Indian J. Pure Appl. Math., **41**(5): 683-700, October 2010

AN EFFICIENT LAGRANGIAN SMOOTHING HEURISTIC FOR MAX-CUT[1]

Yong Xia* and Zi Xu**

**LMIB of the Ministry of Education; School of Mathematics and System Sciences, Beihang University, Beijing*, 100083,
Peoples' Republic of China
***Department of Mathematics, Shanghai University, Shanghai*, 200444
Peoples' Republic of China
e-mail : xuzi@shu.edu.cn

(*Received* 19 *September* 2008; *after final revision* 25 *August* 2010; *accepted* 30 *August* 2010)

Max-Cut is a famous NP-hard problem in combinatorial optimization. In this article, we propose a Lagrangian smoothing algorithm for Max-Cut, where the continuation subproblems are solved by the truncated Frank-Wolfe algorithm. We establish practical stopping criteria and prove that our algorithm finitely terminates at a KKT point, the distance between which and the neighbour optimal solution is also estimated. Additionally, we obtain a new sufficient optimality condition for Max-Cut. Numerical results indicate that our approach outperforms the existing smoothing algorithm in less time.

Key words : Max-Cut; Lagrangian smoothing; Frank-Wolfe algorithm; heuristic.

[1]This work was partially supported by the fundamental research funds for the central universities under grant YWF-10-02-021 and by National Natural Science Foundation of China under grant 11001006 and partially supported by Shanghai Leading Academic Discipline Project, Project Number: S30104 and the Excellent Young Teachers Program of Shanghai Municipality (B.37-0101-08-017).

1. Introduction

Given an undirected graph $G = (V, E)$, V is the set of n nodes and E is the set of all the edges e_{ij} with nonnegative weights ω_{ij}. The Max-Cut problem consists of finding a subset of the nodes $S \subset V$ that maximizes the following cut function

$$cut(S) = \sum_{ij \in \delta(S)} \omega_{ij}, \tag{1.1}$$

where the incidence function $\delta(S) = \{ij \in E \text{ such that } i \in S \text{ and } j \notin S\}$. Let $x_i \in \{-1, 1\}$ represent whether node $i \in S$, then the Max-Cut problem can be equivalently expressed in the mathematical formulation

$$\max \quad \sum_{i<j} \omega_{ij} \frac{1 - x_i x_j}{2} \tag{1.2}$$

$$s.t. \quad x_i^2 = 1, \ i = 1, \cdots, n, \tag{1.3}$$

It is also equivalent to the following quadratic nonconvex problem,

$$\min \quad x^T A x \tag{1.4}$$

$$s.t. \quad x_i^2 = 1, \ i = 1, \cdots, n, \tag{1.5}$$

where A is a matrix with $A_{ij} = \frac{1}{4}\omega_{ij}$ for $i \neq j$ and $A_{ii} = 0$.

Since the Max-Cut problem is NP-hard, many heuristics or approximation algorithms have been proposed. Typical approaches to solving this problem are to find a solution at an approximation factor ρ. Among which the most famous is the algorithm proposed by Goemans and Williamson [8] with $\rho = 0.87856$. The algorithm rewrites the Max-Cut problem as a semidefinite programming problem and relaxes it by removing the rank-1 constraint, then an approximate solution of the max-cut problem is generated from the optimal solution of the relaxed semidefinite programming problem (SDP) using some rounding techniques. Bertsimas and Ye [3] also got a 0.87856 solution by using a different rounding approach. Unfortunately, solving large scale semidefinite problems requires quite a long time. The rank-two relaxation (Burer, Menteiro and Zhang [5]) of Max-Cut problem is the modification of Goemans and Williamson's work. In their heuristic, they relax the binary vector into a vector of angles, and work with an angular representation of the cut. Besides, Festa *et al.* [6] proposed heuristics based on greedy randomized adaptive search and variable neighbourhood search (GRASP-VNS) that work well for the Max-Cut problem.

Another promising approach to solve the Max-Cut problem is the continuation method. Recently, Alperin and Nowak [1] proposed an efficient Lagrangian smoothing truncated projected gradient (LS-TPG) method. They formulated the Max-Cut problem to be a parametric optimization problem defined as a convex combination between a Lagrangian relaxation and the original problem, and used the truncated projected gradient method to solve the subproblems. Their numerical experiments showed that the LS-TPG algorithm outperforms the 0.87856 SDP (Goemans and Williamson [8]) method and is competitive with GRASP-VNS method Festa *et al.* [6] for many typical test problems, but it performs worse than rank 2 (Burer, Monteiro and Zhang [5]) method. A more detailed presentation of the LS-TPG algorithm will be given in Section 2.

In this article, a more efficient Lagrangian smoothing method is proposed for Max-Cut. The continuation subproblems are solved by the truncated Frank-Wolfe algorithm. We establish practical stopping criteria and prove that our algorithm finitely terminates at a KKT point. We also estimate the distance between the objective function of our solution and that of the neighbor optimal solution. It implies that the smaller the outer iteration, the higher the quality of the returned solution. Furthermore, we give sufficient upper bounds on the outer iteration to make the obtained solution at different optimization level. Additionally, we obtain a new sufficient optimality condition for Max-Cut. A simple example is given to understand that our algorithm can be much faster than the LS-TPG algorithm. Finally we do numerical experiments. The computational results show that our algorithm always produces better-quality solutions than the LS-TPG algorithm in less time.

