition, that

$$r_1 \geqslant \ldots \geqslant r_{i-1} \geqslant r_i \geqslant \ldots \geqslant r_n. \tag{4.1}$$

So, Problem (PL_i) takes the form:

$$\left.\begin{aligned} h(x) = \frac{1}{2}\sum_1^n d_i x_i^2 - \sum_1^n r_i x_i \downarrow \min, \\ x \in S = \{x \in R^n \ / \ \sum_1^n x_i = 1 \ \ x_i \geqslant 0, \ \ i = 1, \ldots, n\}. \end{aligned}\right\} \tag{4.2}$$

Recall that due to (3.3) the objective function of (4.2) is strongly convex, and we might apply the standard methods of convex quadratic programming. However, in order to find the unique solution of (4.2) in a fast way, we prefer to use a special finite solving method. Let us begin with

LEMMA 1. *Let $u = (u_1 \ldots u_n)$ be the solution to (4.2) $(u \in Sol(4.2))$. Then only one of the two following alternatives takes place:*
(a) $u_i > 0$ for every $i = \overline{1, n}$;
(b) there exists a number p, $1 \leqslant p < n$, such that

$$u_i > 0 \ \ \textit{for every} \ \ i, \ \ 1 \leqslant i \leqslant p,$$
$$u_i = 0, \ \ \textit{when} \ \ i > p.$$

Proof. Suppose, that there exist some numbers s and t, such that $s < t$, but $u_s = 0$, $u_t > 0$. Since the problem (4.2) is convex and regular, from KKT-theorem one deduces

$$\left.\begin{aligned} d_t u_t - r_t \qquad + \mu = 0, \\ - r_s - \mu_s + \mu = 0. \end{aligned}\right\}$$

Since $r_s \geqslant r_t$, one gets

$$\mu = \mu_s + r_s \geqslant \mu_s + r_t = \mu_s + d_t u_t + \mu$$

or $\mu_s + d_t u_t \leqslant 0$, which is wrong, since $\mu_s \geqslant 0$, $d_t > 0$, $u_t > 0$. □

LEMMA 2. *If $r_i = r_j$ $1 \leqslant i, j \leqslant n$, then only one of the two alternatives takes place:*

(a) $u_i > 0, \ \ u_j > 0$;
(b) $u_i = u_j = 0$.

Proof. Due to $r_i = r_j$ the case $u_i > 0$, $u_j = 0$ is equivalent to the case $u_i = 0$, $u_j > 0$ which is impossible according to Lemma 1. □

In order to develop the solving method for Linearized Problem (4.2), one needs to consider some auxiliary relaxed problems depending on the integer parameter $l,\ 1 \leqslant l \leqslant n$:

$$\left.\begin{aligned} h_l(x) = \tfrac{1}{2}\sum_1^l d_i x_i^2 - \sum_1^l r_i x_i \downarrow \text{min}, \\ \sum_1^l x_i = 1. \end{aligned}\right\} \tag{4.3}$$

Let $u^l = (u_1^l, ..., u_l^l) \in R^l$ be a solution to (4.3) . Applying the Lagrange rule to (4.3) one gets for $i = 1, \dots, l$:

$$u_i^l = \frac{\sum_{s=1}^l (r_i - r_s)/d_s + 1}{d_i \sum_{s=1}^l 1/d_s}. \tag{4.4}$$

LEMMA 3. *Only one of the two alternatives takes place:*
(a) $u_i^l > 0$ *for every* $i = 1, \dots, l$;
(b) one can find a number $p = p(l) < l$ *such that*

$$u_i^l > 0 \ \ for\ \ 1 \leqslant i \leqslant p,$$
$$u_i^l \leqslant 0,\ \ when\ i > p.$$

Proof. The representation (4.4) is deduced from

$$u_i^l = (r_i - \mu)/d_i, \ \ i = 1, \dots, l,$$

where μ is the Lagrange multiplier in Problem (4.3). If $j < l$ and $u_j^l \leqslant 0$, then $\mu \geqslant r_j$, since $d_i \geqslant 1$, $i = 1, \dots n$. Hence, due to (4.1) $\mu \geqslant r_i$ and $u_i^l \leqslant 0$ for all $i :\ j < i \leqslant l.$ □

Now we are able to describe the solution to Linearized Problem (4.2).

PROPOSITION 1. *Let* $u = (u_1, ..., u_n)$ *be the solution to (4.2).*

(i) *Then the alternative a) of Lemma 1 takes place iff* $u_i^n > 0$ *for every* $i = 1, \dots, n$, *where* $u^n = (u_1^n, ..., u_n^n)$ *is the solution to (4.3) with* $l = n$. *Besides,* $u = u^n$ *and the components* u_i *of* u *can be found from (4.4) with* $l = n$.

(ii) *The alternative b) of Lemma 1 takes place iff the positive components of* u *form the solution* u^p *of Problem (4.3) with* $l = p$, *and the number* p *verifies the inequality:*

$$1 + \sum_{s=1}^p (r_{p+1} - r_s)/d_s \leqslant 0. \tag{4.5}$$

In this case the solution $u = (u_1, ..., u_n)$ is described as follows:

$$\left.\begin{aligned} u_i &= \left(1 + \sum_{s=1}^{p}(r_i - r_s)/d_s\right) \Big/ \left(d_i \sum_{s=1}^{p} 1/d_s\right), \quad i = \overline{1, p}, \\ u_i &= 0, \quad i = \overline{p+1, n}. \end{aligned}\right\} \tag{4.6}$$

Proof. (i) If $u \in Sol(4.2)$ then in virtue of the Lagrange rule [1] we have

$$d_i u_i - r_i - \mu_i + \mu = 0, \;\; \mu_i u_i = 0, \;\; i = \overline{1, n}. \tag{4.7}$$

Since $u_i > 0$, one gets $\mu_i = 0$. Then Eq. (4.7) gives the necessary and sufficient conditions for the point u to be a solution to (4.3). On the other hand, since the canonical simplex S is included into the feasible set of Problem (4.3) with $l = n$, then u^n with all positive components turns out to be also a solution to (4.2).

