www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(3), Sep-Dec 2024, 171-182. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2024.v15i2.204

Solving Max-cut Problem with a mixed penalization method for Semidefinite Programming

Derkaoui Orkia[1], Lehireche Ahmed[1], khobzaoui Abdelkader[2]
[1] *University Djillali Liabes of Sidi bel abbes, Computer Science Department, BP 89 City Sidi bel abbes 22000, Algeria*
derkaouiorkia@univ-saida.dz, elhir@yahoo.fr
[2]*University Djillali Liabes of Sidi bel abbes, Mathematics Laboratory, Algeria.*
akhobzaoui@yahoo.fr

Abstract. This paper is an attempt to exploit the opportunity of Semi Definite Programming (SDP), which is an area of convex and conic optimization. Indeed, Numerous NP-hard problems can be solveby using this approach. Hence, we intend to investigate the strength of SDP to model and provide tight relaxations of combinatorial and quadratic problems in order to present a new polynomial time algorithm for solving this robust model. This algorithm was firstly use to solve the nonlinear programs, the reason for which we search to extend it to the SDP programs. Actually, this algorithm designs the combination of two penalization methods. The first one is a primal-dual interior point (PDIM) method while the second one is a primal-dual exterior point (PDEM) method. Unlike the first method, which converges globally, the second one, also called the primal dual nonlinear rescaling method, has local super linear/quadratic convergence. Therefore, it seems appropriate to use a mixed algorithm based on the interior-exterior point method (IEPM). In fact, this resolution starts from the interior method, and at a certain level of execution, it proceeds to exterior method. Hence, a convergence evaluation function is use to know the level of permutation. Through evaluation, it has been approve that our approach is use to solve some instances of max-cut problem. This problem is a central graph theory model that occurs in many real problems and it is one of many NP-hard problems, which has attracted many researchers over the years. Then, we have used a semi definite programming solver SDPA (Semi Definite Programming Algorithm) that is modify to include the exterior point method subroutine. From the computational performance, we conclude that as the problem size increases, interior-exterior point algorithm gets relatively faster. The numerical results obtained are promising.

Keywords: Semidefinite programming, Max-cut Problem, Primal-dual interior point method, Exterior point method, search direction, nonlinear rescaling.

Article Info
Received Oct 20, 2020
Accepted May 2, 2021

1 Introduction

Our problem is an optimization problem (OP). It is non-linear model. We will focus on the minimization problem with inequality constraints as follows:

$$OP = \begin{cases} \text{minimize } f(x) \\ \text{subject to } g(x) \leq 0 \\ x \in R^n \end{cases} \qquad \textbf{(1)}$$

Where $f(x)$ and $g(x) = g_1(x) \dots g_m(x)$ are vectors function. We assume that $f: R^n \to R$ and all $g_i: R^n \to R$, $i = 1 \dots m$ are smooth.We use the standard primal form of semidefinite program **(2)** and its dual **(3)** in block diagonal form, for the original problem **(1)**:

$$SDP = \begin{cases} p^* = minimize\ A_0 \bullet X \\ subject\ to\ A_i \bullet X = b_i\ , i = 1 \dots m \\ X \in S_+^n \end{cases} \quad \textbf{(2)}$$

And its dual :

$$DSDP = \begin{cases} d^* = maximize\ b^t\ y \\ subject\ to\ Z = A_0 - \sum_{i=1}^{m} A_i y_i \\ Z \in S_+^n \end{cases} \quad \textbf{(3)}$$

Where the matrix A_i, $i = 0, \dots, m$ are linearly independent matrices, i.e., $A_i, i = 0, \dots, m$ span an m dimensional linear space in S^n, the space of $n * n$ real symmetric matrices. The vector $b = (b_1, \dots b_m)^T \in R^m$ is the cost vector where R^m is the m-dimentionnel Euclidean space. The matrix $X \in S_+^n$ means that X is positive semidefinite with S_+^n is the cone of SDP matrices.The operator $\bullet$ denotes the standard inner product in S^n: $A \bullet B = trace(AB)$ for $A, B \in S^n$. The vector $y \in R^m$ and the matrix $Z \in S_+^n$ are the dual variables. The values p^* and d^* are the optimal value of the primal objective function and the optimal value of the dual objective function respectively for **(2)** and **(3).**

The duality theory for semidefinite programming is similar to its linear programming counterpart, but more subtle (see for example [1-3]. The programs **(2)** and **(3)** satisfy the weak duality condition: $p^* \leq d^*$ at the optimum, the primal objective $A_0 \bullet X$ is equal to the dual objective $b^T y$.