The article is organized as follows. The next section reviews the LS-TPG algorithm presented in Alperin and Nowak [1]. In section 3, our algorithm is proposed and analyzed. We establish practical stopping criteria and prove that our algorithm finitely terminates at a KKT point, the distance between which and the neighbor optimal solution is also estimated. Additionally, we obtain a new sufficient optimality condition for Max-Cut. Numerical results are presented in section 4. The last section makes some concluding remarks.

Notation : Diag(a) is a diagonal matrix with diagonal components a_i. e denotes the vector with all components equal to one and I denotes the identity matrix. $A \succeq 0$ $(A \prec 0)$ means that A is a semidefinite positive (definite negative) matrix. $\lambda_{min}(A)$ and $\lambda_{max}(A)$ denote the minimal and maximal eigenvalues of A, respectively. For n-dimensional vectors a and b, $a \leq b$ denotes $a_i \leq b_i$, $i = 1, \cdots, n$. The 2-norm of a vector a is defined by $\|a\|_2 = \sqrt{a^T a}$ and the infinity norm of an $n \times n$ matrix A is $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{i=1}^n |A_{ij}|$. $sign(a)$ denotes the sign vector of a whose i-th element is $sign(a_i) = \begin{cases} 1, & \text{if } a_i \geq 0 \\ -1, & \text{otherwise.} \end{cases}$

2. Lagrangian Smoothing Heuristic

Smoothing methods deform the original objective function into a function whose smoothness is controlled by a parameter. The original problem is then solved by a sequence of problems in the same spirit as homotopy methods. The success of this approach depends on finding a good smoothing function.

In [1], the Lagrangian function

$$L(x; \mu) = -n\mu + x^T (A + \mu I) x \tag{2.1}$$

was introduced as a global smoothing function for $x^T Ax$, where μ is a Lagrangian multiplier such that $A + \mu I \succeq 0$. Combining it with the exact penalty function [4]

$$P(x; t) = x^T Ax - (1 - t)^{-1} x^T x, \tag{2.2}$$

they obtained a sequence of problems

$$\begin{aligned} H(x; \mu; t) &= tP(x; t) + (1 - t)L(x; \mu) & (2.3) \\ &= -(1 - t)n\mu + x^T (A - t(1 - t)^{-1} I + (1 - t)\mu I)x. & (2.4) \end{aligned}$$

Then they solved the following parametric optimization problem, denoted by $Q(t; \mu)$:

$$\min_{x \in [-e, e]} H(x; \mu; t), \tag{2.5}$$

for an increased sequence $t_k \in [0, 1)$.

Lemma 2.1 (Alperin and Nowak [1]) — There exists $t^* \in (0, 1)$ such that the optimal function values of (1.4)-(1.5) and (2.5) are equal for all $t \in [t^*, 1)$.

In that paper, they recommend a uniform sequence $t_k = k/(M + 1)$ with M being a positive integer and a geometric sequence $t_k = 1 - \rho^k$ with

$\rho \in (0, 1)$. According to our computational experience, the uniform sequence performs better. In this article, we only consider this case.

Solving the parametric optimization problem (2.5) is as difficult as (1.4)-(1.5) if only t is large enough. A truncated projected gradient algorithm was employed to approximate (2.5) in Alperin and Nowak [1]. The projector operator $\Pi_{[-e,e]}(y)$ they used is boxlike:

$$\Pi_{[-e,e]}(y)_i = \begin{cases} -1, & \text{if } y_i < -1, \\ y_i, & \text{if } -1 \le y_i \le 1, \\ 1, & \text{if } y_i > 1. \end{cases}$$

The detailed algorithm is described as follows:

Algorithm 2.1 (LS-TPG).

Step 1: Initialize $k := 0$, $t_0 := 0$, $x_0 \in [-e, e]$. Set m, M and $\mu := -\lambda_{min}(A)$.

Step 2: For $j := 0$ to $m - 1$ do

$$x_{j+1} := \Pi_{[-e,e]}\left(x_j - \beta_j \frac{\nabla H(x_j; \mu, t_k)}{\|\nabla H(x_j; \mu, t_k)\|} \right), \tag{2.6}$$

where β_j is an appropriate stepsize.

Step 3: If some given stopping criteria are fulfilled or $k = M$, then stop and return $x^* = sign(x_m)$; otherwise update $k := k + 1$, $t_k := \dfrac{k}{M+1}$, $x_0 := x_m$ and goto Step 2.

The authors Alperin and Nowak [1] gave clues to propose stopping criteria in Step 3 of Algorithm 2.1 based on the following proposition. Unfortunately, it is difficult to find satisfied stopping criteria because the practical m is usually set to be small, for example, $m = 10$.

Proposition 2.1 — (Alperin and Nowak [1]) If m is large enough, Algorithm 2.1 can be stopped if $t_k \ge t^*$ without changing the final result, where t^* is defined in Lemma 2.1.

