McMaster University

Advanced Optimization Laboratory

Title:

Interior Point and Semidefinite Approaches in Combinatorial Optimization

Authors:

Kartik Krishnan, Tamás Terlaky

AdvOL-Report No. 2004/2

March 2004, Hamilton, Ontario, Canada

Interior Point and Semidefinite Approaches in Combinatorial Optimization

Kartik Krishnan * and Tamás Terlaky †
Department of Computing & Software
McMaster University
Hamilton, Ontario, L8S 4K1

Dedicated to the memory of Jos F. Sturm
April 5, 2004

Abstract

Interior-point methods (IPMs), originally conceived in the context of linear programming have found a variety of applications in integer programming, and combinatorial optimization. This survey presents an up to date account of IPMs in solving NP-hard combinatorial optimization problems to optimality, and also in developing approximation algorithms for some of them. The surveyed approaches include non-convex potential reduction methods, interior point cutting plane methods, the generic interior point method for the semidefinite programming (SDP) problem, branch and cut approaches based on SDP relaxations, approximation algorithms based on SDP formulations, and finally methods employing successive convex approximations of the underlying combinatorial optimization problem.

Keywords: Interior Point Methods, Integer Programming and Combinatorial Optimization, Semidefinite Programming, Branch and cut, Approximation Algorithms, Successive convex approximations.

*Email: kartik@optlab.mcmaster.ca
†Email: terlaky@mcmaster.ca

1 Introduction

Optimization problems seem to divide naturally into two categories: those with *continuous* variables, and those with *discrete* variables, which we shall hereafter call combinatorial problems. In continuous problems, we are generally looking for a set of real numbers or even a function; in combinatorial optimization, we are looking for certain objects from a finite, or possibly countably infinite set, typically an integer, graph etc. These two kinds of problems have different flavors, and the methods for solving them are quite different too. In this survey paper on interior point methods (IPMs) in combinatorial optimization, we are in a sense at the boundary of these two categories, i.e., we are looking at IPMs, that represent continuous approaches towards solving combinatorial problems usually formulated using discrete variables.

To better understand why one would adopt a continuous approach to solving discrete problems, consider as an instance the linear programming (LP) problem. The LP problem amounts to minimizing a linear functional over a polyhedron, and arises in a variety of applications in combinatorial optimization. Although the LP is in one sense a continuous optimization problem, it can be viewed as a combinatorial problem. The set of candidate solutions are extreme points of the underlying polyhedron, and there are only a finite (in fact combinatorial) number of these. Before the advent of IPMs, the classical algorithm for solving LP's was the simplex algorithm. The simplex algorithm can be viewed as a combinatorial approach to solving an LP, and it deals exclusively with the extreme point solutions; at each step of the algorithm the next candidate extreme point solution is chosen in an attempt to improve some performance measure of the current solution, say the objective value. The improvement is entirely guided by local search, i.e., the procedure only examines a neighboring set of configurations, and greedily selects one that improves the current solution. As a result the search is quite myopic, with no consideration given to evaluate whether the current move is actually useful globally. The simplex method simply lacks the ability for making such an evaluation. Thus, although, the simplex method is quite an efficient algorithm in practice, there are specially devised problems on which the method takes a disagreeably exponential number of steps. In contrast, all polynomial-time algorithms for solving the LP employ a continuous approach. These include the ellipsoid method [55], or IPMs that are subsequent variants of the original method of Karmarkar [73]. It must be emphasized here that IPMs have both better complexity bounds than the ellipsoid method (we will say more on this in the subsequent sections), and the further advantage of being very efficient in practice. For LP

it has been established that for very large, sparse problems IPMs often outperform the simplex method. IPMs are also applicable to more general conic (convex) optimization problems, with efficiently computable self-concordant barrier functions (see the monographs by Renegar [136] and Nesterov & Nemirovskii [113]). This includes important classes of optimization problems such as second order cone programming (SOCP) and semidefinite programming (SDP). For such problems, IPMs are indeed the algorithm of choice.

We now present the underlying ideas behind primal-dual IPMs (Roos et al. [137], Wright [159], Ye [163], and Andersen et al. [3]) the most successful class of IPMs in computational practice. For ease of exposition, we consider the LP problem. We will later consider extensions to convex programming problems, especially the SDP, in Section 4.2. Consider the standard linear programming problem (LP)

$$\begin{array}{lrcl} \min & c^T x & & \\ \text{s.t.} & Ax & = & b, \\ & x & \geq & 0, \end{array} \tag{LP}$$

with dual

$$\begin{array}{lrcl} \max & b^T y & & \\ \text{s.t.} & A^T y + s & = & c, \\ & s & \geq & 0, \end{array} \tag{LD}$$

where m and n represent the number of constraints and variables in the primal problem (LP), with $m < n$. Also, c, x, and s are vectors in $\mathcal{R}^n$, b and y are vectors in $\mathcal{R}^m$, and A is an $m \times n$ matrix with full row rank. The constraints $x, s \geq 0$ imply that these vectors belong to $\mathcal{R}^n_+$, i.e., all their components are non-negative. Similarly, $x > 0$ implies that $x \in \mathcal{R}^n_{++}$ (the interior of $\mathcal{R}^n_+$), i.e., all components of x are strictly positive.

The optimality conditions for LP include primal and dual feasibility and the complementary slackness conditions, i.e.,

$$\begin{array}{rcl} Ax & = & b, \ x \geq 0, \\ A^T y + s & = & c, \ s \geq 0, \\ x \circ s & = & 0, \end{array} \tag{1}$$

where $x \circ s = (x_i s_i)$, $i = 1, \ldots, n$ is the Hadamard product of the vectors x and s.

Consider perturbing the complementarity slackness conditions in (1) to $x \circ s = \mu e$, where e is the all-ones vector and $\mu > 0$ is a given scalar. Neglecting the inequality

constraints in (1) for the moment this gives the following system:

$$\begin{aligned} Ax &= b, \\ A^T y + s &= c, \\ x \circ s &= \mu e. \end{aligned} \tag{2}$$

A typical feasible primal-dual IPM for LP starts with a strictly feasible (x, y, s) solution in $\mathcal{R}^n_{++}$, i.e. $x, s > 0$. The perturbed system (2) has an unique solution (x_μ, y_μ, s_μ) for each $\mu > 0$. Moreover, the set $\{(x_\mu, y_\mu, s_\mu), \mu > 0\}$, also called the *central path*, is a smooth, analytic curve converging to an optimal solution (x^*, y^*, s^*) as $\mu \to 0$. In fact, this limit point is in the relative interior of the optimal set, and is a strictly complementary solution, i.e., $x^* + s^* > 0$ and $x^* \circ s^* = 0$.