The semidefinite programming is a convex optimization technique, see [2]. It is an extension of LP (Linear Programming) in the Euclidean space to the space of symmetric matrices. SDP problems are linear. Their feasible sets which involve the cone of positive semidefinite matrices, a non-polyhedral convex cone and they are called linear semidefinite programs. Such problems are the object of a particular attention in the papers by [2], as well on a theoretical or an algorithmically aspect, see for instance, the following references [2, 4]. The figure **1** below shows the geometrical representation of Max-Cut problem, which is used below in experimentation. It is an elliptope for $n = 3$.

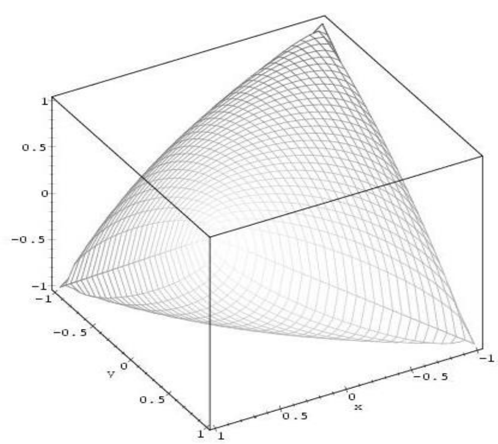

Fig. 1. Elliptope for $n = 3$, Boundary of the set of SDP matrices in S^3. [5]

SDP is not only an extension of LP but also includes convex quadratic optimization problems and some other convex optimization problems. The formulation problems as a semidefinite program have been well studied, although before the development of interior point methods (IPMs). We distinguish two cases:

- The semidefinite programming to model the problem exactly. The optimum provided by IPMs is the real optimum of the problem.
- The semidefinite programming to model a relaxation of a problem. The optimum provided by the IPMs is a lower bound of the true optimum problem (in the case of minimization).

SDP has many applications in various fields such as combinatorial optimization [6], control theory [7], robust optimization [5, 6]and quantum chemistry [6, 8]. See [9, 10-11] for a survey on SDPs and the papers in their references.

Interior-points methods for SDP have sprouted from the seminal work of Nesterov and Nemirovksi [12] who stated the theoretical basis for an extension of interior-methods to conic programming and have suggested three extensions of IPM to SDP (Affine-scaling algorithms, Projective methods with a potential function, Path-following algorithms) [13-15].

Interior point methods for nonlinear optimization problems (see [2, 4]) have good theoretical properties and practical performance for many problems. They use the sequential unconstrained minimization technique developed by Fiacco and McCormick [15] for solving constrained optimization problem with inequality constraints. It is related to a sequence of unconstrained minimizations of the classical log-barrier function used the barrier parameter update.

The majority of SDP solvers implement the primal-dual interior point algorithm which is the most efficient. The theory and numerical experiments of this algorithm demonstrates its excellent performance for wide scale practical problems [16]. Nevertheless, iterates should be kept strictly inside the interior region. If the feasible region is "narrow" iterates that start from a point far from a solution may take many iterations to arrive at the region near the solution. If an iterate happens to be near the boundary of the feasible region which is not close to a solution, it may not be easy to escape from the region and to arrive at the near center trajectory because of possible numerical difficulties when the barrier parameter is small [17]. In this fact, we can use the PDEM when the PDIM method stops making progress to find the solution.

Recent research on exterior point methods for convex optimization problems (see [14, 17–20]) show good theoretical properties and practical performance for a wide range of problems. Polyak [17-18,19] introduces a study about the primal-dual exterior point (PDEM) method for convex optimization problems. The exterior point method has the same computation of the primal-dual interior approximations PDIM. In addition, the PDEM has local convergence properties and it is faster in practice. The work of[14, 19-20]demonstrates the performance of this approach for non-linear optimization problem.

In this paper, we were inspired by the work of polyak and Griva [14, 19-20] on exterior point methods and Semidefinite programming. We extend the work of [14, 17–20] for nonlinear problems to SDP problems. We exploit the robustness and the global convergence of the interior point method and the fast local convergence of the exterior point method. We implement an interior-exterior point method (IEPM) which is the combination of the two methods mentioned above.

The paper is organized as follows: In the next section, we describe briefly the interior point algorithm implemented in SDPA solver [21]. In section 3, we discuss the exterior point method in connection to the nonlinear rescaling principle. Section 4 describes the interior-exterior point method (IEPM). Section 5 contains the numerical results and concluding remarks. Section 6 is the conclusion and future work.

2 The interior point method

A The interior methods are (also referred to as barrier methods) crucial for convex optimization. For a convex optimization problem, we can take the barrier function defined on the feasible set that tends to approach the boundary of feasible set. We apply the log-barrier function to problem **(1)** with $\mu \geq 0$ where μ is a barrier parameter.