3. Our Approach

3.1 *A New Algorithm* : In Xia [10], we proposed an efficient Lagrangian smoothing algorithm for the general nonlinear binary optimization problems with linear equality constraints. We apply it to solve Max-Cut (1.4)-(1.5). Different from LS-TPG, we use the canonical Frank-Wolfe algorithm [7] to solve the parametric optimization subproblem (2.5) for given t. It approximates the objective function with its first order Taylor expansion at any given iteration point x_k, resulting in the linear programming subproblem

$$\begin{array}{ll} \min & \nabla_x H(x_k;\mu;t)^T x \\ s.t. & -e \leq x \leq e, \end{array} \tag{3.1}$$

where the constant terms have been dropped from the objective function. We notice that the linear programming problem (3.1) can be solved in $O(n)$ time. Actually, its optimal solution x_k^* is explicit:

$$(x_k^*)_i = \begin{cases} -1, & \text{if } (\nabla_x H(x_k;\mu;t))_i \geq 0, \\ 1, & \text{otherwise.} \end{cases} \quad (i = 1, \cdots, n) \tag{3.2}$$

This optimal solution x_k^* is used to construct the descent search direction $d_k = x_k^* - x_k$. A line search

$$\alpha^* = arg \min_{\alpha \in [0,1]} H(x_k + \alpha d_k; \mu; t) \tag{3.3}$$

furnishes the next iterate

$$x_{k+1} = x_k + \alpha^* d_k. \tag{3.4}$$

and the process is repeated.

It is easy to verify that the point sequence $\{x_k\}$ generated by the above Frank-Wolfe algorithm converges to x^*, a KKT point of (2.5). But the convergence is slow and hence it is quite time-consuming to obtain x^*. Thus we approximate x^* using a truncated Frank-Wolfe algorithm, which only generates the first m iterative points.

The outer process is repeated for an increased sequence $\{t_k\}$. To introduce a practical stopping criteria, we need some lemmas.

Lemma 3.1 — $x^* \in \{-1, 1\}^n$ is a KKT point of $Q(t;\mu)$ (2.5) if and only if

$$\mathrm{Diag}(x^*) \nabla_x H(x^*;\mu;t) \leq 0. \tag{3.5}$$

Proof : The KKT conditions for $Q(t;\mu)$ (2.5) are

$$\begin{cases} (\nabla_x H(x^*;\mu;t))_i + \alpha_i^* - \beta_i^* = 0, & i = 1, \cdots, n, \\ \alpha_i^* \geq 0, & i = 1, \cdots, n, \\ \beta_i^* \geq 0, & i = 1, \cdots, n, \\ -1 \leq x_i^* \leq 1, & i = 1, \cdots, n, \\ \alpha_i^*(1 - x_i^*) = 0, & i = 1, \cdots, n, \\ \beta_i^*(1 + x_i^*) = 0, & i = 1, \cdots, n. \end{cases} \tag{3.6}$$

Since $x^* \in \{-1, 1\}^n$, the above KKT conditions are equivalent to

$$\begin{cases} \alpha_i^* = \frac{1}{2}(\nabla_x H(x^*;\mu;t))_i(-1 - x_i^*) \geq 0, & i = 1, \cdots, n, \\ \beta_i^* = \frac{1}{2}(\nabla_x H(x^*;\mu;t))_i(1 - x_i^*) \geq 0, & i = 1, \cdots, n. \end{cases} \tag{3.7}$$

It is easy to check that the conditions $x^* \in \{-1, 1\}^n$ and (3.7) imply $x_i^*(\nabla_x H(x^*;\mu;t))_i \leq 0$ for $i = 1, \cdots, n$. On the other hand, if (3.5) holds, α^* and β^* defined in (3.7) satisfy $\alpha \geq 0$ and $\beta \geq 0$. The proof is completed. □

Corollary 3.1 — Let $x^* \in \{-1, 1\}^n$ be any local minimizer of $Q(t;\mu)$ (2.5), then

$$\text{Diag}(x^*)\nabla_x H(x^*;\mu;t) \leq 0. \tag{3.8}$$

Proof : Since the linear function constraint qualification holds for $Q(t;\mu)$, any local minimizer x^* is also a KKT point. □

Lemma 3.2 — If $x \in \{-1, 1\}^n$ is a KKT point of $Q(t;\mu)$ (2.5), then it remains a KKT point of $Q(t';\mu)$ for all $t' > t$.

Proof : $x \in \{-1, 1\}^n$ is a KKT point of $Q(t;\mu)$, which implies that $x_i(\nabla_x H(x;\mu;t))_i \leq 0$ for all $i = 1, \cdots, n$ due to Lemma 3.1. Therefore

$$\begin{aligned} x_i(\nabla_x H(x;\mu;t'))_i &= x_i(\nabla_x H(x;\mu;t))_i - 2[(\frac{t'}{1-t'} - (1-t')\mu) - \\ &\quad (\frac{t}{1-t} - (1-t)\mu)]x_i^2 \\ &= x_i(\nabla_x H(x;\mu;t))_i - 2[(\frac{t'}{1-t'} - (1-t')\mu) - \\ &\quad (\frac{t}{1-t} - (1-t)\mu)] \\ &\leq x_i(\nabla_x H(x;\mu;t))_i \leq 0. \end{aligned}$$

x is also a KKT point of $Q(t';\mu)$ according to Lemma 3.1. □

Notice that if a KKT point is obtained in $\{-1,1\}^n$, it is not needed to update t due to Lemma 3.2. We take the condition $x \in \{-1,1\}^n$ and (3.5) as stopping criteria.