If we solve (2) by Newton's method, we get the following linearized system

$$\begin{aligned} A\Delta x &= 0, \\ A^T \Delta y + \Delta s &= 0, \\ s\Delta x + x\Delta s &= \mu e - x \circ s. \end{aligned}$$

This system has a unique solution, namely

$$\begin{aligned} \Delta y &= (AXS^{-1}A^T)^{-1}(b - \mu A s^{-1}), \\ \Delta s &= -A^T \Delta y, \\ \Delta x &= \mu s^{-1} - x - x \circ s^{-1} \circ \Delta s, \end{aligned} \tag{3}$$

where $X = \mathrm{Diag}(x)$ and $S = \mathrm{Diag}(s)$ are diagonal matrices, whose entries are the components of x and s, respectively. Since the constraints $x, s > 0$ were neglected in (2), one needs to take *damped* Newton steps. Moreover, the central path equations (2) are nonlinear and so it is impossible to obtain the point (x_μ, y_μ, s_μ) on the central path via damped Newton iterations alone. One requires a proximity measure $\delta(x, s, \mu)$ (see Roos et al. [137], Wright [159]) that measures how *close* the given point (x, y, s) is to the corresponding point (x_μ, y_μ, s_μ) on the central path. Finally, IPMs ensure that the sequence of iterates $\{(x, y, s)\}$ remain in some neighborhood of the central path by requiring that $\delta(x, s, \mu) \leq \tau$ for some $\tau > 0$, where τ is either an absolute constant or may depend on n.

We are now ready to present a generic IPM algorithm for LP.

Generic Primal-Dual IPM for LP

Input: A, b, c, a starting point (x^0, y^0, s^0) satisfying the interior point condition (see Roos et al. [137] and Wright [159]), i.e., $x^0, s^0 > 0$, $Ax^0 = b$, $A^T y^0 + s^0 = c$, and

$x^0 \circ s^0 = e$, a barrier parameter $\mu = 1$, a proximity threshold $\tau > 0$ such that $\delta(x^0, s^0, \mu) \leq \tau$, and an accuracy parameter $\epsilon > 0$.

1. Reduce the barrier parameter μ.
2. **If** $\delta(x, s, \mu) > \tau$ compute $(\Delta x, \Delta y, \Delta s)$ using (3).
3. Choose some $\alpha \in (0, 1]$ such that $x + \alpha\Delta x, s + \alpha\Delta s > 0$, and proximity $\delta(x, s, \mu)$ appropriately reduced.
4. Set $(x, y, s) = (x + \alpha\Delta x, y + \alpha\Delta y, s + \alpha\Delta s)$.
5. **If** the duality gap $x^T s < \epsilon$ **then** stop,
   else if $\delta(x, s, \mu) \leq \tau$ **goto** step 1,
   else goto step 2.

We can solve an LP problem with rational data, to within an accuracy $\epsilon > 0$, in $O(\sqrt{n}\log(\frac{1}{\epsilon}))$ iterations (see Roos et al. [137]) for more details). This is the best iteration complexity bound for a primal-dual interior point algorithm. Most combinatorial optimization problems, other than flow and matching problems are NP-complete, all of which are widely considered unsolvable in polynomial time (see Garey & Johnson [46] and Papadimitriou & Steiglitz [119] for a discussion on intractability and the theory of NP completeness). We are especially interested in these problems. One way of solving such problems is to consider successively strengthened convex relaxations (SDP/SOCP) of these problems in a branch-cut framework, and employing IPMs to solving these relaxations. On the other hand, semidefinite programming (SDP) has been applied with a great deal of success in developing approximation algorithms for various combinatorial problems, the showcase being the Goemans & Williamson [47] approximation algorithm for the maxcut problem. The algorithm employs an SDP relaxation of the maxcut problem which can be solved by IPMs, followed by an ingenious randomized rounding procedure. The approximation algorithm runs in polynomial time, and has a worst case performance guarantee. The technique has subsequently been extended to other combinatorial optimization problems.

We introduce two canonical combinatorial optimization problems, namely the maxcut and maximum stable set problems, that will appear in the approaches mentioned in the succeeding sections.

1. **Maxcut Problem:** Let $G = (V, E)$ denote an edge weighted undirected graph without loops or multiple edges. Let $V = \{1, \ldots, n\}$, $E \subset \{\{i, j\} : 1 \leq i < j \leq$

$n\}$, and $w \in \mathcal{R}^{|E|}$, with $\{i,j\}$ the edge with endpoints i and j, with weights w_{ij}. We assume that $n = |V|$, and $m = |E|$. For $S \subseteq V$, the set of edges $\{i,j\} \in E$ with one endpoint in S and the other in $V \backslash S$ form the cut denoted by $\delta(S)$. We define the weight of the cut as $w(\delta(S)) = \sum_{\{i,j\} \in \delta(S)} w_{ij}$. The maximum cut problem, denoted as (MC), is the problem of finding a cut for which the total weight is maximal.

2. **Maximum Stable Set Problem:** Given a graph $G = (V, E)$, a subset $V' \subset V$ is called a stable set, if the induced subgraph on V' contains no edges. The maximum stable set problem, denoted by (MSS), is to find the stable set of maximum cardinality.

It must be mentioned that although (MC) and (MSS) are NP-complete problems, the maxcut problem admits an approximation algorithm, while no such algorithms exist for the maximum stable set problem unless P = NP (see Arora & Lund [10] for a discussion on the hardness of approximating various NP-hard problems).

This paper is organized as follows: Section 2 deals with non-convex potential function minimization, among the first techniques employing IPMs in solving difficult combinatorial optimization problems. Section 3 deals with interior point cutting plane algorithms, especially the analytic center cutting plane method (ACCPM), and the volumetric center method. These techniques do not require a knowledge of the entire constraint set, and consequently can be employed to solve integer programs (IPs) with exponential or possibly infinite number of constraints. They can also be employed as a certificate to show certain IPs can be solved in polynomial time, together with providing the best complexity bounds. Section 4 discusses the complexity of SDP, and provides a generic IPM for SDP. This algorithm is employed in solving the SDP formulations and relaxations of integer programming problems discussed in the succeeding sections. Although IPMs are the algorithms of choice for an SDP, they are fairly limited in the size of problems they can handle in computational practice. We discuss various first order methods that exploit problem structure, and have proven to be successful in solving large scale SDP's in Section 5. Section 6 discusses branch and cut SDP approaches to solving IPs to optimality, advantages and issues involved in employing IPMs in branching, restarting, and solving the SDP relaxations at every stage. Section 7 discusses the use of SDP in developing approximation algorithms for combinatorial optimization. Section 8 discusses approaches employing successive convex approximations to the underlying IP, including recent techniques based on polynomial and copositive programming. We wish to emphasize that the

techniques in Section 8 are more of a theoretical nature, i.e., we have an estimate on the number of liftings needed to solve the underlying IP to optimality, however the resulting problems grow in size beyond the capacity of current state of the art computers and software; this is in sharp contrast to the practical branch and cut approaches in Section 6. We conclude with some observations in Section 9, and also highlight some of the open problems in each area.