(ii) If u is a solution of (4.2) then one can show similarly to the above that the positive components of u form a solution to (4.3) with $l = p$. The inequality (4.5) can be proved by using (4.7) and the property $\mu_{p+1} \geqslant 0$. □

Based on these results we propose two kinds of algorithms for Linearized Problems Solving, the ‘lower’ and the ‘upper’ respectively. The ‘lower’ algorithm starts with $p = 1$, while the ‘upper’ one begins with $p = n$.

After the analysis of computational experiments we have put up on the ‘upper’ way.

Let us describe the ‘upper’ algorithm step by step.

‘Upper’-algorithm.

```
Step 0.  Set $l := n$.
Step 1.  Find the solution $u^l$ to (4.3) by (4.4).
Step 2.  (Stopping criterion). If $u_i \geqslant 0$ for every $i = 1, \dots, l$, then go to Step 5.
Step 3.  Define the number $p = p(l) < l$ such that
```

$$\left.\begin{aligned} u_i^l > 0 \;\; &for\; i:\; 1 \leqslant i \leqslant p, \\ u_i^l \leqslant 0, \;\; &for\;\; i:\; p < i \leqslant l. \end{aligned}\right\}$$

```
Step 4.  Set $l := p$ and loop to Step 1.
Step 5.  Construct $u$ as follows
```

$$u = \begin{cases} u_i^l, & 1 \leqslant i \leqslant l, \\ 0, & l < i \leqslant n. \end{cases} \tag{4.8}$$

Stop. u is the solution to (4.2). □

PROPOSITION 2. *The ‘upper’ algorithm allows to find the solution of Linearized Problem (4.2) in a finite number of steps.*

Proof. The finiteness of the algorithm follows from the description of steps 0,2,3,4 and the fact that at step 3 $p < l$. □

5. Local Search

In this section we present our choice of Local Search Algorithm (LSA) for seeking a local solution in MCP.

The first approach can be described as follows.

Given a feasible point $x^s \in D$, one looks for the next iteration $x^{s+1} \in D$ as an approximate solution of the following Linearized (at x^s) problem

$$g(x) - \left\langle \nabla f(x^s), x \right\rangle \downarrow \min, \quad x \in S.$$

It is clear that the problem can be solved by the 'upper' algorithm of preceding section.

Let us call the above algorithm the L-procedure.

The second approach was proposed by Bomze (1997) and constructs the sequence $\{x^s\} \subset S$ according to the rule

$$x_i^{s+1} = x_i^s \frac{(Ax^s)_i}{\langle x^s, Ax^s \rangle}. \tag{5.1}$$

As shown in Bomze (1997), if the matrix A in the problem

$$\Phi(x) = \frac{1}{2}\langle x, Ax \rangle \uparrow \max, \quad x \in S,$$

is symmetric, $A = A^T$, then the sequence $\{x^s\}$ generated by the rule (5.1) and beginning at any feasible point x^0, turns out to be relaxing, i.e. $\Phi(x^{s+1}) > \Phi(x^s),\ s = 0, 1, 2, \ldots$, and converges to a stationary point. In the sequel we call this method B-procedure. Note that B-procedure presents only one part of the local search of I. Bomze in Bomze (1997).

Since the third algorithm allows to find a maximal clique C, in the sequel we call it C-procedure. In order to describe C-procedure, let begin with

LEMMA 4. *Suppose for some point $x \in S$ the set*

$$Supp(x) = \{i / x_i > 0\}$$

is not a clique. There then exists a pair of vertices, say, k and p; $k\ p \in Supp(x)$, such that the point

$$y = x + x_p(e^k - e^p)$$

is better than x: $F(y) > F(x)$.

Proof. Since $Supp(x)$ is not a clique, then there exist numbers k and p, s.t. $a_{kk} = a_{pp} = 0.5$, $a_{kp} = 0$. Suppose, $(Ax)_k \geqslant (Ax)_p$. Then for $y = x + x_p(e^k - e^p)$ we have

$$F(y) = F(x) + x_p((Ax)_k - (Ax)_p) + (x_p)^2/2.$$

Since $x_p > 0$, one gets $F(y) - F(x) > 0$. □

LEMMA 5. *(Bomze, 1997) Suppose C is a maximal clique of size $k = |C|$ and $z = \frac{1}{k}\sum_{i \in C} e^i$,*

$$S(C) = \{x \in S / x_i = 0, i \notin C\} \subset S.$$

Then $F(x) < F(z)$ for all $x \in S(C)$ and $x \neq z$.

Now let us describe C-procedure which consists of two parts.
(1) The first one begins at some point $x^0 \in S$ and generates the sequence $\{x^m\}$ of feasible points x^m strictly improving the value of the goal function at each iteration:

$$F(x^m) < F(x^{m+1}), \quad m = 0, 1, 2, \ldots$$

The work of the first part is finished when the set $Supp(x^m)$ is a clique.
(2) In the second part one constructs a maximal clique C containing $Supp(x)$, where x is the final point of the first part.