Thus, our algorithm can be formally proposed as follows, denoted by LS-TFW.

Algorithm [LS-TFW]

Step 1: Initialize $k := 0$, $t_0 := 0$, $x_0 \in [-e, e]$. Set m, M and $\mu := -\lambda_{min}(A)$.

Step 2: For $j := 0$ to $m-1$, compute x_j^* and α^* according to (3.2) and (3.3) respectively; update

$$x_{j+1} := x_j + \alpha^*(x_j^* - x_j); \tag{3.9}$$

if $x_{j+1} \in \{-1,1\}^n$ and $\mathrm{Diag}(x_{j+1})\nabla_x H(x_{j+1}; \mu; t) \leq 0$, then stop and return $x^* = x_{j+1}$, otherwise goto Step 3.

Step 3: If $k = M$, then stop and return $x^* = sign(x_m)$; otherwise update $k := k+1$, $t_k := \dfrac{k}{M+1}$, $x_0 := x_m$; goto Step 2.

3.2 *Convergence Results* : Below we study the properties of Algorithm LS-TFW.

Lemma 3.3 — If $\mu > 0$ and

$$\frac{2\mu + \|A\|_\infty + 1 - \sqrt{(\|A\|_\infty + 1)^2 + 4\mu}}{2\mu} < t < 1, \tag{3.10}$$

any $x \in \{-1,1\}^n$ is a KKT point of $Q(t;\mu)$ (2.5).

Proof : The given condition (3.10) implies

$$\frac{t}{1-t} - (1-t)\mu > \|A\|_\infty. \tag{3.11}$$

For each i, if $x_i = 1$,

$$(\nabla_x H(x;\mu;t))_i = 2(Ax)_i - 2\left(\frac{t}{1-t} - (1-t)\mu\right) < 2(Ax)_i - 2\|A\|_\infty$$
$$\leq 2\sum_{j=1}^n |A_{ij}| - 2\|A\|_\infty \leq 0,$$

otherwise $x_i = -1$,

$$(\nabla_x H(x;\mu;t))_i = 2(Ax)_i + 2(\frac{t}{1-t} - (1-t)\mu) > 2(Ax)_i + 2\|A\|_\infty \\ \geq -2\sum_{j=1}^{n} |A_{ij}| + 2\|A\|_\infty \geq 0.$$

In either case, it holds that $x_i(\nabla_x H(x;\mu;t))_i \leq 0$. Due to Lemma 3.1, $x \in \{-1,1\}^n$ is a KKT point. □

Lemma 3.4 — To solve $Q(t;\mu)$ (2.5) with $\mu > 0$ and

$$\frac{2\mu + \lambda_{max}(A) + 1 - \sqrt{(\lambda_{max}(A)+1)^2 + 4\mu}}{2\mu} < t < 1, \tag{3.12}$$

for any given initial point $x_0 \in [-e, e]$, suppose the next iterative point generated by Frank-Wolfe algorithm is denoted by x_1, then $x_1 \in \{-1,1\}^n$.

Proof : The condition (3.12) implies

$$A - (\frac{t}{1-t} - (1-t)\mu)I \prec 0, \tag{3.13}$$

hence $H(x;\mu;t)$ is concave for such given μ and t.

Applying Frank-Wolfe algorithm to solve $Q(t;\mu)$, we first solve the linear programming subproblem (see also (3.1))

$$\begin{array}{ll} \min & \nabla_x H(x_0;\mu;t)^T(x - x_0) \\ s.t. & -e \leq x \leq e, \end{array} \tag{3.14}$$

and get an optimal solution $x_0^* \in \{-1,1\}^n$ (see (3.2)). Since x_0 is feasible in (3.14), we have

$$\nabla_x H(x_0;\mu;t)^T(x_0^* - x_0) \leq 0. \tag{3.15}$$

Then $d_0 = x_0^* - x_0$ is a descent direction at x_0. We do the line search

$$\alpha^* = arg \min_{\alpha\in[0,1]} H(x_0 + \alpha d_0;\mu;t) \tag{3.16}$$

and obtain

$$x_1 = x_0 + \alpha^* d_0 = x_0 + \alpha^*(x_0^* - x_0). \tag{3.17}$$

To complete the proof, it is sufficient to show $\alpha^* = 1$ is the strict global optimal solution of (3.16) in the nontrivial case of $x_0 \neq x_0^*$.

