

# A Continuation Approach Using NCP Function for Solving Max-Cut Problem\*

Xu Fengmin<sup>†</sup>   Xu Chengxian<sup>‡</sup>   Ren Jiuquan<sup>§</sup>

## Abstract

A continuous approach using NCP function for approximating the solution of the max-cut problem is proposed. The max-cut problem is relaxed into an equivalent nonlinearly constrained continuous optimization problem and a feasible direction method without line searches is presented for generating an optimal solution of the relaxed continuous optimization problem. The convergence of the algorithm is proved. Numerical experiments and comparisons on some max-cut test problems show that we can get the satisfactory solution of max-cut problems with less computation time. Furthermore, this is the first time that the feasible direction method is combined with NCP function for solving max-cut problem, and similar idea can be generalized to other combinatorial optimization problems.

**Keywords:** Max-Cut problem, Feasible direction algorithm, NCP function, Continuation approach, Convergence.

**AMS subject classification :** 90C22, 90C25

## 1 Introduction

The max-cut problem is a discrete optimization problem on undirected graphs with weighed edges. Given a undirected graph, the problem is to find a partition of the set of nodes into two parts while maximizes the sum of the weighes on the edges that are cut by the partition. This problem is of fundamental importance

---

\*This work is supported by National Key Laboratory of Mechanical Systems and National Natural Key Product Foundations of China 10231060 and 10671152.

<sup>†</sup>Xu Fengmin, School of Electronic and Information Engineering, Xi'an Jiaotong University, xi'an, 710049, P.R.China, e-mail: fengminxu@mail.xjtu.edu.cn;

<sup>‡</sup>Xu Chengxian, Department of mathematics of Faculty of Science, Xi'an Jiaotong University, xi'an, 710049, P.R.China, e-mail: mxxu@xjtu.edu.cn.

<sup>§</sup>Ren Jiuquan, Department of mathematics of Faculty of Science, Xi'an Jiaotong University, xi'an, 710049, P.R.China, e-mail:jqren@mail.xjtu.edu.cn.

in combinatorial optimization and has wide applications in network, statistical physics and VLSI designs.

Let  $G(V, E)$  be a given undirected graph with  $n$  nodes where  $V$  and  $E$  are the sets of nodes and edges in the graph. Let  $W = (w_{ij})_{n \times n}$  be the symmetric weighted adjacency matrix of the graph with  $w_{ij} \neq 0$  for  $(i, j) \in E$  and  $w_{ij} = 0$  for  $(i, j) \notin E$ . Let  $L = \frac{1}{4}(\text{Diag}(We) - W)$  denote the Laplacian matrix associated with the graph ( $L \succeq 0$ ), where the linear operator  $\text{Diag}$  returns a diagonal matrix with diagonal entry obtained by the corresponding entry in the vector, and  $e$  denotes the vector of all ones. Let the vector  $x \in \{\pm 1\}^n$  represent any cut in the graph via the interpretation that the sets  $\{i : x_i = 1\}$  and  $\{i : x_i = -1\}$  form a partition of the node set of the graph. It is well known that the formulations over  $\{0, 1\}^n$  and  $\{\pm 1\}^n$  are equivalent, see for example Helmbert et al (1995).

Following Mohar and Poljak (1998), we can formulate the max-cut problem as:

$$(MC) : \begin{cases} \mu^* = \text{Max } x^T L x \\ \text{s.t. } x_i \in \{-1, 1\}, \quad i = 1, \dots, n. \end{cases}$$

In this paper  $\mu^*$  denotes the optimal value of the max-cut problem. Note that the following model has the different objective value with no effects on the optimal solutions of  $(MC)$  :

$$\begin{cases} \text{Max } x^T (L + \sigma I) x \\ \text{s.t. } x_i \in \{-1, 1\}, \quad i = 1, \dots, n. \end{cases}$$

Without loss of generality, we will assume, in the rest of the paper, that  $L = L + \sigma I$  is positive definite and  $L_{ii} > 0, i = 1, \dots, n$ .