Now let us describe C-procedure step by step.

```
C-procedure
Step 0.  Set $m := 0$, choose $x^0 \in S$.
Step 1.  (Stopping criterion) If $Supp(x^m)$ is a clique, i.e.

             $a_{ij} = 1, \quad \forall i, j \in Supp(x^m), \; i \neq j.$

         then set $x := x^m$ and go to step 4.
Step 2.  Find $k, p$ from $Supp(x^m)$ such that $(k, p) \notin E$, i.e.

             $a_{kp} = 0, \; (Ax^m)_k \geqslant (Ax^m)_p.$          (5.2)

Step 3.  Set

             $x^{m+1} := x^m + x^m_p(e^k - e^p),$

         $m := m + 1$ and loop to Step 1.
Step 4.  Find a maximal clique $C \supset Supp(x^m)$, set $k := |C|$.
Step 5.  Construct the point $z := \frac{1}{k}\sum_{i \in C} e^i$ and Stop.  □
```

PROPOSITION 3. *Suppose, some feasible point x^0 is given. The sequence $\{x^m\}$ generated by C-procedure converges to a point z of strict local maximum to Problem (P) in a finite number of iterations not larger that n. Besides, if the initial point x^0 is not a local maximum to Problem (P), one has the strict improvement, that is*

$$F(x^0) < F(z), \quad x^0 \neq z.$$

Proof. Due to the constraint $\sum_{i=1}^{n} x_i = 1$ one has $Supp(x^m) \neq \emptyset$. On the other hand it can be readily seen that $x^{m+1} \in S$ and $Supp(x^{m+1}) = Supp(x^m)\backslash\{p\}$. Whence one gets the finiteness of C-procedure. The inequality $F(x^0) < F(z)$ follows from Lemmas 4 and 5. □

Further, let move to a more precise description of Step 2 of C-procedure. Assume, that $Supp(x^m) = \{1, \dots, q\}$ and $(Ax^m)_1 \geqslant \dots \geqslant (Ax^m)_q$.

```
(2)-Algorithm.
Step 0.  Set $i := 1$.
Step 1.  Construct the set $J = \{j/i < j \leqslant q,\ a_{ij} = 0\}$.
Step 2.  If $J = \emptyset$, then set $i := i + 1$ and loop to Step 1.
Step 3.  Set $r := \max\{j/j \in J\}$, $k := i$,  $p := r$. Stop.
```

LEMMA 6. *Suppose, $Supp(x^m)$ is not a clique. Then for the pair of vertices (k, p), $k, p \in Supp(x^m)$, obtained by (2)-algorithm the property (5.2) holds.*

The proof is similar to that of Lemma 4. □

Now, turn our attention to Step 5 of C-procedure, which aims to construct a maximal clique (MC) C including $Supp(x)$. Recall, that according to the definition a clique C is MC, if C is not contained in a clique of larger cardinality.

The following algorithm enables us to construct such a MC.

```
(5)-Algorithm.
Step 0.  Set $C = Supp(x)$.
Step 1.  Construct the set $K = \{k \notin C :\ a_{ki} = 1,\ \forall i \in C\}$.
Step 2.  (Stopping criterion). If $K = \emptyset$, then Stop. $C$ is the sought-after clique.
Step 3.  From the set $K$ choose the number $j$ such that $d_j = \max_{k \in K} d_k$, where $d_k$
         is the degree of the vertex $k$ in the graph generated by $K$.
Step 4.  Set $C = C \cup \{j\}$ and loop to Step 1.  □
```

LEMMA 7. *(5)-Algorithm finds MC, containing the set $Supp(x)$, in a finite number of iterations.*

As above, for instance, in the proof of Proposition 3, the proof of Lemma 7 uses the fact that the set $Supp(x) \subset \{1, \dots, n\}$ is finite.

Now let us consider the results of comparative computational testing of L-, B-, and C-procedures presented in Table 1. Recall, that L- and B-procedures are not

Table 1.

Graph	n	Dens	F_0	F_{max}			F_*	$\mathcal{K}$		
				B	L	C		B	L	C
data_17_1	17	0.279	0.146	0.450	0.450	0.450	0.450	5	5	5
data_17_2	17	0.514	0.256	0.406	0.406	0.416	0.437	–	–	3
data_20_1	20	0.326	0.167	0.416	0.416	0.450	0.450	4	4	5
data_20_2	20	0.447	0.225	0.416	0.416	0.416	0.437	3	3	3
data_25_1	25	0.333	0.170	0.431	0.431	0.437	0.450	–	–	4
data_25_2	25	0.443	0.222	0.437	0.437	0.437	0.450	4	4	4
data_30_1	30	0.455	0.228	0.260	0.260	0.416	0.437	–	–	3
data_30_2	30	0.455	0.228	0.437	0.437	0.437	0.437	4	4	4
data_35_1	35	0.556	0.277	0.425	0.425	0.437	0.450	–	–	4
data_35_2	35	0.554	0.276	0.425	0.425	0.437	0.450	–	–	4
data_40_1	40	0.510	0.255	0.458	0.458	0.458	0.458	6	6	6
data_40_2	40	0.506	0.253	0.422	0.422	0.437	0.450	–	–	4
data_40_3	40	0.506	0.253	0.422	0.422	0.437	0.450	–	–	4
data_40_4	40	0.519	0.259	0.380	0.380	0.416	0.450	–	–	3
data_45_1	45	0.486	0.243	0.425	0.425	0.437	0.458	–	–	4
data_45_2	45	0.449	0.225	0.392	0.380	0.416	0.450	–	–	3
data_50_1	50	0.582	0.290	0.425	0.425	0.458	0.458	–	–	6
data_50_2	50	0.582	0.290	0.446	0.425	0.458	0.464	–	–	6
data_50_3	50	0.581	0.289	0.425	0.425	0.458	0.458	–	–	6
data_50_4	50	0.578	0.288	0.425	0.425	0.437	0.458	–	–	4
data_50_5	50	0.578	0.288	0.425	0.425	0.437	0.458	–	–	4
data_50_6	50	0.574	0.286	0.425	0.425	0.437	0.458	–	–	4
data_50_7	50	0.579	0.289	0.425	0.425	0.437	0.464	–	–	4

finite and converge to a stationary point. The barycenter $x^0 = (1/n, \dots, 1/n)^T$ of the canonical simplex S has been always chosen as initial point. All the testing examples were invented by the first author and can be found by the address: http://dol.iitam.omsk.net.ru/

In Table 1 n is the dimension of the graph G; Dens stands for the density of graph; F_0, F_{max} are the values of the goal function at the initial and final points, respectively; F_* is the global maximum of $F(x)$ over S.