Since $H(x + \alpha d_0; \mu; t)$ is a quadratic function of x,

$$\begin{aligned} H(x_0^*; \mu; t) &= H(x_0; \mu; t) + \nabla_x H(x_0; \mu; t)^T (x_0^* - x_0) \\ &\quad + \frac{1}{2}(x_0^* - x_0)^T (A - (\frac{t}{1-t} - (1-t)\mu) I)(x_0^* - x_0) \\ &< H(x_0; \mu; t), \end{aligned}$$

where the final inequality follows from (3.15) and (3.13).

Therefore, for all $0 \leq \alpha < 1$,

$$\begin{aligned} H(x_0 + \alpha d_0; \mu; t) &= H((1-\alpha)x_0 + \alpha x_0^*; \mu; t) \\ &\geq (1-\alpha) H(x_0; \mu; t) + \alpha H(x_0^*; \mu; t) \\ &> H(x_0^*; \mu; t), \end{aligned}$$

where the first inequality follows from the concavity of $H(x; \mu; t)$. □

Lemma 3.5 — Suppose $\mu > 0$ and A is a symmetric matrix with zero diagonal elements, then

$$\begin{aligned} &\frac{2\mu + \lambda_{\max}(A) + 1 - \sqrt{(\lambda_{\max}(A) + 1)^2 + 4\mu}}{2\mu} \\ &\leq \frac{2\mu + \|A\|_\infty + 1 - \sqrt{(\|A\|_\infty + 1)^2 + 4\mu}}{2\mu} < 1 \end{aligned}$$

PROOF : Let $Ax = \lambda_{\max}(A)x$ ($x \neq 0$), we have

$$0 \leq \lambda_{\max}(A) = \frac{\|\lambda_{\max}(A)x\|_\infty}{\|x\|_\infty} = \frac{\|Ax\|_\infty}{\|x\|_\infty} \leq \|A\|_\infty.$$

The proof is completed since

$$f(x) = \frac{2\mu + x + 1 - \sqrt{(x+1)^2 + 4\mu}}{2\mu}$$

is a monotone increasing function for $x \geq 0$ and $\lim_{x\to\infty} f(x) = 1$. □

***Theorem* 3.1** — *Suppose $A \neq 0$ is a symmetric matrix with zero diagonal elements and the finite integer parameters fulfill $m \geq 1$ and*

$$M > \frac{2\mu}{\sqrt{(\|A\|_\infty + 1)^2 + 4\mu} - \|A\|_\infty - 1} - 1, \qquad (3.18)$$

Then Algorithm LS-TFW finitely terminates at a KKT point of $Q(t;\mu)$ (2.5).

PROOF : $A \neq 0$ is a symmetric matrix with zero diagonal elements implies $\mu = -\lambda_{min}(A) > 0$. (3.18) implies

$$\frac{M}{M+1} > \frac{2\mu + \|A\|_\infty + 1 - \sqrt{(\|A\|_\infty + 1)^2 + 4\mu}}{2\mu}.$$

Let $t = \frac{M}{M+1}$ due to Lemma 3.5 and Lemma 3.4, after one Frank-Wolfe iteration, we obtain an $x^* \in \{-1,1\}^n$, which is a KKT point of $Q(t;\mu)$ according to Lemma 3.3. Due to Lemma 3.1, it satisfies the stopping criteria (3.15), i.e., Algorithm LS-TFW terminates at Step 2. □

Remark 3.1 : The computational complexity of Algorithm 3.1 is at most $O(Mmn^2)$.

3.3. *A New Sufficient Optimality Condition* : To further study the quality of x^* returned by Algorithm LS-TFW, we consider sufficient optimal conditions for Max-Cut (1.4)-(1.5).

Lemma 3.6 (Beck and Teboulle [2]) — If $x \in \{-1,1\}^n$ satisfy

$$\lambda_{\min}(A)e \geq \mathrm{Diag}(x)Ax, \tag{3.19}$$

then x is a globally optimal solution for Max-Cut (1.4)-(1.5).

Lemma 3.7 (Xia [11]) — Let $x \in \{-1,1\}^n$ and γ_p be a suitable lower bound of the following problem

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}y_iy_j, \tag{3.20}$$

$$s.t. \quad \sum_{i=1}^{n} y_i^2 = p, \tag{3.21}$$

$$y_i \in \{-1,0,1\},\ i = 1,\cdots,n. \tag{3.22}$$

If

$$\left(\min_{p\in\{1,\cdots,n\}} \frac{\gamma_p}{p}\right) e \geq \mathrm{Diag}(x)Ax, \tag{3.23}$$

then x is a globally optimal solution for Max-Cut (1.4)-(1.5).

Lemma 3.7 implies Lemma 3.6 if we choose

$$\gamma_p = \min \quad y^T A y \tag{3.24}$$
$$\text{s.t.} \quad y^T y = p, \tag{3.25}$$

whose optimal solution is the eigenvector corresponding to the minimal eigenvalue of A, i.e.,

$$\gamma_p = p \cdot \lambda_{\min}(A). \tag{3.26}$$

More tight choices of γ_p are discussed in [11]. Here we show that Lemma 3.7 can be further strengthened.