The survey is by no means complete; it represents the authors biased view of this rapidly evolving research field. The interested reader is referred to the books by Chvátal [30], Papadimitriou & Steiglitz [119], and Schrijver [138] on combinatorial optimization and LP & IP. The books by Roos et al. [137], Wright [159], and Ye [163] contain a treatment of IPMs in linear optimization. A recent survey on SOCP appears in Alizadeh & Goldfarb [1]. Excellent references for SDP include the survey papers by Vandenberghe & Boyd [155], Todd [147], the SDP handbook edited by Wolkowicz et al. [160], and the recent monograph by De Klerk [33]. A repository of recent papers dealing with interior point approaches to solving combinatorial optimization problems appear in the following websites: Optimization Online [115], Interior Points Online [67], and the SDP webpage [64] maintained by Helmberg. Finally, recent surveys by Laurent & Rendl [89] and Mitchell et al. [98] also complement the material in this survey.

2 Non-convex potential function minimization

The non-convex potential function approach was introduced by Karmarkar et al. [74, 75] as a nonlinear approach for solving integer programming problems. Warners et al. [157, 158] also utilized this approach in solving frequency assignment problems (FAP), and other structured optimization problems. We present a short overview of the approach in this section.

Consider the following binary $\{-1, 1\}$ feasibility problem:

$$\text{find } \bar{x} \in \{-1, 1\}^n \text{ such that } \bar{A}\bar{x} \leq \bar{b}. \tag{4}$$

Let $\mathcal{I}$ denote the feasible set of (4). Binary feasibility problems arise in a variety of applications. As an example we can consider the stable set problem on the graph $G = (V, E)$ with $n = |V|$. The constraints $\bar{A}x \leq \bar{b}$ are given by $x_i + x_j \leq 0$, $\{i, j\} \in E$, where the set of $\{-1, 1\}$ vectors $x \in \mathcal{R}^n$ correspond to incidence vectors of stable sets in the graph G, with $x_i = 1$ if node i is in the stable set, and $x_i = -1$ otherwise.

The problem (4) is NP-complete, and there is no efficient algorithm that would solve it in polynomial time. Therefore, we consider the following polytope $\mathcal{P}$, which is a relaxation of $\mathcal{I}$.

$$\mathcal{P} \;=\; \{x \in \mathcal{R}^n : Ax \leq b\},$$

where $A = \begin{pmatrix} \bar{A} \\ I \\ -I \end{pmatrix}$, and $b = \begin{pmatrix} \bar{b} \\ e \\ e \end{pmatrix}$. Here I is the $n \times n$ identity matrix, and e is the vector of all ones. Finding a vector $x \in \mathcal{P}$ amounts to solving an LP problem, and can be done efficiently in polynomial time. Let $\mathcal{P}'$ denote the relative interior of $\mathcal{P}$, i.e., $\mathcal{P}' = \{x \in \mathcal{R}^n : Ax < b\}$. Since $-e \leq x \leq e$, $\forall x \in \mathcal{P}$, we have $x^T x \leq n$, with equality occurring if and only if $x \in \mathcal{I}$. Thus, (4) can also be formulated as the following concave quadratic optimization problem with linear constraints.

$$\begin{array}{ll} \max & \sum_{i=1}^{n} x_i^2 \\ \text{s.t.} & x \in \mathcal{P}. \end{array} \tag{5}$$

Since the objective function of (5) is non-convex, this problem is NP-complete as well. However, the global optimum of (5), when (4) is feasible, corresponds to ± 1 binary solutions of this problem.

Consider now a non-convex potential function $\phi(x)$, where

$$\phi(x) \;=\; \log \sqrt{n - x^T x} - \tfrac{1}{m} \sum_{i=1}^{m} \log s_i$$

and

$$s_i \;=\; b_i - \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, \ldots, m$$

are the slacks in the constraints $Ax \leq b$. We replace (5) in turn by the following non-convex optimization problem

$$\begin{array}{ll} \min & \phi(x) \\ \text{s.t.} & x \in \mathcal{P}. \end{array} \tag{P_ϕ}$$

Assuming $\mathcal{I} \neq \phi$, a simple observation reveals that x^* is a global minimum of (P_ϕ) if and only if $x^* \in \mathcal{I}$. To see this, note that since $\phi(x) = \log \left(\frac{(n - x^T x)^{\frac{1}{2}}}{\Pi_{i=1}^{m} (b_i - a_i^T x)^{\frac{1}{m}}} \right)$, the denominator of the log term of $\phi(x)$ is the geometric mean of the slacks, and is maximized at the analytic center of the polytope $\mathcal{P}$, whereas the numerator is minimized when $x \in \mathcal{I}$, since $-e \leq x \leq e$, $\forall x \in \mathcal{P}$. Karmarkar et al. [74, 75] solve

(P_ϕ) using an interior point method. To start with, we will assume a strictly interior point, i.e., $x^0 \in \mathcal{P}'$. The algorithm generates a sequence of points $\{x^k\}$ in $\mathcal{P}'$. In every iteration we perform the following steps:

1. Minimize a quadratic approximation of the potential function over an inscribed ellipsoid in the feasible region $\mathcal{P}$ around the current feasible interior point, to get the next iterate.

2. Round the new iterate to an integer solution.
   If this solution is feasible the problem is solved,
   else goto step 1.

3. When a local minimum is found, modify the potential function to avoid running into this minimum again, and restart the process.

These steps will be elaborated in more detail in the subsequent sections.

2.1 Non-convex quadratic function minimization

We elaborate on Step 1 of the algorithm in this subsection. This step is an interior point algorithm to solve (P_ϕ). It mimics a trust region method, except that the trust region is based on making good global approximations to the polytope $\mathcal{P}$.

Given $x^k \in \mathcal{P}'$, the next iterate x^{k+1} is obtained by moving in a descent direction Δx from x^k, i.e., a direction such that $\phi(x^k + \alpha\Delta x) < \phi(x^k)$, where α is an appropriate step length. The descent direction Δx is obtained by minimizing a quadratic approximation of the potential function about the current point x^k over the Dikin ellipsoid, which can be shown to be inscribed in the polytope $\mathcal{P}$. The resulting problem (P_r) solved in every iteration is the following:

$$\begin{array}{ll} \min & \frac{1}{2}(\Delta x)^T H(\Delta x) + h^T(\Delta x) \\ \text{s.t.} & (\Delta x)^T A^T S^{-2} A(\Delta x) \leq r^2, \end{array} \qquad (P_r)$$

for some $0 \leq r \leq 1$. Here $S = \text{Diag}(s)$ and H and h are the Hessian, and the gradient of the potential function $\phi(x)$, respectively.