The PDIM algorithm presented in this paper corresponds to the predictor-corrector variant of the primal-dual barrier method developed by [22]. The figure **2** above shows the different steps of the PDIM method.

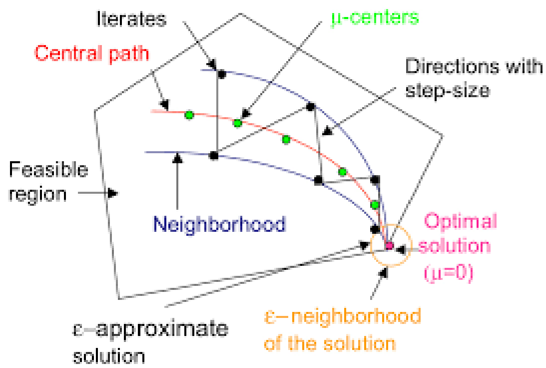

Fig. 2. The graphical representation of the interior points algorithm in S^2 [22].

The PDIM tries to solve the below first order necessary optimality conditions Karush-Kuhn-Tucker (KKT) applied to the barrier problem by iterative methods. Usually, the search direction uses the Newton step for solving the equality part of the barrier KKT conditions. Iterates are kept in the interior region that satisfies $p^* = 0$ and $d^* = 0$ by definition. However, in extending primal-dual interior-point methods from LP to SDP, certain choices have to be made and the resulting search direction depends on these choices. As a result, there can be several search directions for SDP corresponding to a single search direction for LP. We can cite the following four search directions:

- HRVW/KSH/M direction (proposed by [22]),
- MT direction (proposed by [23]),
- AHO direction (proposed by [24]),
- NT direction (proposed by [25]).

The convergence property of the interior-point methods algorithm varies depending on the choice of direction. To compute the search direction, the SDPA employs Mehrotra type predictor-corrector procedure [26] with use of the HRVW/KSH/M search direction [22, 27-28].

Then, KKT provides necessary and sufficient conditions for optimality:

$$\begin{cases} X \succcurlyeq 0\,,\ A{\bullet}X = b_i, i = 1 \dots m & \text{(primal feasibility)} \\ Z \succcurlyeq 0,\ Z = A_0 - \sum_{i=1}^{m} A_i y_i & \text{(dual feasibility)} \\ XZ = \mu I & \text{(complementary \textbackslash\ slackness)} \end{cases} \quad \textbf{(4)}$$

Where I is the identity matrix. The set of solutions $(X, y, Z)_{\mu\geq 0}$ constitutes the central path when μ varies. When μ tends to 0, the central path converges to an optimal solution of the problem. A summary of the different steps of PDIM algorithm is displayed in Figure **3**, [1-3, 22].