Definition 3.1 (Xia [11]) — Define the p-distance ($1 \le p \le n$) ring solution $x' \in \{-1,1\}^n$ as

$$\begin{aligned} (x')^T A x' \le \quad \min \quad & x^T A x \\ \text{s.t.} \quad & \|x - x'\|_0 = p, \\ & x \in \{-1,1\}^n, \end{aligned}$$

where $\|x - x'\|_0$ is the Hamming distance, i.e., the number of different components between x and x'.

***Theorem* 3.2** — *Let $x \in \{-1,1\}^n$ and γ_p be a suitable lower bound of Problem (3.20)-(3.22). If*

$$\left(\min_{p \in \{1,\cdots,\left[\frac{n}{2}\right]\}} \frac{\gamma_p}{p} \right) e \ge \operatorname{Diag}(x) A x, \tag{3.27}$$

where $\left[\frac{n}{2}\right]$ denotes the largest integer smaller than or equal to $\frac{n}{2}$, then x is a globally optimal solution for Max-Cut (1.4)-(1.5).

PROOF : As shown in [11], first it is not difficult to verify that $x \in \{-1,1\}^n$ is a p-distance ring solution of Max-Cut (1.4)-(1.5) only if

$$\frac{\gamma_p}{p} e \ge \operatorname{Diag}(x) A x. \tag{3.28}$$

Then Lemma 3.7 follows from the fact that x is globally optimal if and only if it is p-distance ring solution for all $p = 1, \cdots, n$.

Let $x' \in \{-1,1\}^n$ satisfy

$$\frac{\gamma_p}{p} e \ge \operatorname{Diag}(x') A x' = \operatorname{Diag}\left(-x'\right) A \left(-x'\right). \tag{3.29}$$

Therefore, $(-x')$ is a p-distance ring solution of Max-Cut (1.4)-(1.5), i.e.,

$$(-x')^T A(-x') \leq \begin{array}{ll} \min & x^T A x \\ s.t. & \|x-(-x')\|_0 = p \\ & x \in \{-1,1\}^n \end{array},$$

which is further equivalent to the following inequality since $\|x-(-x')\|_0 = p$ if and only if $\|x-x'\|_0 = n-p$ for any $x, x' \in \{-1,1\}^n$:

$$x'^T A x' \leq \begin{array}{ll} \min & x^T A x \\ s.t. & \|x-x'\|_0 = n-p \\ & x \in \{-1,1\}^n \end{array}.$$

Now we can see that x' is an $(n-p)$-distance ring solution of Max-Cut (1.4)-(1.5). In sum, the inequality (3.28) is a sufficient condition under which x is not only a p-distance but also an $(n-p)$-distance ring solution. □

3.4 *Error Estimation* : In this subsection, we first establish some conditions under which the returned solution is globally or locally optimal. We then estimate the distances between the returned solution and some local minimizers.

Theorem 3.3 — *Suppose Algorithm LS-TFW stops in k outer iterations and*

$$k \leq \min_{p \in \{1,\cdots,[\frac{n}{2}]\}} \left(1 - \frac{2}{1+\frac{\gamma_p}{p}+\sqrt{(1+\frac{\gamma_p}{p})^2+4\mu}}\right)(M+1) \tag{3.30}$$

where γ_p is a suitable lower bound of Problem (3.20)-(3.22), the returned x^* is the globally optimal solution of Max-Cut (1.4)-(1.5).

Proof : The termination x^* satisfies $x^* \in \{-1,1\}^n$ and $\mathrm{Diag}(x^*)$ $\nabla_x H(x^*;\mu;t) \leq 0$. That is, for every i,

$$\mathrm{Diag}(x^*)Ax^* + \mathrm{Diag}(x^*)\left(-\frac{t}{1-t}+(1-t)\mu\right)x^* \leq 0, \tag{3.31}$$

or equivalently,

$$\mathrm{Diag}(x^*)Ax^* \leq \left(-\frac{t}{1-t}+(1-t)\mu\right)e, \tag{3.32}$$

which implies Inequality (3.27) under the assumption (3.33). The proof is then completed due to Theorem 3.2. □

Corollary 3.2 — If Algorithm LS-TFW stops in the first outer iteration (i.e., k remains 0), then the returned x^* is also a globally optimal solution for Max-Cut (1.4)-(1.5).

PROOF : Choose $\gamma_p = p \cdot \lambda_{\min}(A)$. Then Inequality (3.33) reduced to $k \leq 0$. □

Definition 3.2 — Define the q-neighbour $(1 \leq q \leq [\frac{n}{2}])$ solution $x' \in \{-1,1\}^n$ as

$$\begin{aligned}(x')^T Ax' \leq \quad \min \quad & x^T Ax \\ s.t. \quad & \|x - x'\|_0 \leq q, \\ & x \in \{-1,1\}^n.\end{aligned}$$

It is trivial to verify that a q-neighbour solution must be p-distance ring solutions (see Definition 3.1) for $p = 1, 2, \cdots, q$ and vice versa. Similarly, we have

***Theorem* 3.4** — *Suppose Algorithm LS-TFW stops in k outer iterations and*

$$k \leq \min_{p\in\{1,\cdots,q\}} \left(1 - \frac{2}{1 + \frac{\gamma_p}{p} + \sqrt{(1 + \frac{\gamma_p}{p})^2 + 4\mu}}\right)(M+1) \tag{3.33}$$

where $1 \leq q \leq [\frac{n}{2}]$ and γ_p is a suitable lower bound of Problem (3.20)-(3.22), the returned x^* is a q-neighbour solution of Max-Cut (1.4)-(1.5).