The problem (P_r), a trust region subproblem for some $r_\ell \leq r \leq r_u$, is approximately solved by an iterative binary search algorithm (see Conn et al. [31], Karmarkar [74], Vavasis [156]), in which one solves a series of systems of linear equations of the form

$$(H + \mu A^T S^{-2} A)\Delta x \;\; = \;\; -h,$$

where $\mu > 0$ is a real scalar. This system arises from the first order KKT optimality condition for (P_r). Since there are two iterative schemes at work, we will refer to the iterations employed in solving (P_ϕ) as *outer* iterations, and the iterations employed in solving (P_r) as *inner* iterations. In this terminology, each outer iteration consists of a series of inner iterations. We concentrate on the outer iterations first. Assume for simplicity that (P_r) is solved exactly in every outer iteration for a solution Δx^*. Let us define the *S-norm* of Δx^* as

$$||\Delta x^*||_S \quad = \quad \sqrt{(\Delta x^*)^T A^T S^{-2} A(\Delta x^*)}.$$

Since H is indefinite, the solution to (P_r) is attained on the boundary of the Dikin ellipsoid, giving $r = ||\Delta x^*||_S$. On the other hand, the computed direction Δx^* need not be a descent direction for $\phi(x)$, since the higher order terms are neglected in the quadratic approximation. Karmarkar et al. [75] however show that a descent direction can always be computed provided the radius r of the Dikin ellipsoid is decreased sufficiently. In the actual algorithm in each outer iteration we solve (P_r) for a priori bound (r_ℓ, r_u) on r, and if the computed Δx^* is not a descent direction, we reduce r_u, and continue with the process. Moreover, we stop these outer iterations with the conclusion that a *local minimum* is attained for (P_ϕ) as soon as the upper bound r_u falls below a user specified tolerance $\epsilon > 0$.

We now discuss the inner iterations, where a descent direction Δx is computed for (P_r): assuming we are in the kth outer iteration we have as input the current iterate x^k, a multiplier μ, lower and upper bounds (r_ℓ, r_u) on r, a flag ID, which is false initially, and is set to true if during any inner iteration an indefinite matrix $(H + \mu A^T S^{-2} A)$ is encountered. The algorithm computes $\Delta x^*(\mu)$ by solving the following system of linear equations.

$$\Delta x^*(\mu) \quad = \quad -(H + \mu A^T S^{-2} A)h.$$

We are assuming that $\mu > 0$ is chosen so that $(H + \mu A^T S^{-2} A)$ is positive definite. If this is not true for the input μ, the value of μ is increased, and the flag ID is set to true. This process is repeated until we have a nonsingular coefficient matrix. Once $\Delta x^*(\mu)$ is computed, we compute $r^* = ||\Delta x^*(\mu)||_S$. One of the following four cases can then occur:

1. If $r^* \leq r_\ell$ and ID is false, an upper bound on μ has been found; set $\mu_{\text{upper}} = \mu$, and μ is decreased either by dividing it by a constant > 1, or if a lower bound μ_{lower} on μ already exists by taking the geometric mean of the current μ and μ_{lower}. The direction Δx is recomputed with this new value of μ.

2. If $r^* \geq r_u$, a lower bound on μ has been found; set $\mu_{\text{lower}} = \mu$, and μ is increased, either by multiplying it with some constant > 1, or if μ_{upper} already exists, by taking the geometric mean of μ and μ_{upper}. The direction Δx is recomputed with this new value of μ.

3. If $r^* \leq r_\ell$, and ID is true, decreasing μ will still lead to an indefinite matrix; in this case the lower bound r_ℓ is reduced, and the direction Δx is recomputed.

4. Finally, if $r_\ell \leq r^* \leq r_u$, the direction Δx is accepted.

2.2 Rounding schemes and local minima

We discuss Steps 2 and 3 of the algorithm in this subsection. These include techniques to round the iterates to ± 1 vectors, and schemes to modify the potential function to avoid running into the same local minima more than once.

1. **Rounding schemes:** In Step 2 of the algorithm, the current iterate x^k is rounded to a ± 1 solution $\bar{x}$. Generally these rounding techniques are specific to the combinatorial problem being solved, but two popular choices include:

   (a) Round to the nearest ± 1 vertex, i.e.,

   $$\begin{aligned} \bar{x_i} &= 1 \quad \text{if } x_i^k \geq 0; \\ \bar{x_i} &= -1 \quad \text{if } x_i^k < 0. \end{aligned}$$

   (b) We can obtain a starting point x^0 by solving a linear relaxation of the problem, using an IPM. The rounding can then be based on a coordinate-wise comparison of the current solution point with the starting point, i.e.,

   $$\begin{aligned} \bar{x_i} &= 1 \quad \text{if } x_i^k \geq x_i^0; \\ \bar{x_i} &= -1 \quad \text{if } x_i^k < x_i^0. \end{aligned}$$

2. **Avoiding the same local minima:** After a number of iterations, the interior point algorithm may lead to a local minimum. One way to avoid running into the same local minimum twice is the following: Let $\bar{x}$ be the rounded ± 1 solution and suppose $\bar{x} \notin \mathcal{I}$. It can be easily seen that

   $$\begin{aligned} \bar{x}^T y &= n, \qquad \text{for } y = \bar{x} \\ \bar{x}^T y &\leq n - 2, \quad \forall y \in \{y \in \mathcal{R}^n : y_i \in \{-1, 1\}, y \neq \bar{x}\}. \end{aligned}$$

Thus we can add the cut $\bar{x}^T y \leq n-2$ without cutting off any integer feasible solution. After adding the cut the process is restarted from the analytic center of the new polytope. Although there is no guarantee that we won't run into the same local minimum again, in practice, the addition of the new cut changes the potential function and alters the trajectory followed by the algorithm.

Warners et al. [157, 158] consider the following improvement in the algorithm arising from the choice of a different potential function: For the potential function $\phi(x)$ discussed earlier in the section, the Hessian H at the point x^k is given by

$$\begin{aligned} H &= \nabla^2 \phi(x^k) \\ &= -\tfrac{1}{f_0} I - \tfrac{2}{f_0^2} x^k x^{k^T} + \tfrac{1}{n} A^T S^{-2} A. \end{aligned}$$

For a general x^k this results in a dense Hessian matrix, due to the outer product term $x^k x^{k^T}$. This increases the computational effort in obtaining Δx since we have now to deal with a dense coefficient matrix. The sparsity of A can be utilized by employing rank 1 updates. Instead, Warners et al. [157, 158] introduce the potential function.

$$\phi_w(x) = (n - x^T x) - \sum_{i=1}^{m} w_i \log s_i,$$

where $w = (w_1, \ldots, w_m)^T$ is a nonnegative weight vector. In this case the Hessian $H_w = -2I + A^T S^{-1} W S^{-1} A$, where $W = \mathrm{Diag}(w)$. Now H_w is a sparse matrix, whenever the product $A^T A$ is sparse, and this fact can be exploited to solve the resulting linear system more efficiently. The weights $w_i^k \to 0$ during the course of the algorithm. Thus, initially when $w^k > 0$, the iterates x^k avoid the boundary of the feasible region, but subsequently towards optimality, these iterates approach the boundary, as any ± 1 feasible vector is at the boundary of the feasible region.

The technique has been applied to a variety of problems including satisfiability [68], set covering [75], inductive inference [69], and variants of the frequency assignment problem [157, 158].

3 Interior point cutting plane methods

In this section we consider interior point cutting plane algorithms, especially the analytic center cutting plane method (ACCPM) (Goffin & Vial [51] and Ye [163]) and the volumetric center method (Vaidya [154] and Anstreicher [7, 9, 8]). These techniques are originally designed for convex feasibility or optimization problems.