Further, the inequality

$$||x^{s+1} - x^s|| \leqslant \varepsilon, \tag{5.3}$$

with $\varepsilon = 10^{-9}$ was chosen as the stopping criterion for L- and B - procedures. If (5.3) is verified then x^{s+1} was taken as the stationary point z, $F_{max} := F(z)$. In addition, we considered the set

$$Supp(z, \varepsilon) = \{i / z_i > \varepsilon\},$$

and if this set turns out to be a clique reached by corresponding procedure, then the size of the clique was denoted by $\mathcal{K}$ in the table.

It can be seen from Table 1 that in all test examples C-procedure turned out to be the most successful.

6. β-Maximization and an approximation of the level surface of $f(\cdot)$

According to Theorem 1 we need the numbers $\beta_- = \inf(g, S)$, $\beta_+ = \sup(g, S)$. Let us calculate them analytically.

It follows from KKT-theorem that the global minimum $x^\star$ of the convex function $g(x) = \frac{1}{2}\sum_1^n d_i x_i^2$ over the canonical simplex S verifies the equalities:

$$x_i^\star = \mu/d_i, \quad i = 1, \dots, n,$$

where the number μ can be found from the equality $\sum_1^n x_i = 1$. Thus,

$$\beta_- = g(x^\star) = \frac{1}{2}\mu^2 \sum_1^n 1/d_i = \frac{1}{2}\left(\sum_1^n 1/d_i\right)^{-1}.$$

On the other hand, taking into account that a convex function reaches its maximum at an extreme point of a feasible set [7], one gets

$$\beta_+ = \frac{1}{2}\max_i\{d_i \ / \ 1 \leqslant i \leqslant n\}.$$

In order to approximate the level surface $\{x \ / \ f(x) = \beta + \zeta\}$ of the function $f(\cdot)$ defined in (3.2), (3.6) consider the set of directions:

$$D = \{e^1, \dots, e^n \in R^n\}.$$

Then the approximation may be constructed as follows:

$$\mathcal{A}(\zeta, \beta) = \{y^1, \dots, y^n \ / \ f(y^i) = \beta + \zeta\},$$

where

$$y^i = \lambda_i e^i. \tag{6.1}$$

Since $f(x) = \frac{1}{2}\langle x, A_1 x\rangle$, it is obvious that

$$\lambda_i = \lambda_i(\beta) = \left(\frac{2(\beta + \zeta)}{d_i + 0.5}\right)^{\frac{1}{2}}. \tag{6.2}$$

Note, the function $\lambda_i(\beta)$ is monotonously increasing over $[\beta_-, \beta_+]$.

Now let us look at the behavior of the solution of Linearized Problem:

$$g(x) - \langle \nabla f(y^i), x\rangle \downarrow \min, x \in S, \tag{PL_i}$$

when the parameter β, $f(y^i) = \beta + \zeta$, is moving over $[\beta_-, \beta_+]$.

Further, let us denote (the neighborhood of a vertex i)

$$V_i = \{j \in \{1, \dots, n\} \;/\; a^G_{ij} = 1\}.$$

PROPOSITION 4. *Let z be a local maximum of $F(\cdot)$ over S, $\zeta = F(z)$, and u^i be a solution to Linearized Problem (PL_i) with y^i satisfying (6.1), (6.2) and verifying*

$$g(y^i) \leqslant \beta \tag{6.3}$$

for some $\beta \in [\beta_-, \beta_+]$.

In this case, the structure of the solution u^i is as follows.

(a) If $\lambda_i^- \leqslant \lambda_i < \lambda_i^+$, where

$$\lambda_i^- = \left(\frac{d_i + 0.5}{d_i} + \sum_{j\in V_i} 1/d_i\right)^{-1}, \tag{6.4}$$

$$\lambda_i^+ = \frac{d_i}{d_i - 0.5}, \tag{6.5}$$

then $u^i_j > 0$ only for $j \in V_i$ and $j = i$.

Besides,

$$u^i_i = (1 + \lambda_i(d_i - 0.5)c_i^0)/d_i c_i; \tag{6.6}$$

$$u^i_j = (1 - \lambda_i(d_i - 0.5)/d_i)/d_j c_i; \tag{6.7}$$

where

$$c_i^0 = \sum_{j\in V_i} 1/d_j; \quad c_i = c_i^0 + 1/d_i. \tag{6.8}$$

(b) If $\lambda_i \geqslant \lambda_+$, then $u^i_i > 0$ and $u^i_j = 0\ \forall j \neq i$.

Proof. 1) Due to (6.1) and (6.2), respectively, one has

$$g(y^i) = \frac{1}{2} d_i \lambda_i^2, \quad \beta = \lambda_i^2 \frac{2d_i + 1}{4} - \zeta.$$

Then, the inequality $g(y^i) \leqslant \beta$ is equivalent to

$$\lambda_i \geqslant 2\sqrt{\zeta}. \tag{6.9}$$

It is well-known that if z is a local solution then $\zeta = \frac{1}{2} - \frac{1}{4\mathcal{K}}$, where $\mathcal{K}$ is the cardinality of the maximal clique corresponding to z. Hence, $\zeta > \frac{1}{4}$, and from (6.9) it follows

$$\lambda_i > 1, \quad i = 1, \ldots, n. \tag{6.10}$$

(2) Without loss of generality one can set $i := 1$, $r := \nabla f(y^1) = \lambda_1 A_1 e^1 = \lambda_1 a^1$. Then, one gets

$$a_j^1 = \begin{cases} d_1 + 0.5, & \text{if } j = 1, \\ 1, & \text{if } j \in V_1, \\ 0, & \text{otherwise} \end{cases}$$