Remark 3.2 : Theorem 3.4 implies that the smaller the k, the better the solution x^*.

Corollary 3.3 — Suppose A is a symmetric matrix with zero diagonal elements and Algorithm LS-TFW stops in k outer iterations. If

$$k \leq \left(1 - \frac{2}{1 + \sqrt{1 + 4\mu}}\right)(M+1) \tag{3.34}$$

then the returned x^* is a 1-neighbour solution of Max-Cut (1.4)-(1.5).

PROOF : This is the special case when $q = p = 1$. The optimal objective function value of the problem (3.20)-(3.22) is zero since $p = 1$ and $A_{ii} = 0$ for all i. Therefore we can set $\gamma_1 = 0$. The proof is completed. □

***Theorem* 3.5** — *Suppose Algorithm LS-TFW stops at x^k in k outer iterations and p is an integer lying between 1 and n. Let γ_p be a lower bound of the problem (3.20)-(3.22), then*

$$(x^k)^T Ax^k - (x^*)^T Ax^* \leq -4p\left(-\frac{k}{M+1-k} + \frac{M+1-k}{M+1}\mu\right) + 4\gamma_p \quad (3.35)$$

where x^* is the p-distance ring solution (see Definition 3.1) of Max-Cut (1.4)-(1.5) at x^k.

PROOF : The termination x^k satisfies $x^k \in \{-1,1\}^n$ and $\mathrm{Diag}(x^k)$ $\nabla_x H(x^k;\mu;t) \leq 0$. That is, for every i,

$$\begin{aligned} 0 &\geq x_i^k(\sum_{j=1}^n A_{ij}x_j^k) + (-\frac{t}{1-t} + (1-t)\mu)(x_i^k)^2 \\ &= x_i^k(\sum_{j=1}^n A_{ij}x_j^k) + (-\frac{t}{1-t} + (1-t)\mu). \end{aligned}$$

Let $N = \{1, 2, \cdots, n\}$. For any $N_p \subseteq N$ with p elements, we have

$$\sum_{i\in N_p} x_i^k(\sum_{j=1}^n A_{ij}x_j^k) + p(-\frac{t}{1-t} + (1-t)\mu) \leq 0. \quad (3.36)$$

Therefore,

$$\begin{aligned} \sum_{i\in N_p} x_i^k(\sum_{j\notin N_p} A_{ij}x_j^k) &\leq -p(-\frac{t}{1-t} + (1-t)\mu) - \sum_{i\in N_p} x_i^k(\sum_{j\in N_p} A_{ij}x_j^k) \\ &\leq -p(-\frac{t}{1-t} + (1-t)\mu) + \gamma_p. \end{aligned}$$

Now for any N_p,

$$\begin{aligned} (x^*)^T Ax^* &= \sum_{i\notin N_p}\sum_{j\notin N_p} A_{ij}x_i^k x_j^k + \sum_{i\in N_p}\sum_{j\in N_p} A_{ij}x_i^k x_j^k - 2\sum_{i\in N_p} x_i^k(\sum_{j\notin N_p} A_{ij}x_j^k) \\ &= (x^k)^T Ax^k - 4\sum_{i\in N_p} x_i^k(\sum_{j\notin N_p} A_{ij}x_j^k) \\ &\geq (x^k)^T Ax^k + 4p(-\frac{t}{1-t} + (1-t)\mu) - 4\gamma_p. \end{aligned}$$

4. Numerical Comparison

In this section, we first use a simple example to show the different performances of our LS-TFW algorithm and the LS-TPG algorithm. Consider the following problem,

$$\min \quad -x^T x \tag{4.1}$$

$$s.t. \quad x_i^2 = 1, \quad i = 1, \dots, n. \tag{4.2}$$

Let $x_0 = \epsilon(1, 1, \dots, 1)^T$, where ϵ is an adjustable parameter. It is easy to see that $\forall \epsilon > 0$, our LS-TFW algorithm terminates at an optimal solution $(1, 1, \dots, 1)^T$ in one step. But when LS-TPG algorithm is used, it only converges linearly if $\epsilon < 1 - \frac{2\beta_j}{\sqrt{n}}$ and m is large enough.

Then we numerically compared our LS-TFW algorithm with the LS-TPG algorithm and the famous rank-two algorithm (Burer, Monteiro and Zhang [5]). The test problems are created with rudy, a machine independent graph generator written by Rinaldi, which is standard for the Max-Cut problem (Helmberg and Rendl [9]). For LS-TFW algorithm and LS-TPG algorithm, 10 random initial points are chosen for each test problem and the best (maximal) cut values over 10 runs are reported, respectively. The parameters of LS-TFW algorithm and LS-TPG algorithm are set to be the same as Alperin and Nowak [1], i.e., $\beta_j \equiv 5$, $m = 10$ and $M = 20$. MATLAB 7.6 software was used to run LS-TFW algorithm and LS-TPG algorithm on a PC machine with an AMD Turion 1.6G×2 MHZ processor. Results of the rank-two algorithm with parameters N = 10 and M = 8 were obtained on a SGI Origin2000 machine with a 300 MHZ R12000 processor (Burer, Monteiro and Zhang [5]).