Further we observe that  $x \in \{\pm 1\}^n$  if and only if  $x_i^2 = 1, i = 1, \dots, n$ . This immediately yields our second formulation for the max-cut problem:

$$(MC1) : \begin{cases} \mu^* = \text{Max } x^T L x \\ \text{s.t. } x_i^2 = 1, \quad i = 1, \dots, n. \end{cases}$$

$(MC1)$  is a well-known and commonly used formulation for the max-cut problem.

Let  $X = xx^T$ , then  $x^T L x = L \cdot X$ , where  $L \cdot X$  denotes the Frobenius inner product of two matrices. The relaxed semidefinite programming of the max-cut problem is given by

$$(SDP) : \begin{cases} \text{Max } L \cdot X, \\ \text{s.t. } \text{diag}(X) = e, \\ X \succeq 0, \end{cases}$$

where  $\text{diag}(X)$  denotes the vector in  $R^n$  consisting of the diagonal elements of the matrix  $X$ ,  $X \succeq 0$  means that  $X$  is symmetric and positive semi-definite. It's well known that a SDP problem can be solved by using the interior-point

algorithms in polynomial time (Alizadeh et al (1998)) and (Helmberg et al (1996)). However, the solution of the relaxed semi-definite programming only provides a suboptimal solution of the max-cut problem with a better upper bound. It is known that the max-cut problem is NP-hard, and algorithms are available to find its approximate solutions. Typical approaches to solve such problems are to find a solution within approximation factor  $\rho$ . Sahni and Gonzales (1976) reported an approximation algorithm with  $\rho = \frac{1}{2}$  for max-cut problems. Since then, various approximation algorithms for max-cut problems are proposed. Among them the most famous is the 0.87856 randomization algorithm proposed by Goemans and Williamson (Goemans and Williamson (1995)). Their algorithm relaxes the max-cut problem as a semi-definite programming problem, and the resulting problem is then solved using any existing semi-definite programming algorithms, for example, interior algorithms. The strengthened semidefinite programming relaxation and rank two relaxation of max-cut problems are modifications of Goemans and Williamson's work (Goemans and Williamson (1995)), for example (Anjos and Wolkowicz (1999)) and (Burer and Monteiro (2001)).

In this paper a continuation approach using NCP function for solving max-cut problems is proposed. Unlike the available relaxation methods, NCP function is employed to convert the max-cut problem to a continuous nonlinear programming, and then the resulting nonlinear programming problem is solved using the feasible direction method without line search. The convergence property of the proposed algorithm is studied, and numerical experiments and comparisons on some problems generated by **rudy** are made to show the efficiency of the proposed algorithm on both the CPU times and solutions.

The rest of the paper is organized as follows. Section 2 gives the continuous relaxation of max-cut problems by applying the NCP continuation function. The feasible direction method without line searches for the solution of the resulting nonlinear programming problem is given in section 3. The convergence property and the finite termination property of the feasible direction method are also studied in section 3. In section 4 numerical experiments and comparisons on some well known test problems are reported. The weighted adjacency matrices of these test max-cut problems are generated by **rudy**. Numerical experiments show that the proposed continuation algorithm generates satisfactory solutions with less computational times on these test problems.

## 2 The Continuation Model of the Max-Cut Problem

In this section, we present the continuation model of the max-cut problem using NCP function.

The NCP function is given by

$$\phi_F(a, b) = \sqrt{a^2 + b^2} - a - b$$

(See Fischer-Burmeister, 1992). Then

$$\phi_F(a, b) = 0 \Leftrightarrow ab = 0, a \geq 0, b \geq 0.$$

Let  $a = 1 - x_i$ ,  $b = 1 + x_i$ , then the problem (MC1) can equivalently be described by the following nonlinear programming problem

$$(NP) : \begin{cases} \text{Max } x^T Lx \\ \text{s.t. } \phi_F(1 - x_i, 1 + x_i) = 0, \quad i = 1, \dots, n. \end{cases}$$