To see how this relates to combinatorial optimization, consider the maxcut problem discussed in Section 1. The maxcut problem can be expressed as the following $\{-1, 1\}$ integer programming problem.

$$\begin{array}{lrcll} \max & & w^T x & & \\ \text{s.t.} & x(C\backslash F) - x(F) & \leq & |C| - 2 & \forall \text{ circuits } C \subseteq E \\ & & & & \text{and all } F \subseteq C \text{ with } |F| \text{ odd}, \\ & & x & \in & \{-1, 1\}^m. \end{array} \tag{6}$$

Here w_{ij} represents the weight of edge $\{i, j\} \in E$. Let CHULL(G) represent the convex hull of the feasible set of (6). We can equivalently minimize the linear functional $w^T x$ over CHULL(G), i.e., we have replaced the maxcut problem via an equivalent convex optimization problem. Unfortunately, an exact description of CHULL(G) is unknown, and besides this may entail an exponential set of linear constraints. However, we can solve such problems by using interior point cutting plane methods discussed in this section.

Although we are primarily interested in optimization, we motivate these cutting plane methods via the convex feasibility problem; we will later consider extensions to optimization. Let $\mathcal{C} \subseteq \mathcal{R}^m$ be a convex set. We want to find a point $y \in \mathcal{C}$. We will assume that if the set $\mathcal{C}$ is nonempty then it contains a ball of radius ϵ for some tolerance $\epsilon > 0$. Further, we assume that $\mathcal{C}$ is in turn contained in the m dimensional unit hypercube given by $\{y \in \mathcal{R}^m : 0 \leq y \leq e\}$, where e is the all ones vector. We also define $L = \log \frac{1}{\epsilon}$.

Since each convex set is the intersection of a (possibly infinite) collection of halfspaces, the convex feasibility problem is equivalent to the following (possibly semi-infinite) linear programming problem.

$$\text{Find } y \text{ satisfying } A^T y \leq c,$$

where A is a $m \times n$ matrix with independent rows, and $c \in \mathcal{R}^n$. As discussed earlier, the value of n could be infinite. We assume we have access to a *separation oracle*. Given $\bar{y} \in \mathcal{R}^m$, the oracle either reports that $\bar{y} \in \mathcal{C}$, or it will return a *separating hyperplane* $a \in \mathcal{R}^m$ such that $a^T y \leq a^T \bar{y}$ for every $y \in \mathcal{C}$. Such a hyperplane which passes through the query point $\bar{y} \notin \mathcal{C}$ will henceforth be referred to as a *central* cut. A weakened version of this cutting plane, hereafter referred to as a *shallow* cut, is $a^T y \leq a^T \bar{y} + \beta$, for some $\beta > 0$.

Generic cutting plane algorithm

Input: Let $\mathcal{P} \supseteq \mathcal{C}$ be a computable convex set.

1. Choose $\bar{y} \in \mathcal{P} \subseteq \mathcal{R}^m$.
2. Present $\bar{y}$ to the separation oracle.
3. **If** $\bar{y} \in \mathcal{C}$ we have solved the convex feasibility problem.
4. **Else** use the constraint returned by the separation oracle to update $P = P \cup \{y : a^T y \leq a^T \bar{y}\}$ and **goto** step 2.

We illustrate the concept of an oracle for the maxcut problem. The maxcut polytope CHULL(G) does not admit a polynomial time separation oracle, but this is true for polytopes obtained from some of its faces. One such family of faces are the odd cycle inequalities; these are the linear constraints in (6). These inequalities form a polytope called the *metric* polytope. Barahona & Mahjoub [17] describe a polynomial time separation oracle for this polytope, that involves the solution of n shortest path problems on an auxiliary graph with twice the number of nodes, and four times the number of edges.

The cutting plane approach to the feasibility problem can be extended to convex optimization problems by cutting on a violated constraint when the trial point is infeasible, and cutting on the objective function when the trial point is feasible but not optimal.

Interior point cutting plane methods set up a series of convex relaxations of $\mathcal{C}$, and utilize the analytic and volumetric centers of these convex sets as test points $\bar{y}$, that are computed in polynomial time by using IPMs. The relaxations are refined at each iteration by the addition of cutting planes returned by the oracle; some cuts may even conceivably be dropped. We will assume that each call to the oracle takes unit time.

We discuss the analytic center cutting plane method in Section 3.1, and the volumetric center method in Section 3.2.

3.1 Analytic center cutting plane methods

A good overview on ACCPM appears in the survey paper by Goffin & Vial [51], and the book by Ye [163]. The complexity analysis first appeared in Goffin et al. [50]. The algorithm was extended to handle multiple cuts in Goffin & Vial [52],

and nonlinear cuts in Mokhtarian & Goffin [103], Luo & Sun [93], Sun et al. [145], Toh et al. [152], Oskoorouchi & Goffin [116, 117]. The method has been applied to a variety of practical problems including stochastic programming (Bahn et al. [14]), multicommodity network flow problems (Goffin et al. [49]). A version of the ACCPM software [53] is publicly available. Finally, ACCPM has also appeared recently within a branch-and-price algorithm in Elhedhli & Goffin [41].

Our exposition in this section closely follows Goffin & Vial [51] and Goffin et al. [50]. We confine our discussion to the convex feasibility problem discussed earlier.

For the ease of exposition, we will assume the method approximates $\mathcal{C}$ via a series of increasingly refined polytopes $\mathcal{F}_D = \{y : A^T y \leq c\}$. Here A is an $m \times n$ matrix, $c \in \mathcal{R}^n$, and $y \in \mathcal{R}^m$. We will assume that A has full row rank, and $\mathcal{F}_D$ is bounded with a nonempty interior. The vector of slack variables $s = c - A^T y \in \mathcal{R}^n$, $\forall y \in \mathcal{F}_D$.

The analytic center of $\mathcal{F}_D$ is the unique solution to the following minimization problem.

$$\begin{aligned} \min \quad & \phi_D(s) = -\sum_{i=1}^{n} \log s_i \\ \text{s.t.} \quad & A^T y + s = c, \\ & s > 0. \end{aligned}$$

If we introduce the notion that $F(y) = \phi_D(c - A^T y)$, then the analytic center y^* of $\mathcal{F}_D$ is the minimizer of $F(y)$.

Assuming that $\mathcal{C} \subseteq \{y : 0 \leq y \leq e\}$, the complete algorithm is the following:

Analytic center cutting plane method

Input: Let $\mathcal{F}_D^0 = \{y : 0 \leq y \leq e\}$, and $F_0(y) = -\sum_{i=1}^{m} \log(y_i(1 - y_i))$. Set $y^0 = \frac{e}{2}$.