Let $V_1 = \{2, \ldots, p\}$, where $p = d_1 + 1$. Then we have

$$r_1 \geqslant r_2 \geqslant \ldots \geqslant r_n, \tag{6.11}$$

or more exactly

$$r_2 = \ldots = r_p = \lambda_1, \quad r_{p+1} = \ldots = r_n = 0. \tag{6.12}$$

(3) On the other hand, from Lemmas 1 and 2 we have only three alternatives:
(i) $u_j > 0, \; j = 1, \ldots, n;$
(ii) $u_j > 0, \; j = 1, \ldots, p; \; u_j = 0, \; p < j \leqslant n.$
(iii) $u_1 > 0, \; u_j = 0, \; j \geqslant 2.$
(i) In this case due to Proposition 1 (see (4.6)) the following system of inequalities takes place:

$$1 + \sum_{s=1}^{p} (r_j - r_s)/d_s > 0, \quad j = 1, \ldots, n. \tag{6.13}$$

Due to (6.11)–(6.12) the system (6.13) is equivalent to only one inequality

$$1 - \sum_{s=1}^{p} r_s/d_s > 0,$$

or, which is the same,

$$1 - \lambda_1 \left((d_1 + 0.5)/d_1 + \sum_{s=2}^{p} 1/d_s \right) > 0.$$

So, it is clear that the case i) takes place iff $0 < \lambda_1 < \lambda_1^-$, where λ_1^- is defined in (6.4). But it can be easily seen, that $\lambda_1^- < 1$. So, due to (6.10) he case i) never takes place.

(ii) As above, due to (4.6) we have the system of inequalities

$$1 + \sum_{s=1}^{p} (r_j - r_s)/d_s > 0, \quad j = 1, \dots, p.$$

which is equivalent to one inequality

$$1 + \sum_{s=1}^{p-1} (r_p - r_s)/d_s > 0.$$

Furthermore, since

$$1 + \sum_{s=1}^{p-1} (r_p - r_s)/d_s = 1 + (r_p - r_1)/d_1 = 1 - \lambda_1 ((d_1 - 0.5)/d_1) = 1 - \lambda_1/\lambda_1^+,$$

it is clear, that the case (ii) takes place when $\lambda_1 < \lambda_1^+$ (sf. (6.5)). Note that $\lambda_1^+ > 1$, therefore the case a) is possible.

(iii) In this case it follows from (4.5)

$$1 + (r_2 - r_1)/d_1 \leqslant 0,$$

whence one deduces $\lambda_1 \geqslant \lambda_1^+$. □

It is clear that there is no need to seek y^i in the form (6.1), and one can avoid the verification of (6.3), since we have the inequality

$$\frac{1}{2} d_i \lambda_i^2 \leqslant \beta$$

instead. In the light of the preceding results, one can transform $\Re$-strategy into the form, called in the sequel the $\Re M$-algorithm. The latter differs from $\Re$-strategy (see Section 3) only by Steps 3 and 4.

Step 3. Set $\lambda_i = \lambda_i(\beta)$ according to (6.2).

Step 4. Introduce the set

$$I_k = I_k(\beta) = \{i \in \{1, ..., N_k\} \;/\; \frac{1}{2} d_i \lambda_i^2 \leqslant \beta, \;\; \lambda_i < \lambda_i^+\}.$$

$\Re M$-algorithm was coded in the Borland Pascal and was tested on PC Pentium 166 MMX. The results of the computational experiments are summarized in Table 2, where $\mathcal{K}_*$ is the corresponding cardinality of a maximum clique. Since for all considered examples the obtained cardinality of the maximal clique turned out to be equal to the maximum clique cardinality, we decided not to place the column $\mathcal{K}$ into the table. Further, St is the number of local maximum obtained and PL stands for the number of the Linearized Problem solved during the simulation. As

Table 2.

Graph	$\mathcal{K}_*$	St			PL			Time (s)		
		$\Re$	$\Re M$	$\Re D$	$\Re$	$\Re M$	$\Re D$	$\Re$	$\Re M$	$\Re D$
data17_1	5	1	1	1	201	85	51	0.22	0.16	0.05
data17_2	4	2	2	2	44	19	49	0.11	0.11	0.06
data20_2	5	1	1	1	163	41	60	0.33	0.16	0.06
data20_2	4	2	2	2	177	39	58	0.38	0.16	0.11
data25_1	5	2	2	2	227	49	73	0.55	0.28	0.16
data25_2	5	2	2	2	213	35	73	0.61	0.27	0.17
data30_1	4	2	2	2	101	13	88	0.49	0.28	0.27
data30_2	4	1	1	1	303	52	90	0.88	0.39	0.27
data35_1	5	2	2	2	914	35	103	3.68	0.55	0.49
data35_2	5	2	2	2	919	35	103	3.68	0.55	0.50
data40_1	6	1	1	1	526	30	120	2.58	0.61	0.60
data40_2	5	2	2	2	548	40	118	2.64	0.82	0.60
data40_3	5	2	2	2	548	40	120	2.64	0.82	0.66
data40_4	5	3	3	3	1370	53	120	5.93	1.10	0.65
data45_1	6	2	2	3	544	45	131	3.46	1.05	0.83
data45_2	5	2	2	3	1759	28	135	8.62	0.77	0.71
data50_1	6	1	1	1	661	42	150	4.89	1.16	1.10
data50_2	7	2	2	2	793	41	148	6.32	1.32	1.04
data50_3	6	1	1	1	673	42	150	5.05	1.15	1.16
data50_4	6	3	3	3	769	57	148	6.05	2.03	1.10
data50_5	6	3	3	3	811	61	147	6.31	2.09	1.05
data50_6	6	3	3	3	787	56	147	6.10	1.92	1.10
data50_7	7	3	3	4	821	43	145	6.37	1.65	0.99

one can see from Table 2, the preceding analytical investigations have no impact on the obtained solutions, which are the same for $\Re$- and $\Re M$-algorithms. The number St is also without change.