Now we focus on the solution of problem (NP). At first, let us consider the following equivalent nonlinear programming

$$(NP1) : \begin{cases} \mu^{**} = \text{Max } x^T Lx \\ \text{s.t. } \phi_F(1 - x_i, 1 + x_i) \leq 0, \quad i = 1, \dots, n, \\ \|x\| = \sqrt{n}. \end{cases}$$

Let

$$F_1 = \{x | \phi_F(1 - x_i, 1 + x_i) = 0, i = 1, \dots, n\},$$

$$F_2 = \{x | \phi_F(1 - x_i, 1 + x_i) \leq 0, i = 1, \dots, n, \|x\| = \sqrt{n}\},$$

be feasible regions of problem (NP) and (NP1), respectively. The following theorem gives the equivalence of  $F_1$  and  $F_2$ .

**Lemma 2.1** *The NCP function  $\phi_F(1 - x_i, 1 + x_i) = \sqrt{(1 - x_i)^2 + (1 + x_i)^2} - 2$  for  $i = 1, 2, \dots, n$  is strictly convex for all  $x_i \in (-1, 1)$ .*

**Proof** Since the first-order derivative and the second-order derivative of the function  $\phi_F(1 - x_i, 1 + x_i)$ ,  $i = 1, 2, \dots, n$  are given by

$$\phi'_F(1 - x_i, 1 + x_i) = \frac{2x_i}{\sqrt{2x_i^2 + 2}},$$

$$\phi''_F(1 - x_i, 1 + x_i) = \frac{4}{(2x_i^2 + 2)\sqrt{2x_i^2 + 2}} > 0.$$

Since the second-order derivative of  $\phi_F(1 - x_i, 1 + x_i)$ , ( $i = 1, 2, \dots, n$ ) is positive for all  $x_i \in (-1, 1)$ , So  $\phi_F(1 - x_i, 1 + x_i)$  for  $i = 1, 2, \dots, n$  is strictly convex.  $\square$

**Theorem 2.2** *Let  $F_1, F_2$  be feasible regions of problem (NP) and (NP1), respectively. Then  $F_1 = F_2$ . That is  $\mu^* = \mu^{**}$ .*

**Proof** Obviously,  $F_1 \subseteq F_2$ .

On the other hand, Let  $x \in F_2$ , Since the definition of  $\phi_F(1 - x_i, 1 + x_i)$ , then  $-1 \leq x_i \leq 1$  and  $\|x\| = \sqrt{n}$ . Suppose  $x \notin F_1$ , there exists indices  $t$  such that  $|x_t| < 1$ . Since  $\|x\| = \sqrt{n}$ , there must exist an index  $s$  such that  $|x_s| > 1$ . This contradicts the fact that  $x \in F_2$ . Then for all  $x \in F_2$ , there is  $x \in F_1$ . So  $F_2 \subseteq F_1$ .

In conclusion, we have  $F_1 = F_2$ . So  $\mu^* = \mu^{**}$ . That is problem (NP) and (NP1) have the same optimal solution.  $\square$

Theorem 2.2 implies that the optimal solution of the problem (NP) can be obtained by solving the problem (NP1).

Now we further relax the constraints in problem (NP1) to get the following continuous nonlinear programming problem

$$(NP2) : \begin{cases} v^* = \text{Max } x^T Lx \\ \text{s.t. } \phi_n(\sqrt{n} - x_i, \sqrt{n} + x_i) \leq 0, i = 1, \dots, n, \\ \|x\| \leq \sqrt{n}, \end{cases}$$

where

$$\phi_n(\sqrt{n} - x_i, \sqrt{n} + x_i) = \sqrt{(\sqrt{n} - x_i)^2 + (\sqrt{n} + x_i)^2} - 2\sqrt{n} \quad i = 1, \dots, n.$$