1. Compute y^k an approximate minimizer of $F_k(y)$.

2. Present y^k to the oracle.
   If $y^k \in \mathcal{C}$ **then** stop,
   else the oracle returns the separating hyperplane with normal a^k passing through y^k. Update
   $$\begin{aligned} \mathcal{F}_D^{k+1} &= \mathcal{F}_D^k \cap \{y : (a^k)^T y \leq (a^k)^T y^k\}, \\ F_{k+1}(y) &= F_k(y) - \log((a^k)^T (y^k - y)). \end{aligned}$$
   Set $k = k + 1$ and **goto** step 1.

The formal proof of convergence of the algorithm is carried out in three steps. We will assume that the algorithm works with exact analytic centers.

1. We first show that a new analytic center can be found quickly after the addition of cuts. This is done in an iterative fashion using damped Newton steps, that are the inner iterations in the algorithm. Goffin & Vial ([51]) show that an analytic center can be found in $O(1)$ iterations when one central cut is added in each iteration. In [52], they also show that it is possible to add p cuts simultaneously, and recover a new analytic center in $O(p\log(p))$ Newton iterations.

2. One then proceeds to establish bounds on the logarithmic barrier function $F(y)$. Let $\bar{y}^k$ be the exact analytic center of the polytope $\mathcal{F}_D^k$, i.e., the minimizer of

$$F_k(y) \;=\; -\sum_{i=1}^{2m+k} \log(c^k - (A^k)^T\bar{y}^k)_i.$$

We now establish upper, and lower bounds on $F_k(\bar{y}^k)$. If we are not done in the kth iteration, the polytope $\mathcal{F}_D^k$ still contains a ball of radius ϵ. If $\bar{y}$ is the center of this ball, then we have $\bar{s} = c - \mathcal{A}^T\bar{y} \geq \epsilon e$, giving

$$\begin{aligned} F_k(\bar{y}^k) &\leq F_k(\bar{y}) \\ &\leq (2m+k)\log(\tfrac{1}{\epsilon}). \end{aligned} \tag{7}$$

This is an upper bound on $F_k(\bar{y}^k)$. We can also obtain a lower bound on $F_k(\bar{y}^k)$ in the following manner. We only outline the main steps, more details can be found in Goffin et al. [50]. Let H_i denote the Hessian of $F(y)$ evaluated at $\bar{y}^i$. We first obtain the bound

$$F_k(\bar{y}^k) \;\geq\; -\tfrac{k}{2}\log\left(\tfrac{1}{k}\sum_{i=1}^{k}(a_i^T H_{i-1}^{-1} a_i)\right) + 2m\log\left(\tfrac{1}{2}\right) \tag{8}$$

by exploiting the following self-concordance property of $F_j(y)$

$$F_j(y) \;\geq\; F_j(\bar{y}^j) + \sqrt{(y-\bar{y}^j)^T H_j (y-\bar{y}^j)} - \log\left(1+\sqrt{(y-\bar{y}^j)^T H_j (y-\bar{y}^j)}\right),$$

and applying this property recursively on $F_k(y)$. The bound is simplified in turn by bounding the Hessian H_i from below by a certain matrix, which is simpler to analyze. This yields the following upper bound on $\sum_{i=1}^{k} a_i^T H_{i-1}^{-1} a_i$

$$\sum_{i=1}^{k} a_i^T H_{i-1}^{-1} a_i \;\leq\; 2m^2\log\left(1+\tfrac{k}{m^2}\right),$$

that is employed in the complexity analysis. Substituting this relation in (8) and simplifying the resulting formulas we have

$$F_k(\bar{y}^k) \;\geq\; -k\log\left(\sqrt{2}\right) + k\log\left(\frac{\frac{k}{m^2}}{\log\left(1+\frac{k}{m^2}\right)}\right) - 2m\log\left(\tfrac{1}{2}\right). \tag{9}$$

3. A comparison of the two bounds (7) and (9) on $F_k(\bar{y}^k)$ yields the following upper bound on the number of outer iterations

$$k \log\left(\frac{\frac{k}{m^2}}{\log\left(1+\frac{k}{m^2}\right)}\right) \leq (2m+k)\log\left(\frac{1}{\epsilon}\right) + k\log\left(\sqrt{2}\right) + 2m\log\left(\frac{1}{2}\right),$$

that provides the proof of global convergence of the algorithm. It is clear from this inequality that the algorithm terminates in a finite number of iterations, since the ratio $\left(\frac{\frac{k}{m^2}}{\log(1+\frac{k}{m^2})}\right)$ tends to infinity as k approaches infinity, i.e., the left hand side grows superlinearly in k. Neglecting the logarithmic terms, an upper bound on the number of outer iterations is given by $O^*(\frac{m^2}{\epsilon^2})$ (the notation O^* means that logarithmic terms are ignored).

The analysis presented above can be extended to approximate analytic centers (see Goffin et al. [50]) to yield a *fully* polynomial time algorithm for the convex feasibility problem. The ACCPM algorithm is not polynomial, since the complexity is polynomial in $\frac{1}{\epsilon}$ not $\log(\frac{1}{\epsilon})$. There is a variant of ACCPM due to Atkinson & Vaidya [11] (also see Mitchell [96] for an easier exposition) which is polynomial with a complexity bound of $O(mL^2)$ calls to the oracle, but the algorithm requires dropping constraints from time to time, and also weakening the cuts returned by the oracle making them shallow. In the next section, we will discuss the volumetric center method which is a polynomial interior point cutting plane method, with a better complexity bound than ACCPM for the convex feasibility problem.

3.2 Volumetric center method

The volumetric center method is originally due to Vaidya [154], with enhancements and subsequent improvements in Anstreicher [7, 8, 9] and Mitchell & Ramaswamy [100].

The complexity of the volumetric center algorithm is $O(mL)$ calls to the oracle, and either $O(mL)$ or $O(m^{1.5}L)$ approximate Newton steps depending on whether the cuts are shallow or central. The complexity of $O(mL)$ calls to the separation oracle is optimal - see Nemirovskii & Yudin [110].

As in Section 3, we approximate the convex set $\mathcal{C}$ by the polytope $\mathcal{F}_{\mathcal{D}}(y) = \{y \in \mathcal{R}^m : A^T y \leq c\} \supseteq \mathcal{C}$, where A is an $m \times n$ matrix, and c is an n dimensional vector. Let y be a strictly feasible point in $\mathcal{F}_{\mathcal{D}}$, and let $s = c - A^T y > 0$. The volumetric barrier function for $\mathcal{F}_{\mathcal{D}}$ at the point y is defined as

$$V(y) = \frac{1}{2}\log(\det(AS^{-2}A^T)).$$

The volumetric center $\hat{y}$ of $F_D(y)$ is the point that minimizes $V(y)$. The volumetric center can also be defined as the point y chosen to maximize the volume of the inscribed Dikin ellipsoid $\{z \in \mathcal{R}^m : (z-y)^T(AS^{-2}A^T)(z-y) \leq 1\}$ centered at y.

The volumetric center is closely related to the analytic center of the polytope discussed in Section 3.1. It is closer to the geometrical center of the polytope, than the analytic center.