Instead, the number PL of Linearized Problems solved by $\Re M$ is considerably reduced w.r.t. $\Re$-algorithm. For instance, in the example 'data45_2.clq' PL decreased from 1759 to 28. The solving time also changes correspondingly.

To summarize, one can say that on the average $\Re M$-algorithm allows to throw away 9 of 10 PL to be solved during the unidimensional maximization $\eta_k(\beta)$ over $[\beta_-, \beta_+]$.

7. Further improvement of $\Re$-strategy and testing on DIMACS benchmark graphs

Impressed by the improvement which $\Re M$-strategy gives comparing with the starting version of Global Search, we decided to work on the scheme refinement. Using the inequality (6.3) and Proposition 4 we can estimate the boundary for λ_i as follows

$$\gamma \leqslant \lambda_i \leqslant \lambda_i^+,$$

where

$$\gamma = \gamma(\zeta) \overset{\triangle}{=} 2\sqrt{\zeta} \tag{7.1}$$

Then the following question naturally arises. May one avoid the choice of the parameter $\beta \in [\beta_-, \beta_+]$ by replacing it with a direct choice of λ_i? When such λ_i is known, β can be calculated as follows

$$\beta = \lambda_i^2(d_i + 0.5)/2 - \zeta. \tag{7.2}$$

Analysing $\Re M$-algorithm, we see that one needs the value of β only on step 7 in order to find $w^i : f(w^i) = \beta + \zeta_k$. Howerver, for quadratic cases the level problem is analytically solvable, and according to Strekalovsky (1993, 2000), we have

$$w^i = \mu_i v^i = \left(\frac{\beta + \zeta}{f(v^i)}\right)^{\frac{1}{2}} v^i = \lambda_i \left(\frac{d_i + 0.5}{2f(v^i)}\right)^{\frac{1}{2}} v^i. \tag{7.3}$$

Thus, it is possible to carry out Steps 0–7 without knowing the value of β. But the crucial obstacle is the necessity to change the stopping criterion on Step 8 (cf.(3.7)). To get over, observe, that in reality there is no need to know the value of $\eta_k(\beta)$, but only the sign. Therefore, we have organized a grid on the segment $[\gamma(\eta_k), \lambda_i^+]$ putting $\Delta\lambda_i = (\lambda_i^+ - \gamma(\zeta_k))/m$ for $m = 2,\ m = 3$, and as a result, obtained the following algorithm.

```
Step 0. Set $k := 0,\ x^k := x^0$.
Step 1. Starting at $x^k \in S$ obtain by $C$-procedure a point $z^k \in S$ of local maximum to $(P)$. Set $\zeta_k := F(z^k)$.
Step 2. Compute $\gamma = \gamma(\zeta_k)$ and $\lambda_i^+$ according to (6.5) and (7.1).
Step 3. Choose an integer $m > 1$. Set $\Delta\lambda_i := (\lambda_i^+ - \gamma)/m,\ l := 0$.
Step 4. Set $\lambda_i := \gamma + l\Delta\lambda_i$.
Step 5. Compute the solution $u^i \in S$ by (6.6)–(6.8).
Step 6. Starting at $u^i$ obtain by $C$-procedure a point $v^i \in S$ of local maximum to $(P)$.
Step 7. Compute $\beta$ and $w^i$ according to (7.2) and (7.3), respectively.
Step 8. If $\langle \nabla f(w^i), v^i - w^i \rangle + \beta - g(v^i) > 0$, then set $k := k + 1,\ z^k := v^i,\ \zeta_k := F(z^k)$ and go to Step 10.
```

Table 3.

Graph	n	Dens	$\mathcal{K}$	$\mathcal{K}_*$	Rel %	St	Time (h:min:s)
MANN_a9	45	0.9273	16	16	0	1	00:00.99
hamming6_2	64	0.9048	32	32	0	1	00:03.13
hamming6_3	64	0.3492	4	4	0	1	00:00.33
hamming8-2	256	0.9686	128	128	0	1	11:43.70
hamming8-4	256	0.6392	16	16	0	1	03:25.53
johnson8_2_4	28	0.5556	4	4	0	1	00:00.11
johnson8_4_4	70	0.7681	14	14	0	1	00:03.41
johnson16_2_4	120	0.7647	8	8	0	1	00:21.37
keller4	171	0.6491	11	11	0	2	00:45.97
c_fat200-1	200	0.0771	12	12	0	1	00:00.44
c_fat200-2	200	0.1626	24	24	0	1	00:01.76
c_fat200-5	200	0.4258	58	58	0	1	00:17.41
san200_0.7_1	200	0.7000	30	30	0	2	01:49.79
san200_0.7_2	200	0.7000	18	18	0	4	01:54.03
san200_0.9_1	200	0.9000	70	70	0	2	03:42.34
san200_0.9_2	200	0.9000	60	60	0	5	03:49.59
san200_0.9_3	200	0.9000	44	44	0	2	03:52.56
sanr200_0.7	200	0.6969	18	18	0	3	01:48.14
sanr200_0.9	200	0.8976	41	$\geqslant 42$	–	4	03:40.36
brock200_1	200	0.7454	20	21	5	3	02:11.76
brock200_2	200	0.4963	11	12	8	4	00:40.32
brock200_3	200	0.6054	14	15	7	4	01:12.94
brock200_4	200	0.6577	15	17	12	2	01:30.68
p_hat300-1	300	0.2438	8	8	0	3	00:37.29
_hat300-2	300	0.4889	25	25	0	6	04:05.13
p_hat300-3	300	0.7445	34	36	6	5	11:44.20

Step 9. If $l < (m-1)$, set $l := l+1$, and go to Step 4.
Step 10. If $i < n$, then set $i := i+1$, and go to Step 2.
Step 11. If $i = n$, then Stop. □

In the sequel we call this variant of $\Re$-strategy $\Re D$-algorithm. $\Re D$ was coded and tested in the same manner as $\Re M$. To estimate the efficiency of the proposed approach, extensive simulations were carried out, first, on the test examples and, second, on DIMACS benchmark graphs. Table 3 contains the results of our testing for $\Re M$- and $\Re D$-algorithms. One stresses that $\Re D$-algorithm also obtained the global solution in all the examples.