Let

$$F_3 = \{x | \phi_n(\sqrt{n} - x_i, \sqrt{n} + x_i) \leq 0, i = 1, \dots, n, \quad \|x\| \leq \sqrt{n}\}$$

be the feasible region of problem (NP2). It can be observed that  $F_2 \subseteq F_3$ , and hence  $v^* \geq \mu^{**}$ , that is, the solution of the problem (NP2) can provide an upper bound on the value of the max-cut. Let  $\bar{x}$  be the optimal solution of the problem (NP2). Then  $y = \text{sign}(\bar{x})$  is a feasible point of the problem (NP1) and hence the problem (MC1). Even though this feasible solution  $y$  can not be guaranteed to be an optimal solution of the problem (MC1), numerical experiments show that such a procedure generally generates a satisfactory solution. In order to generate a tighter lower bound on the optimal value of the max-cut, a local search procedure at  $y$  is employed to either generate an improved local solution or to ensure that  $y$  is already a local solution of the max-cut problem.

The NP2 relaxation is a continuous nonlinear programming problem and any effective nonlinear optimization algorithm can be employed or modified to find its solution (See Xu CX and Zhang JZ, 2001 for quasi-Newton methods). However, unlike the semi-definite programming relaxation (SDP) that is a convex programming problem, the NP2 relaxation is a concave nonlinear programming problem, and hence, there exist local solutions. The solution obtained from any local optimization algorithm can not be guaranteed to be a global solution, and hence, it is impossible to get a performance guarantee for the solution of max-cut problems based on the NP2 relaxation. On the other hand, the number of variables in the NP2 relaxation is the same as that of the original max-cut

problem, while the SDP relaxation increases the problem variables from  $n$  to  $n^2$ . Also the characters of the NP2 relaxation can be used to design effective algorithms for the solution of the problem (NP2) so that the computational time can be greatly reduced (see next sections for the algorithm and numerical experiments).

### 3 Feasible Direction Algorithm

In this section, the feasible direction algorithm without linear search is presented for the solution of problem (NP2). The algorithm employs no line search and no calculation on matrices, and thus greatly reduces the calculation expenses. Before we derive the algorithm, some basic properties of problem (NP2) are discussed in the following lemmas.

First we recall that Lemma 2.1 implies that function  $\phi_n(\sqrt{n}-x_i, \sqrt{n}+x_i)$ , ( $i = 1, 2, \dots, n$ ) is also convex for all  $x_i \in [-\sqrt{n}, \sqrt{n}]$ , So the feasible region  $F_3$  of problem (NP2) is convex.

Let  $x^k$  be a feasible point of problem (NP2) and  $f(x) = x^T Lx$ .  $g^k = 2Lx^k$  is the gradient of objective function  $f(x)$  at point  $x^k$ . Let  $x^{k+1} = \frac{g^k \|x^k\|}{\|g^k\|}$ , then  $x^{k+1} \in F_3$ . Define  $d^k = \frac{g^k \|x^k\|}{\|g^k\|} - x^k$  as a search direction, the next lemmas show that if  $d^k = 0$ , then  $x^k$  is a KKT point of problem (NP2), and if  $d^k \neq 0$ , then  $d^k$  is a feasible ascent direction of problem (NP2) at point  $x^k$ .

**Lemma 3.1** *Suppose  $x^*$  is an optimal solution of problem (NP2), then  $x^*$  is an eigenvector of the matrix  $L$  that satisfies the constraints of problem (NP2), that is, there exists an eigenvalue  $\lambda_1$ , such  $Lx^* = \lambda_1 x^*$ .*

**Proof** Let  $x^*$  is an optimal solution of problem (NP2), then  $x^* \in F_3$ . It is clear that  $Lx^* \neq 0$ , otherwise,  $x^*$  can not be the optimal solution of the problem.

At first, we prove that all the first  $n$  constraints in (NP2) are inactive and only last constraint is active at point  $x^*$ .

The convexity of the feasible region and the convexity of the objective function imply that the optimal solution of the problem will be achieved at some extreme points of the feasible region, that is, some constraints will be active at the solution  $x^*$ .

Now, we prove that all the first  $n$  constraints in (NP2) are inactive and only the last constraint is active at point  $x^*$ .