We also define variational quantities (see Atkinson & Vaidya [11]) for the constraints $A^T y \leq c$ as follows:

$$\sigma_j \quad = \quad \frac{a_j^T (AS^{-2}A^T)^{-1} a_j}{s_j^2}, \quad j = 1, \ldots, n.$$

These quantities give an indication of the relative importance of the inequality $a_j^T y \leq c_j$. The larger the value of σ_j, the more important the inequality. A nice interpretation of these quantities appears in Mitchell [96]. The variational quantities are used in the algorithm to drop constraints that are not important.

We present the complete algorithm below.

Volumetric center IPM

Input: Given $F_D^0(y) = \{y \in \mathcal{R}^m : 0 \leq y \leq e\}$ with $C \subseteq F_D^0(y)$ and $n = 2m$ be the total number of constraints. Set $y^0 = \frac{e}{2}$, and let $0 < \epsilon < 1$ be the desired tolerance.

1. **If** $V(y^k)$ is sufficiently large then stop with the conclusion that $\mathcal{C}$ is empty.
   Else goto step 2.

2. Compute σ_i for each constraint.
   If $\sigma_{\bar{i}} = \min\limits_{i=2m+1,\ldots,n} \sigma_i > \epsilon$ **goto** step 4,
   else goto step 3.

3. Call the oracle at the current point y^k.
   If $y^k \in \mathcal{C}$ then stop,
   else the oracle returns a separating hyperplane with normal a^k passing through y^k.
   Update $F_D^{k+1} = F_D^k \cap \{y : (a^k)^T y \leq (a^k)^T y^k\}$, $n = n + 1$, and **goto** step 5.

4. Drop the $\bar{i}$th constraint from the current feasible set, i.e., $F_D^{k+1} = F_D^k \backslash \{y : a_{\bar{i}}^T y \leq c_{\bar{i}}\}$, update $n = n - 1$, and **goto** step 5.

5. Take a series of damped Newton steps to find a new approximate volumetric center. Set $k = k + 1$ and **goto** step 1.

We note that the box constraints $0 \leq y \leq e$ defining the initial polyhedral approximation are never dropped, and hence the polyhedral approximations have at least $2m$ constraints. In every iteration we either add or drop a constraint. It follows that in the kth iteration, the algorithm must have previously visited Step 4 where we add a constraint at least $\frac{k}{2}$ times, and Step 5 where we drop a constraint on no more than $\frac{k}{2}$ occasions. Else, the number of constraints would fall below $2m$. The formal proof of convergence of the algorithm proceeds in the following way:

1. First, we show that the number of Newton iterations in one call to Step 6 of the algorithm to find an approximate volumetric center is bounded. These are the inner iterations in the algorithm. The condition for a point to be an approximate volumetric center can be expressed as a condition on the norm of the gradient of the volumetric barrier function in the norm given by an approximation to the Hessian of the volumetric barrier function. Formally, a point y is an approximate volumetric center if

$$\beta ||g(y)||_{P(y)^{-1}} \quad \leq \quad \gamma,$$

for some appropriate $\gamma \leq \frac{1}{6}$, where

$$\beta \quad = \quad \min \left\{ (2\sqrt{\sigma_{\bar{i}}} - \sigma_{\bar{i}})^{-\frac{1}{2}}, \sqrt{\tfrac{1+\sqrt{m}}{2}} \right\},$$

$g(y)$, and $P(y)$ are the gradient and an approximation to the Hessian of the volumetric barrier function $V(y)$ at the point y, respectively. In Step 6 one take a series of damped Newton steps of the form $\bar{y} = y + \alpha d$, where $P(y)d = -g(y)$. Anstreicher [8] shows that when a central cut is added in Step 4, then an approximate volumetric center satisfying (1) could be recovered in $O(\sqrt{m})$ Newton steps. In this case, the direction first proposed in Mitchell & Todd [101] is used to move away from the added cut, and the damped Newton iterations described above are used to recover an analytic center. On the other hand, when a cut is dropped in Step 5, Vaidya [154] showed that an approximate volumetric center could be obtained in just one Newton iteration. In the original volumetric barrier [154], Vaidya weakened the cuts returned by the oracle (shallow cuts), and showed that a new approximate volumetric center could be obtained in $O(1)$ Newton steps (these are the number of Newton steps taken to recover an approximate analytic center in ACCPM with central cuts).

2. The global convergence of the algorithm is established by showing that eventually the volumetric barrier function becomes too large for the feasible region to

contain a ball of radius ϵ. This establishes an upper bound on the number of iterations required. For ease of exposition we shall assume that we are dealing with the exact volumetric center of the polyhedral approximation in every iteration. In reality this is not possible, however the analysis can be extended to include approximate volumetric centers. For example, Anstreicher [7, 8] shows that if the current polyhedral approximation F_D of $\mathcal{C}$ has n constraints, then if the value of the barrier functional at the volumetric center y of F_D is greater than $V_{max} = mL + m \log n$, then the volume of $\mathcal{C}$ is smaller than that of an m dimensional sphere of radius ϵ. He then establishes that the increase in the barrier function, when a constraint is added, is at least ΔV^+, and also the decrease is no more than ΔV^-, for constants ΔV^+ and ΔV^- satisfying $0 < \Delta V^- < \Delta V^+$, and where $\Delta V = \Delta V^+ - \Delta V^- > 0$ is $O(1)$. Thus, we can bound the increase in the value of the volumetric barrier functional in the kth iteration as follows:

$$\begin{aligned} V(y^k) - V(y^0) &\geq \text{(no of constraints added and still in relaxation)} \times \Delta V^+ + \\ &\quad \text{(no of constraints added and subsequently dropped)} \times \Delta V \\ &\geq \Delta V \times \text{(total no of constraints added)} \\ &\geq \tfrac{k \times \Delta V}{2}, \end{aligned}$$

where the last inequality follows from the fact that the algorithm must have visited the separation oracle in Step 4 previously at least on $\frac{k}{2}$ occasions. Combining this with the maximum value V_{max}, gives the complexity estimate that the volumetric center cutting plane algorithm either finds a feasible point in $\mathcal{C}$, or proves that it is empty in $O(mL)$ calls to the oracle, and $O(m^{1.5}L)$ Newton steps. The actual results in Anstreicher [7] deal with approximate volumetric centers. The number of Newton steps can be brought down to $O(mL)$ if shallow cuts are employed as in Vaidya [154].

The overall complexity of the volumetric center method is $O(mLT + m^{4.5}L)$ arithmetic operations, where T is the complexity of the oracle, for central cuts, and $O(mLT + m^4L)$ for shallow cuts. The ellipsoid method (see Grötschel et al. [55]) on the other hand takes $O(m^2LT + m^4L)$ arithmetic operations to solve the convex feasibility problem. Although the original algorithm due to Vaidya [154] had the best complexity, it was not practical since the constants involved in the complexity analysis were very large, of the order of 10^7. The algorithm was substantially refined in Anstreicher [7, 8] significantly bringing down the maximum number of constraints required in the polyhedral approximation to $25n$ in Anstreicher [8]. Also, since the algorithm employs central cuts the number of Newton steps required in Step 6 is

$O(\sqrt{m})$, which is significantly more than the $O(1)$ steps employed in the ACCPM algorithm in Section 3.1; whether this can be achieved for the volumetric center method is still an open question. Finally, we must mention that the computational aspects of the volumetric center method have not yet been entirely tested.