PDIM Algorithm Description

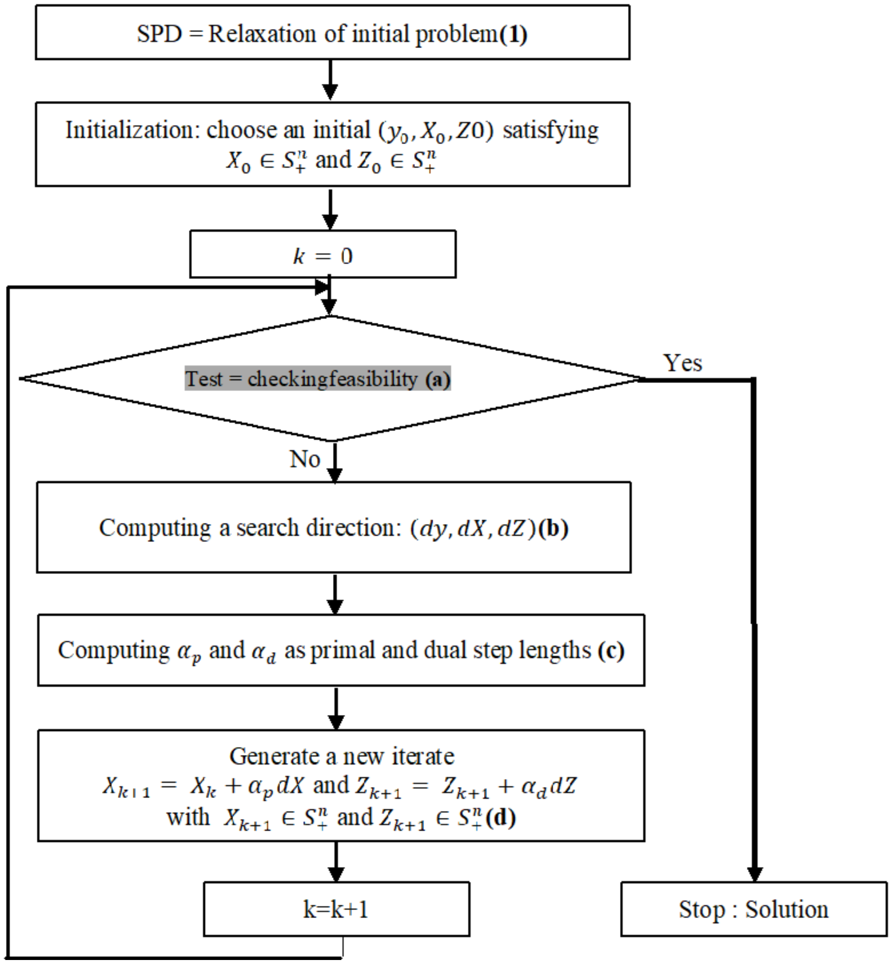

Fig. 3. The PDIM Flow chart

We present the details of **(a)**, **(b)**, **(c)**, **(d)** in the algorithm:

(a): (Checking Feasibility): If y_k, X_k, Z_k is an $\varepsilon -$ approximate optimal solution of the (1,2)and (1,3), stop the iteration.
(b) : (Computing a search direction):As described in [17, 26], apply Mehrotra type predictor-corrector procedure to generate a search direction.

(c) :(Computing α_p and α_d): To compute α_p and $\alpha_d$as primal and dual step lengths, we use the procedure proposed in a previous work entitled -Numerical Experiments with a Primal-Dual Algorithm for Solving Quadratic Problems- in order to minimize the computation cost the procedure. It gives efficient results in practice.
(d) : (Generating a new iterate):

- (Computing μ):μ is compute as half the duality measure $\mu = ZX/m$. This choice is justified by good practical results obtained with this simple heuristic for LP
- (Linearization): There are several possibilities for the linearization of the optimality condition$ZX = \mu I$. In the present method, the chosen condition is ZX-$\mu I = 0$. It does not preserve symmetry and therefore, only the symmetric part of the obtained search direction is kept.

Method PDIM is efficient in practice; its polynomiality is prove by showing that μ_k converges linearly [9]. We can augment this performance when we integrate it in the switch system with PDEM.

3 The exterior point method

The exterior-point method is the generalization of the primal-dual nonlinear rescaling approach. The basic principle of the PDEM is to use the Newton nonlinear rescaling method, which consists of using Newton method for finding an approximation of the primal minimizer followed by the Lagrange multipliers update, for more details see [18].

Here, we just review its basic principles. The external point method calculates simultaneously the primal and the dual problems. The (PDEM) also has interesting local convergence properties. We used the convergence demonstrations of the PDEP method presented in the reference. The work of Polyak.R,Roman.A and Griva.I [14, 19-20, 30] has given a very advanced step to the use of positive semi-definite programming in solving non-linear problems.

This approach is to replace the original problem by a sequence of unconstrained problems whose objective function a combination is of f and a function measuring the violation of constraints. The "exterior" qualifier comes from the property (iii) in **(6)** below, which states that ψ modifies f at the exterior of the admissible set. External penalty methods try to approach the minimum from outside of admissible set of x. This is an example (a variable and a constraint):

$$Example = \begin{cases} minimize\ f(x) = 1 + x + \frac{1}{3}x^3 \\ x \geq 0 \end{cases} \quad \textbf{(5)}$$

An external ψ penalty function said quadratic, is associated to **(5)** where $\psi(x) = (x^-)^2$, with$(x^- = \max(-x, 0))$. So the problem **(5)** is replaced by the penalty problem, it is equivalent to $\min 1 + x + \frac{1}{3}x^3 + r(x^-)^2$, where $r \geq 0$ is a barrier parameter. The effect of this penalty can be observed in Figure **4**.

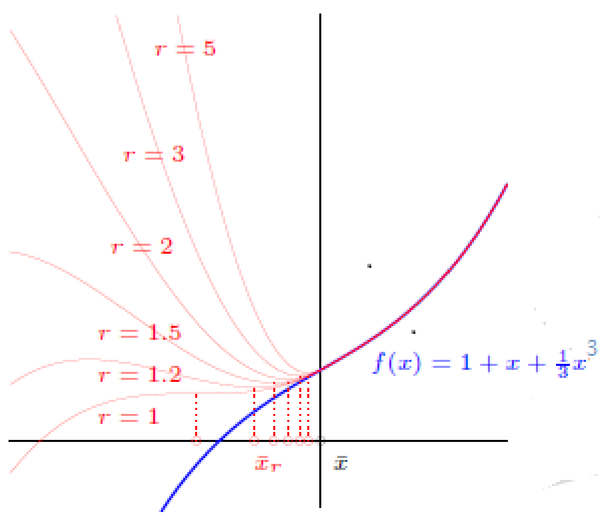

Fig. 4. Quadratic Penalization for r=1,1.2,1.5,2,3,5. [18].