Suppose that there exists an index,  $t$  say, satisfying  $x_t^* = \pm\sqrt{n}$  and  $x_i^* = 0$ ,  $i \neq t$ ,  $i = 1, \dots, n$ , that is, the constraint  $\phi_n(\sqrt{n} - x_t, \sqrt{n} + x_t) \leq 0$  is active

at point  $x^*$ . Define  $d = \sqrt{n} \frac{g^*}{\|g^*\|} - x^*$ , then  $d$  is feasible and

$$\nabla f(x^*)^T d = 2n \sqrt{\sum_{i=1}^n L_{ti}^2} - 2nL_{tt}.$$

It is clear that there exists at least one index  $i \neq t$  such that  $L_{ti} \neq 0$ , otherwise, node  $t$  is an isolated point of the graph. Hence  $\nabla f(x^*)^T d > 0$ , and  $d$  is a feasible ascent direction of problem (NP2) at  $x^*$ . This gives a contradiction to the fact that  $x^*$  is the optimal solution of (NP2)

It then follows from the inactivity of the first  $n$  constraints of problem (NP2) at  $x^*$  that the last constraint is active at  $x^*$ . Applying the KKT condition to problem (NP2), there exists Lagrange multiplier  $\lambda_1$  (replacing the constraint  $\|x\| = \sqrt{n}$  by  $\|x\|^2 = n$  gives no any effect to the problem) such that

$$g^* - 2\lambda_1 x^* = 0$$

holds at  $x^*$ . Then

$$g^* - 2\lambda_1 x^* = 0 \Leftrightarrow Lx^* - \lambda_1 x^* = 0.$$

This indicates that  $x^*$  is an eigenvector of the matrix  $L$ . The proof is completed.  $\square$

Lemma 3.1 shows that optimal solutions of problem (NP2) can be found from the eigenvectors of the matrix  $L$  that satisfies the constraints of problem (NP2). The algorithm proposed in this section either terminates or converges to an eigenvector of the matrix  $L$  satisfying constraints in (NP2). It can also be observed from the proof of Lemma 3.1 that the optimal solution of problem (NP2) will not change if we replace the matrix  $L$  by a positive definite matrix  $L + \sigma I$  where  $\sigma > 0$  is a constant. Therefore, in the following we will assume that the matrix  $L$  is positive definite by adding a matrix  $\sigma I$ , and hence  $g^k = 2Lx^k$  will not be zero for all  $k \geq 0$ .

**Lemma 3.2** *Let  $x^k \in F_3$ . If  $d^k = 0$ , then  $x^k$  is an eigenvector of the matrix  $L$  that satisfies the constraints of problem (NP2), that is,  $x^k$  is a KKT point of problem (NP2).*

**Proof** From the definition of  $d^k$ , we have

$$d^k = \frac{\sqrt{n}g^k}{\|g^k\|} - x^k = 0,$$

that is

$$Lx^k - \frac{\|g^k\|}{2\sqrt{n}}x^k = 0.$$

This shows that  $x^k$  is an eigenvector of the matrix  $L$  satisfying constraints of problem (NP2). This completes the proof of the lemma.  $\square$

**Lemma 3.3** *Let  $x^k \in F_3$ . Suppose  $d^k \neq 0$ , then  $d^k$  is a feasible ascent direction of problem (NP2) at  $x^k$ .*

**Proof** The feasibility of the direction  $d^k$  comes from the feasibility of points  $x^k$  and the convexity of the feasible region  $F_3$ .

$$\begin{aligned} (\nabla f(x^k))^T d^k &= (g^k)^T \left( \frac{g^k \|x^k\|}{\|g^k\|} - x^k \right) \\ &= \|g^k\| \|x^k\| - (g^k)^T x^k \geq 0 \end{aligned}$$

If  $(\nabla f(x^k))^T d^k > 0$ , then  $d^k$  is an ascent direction. If  $(\nabla f(x^k))^T d^k = 0$ , then

$$f(x^k + \alpha d^k) = f(x^k) + \alpha (\nabla f(x^k))^T d^k + \alpha^2 (d^k)^T L d^k, \quad (3.1)$$

and the positive definiteness of the matrix  $L$  also shows that  $d^k$  is ascent. The proof is completed.  $\square$

Lemma 3.3 implies that  $\alpha = 1$  is the best choice for the step length in the direction  $d^k$ . It is the reason why we adopt the above iterative format without line searches. No line search in iterations greatly reduces the computational cost, and increase the speed of the algorithm to achieve the solution.