The numerical results are reported in Table 1, where the first two columns contain information concerning the tested graphs, 'n' denotes the dimension, '$\hat{m}$' denotes the number of the nonzero elements of A, column 'value' gives the best (maximal) cut values detected by all algorithms and column 'times (s)' shows the average running CPU time in seconds. For simplicity, in Table 1, the LS-TFW algorithm, the LS-TPG algorithm and the rank-two algorithm are denoted by FW, PG and RT respectively. We can see that our LS-TFW algorithm always produces better-quality solutions than the LS-TPG algorithm in less time.

Table 1: Comparison of our algorithm with the existing smoothing algorithm

Test	Size	Time (s)			Value		
Problem	(n, $\hat{m}$)	RT	FW	PG	RT	FW	PG
g11	(800, 1600)	3.88	0.02	0.03	554	556	556
g12	(800, 1600)	3.76	0.02	0.04	552	550	548
g13	(800, 1600)	3.45	0.02	0.03	572	580	570
g14	(800, 4694)	5.53	0.04	0.06	3053	3034	3016
g15	(800, 4661)	5.91	0.03	0.06	3039	3022	3007
g20	(800, 4672)	5.56	0.04	0.10	939	931	911
g22	(2000, 19990)	22.31	0.10	0.38	13331	13288	13245
g24	(2000, 19990)	27.30	0.12	0.42	13287	13269	13209
g31	(2000, 19990)	19.61	0.10	0.37	3255	3255	3204
g32	(2000, 4000)	13.09	0.04	0.08	1380	1378	1376
g34	(2000, 4000)	9.82	0.04	0.07	1358	1356	1356

Conclusions

The Max-Cut problem is a famous NP-hard problem in combinatorial optimization and widely used in network, statistical physics and many other fields. The Lagrangian smoothing heuristic (LS-TPG) is a recent efficient approach to solve the Max-Cut problem.

In this article, we proposed a more efficient Lagrangian smoothing algorithm (LS-TFW). The continuous subproblems arising from the smoothed functions were solved by the truncated Frank-Wolfe algorithm. We established practical stopping criteria and proved that our LS-TFW algorithm finitely terminates at a KKT point, the distance between which and the neighbour optimal solution is also estimated. We further gave sufficient upper bounds on the outer iteration to make the obtained solution at different optimization level. Additionally, we obtain a new sufficient optimality condition for Max-Cut, see Theorem 3.2. A simple example was given to understand that our LS-TFW algorithm can be much faster than the LS-TPG algorithm. Finally we did numerical experiments using the same test problems and the same parameters as the LS-TPG algorithm. Numerical results indicated that our LS-TFW algorithm always obtained better solutions in less time than that of the LS-TPG algorithm.

References

1. H. Alperin and I. Nowak, "Lagrangian smoothing heuristics for Max-Cut", *Journal of Heuristics*, **11** (2005), 447-463.

2. A. Beck and M. Teboulle, "Global optimality conditions for quadratic optimization problems with binary constraints", *SIAM Journal on Optimization*, **11**(1) (2000), 179-188.

3. D. Bertsimas and Y. Ye, "Semidefinite Relaxations, Multivariate Normal Distributions, and Order Statistics," In D.-Z. Du and P. Pardalos, (eds.), *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, 1-19, (1998).

4. M. Borchardt, "An exact penalty approach for solving a class of minimization problems with boolean variables", *Optimization*, **19**(6) (1988), 829-838.

5. S. Burer, R. D. C. Monteiro and Y. Zhang, "Rank-two relaxation heuristics for MAX-CUT and other binary quadratic programs", *SIAM Journal on Optimization*, **12** (2001), 503-521.

6. P. Festa, P. M. Pardalos, M. G. C. Resende and C. C. Ribeiro, "Randomized Heuristics for the Max-Cut Problem." *Optimization Methods and Software*, **7** (2002), 1033-1058.

7. M. Frank and P. Wolfe, "An algorithm for quadratic programming", *Naval Research Logistics Quaterly*, **3** (1956), 95-110.

8. M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satifiablity problemsusing semidefinite programming", *Journal of the ACM*, **42** (1995), 1115-1145.

9. C. Helmberg and F. Rendl, "A Spectral Bundle Method for Semidefinite Programming", *SIAM Journal on Optimization*, **10**(3) (2000), 673-695.

10. Y. Xia, "A New Continuation Approach to Quadratic Assignment and Related Problems", In Ya-Xiang Yuan et al (Eds) *Proceedings of the Eighth National Conference of Operations Research Society of China*, Global-Link Informatics Limited, HongKong, 262-269 (2006).

11. Y. Xia, "New Optimality Conditions for Quadratic Optimization Problems with Binary Constraints", *Optimization Letters*, **3**(2) (2009), 253-263.