In our work, we use one popular exterior penalty method which is the quadratic penalty function: $\psi(x) = \frac{1}{2r}(p(x))^2$with the barrier parameter $r \geq 0$. The function $\psi(x)$ has the following properties (see [30]):

$$\begin{cases} \psi \quad \text{is continue and convexe function} & \text{(i)} \\ \psi \geq 0, \ \forall x \in R^n & \text{(ii)} \\ \psi = 0, \ \text{if } x \in U, \quad U \text{ is feasible set} & \text{(iii)} \end{cases} \quad \textbf{(6)}$$

The Classical Lagrangian L, for the equivalent problem, which is our main tool, is given by formula :
$L(x,y,r) = f(x) + \sum_{i=1}^{m} y_i \, \psi(rp_i(x))$Knowing that y_i are the Lagrange multipliers.

Then, we solve the SDP problem associated to problem **(5)** with this penalty. The computation of the Hessian assembling is necessary.The Hessian assembling matrix is a square matrix of second-order partial derivatives of a scalar-valued function, or scalar field. It describes the local curvature of a function of many variables. Then the Hessian matrix H of f is a square $n * n$ matrix, usually defined by$\frac{\partial^2}{\partial x_i \partial y_i}$. However, the choice of quadratic penalty function allows direct computation of the Hessian.The exterior point method allows the simultaneous calculation of the primal and dual approximations. Infeasible points are generated in the infeasible region. The limit of these points is an optimal solution to the initial problem. We start with a selected infeasible point. A good property of the method is to start at an unfeasible point (outside the feasible region). It has a fat convergence.Let us give the general procedure:

Procedure PDEM

Choose the following initially:
a tolerance ε,
an increase factor β,
a starting point y_0, X_0, Z_0,
An initial penalty parameter r,
$k = 0$.

At Iteration $k = 0$
1. Solve the SDP relaxation of the problem **(5)**:
   - Calculate the scaling parameter,
   - Find the primal-dual Newton direction from the system, the computation of the Hessian is considered,
   - Find the new primal-dual vector.
2. If the test is attained STOP, else iterate, $k = k + 1$ and go to 1.

For more details about the algorithm, see [18].

The primal-dual iteration takes places outside the primal interior region. The method has the similar performance of an interior point method. The PDEM utilizes an infeasible start point. We propose to combine between the twice methods (the PDIM end PDEM), it has more performance.

4 The Mixed penalization method: The interior-exterior point method

In short, both the robustness of the interior point method and the local convergence properties of the exterior point method encourage us to examine the combination of the methods. The investigation has revealed that the methods can augment each other. Indeed, the interior point method can bring the trajectory to the area of a super linear convergence of the exterior point method, while the exterior point method can improve the convergence in case the interior point method experiences numerical problems.

The interior and exterior point methods are mainly distinct in their driving force of convergence. While, the former requires the decrease to zero of the barrier parameter $\mu \geq 0$, the latter converges due to the information carried by the vector of the Lagrange multipliers y, and then the exterior penalty parameter is so large.

On the other hand, the interior point method, which has global convergence properties, exhibits robust behavior that brings its trajectory to the neighborhood of the solution. If the barrier parameter is decreased, the exterior point method converges locally with the super linear rate [31]. Therefore, both the robustness of the interior point method and the local convergence properties of the exterior point method encourage us to examine the combination of the methods. The investigation has revealed that the methods can augment each other. Indeed, the interior point method can bring the trajectory to the area of a super linear convergence of the exterior point method, while the exterior point method can improve the convergence in case the interior point method experiences numerical problems.

We consider the two SDP problems, the primal one **(2)** and its dual **(3)**. We define the lagrangian as follows $L(x,Z,r) = f(x) + \sum_{i=1}^{m} Z{\bullet}\psi\,(rA(x))$. Where the operator $\bullet$ denotes the inner product in S_+^n, $Z \in S_+^n$, $\psi(x)$=$(A(x))^2$is the quadratic penalty function on S_+^n and r is a barrier parameter. As cited above, we choose the quadratic penalty function in order to minimize the computation of the Hessian assembling. The quadratic penalty function allows direct computation of the Hessian.

The algorithm consists of a switch between the PDIM and PDEM methods. This switching must be controlled. We consider a function $t(X,Z,y)$ of measuring the degree of violation of KKT conditions (that controls convergence of the algorithm) [14]. We begin with the interior search by executing the PDIM procedure. If the measure of the violation is minimal, we continue with the interior search. If this violation is important, the interior point method stops making progress then we switch towards the PDEM method.