The following theorem gives the convergence of the algorithm to KKT points of problem (NP2).

**Theorem 3.4** *Suppose  $d^k \rightarrow 0$ . Then any accumulation point  $x^*$  of the sequence  $\{x^k\}$  is an eigenvector of the matrix  $L$  that satisfies the constraints of problem (NP2), that is,  $x^*$  is a KKT point of (NP2).*

**Proof** Let  $x^*$  be an accumulation point of the sequence  $\{x^k\}$ . Without loss of generality, assume that  $x^k \rightarrow x^*$ . It follows from the definition of  $d^k$ , and the continuity of  $g(x)$ , we have

$$\lim d^k = \lim \frac{g^k \sqrt{n}}{\|g^k\|} - x^k = \frac{g^* \sqrt{n}}{\|g^*\|} - x^* = 0.$$

That is,

$$Lx^* - \frac{\|g^*\|}{2\|x^*\|} x^* = 0.$$

This shows that  $x^*$  is an eigenvector of the matrix  $L$  satisfying constraints of (NP2), and the proof is completed.  $\square$

The rest of this section is devoted to the proof of the convergence of the infinite sequence  $\{d^k\}$ , generated by the proposed algorithm, to zero vector.



**Lemma 3.5** *Let  $d^k \neq 0$ , then the following inequalities hold*

$$\lambda_{\min}(L)\|d^k\|_2^2 \leq f(x^{k+1}) - f(x^k) \leq \|g^k\|_2\|d^k\|_2 + \lambda_{\max}(L)\|d^k\|_2^2.$$

**Proof** Since  $f(x)$  is a quadratic function, we have

$$f(x^{k+1}) - f(x^k) = (g^k)^T d^k + (d^k)^T L d^k. \quad (3.2)$$

From  $(g^k)^T d^k \leq \|g^k\|_2\|d^k\|_2$  and the positive definiteness of the matrix  $L$ , we obtain

$$f(x^{k+1}) - f(x^k) \leq \|g^k\|_2\|d^k\|_2 + \lambda_{\max}(L)\|d^k\|_2^2, \quad (3.3)$$

where  $\lambda_{\max}(L)$  is the largest eigenvalue of the matrix  $L$ . Furthermore, since  $(g^k)^T d^k \geq 0$  and the property of matrix  $L$ , we have

$$f(x^{k+1}) - f(x^k) \geq \lambda_{\min}(L)\|d^k\|_2^2, \quad (3.4)$$

where  $\lambda_{\min}(L)$  is the smallest eigenvalue of the matrix  $L$ . Inequalities (3.3) and (3.4) give the conclusion of the Lemma.  $\square$

**Theorem 3.6** *If  $d^k \neq 0$  for any  $k > 0$ , then  $\|d^k\|_2 \rightarrow 0$ .*

**Proof** From Lemma 3.5, for any  $m > 0$  we have

$$\begin{aligned} \sum_{i=0}^m \|d^i\|_2^2 &\leq \frac{1}{\lambda_{\min}(L)} \sum_{i=0}^m (f(x^{i+1}) - f(x^i)) \\ &= \frac{1}{\lambda_{\min}(L)} [f(x^m) - f(x^0)] \\ &\leq \frac{1}{\lambda_{\min}(L)} (x^*)^T L x^* \\ &\leq \frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} \|x^*\|_2^2 \\ &\leq \frac{\lambda_{\max}(L)}{\lambda_{\min}(L)} n. \end{aligned}$$

That is,  $\sum_{i=0}^{+\infty} \|d^i\|_2^2$  is convergent, and hence  $\|d^k\|_2 \rightarrow 0$  holds.  $\square$

When the algorithm is implemented to solve problem (NP2), the condition  $\|d^k\| \leq \epsilon$  or  $f(x^{k+1}) - f(x^k) \leq \epsilon$  is used to terminate the iteration.