Table 4.

Graph	n	Dens	$\mathcal{K}$	$\mathcal{K}_*$	Rel (%)	St	Time (h:min:s)
MANN_a27	378	0.9901	125	126	1	1	1:27:21.05
johnson32-2-4	496	0.8788	16	16	0	1	1:43:19.66
p_hat500-1	500	0.2531	9	9	0	3	0:04:15.57
p_hat500-2	500	0.5046	35	36	3	6	0:33:45.81
p_hat500-3	500	0.7519	49	$\geqslant 49$	–	3	0:26:41.88
p_hat700-1	700	0.2493	11	11	0	5	0:17:58.12
p_hat700-2	700	0.4976	44	44	0	4	2:12:25.07
p_hat700-3	700	0.7480	62	$\geqslant 62$	–	7	6:08:32.32
keller_5	776	0.7515	25	27	1	7	8:22:29.07
c_fat500-1	500	0.0357	14	14	0	1	0:00:02.42
_fat500-2	500	0.0733	26	26	0	1	0:00:09.17
c_fat500-5	500	0.1859	64	64	0	1	0:01:06.68
c_fat500-10	500	0.3738	126	126	0	1	0:06:36.94
san400_0.5_1	400	0.5000	13	13	0	2	0:10:58.94
san400_0.7_1	400	0.7000	40	40	0	3	0:28:49.71
san400_0.7_2	400	0.7000	30	30	0	2	0:28:09.01
san400_0.7_3	400	0.7000	19	22	14	4	0:28:53.78
san400_0.9_1	400	0.9000	100	100	0	5	1:00:28.49
sanr400_0.5	400	0.5011	12	13	8	3	0:10:10.34
sanr400_0.7	400	0.7001	20	$\geqslant 21$	–	2	0:27:03.43
brock400_1	400	0.7492	24	27	11	4	0:34:10.59
brock400_2	400	0.7492	24	29	17	3	0:34:30.37
brock400_3	400	0.7479	24	31	23	4	0:33:52.41
brock400_4	400	0.7489	24	33	27	3	0:34:05.37
brock800_1	800	0.6493	21	23	9	4	6:11:13.76
rock800_2	800	0.6513	20	24	17	3	6:10:06.53
brock800_3	800	0.6487	20	25	20	4	6:08:53.25
brock800_4	800	0.6497	20	26	23	4	6:13:03.25

Note, in Tables 3 and 4 Rel stands for

$$\frac{\mathcal{K}_\star - \mathcal{K}}{\mathcal{K}_\star} \times 100. \tag{7.4}$$

Recall that in all examples $\Re D$-algorithm was run by starting the process from the vector $x(0) = (1/n, \dots, 1/n)^T$, which corresponds to the barycenter of the domain S.

Table 5.

Graph	n	Dens	$\mathcal{K}_*$	$\mathcal{K}$			
				COM	M/S	CBH	$\Re D$
c_fat200-1	200	0.077	12	12	8	12	12
c_fat200-2	200	0.163	24	24	24	24	24
c_fat200-5	200	0.426	58	58	58	58	58
c_fat500-1	500	0.036	14	14	14	14	14
c_fat500-2	500	0.073	26	26	26	26	26
c_fat500-5	500	0.186	64	64	64	64	64
c_fat500-10	500	0.374	126	126	126	126	126
p_hat300-1	300	0.244	8	6	6	8	8
p_hat300-2	300	0.489	25	22	24	25	25
p_hat300-3	300	0.744	36	32	33	36	34
p_hat500-1	500	0.253	9	8	8	9	9
p_hat500-2	500	0.505	36	33	35	35	35
p_hat500-3	500	0.752	$\geqslant 49$	47	48	49	49
p_hat700-1	700	0.249	11	7	9	11	11
p_hat700-2	700	0.497	44	43	43	44	44
_hat700-3	700	0.748	$\geqslant 62$	57	59	60	62

Table 6.

Graph	n	Dens	$\mathcal{K}_*$	$\mathcal{K}$			
				COM	M/S	CBH	$\Re D$
san200_0.7_1	200	0.700	30	15	15	15	30
san200_0.7_2	200	0.700	18	12	12	12	18
san200_0.9_1	200	0.900	70	45	45	46	70
san200_0.9_2	200	0.900	60	36	35	36	60
san200_0.9_3	200	0.900	44	32	33	30	44
san400_0.5_1	400	0.500	13	7	7	8	13
san400_0.7_1	400	0.700	40	20	20	20	40
san400_0.7_2	400	0.700	30	15	15	15	30
san400_0.7_3	400	0.700	22	12	12	14	19
san400_0.9_1	400	0.900	100	40	55	50	100
sanr200_0.7	200	0.697	18	14	16	18	18
sanr200_0.9	200	0.898	$\geqslant 42$	37	40	41	41
sanr400_0.5	400	0.501	13	11	11	12	12
sanr400_0.7	400	0.700	$\geqslant 21$	18	18	20	20

Table 7.

Graph	n	Dens	$\mathcal{K}_*$	$\mathcal{K}$			
				COM	M/S	CBH	$\Re D$
MANN_a9	45	0.927	16	12	12	16	16
MANN_a27	378	0.990	126	117	117	121	125
keller4	171	0.649	11	7	7	10	11
keller_5	776	0.751	27	15	15	21	25
brock200_1	200	0.745	21	17	18	20	20
brock200_2	200	0.496	12	8	8	12	11
brock200_3	200	0.605	15	9	10	14	14
brock200_4	200	0.658	17	12	13	16	15
brock400_1	400	0.748	27	21	21	23	24
brock400_2	400	0.749	29	20	22	24	24
brock400_3	400	0.748	31	18	20	23	24
brock400_4	400	0.749	33	19	21	24	24
brock800_1	800	0.649	23	16	17	20	21
brock800_2	800	0.651	24	15	17	19	20
brock800_3	800	0.649	25	16	18	20	20
brock800_4	800	0.650	26	15	17	19	20

As one can see from Tables 2–4 the results obtained are rather encouraging since the average solving time is really reduced (2–3 times w.r.t. $\Re$-algorithm, Table 2).