The proposed algorithm is carried out by modifying the code of the method implemented in the SDPA solver. Globally, the procedure is the switching between the PDIM and PDEM described below. The algorithm of the mixed penalization methods uses the barrier penalization for the interior method and the quadratic penalization of exterior method.

IEPM Algorithm Description

Initialization:

Propose an initial solution X_0, Z_0, y_0 where $X_0 \in S_+^n$, $Z_0 \in S_+^n$, $y_0 \in R_+^n$,

Propose values of these parameters:

$\varepsilon \geq 0$: a precision parameter,

$t(X,Z,y)$: a degree of violation KKT **(4)**,

$\beta \geq 0$:a precision parameter,

k_1 is the number of inner iterations and k_2 is the number of outer iterations,

Set $k_1 = 0$, $k_2 = 0$.

1. If (X,Z,y) is an $\varepsilon -$ approximate optimal solution, Stop, Output: $X,Z,y)$.
2. If $t(X,Z,y) \leq 0$, Goto 6.
3. Find new $(X^*,Z^*,y^*) = \mathrm{PDIM}(X,Z,y)$, $k_1 := k_1 + 1$.
4. If (X^*,Z^*,y^*) is an $\varepsilon -$ approximate optimal solution, Stop, Output: (X^*,Z^*,y^*) is the solution, end.
5. If $t(X^*,Z^*,y^*) \leq 0.99\ (X^*,Z^*,y^*)$, Set $(X,Z,y) = (X^*,Z^*,y^*)$,Goto Step 3.
6. Find new $t(X^*,Z^*,y^*) = \ \mathrm{PDEM}(X,Z,y)$, $k_2 := k_2 + 1$.
7. If $t(X^*,Z^*,y^*) \leq\ \beta$, Stop, Output:$(X^*,Z^*,y^*)$ is the solution, end.
8. Goto Step 6.

The implementation of this algorithm makes a new version of solver of SDP programs.

5 Numerical experiments

A Now, we will describe the computational experience that we have done to compare the new version of our algorithm and the classical one. We use the library SDPLIB [32], it is collection of semidefinite programming test problems. All problems are stored in SDPA format [33-34]. From the collection SDPLIB, we are particularly interested to solve the Max-cut problem. It is one of many NP-hard graph theory problems, which attracted many researchers over the years. However, there is almost no hope in finding a polynomial time algorithm for max-cut problem, various heuristics, or combination of optimization and heuristic methods have been developed to solve this problem. We use its relaxation to produce an approximate solution to the max-cut problem. This approximate solution, however, can be integrated in the branch and bound algorithm to resolve the problem to optimality.

In this section, we first define the max-cut problem, and then we present its SDP formulation. We generate big instances for particular Max cut problem and we use the instances from the SDPLIB sets.

5.1 Max-cut Problem

A cut in a weighted undirected graph $G_w = (V, E)$, is defined as partition of the vertices of Ginto two sets; and the weight of a cut is the summation of weights of the edges that has an end point in each set (i.e. the edges that connect vertices of one set to the vertices of the other). Trivially, one can define the max-cut problem as the problem of finding a cut in $G$with maximum weight [35].

a. Notation

In this paper, $G = (V, E)$stands for a weighted undirected graph. $V = \{1, \ldots, n\}$is the set of nodes andE is edge set. We consider the weight on edge (i, j), for $(i, j) \in E$with $w_{ij} = w_{ji}$. We assume that $w_{ij} \geq 0$ for all $(i, j) \in E$.

5.3 Formulation of the Max-cut Problem

In this section, we first model the max-cut problem, and then we show how we can obtain its semidefinite relaxation. This relaxation is convex and semidefinite. Let us assign a variable $x_i$to each node of $G = (V, E)$, and define this variable as follows: $x_i = 1$ if the ith node is $\in Q$ and $x_i = -1$if the ith node is in $\bar{Q}$, where Q is a subset of$V$and $\bar{Q}$ is the complement of Q.

Now, we can model the max-cut problem as:

$$MC = \begin{cases} maximize \ \frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) \\ subject\ x_i \ \in \{-1, 1\},\ i = 1 \ldots n \end{cases} \quad \textbf{(7)}$$

The feasible region of model **(7)** obviously defines a cut, so we only need to show that the objective function gives the value of this cut. Notice that for any i and j,$(1 - x_i x_j) = 0$ if nodesx_i and x_j are in the same set, and 2 otherwise. This means that $\sum_{i<j} w_{ij}(1 - x_i x_j)$ is twice as much as the weight of the cut. Dividing this value by 2 gives the weight of the cut. It is worth mentioning here that we can write the same objective function as $\frac{1}{4}\sum_{i<j} w_{ij}(1 - x_i x_j)$. Model **(7)** can be translated into vector notation as follows:

$$MC = \begin{cases} maximize \ \frac{1}{2}x^T L x \\ subject\ to\ x^2 = 1, i = 1 \ldots n \end{cases} \quad \textbf{(8)}$$

Where L is the weighted Laplacian $n * n$ matrix of the graph G with n nodes. For more details from the computing of L see [37].