## 4 Numerical Experiments

In this section we report numerical results and comparisons to show the effectiveness and efficiency of the proposed feasible direction method. The algorithm is programmed in Matlab 6.0, and experiments are implemented on a 1.6GHz Pentium IV personal computer with 256Mb of Ram. The value  $\epsilon = 0.0001$  is used in the termination conditions  $\|d^k\| \leq \epsilon$  or  $f(x^{k+1}) - f(x^k) \leq \epsilon$ . The value  $\sigma = 10$  is selected to make sure  $L + \sigma I$  is positive definite, but the objective

function value at the termination point is still calculated as  $(x^k)^T Lx^k$  in the implementation. The initial points for all test problems are randomly generated by  $x^0 = \text{sign}(\text{rand}(n, 1))$  which satisfies  $\|x^0\| = \sqrt{n}$ , and is feasible to problem (NP2).

The set of test problems are randomly created using the procedure **rudy**, a machine independent graph generator written by Ginaldi, that generates middle or large scale test problems for max-cut (Helmberg and Rendl (2000)). Table 4.1 gives the results on 6 test problems, where Size gives the number of nodes in test graphs,  $B_{sol}$  and  $B_{time}$  that cited in Helmberg and Rendl (2000) give an upper bound of the optimal value of these max-cut problems and computation time that is generated using spectral bundle method(SB) , and GW-cut gives the results generated using Goemans and Williamson's 0.87856 randomized approximation algorithm, which generates a lower bound to the optimal value of test problems by using interior point software SDPpack (Alizadeh, Haeberly, Nayakkankuppam, Overton and Schmieta (1997), and  $f^*$  and Time(Sec) are results generated by the continuation feasible direction method. It can be observed from Table 4.1 that the proposed algorithm (FA) provides better solutions than the method GW for all this set of test problems.

Table 4.1: Comparison of solution quality

Problem	size	SB		GW-cut	FA	
		$B_{sol}$	$B_{time}$		$f^*$	Time(Sec)
G03	800	12084	4:38	10610	11368	0.341
G14	800	3192	14:30	2803	2941	1.331
G38	2000	8015	4:03:53	7037	7341	1.843
G44	1000	7028	5:06:31	6170	6423	1.332
G50	3000	5988	5:17:51	5257	5803	12.76
G52	1000	4009	5:09:02	3520	3698	0.49

Table 4.2 shows a comparison of time and result among these three algorithms, that are Rank-2 heuristic, GRASP-VNS method and random primal sampling  $S_{RP}$  with sample size(Hernn Alperin , Ivo Nowak (2002)). The results and the time of these algorithms in Table 4.2 are copied from Hernn Alperin , Ivo Nowak (2002). The CPU time is presented in  $ss$  seconds or  $ss.ddd$  seconds.

From the table 4.2 it can be observed that the continuous feasible direction method has similar performance on these large scale test problems as the rank-2 algorithm, and is better than GRASP-VNS and  $S_{RP}$  method.

Table 4.2: Comparison with other algorithm

Problem	size	$R_{SP}$		GRASP-VNS		rank-2		FA	
		$ss$	$f^*$	$ss.dd$	$f^*$	$ss.dd$	$f^*$	$f^*$	Time
G11	800	4	550	10.00	552	0.06	524	542	0.33
G12	800	3	542	9.36	532	0.06	512	532	0.34
G13	800	4	570	12.41	564	0.06	536	554	0.48
G14	800	4	3006	12.89	3040	0.09	3016	2955	0.55
G15	800	5	3002	18.09	3017	0.09	3011	2944	0.8
G22	2000	21	13193	56.98	13087	0.37	13148	13148	1.74
G24	2000	25	13165	192.81	13209	0.30	131954	13236	2.45
G32	2000	144	1346	99.91	1368	0.18	1306	1338	5.43
G34	2000	14	1334	55.22	1340	0.12	1276	1292	2.32

Table 4.3 gives comparisons of the proposed method on 4 large scale test max-cut problems with negative weights that are also generated using **rudy**. The value of  $\sigma = 100$  is selected to ensure the matrix  $L + \sigma I$  is positive definite. The efficiency of the proposed continuous feasible direction method can also be observed from Table 4.3.