Therefore, we decided to test $\Re D$-algorithm on DIMACS benchmark graphs of dimension up to 800. As for the solving time, for the most of the problems of size up to 500 it varies from 30 to 40 min on the average, while the maximal time is 1 h. 43 min. For the size of 700–800 the solving time balances between 17 min and 8 h 22 min, which is apparently suitable considering the PC computer used.

As the first conclusion on disadvantages of $\Re D$-algorithm, one may say, that $\Re D$ did not find the global solution in the dense graphs with relatively small cardinality of maximum clique (Tables 3 and 4).

In order to compare the computational efficiency of the approach based on GOC with other approaches, we used the paper of Bomze et al. (1997), where one can find the computational results of solving MCP by three different approaches. Besides, all three methods, as well as ours, used a reducing of the combinatorial problem to a continuous quadratic maximization over the canonical simplex S.

The first method is due to Bomze (1997) and was denoted by COM. The second one (M/S) is due to Pelillo (1995) and based upon the non-regularized version of Motzkin-Straus relaxation. Finally, the third approach (CBH, [6]) is developed upon the powerful ‘continuous based heuristic.’

As Tables 5–7 show, the computational results obtained by $\Re D$ are quite encouraging since there are no examples where $\Re D$-algorithm conceded to COM and M/S procedures, while there are only three examples in which $\Re D$ did worse than CBH.

We have to point out (Table 6), that for Sanchis' graphs $\Re D$ was successful to reach the maximal cliques of cardinality almost 1.5–2 times as large as those found by other methods.

So, as to attainability of global solutions, $\Re D$-algorithm shows itself as rather competitive w.r.t. the procedures based upon different ideologies. Finally, we guess that the possibilities of $\Re D$ are not exhausted.

8. Conclusion

In this paper we considered the well-known combinatorial problem of finding a maximum clique in the regularized continuous form due to Motzkin/Straus/Bomze.

For solving the problem we applied an approach based on Global Optimality Conditions for d.c. maximization.

Developing the proposed Global Search Strategy, we obtained almost discrete version of GS-algorithm.

The extensive computational experiments were carried out on the test examples and the DIMACS benchmark graphs. The obtained computational results stimulate the future investigations.

Acknowledgement

The investigation was carried out under financial support of RFBR (the grant number 98-01-00043

References

1. Bazaraa, M.S. and Shetty, C.M. (1979), *Nonlinear Programming Theory and Algorithms*, John Wiley and Sons, New York.
2. Bomze, I. (1997), Evolution towards the maximum clique, *Journal of Global Optimization*, **10**, 143–164.
3. Bomze, I.M., Budinich, M., Pardalos, P.M. and Pelillo, M. (1999), The maximum clique problem, In: Du, D.-Z. and Pardalos, P.M. (eds.), *Handbook of Combinatorial Optimization*, Suppl. Vol. A., Kluwer Academic Publishers, Boston, pp. 1–74.
4. Bomze, I.M., Pelillo, M. and Giacomini, R. (1997), Evolutionary Approach to the Maximum Clique Problem: Empirical Evidence on a Large Scale, In: Bomze, I.M., Csendes, T., Horst, R. and Pardalos, P.M. (eds.), *Developments in Global Optimization*, 18, Kluwer Academic Publishers, Dordrecht, pp. 95–108.
5. Garey, M. and Johnson, D. (1979), *Computer and Intractability, A Guide to The Theory of NP-Completeness*, Freeman, San Francisco.
6. Gibbons, L.E., Hearn, D.W. and Pardalos P.M. (1995), A continuous based heuristic for the maximum clique problem, In: Johnson, D.S., Trick, M. (eds.), *Clique, Graph Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, 26, 103–124.

7. Horst, R., Pardalos, P.M. and Thoai, V. (1995), *Introduction to Global Optimization*, 3, Kluwer, Dordrecht.
8. Motzkin, T.S. and Straus, E.G. (1965), *Maxima for graphs and a new proof of a theorem of Turán*, Cand. J. Math., 17, 533–540.
9. Pelillo, M. (1995), Relaxation labeling networks for the maximum clique problem, *Journal of Artificial Neural Networks*, 2, 313–327.
10. Pelillo, M. and Jagota A. (1995), Feasible and infeasible maxima in a quadratic program for maximum clique, *Journal of Artificial Neural Networks*, 2, 411–420.
11. Strekalovsky, A.S. (1997), On Global Optimality Conditions for D.C. Programming Problems, Irkutsk University Press, Irkutsk.
12. Strekalovsky, A.S. (1993), The search for a global maximum of a convex functional on an admissible set, *Comput. Math. and Math. Physics*, 33, 315–328, Pergamon Press.
13. Strekalovsky, A.S. and Tsevendorj, I. (1998), Testing the $\mathfrak{R}$-strategy for a Reverse Convex Problem, *Journal of Global Optimization* 13, 61–74.
14. Kuznetsova, A.A., Strekalovsky A.S. and Tsevendorj I. (1999), An approach to the solution of integer optimization problem, *Comput. Math. and Math. Physics*, 39, 6–13.
15. Strekalovsky A.S., (2000) One way to Construct a Global Search Algorithm for d.c. Minimization Problems In: Pillo, G. Di., Giannessi, F. (eds.), *Nonlinear Optimization and Related Topics*, **36**, Kluwer, Dordrecht, pp. 429–443.