Lets define a new variable $X = xx^T$. It is easy to show that $X \in S_+^n$ with $X_{ij} = x_i x_j, i = 1 \dots n, j = 1 \dots n$. $x_i \in \{-1, 1\}$is equivalent to $X_{ii} = 1, i = 1 \dots n$. As $X = x^T Lx = L \bullet X$, now we can write model **(8)** as follows:

$$MC = \begin{cases} \text{maximize } \frac{1}{4} L \bullet X \\ \text{subject to } X_{ii} = 1, i = 1 \dots n \\ Rank(X) = 1 \\ X \in S_+^n \end{cases} \quad \textbf{(9)}$$

This problem **(9)** is still hard to solve because of the rank constraint. We Relax the problem by deleting this constraint, (we get elliptope figure **1** above). We obtain the following relaxation:

$$MC\ Relax = \begin{cases} \text{maximize } \frac{1}{4} L \bullet X \\ \text{subject to } X_{ii} = 1, i = 1 \dots n \\ X \in S_+^n \end{cases} \quad \textbf{(10)}$$

We see that MC Relax provides an upper bound on MC. To obtain the SDP format **(2)** and **(3)** of **(10)**, we have:

$$SDP = \begin{cases} p^* = \text{minimize } A_0 \bullet X \\ \text{subject to } A_i \bullet X = b_i, i = 1 \dots m \\ X \in S_+^n \end{cases} \quad \textbf{(11)}$$

And its dual :

$$DSDP = \begin{cases} d^* = \text{maximize } b^t y \\ \text{subject to } Z = A_0 - \sum_{i=1}^m A_i y_i \\ Z \in S_+^n \end{cases} \quad \textbf{(12)}$$

Where $A_0 = \frac{-1}{4} L$.

Model **(11)** and its dual **(12)** is an SDP problem, which can efficiently be solved in polynomial-time, and gives us an upper bound on the max-cut problem. This relaxation of max-cut is well known and studied e.g. in [7, 29, 35]. Goemans and Williamson [7] have recently shown that the optimal value of this relaxation is approximate to the value of the maximum cut with little gap. For the tests, firstly we consider random graphs. We consider a complete graph, the variable Xcan be interpreted as being defined on the edge set of the graph. Secondly, we test the max-cut instances of the SDPLIB set.

5.3 Numerical results

Here is a brief description of the Used Tools. The computational tests were performed in Intel(R) Coreâ„ i5 2.50 GHz with 4Go memory under Linux 11. To implement the new predictor-corrector variant we used the 6.0 version of the source code of the package SDPA by Makoto Yamashita, Katsuki Fujisawa, Mituhiro Fukuda, Kazuhiro Kobayashi, Kazuhide Nakata and Maho Nakata [34]. The code was modified to achieve two main purposes: it was adapted to implement the EIPM variant and it was optimized to become faster and more robust. We use the library LAPACK [36] for dense case and the library Lanczos [36] for sparse case.

To compare the performance of the algorithm corresponding to PDIM in SDPA classic and IEPM in new version of SDPA, we generated automatically big instances of max-cut problem and then we test the SDPLIB max-cut problem (maxG11, maxG32, maxG51). The motivation to consider this example is to show the effectiveness and the realizability of our procedure and to generate big instances. These results are preliminaries.

Table I below presents the results of our experimentations. In this table, Problem is the problem to solve, m is the number of variables, inner iter is the number of iterations with PDIM (SDPA classic), outer iter is the number of iterations with IEPM (SDPA new variant), and CPU is the time in second required to solve the problem.

Table 1. Comparison of the number of iterations and CPU with SDPA classic and SDPA new variant of max-cut problem

		SDPA Classic		SDPA new variant		
Problem	n	Inner iter	CPU	Inner iter	Outer iter	CPU
Max200	200	10	0.111	6	3	0.105
Max400	400	12	3.356	7	6	2.950
Max500	500	14	5.378	11	3	4.512
Max700	700	13	18.560	7	3	16.987
Max800	800	11	20.525	7	4	19.109
Max900	900	14	40.568	6	7	40.319
Max1000	1000	13	134.256	10	3	134.165
Max1500	1500	15	295.235	12	3	290.191
MaxG11	800	10	47.23	4	4	47.200
MaxG32	2000	17	682.235	7	8	679.652
MaxG51	1000	17	46.25	12	4	42.257

The various numerical experiments show that the IEPM is efficient in practice. The PDIM algorithm is known to converge after very little iterations, but with large computation cost for each one. The cost of the iteration is improved with the IEPM and this is proved by the decrease of the CPU. The solver is particularly suitable for large sparse problems. Our experimentation is preliminary but promising.