Table 4.3: Max-cut problems with negative weights

Problem	size	SB		rank-2		FA	
		$B_{sol}$	$B_{time}$	$f^*$	Time	$f^*$	Time
G23	2000	14146	38:11	13197	0.37	13148	4.8230
G30	2000	4215	1:02:39	3234	0.32	3338	1.840
G31	2000	4117	26:11	3146	0.33	3221	2.56
G33	2000	1544	6:04:22	1290	0.14	1325	2.54

## 5 Conclusion and Discussion

A new continuation relaxation model for max-cut problems are proposed. NCP function is employed to relax the discrete maximum cut problem into a continuous nonlinear programming problem. Then a feasible direction method is proposed to find the solution of the resulting nonlinear programming problem. The algorithm employs only the gradient values of the objective function in the resulting programming problem and no matrix calculation, and no line searches. This greatly reduces the computational cost to achieve the solution. The convergence of the proposed algorithm to a KKT point of the nonlinear programming is proved. Numerical experiments and comparisons with Goemans and Williamson's algorithm and rank-2 algorithm on some test max-cut problems are performed to show that the proposed algorithm is efficient to get satisfactory solutions of max-cut problems.

**Acknowledgments** The authors wish to express their gratitude to the referee for very helpful and detailed comments.

## References

- [1] Alizadeh, F, Haeberly, J P and Overton, M (1998), Primal dual interior point methods for Semidefinite Programming:convergence rates,stability and numerical results, *SIAM Journal of Optimization*, 8, 746-768.
- [2] Alizadeh, F, Haeberly, J P, Nayakkankuppam, M V, Overton M L and Schmieta, S (1997), SDPpack user's guide -version 0.9Beta. Technical Report TR1997-737, Courant Institute of Mathematical Science, NYU, New York, NY, June.
- [3] Anjos, W and Wolkowicz, H (1999), A strengthened SDP relaxation via a second lifting for the Max Cut problem, Technical Report Research Report CORR 95-55, University of Waterloo, Waterloo, Onratio.
- [4] Burer, S, Monteiro, R D C and Zhang, Y (2001), Rank-two relaxation heuristics for MAX-CUT and other binary quadratic programs, *SIAM Journal on Optimization*, 12, 503-521.
- [5] Fischer, A (1992), A special Newton-type optimization method, *Optimization*, 24, 269-284.
- [6] Goemans, M X, and Williamson, D P (1995), Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *Journal of ACM*, 42, 1115-1145.
- [7] Helmberg, C and Rendl, F (2000), A spectral bundle method for semidefinite programming, *SIAM Journal of Optimization*, 10, 673-695.
- [8] Helmberg, C, Rendl, F, Vanderbei, R J and Wolkowicz, H (1996), An interior point methods for semidefinite programming, *SIAM Journal of Optimization*, 6, 342-361.
- [9] Helmberg, C, Poljak, S and Wolkowicz, W (1995), Combining semidefinite and polyhedral relaxations for interger programs, In *Interger Programming and Combinatorial optimization*, pages 124-134, Springer, Berlin.
- [10] Hernan Alperin , Ivo Nowak (2002), Lagrangian smoothing heuristic for Max-Cut , Humboldt Universitt zu Berlin, Institut fr Mathematik, technical report NR-2002-6.
- [11] Mohar, B and Poljak, S(1998), Gigenvalue in combinatorial optimization, In *combinatorial and graph theoretical problems in linear algebra*, pages 107-151, Springer, New York.
- [12] Sahni, S and Gonzalez, T (1976), P-Complete approximation problem, *Journal of ACM*, 23, 555-565.
- [13] Xu CX and Zhang JZ (2001), Survey of quasi-Newton equations and quasi-Newton methods for optimization, *Annals of Operations Research*, 103, 213-234.