6 Conclusions

This paper presents SDP relaxation and its resolution of max-cut problem with a new algorithm. We present comparative computational study between the classic PDIM implemented in SDPA solver and our combination of the PDIM and PDEM.The numerical testing of the interior-exterior point method (IEPM) has shown that the interior point method and the exterior point method are capable of augmenting each other. Their combined performance is better than either method can achieve individually. Our proposition has three profits; the first one is the exploitation of the performance of each method. Secondly, the tightly bound of the SDP relaxation of max-cut and the numerical performance obtained from the combination of the two methods give us the opportunity to integrate this relaxation in a branch and bound algorithm to resolve this problem and other nonlinear problem to optimality. Finally, we can already use other penalty functions and test them to augment the efficacy of the implementation.

References

1. Alizadeh, F. (1995). Interior point methods in semidefinite programming with application to combinatorial optimization. *SIAM Journal on Optimization, 5*, 1351-1379.
2. Alizadeh, F., Haberly, J.-P., & Overton, M.-L. (1998). Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability, and numerical results. *SIAM Journal on Optimization, 8*(3), 746-768.
3. Alizadeh, F., Haberly, J.-P., & Overton, M.-L. (1994). Primal-dual interior point methods for semidefinite programming. *Technical Report*.
4. Benterki, D., Crouzeix, J.-P., & Merikhi, B. (2003). A numerical implementation of an interior point method for semi-definite programming. *Pesquisa Operacional, 23*(1), 499-511.
5. Boyd, S., Ghaoui, L.-E., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. Society for Industrial and Applied Mathematics.
6. Nesterov, Y.-E., & Nemirovskii, A.-S. (1993). *Interior point polynomial methods in convex programming: Theory and algorithms*. SIAM Publications.
7. Goemans, M.-X., & Williamson, D.-P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the Association for Computing Machinery, 42*(6), 1115-1145.

8. Wolkowicz, H., Saigal, R., & Vandenberghe, L. (2000). *Handbook of semidefinite programming: Theory, algorithms, and applications*. Kluwer Academic Publishers.
9. Nakata, M., Nakatsuji, H., Ehara, M., Fukuda, M., Nakata, K., & Fujisawa, K. (2001). Variational calculations of fermion second-order reduced density matrices by semidefinite programming algorithm. *Journal of Chemical Physics, 114*, 8282-8292.
10. Nakata, M., Ehara, M., & Nakatsuji, H. (2002). Density matrix variational theory: Application to the potential energy surfaces and strongly correlated systems. *Journal of Chemical Physics, 116*, 5432-5439.
11. Todd, M.-J. (2001). Semidefinite optimization. *Acta Numerica, 10*, 515-560.
12. Nesterov, Y.-E., & Nemirovskii, A.-S. (1990). Optimization over positive semidefinite matrices: Mathematical background and user manual. *Technical Report*, Central Economic and Mathematical Institute, USSR Acad. Sci., Moscow.
13. Vandenberghe, L., & Boyd, S. (1995). Primal-dual potential reduction method for problems involving matrix inequalities. *Mathematical Programming, 69*, 205-236.
14. Griva, I. (2004). Numerical experiments with an interior-exterior point method for nonlinear programming. *Computational Optimization and Applications, 29*, 173-195.
15. Ben-Tal, A., & Nemirovskii, A. (2001). *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*. SIAM.
16. Wright, S.-J. (1997). Stability of augmented system factorizations in interior-point methods. *SIAM Journal on Matrix Analysis and Applications, 18*, 191-222.
17. Polyak, R. (1992). Modified barrier functions. *Mathematical Programming, 54*, 177-222.
18. Polyak, R. (2002). Nonlinear rescaling vs. smoothing technique in convex optimization. *Mathematical Programming, Series A, 92*, 197-235.
19. Griva, I., & Polyak, R. (2008). 1.5-Q-superlinear convergence of an exterior-point method for constrained optimization. *Journal of Global Optimization, 40*(4), 679-695.
20. Polyak, R., & Griva, I. (2004). Primal-dual nonlinear rescaling method for convex optimization. *Journal of Optimization Theory and Applications*, *122*(1), 111-156.
21. Fujisawa, K., & Kojima, M. (1995). SDPA (Semidefinite Programming Algorithm) user's manual. *Technical Report B-308*, Tokyo Institute of Technology.