4 Complexity and IPMs for SDP

We consider the complexity of SDP in Section 4.1, and a generic interior point method (IPM) for solving the SDP, together with issues involved in an efficient implementation is presented in Section 4.2. This algorithm is employed in solving the SDP relaxations of combinatorial problems as discussed in the subsequent sections. Our exposition in this section is sketchy, and for details we refer the interested reader to the excellent surveys by De Klerk [33], Todd [147], Monteiro [105], the habilitation thesis of Helmberg [57], and the Ph.D. dissertation of Sturm [143].

Consider the semidefinite programming problem

$$\begin{array}{lrcl} \min & C \bullet X & & \\ \text{s.t.} & \mathcal{A}(X) & = & b, \\ & X & \succeq & 0, \end{array} \tag{SDP}$$

with dual

$$\begin{array}{lrcrcl} \max & b^T y & & & & \\ \text{s.t.} & \mathcal{A}^T y & + & S & = & C, \\ & & & S & \succeq & 0, \end{array} \tag{SDD}$$

where the variables $X, S \in \mathcal{S}^n$ the space of real symmetric $n \times n$ matrices, $b \in \mathcal{R}^m$. Also $C \bullet X = \sum_{i,j=1}^{n} C_{ij} X_{ij}$ is the Frobenius inner product of matrices in $\mathcal{S}^n$. The linear operator $\mathcal{A} : \mathcal{S}^n \to \mathcal{R}^m$, and its adjoint $\mathcal{A}^T : \mathcal{R}^m \to \mathcal{S}^n$ are:

$$\mathcal{A}(X) \;=\; \begin{bmatrix} A_1 \bullet X \\ \vdots \\ A_m \bullet X \end{bmatrix} \quad \text{and} \quad \mathcal{A}^T y \;=\; \sum_{i=1}^{m} y_i A_i,$$

where the matrices $A_i \in \mathcal{S}^n$, $i = 1, \ldots, m$, and $C \in \mathcal{S}^n$ are the given problem parameters. The constraints $X \succeq 0$, $S \succeq 0$ are the only nonlinear (actually convex) constraints in the problem requiring that these matrices X and S are symmetric positive semi-definite matrices. We will hereafter assume that the matrices A_i, $i = 1, \ldots, m$ are linearly independent, that implies $m \leq \binom{n+1}{2}$.

If both the primal (SDP) and the dual (SDD) problems have strictly feasible (Slater) points, then both problems attain their optimal solutions, and the duality gap $X \bullet S = 0$ is zero at optimality. Most SDPs arising in combinatorial optimization satisfy this assumption. For more on strong duality we refer the reader to Ramana et al. [134], and De Klerk et al. [37] who discuss how to detect all cases that occur in SDP.

4.1 The complexity of SDP

In this section, we briefly review the complexity of SDP. Most results mentioned here can be found in the book by Grötschel et al. [55], the Ph.D. thesis of Ramana [132], the review by Ramana & Pardalos in the IPM handbook edited by Terlaky [146], Krishnan & Mitchell [81], and Porkoláb & Khachiyan [127].

We will assume that the feasible region of the SDP is contained in a ball of radius $R > 0$. The ellipsoid algorithm (see Theorem 3.2.1 in Grötschel et al. [55]) can find a solution X^* to this problem such that $|C \bullet X^* - \mathrm{OPT}| \leq \epsilon$ (OPT is the optimal objective value), in a number of arithmetic operations that is polynomial in m, n, $\log R$, and $\log \frac{1}{\epsilon}$ in the bit model. In Krishnan & Mitchell [81], for the particular choice of $R = \frac{1}{\epsilon}$, it is shown that the ellipsoid method, together with an oracle that computes the eigenvector corresponding to the most negative eigenvalue of S during the course of the algorithm, takes $O((m^2n^3 + m^3n^2 + m^4)\log(\frac{1}{\epsilon}))$ arithmetic operations. We can employ the volumetric barrier algorithm, discussed in Section 3, to improve this complexity. In Krishnan & Mitchell [81] it is shown that such an algorithm, together with the oracle mentioned above, takes $O((mn^3 + m^2n^2 + m^4)\log(\frac{1}{\epsilon}))$ arithmetic operations. This is also slightly better than the complexity of primal-dual interior point methods to be discussed in Section 4.2, when there is no structure in the underlying SDP.

On the other hand, no polynomial bound has been established for the bit lengths of the intermediate numbers occurring in interior point methods solving an SDP (see Ramana & Pardalos in [146]). Thus, strictly speaking, these methods for SDP are not polynomial in the bit model.

We now address the issue of computing an exact optimal solution of an arbitrary SDP, when the problem data is rational. Rigorously speaking, this is not a meaningful question since the following pathological cases can occur for a feasible rational semidefinite inequality, that cannot occur in the LP case.

1. It only has irrational solutions.

2. All the rational solutions have exponential bitlength.

As a result, the solution may not be representable in polynomial size in the bit length model. However we can still consider the following semidefinite feasibility problem (SDFP).

Definition 1 *Given rational symmetric matrices $A_0, \ldots, A_m$ determine if the semidefinite system*

$$\sum_{i=1}^{m} x_i A_i \quad \preceq \quad A_0$$

is feasible for some real $x \in \mathcal{R}^m$.

Ramana [133] established that SDFP cannot be an NP-complete problem, unless NP = co-NP. In fact, Porkolab & Khachiyan [127] have shown that SDFP can actually be solved in polynomial time, if either m or n is a fixed constant. The complexity of SDFP remains one of the unsolved problems in SDP.

4.2 Interior Point Methods for SDP

In this section we consider primal-dual IPMs for SDP. These are in fact extensions of the generic IPM for LP discussed in Section 1.

The optimality conditions for the SDP problem (compare with (1) for LP in Section 1) include the following:

$$\begin{array}{rcl} \mathcal{A}(X) & = & b, \;\; X \;\; \succeq \;\; 0, \\ \mathcal{A}^T y + S & = & C, \;\; S \;\; \succeq \;\; 0, \\ XS & = & 0. \end{array} \tag{10}$$

The first two conditions represent primal and dual feasibility while the third condition gives the complementary slackness condition. Consider perturbing the complementary slackness conditions to $XS = \mu I$ for some $\mu > 0$. Ignoring the inequality constraints $X, S \succeq 0$ for the moment this gives the following system:

$$\begin{array}{rcl} \mathcal{A}(X) & = & b, \\ \mathcal{A}^T y + S & = & C, \\ XS & = & \mu I. \end{array} \tag{11}$$

We denote the solution to (11) for some fixed $\mu > 0$ by (X_μ, y_μ, S_μ). The set $\{(X_\mu, y_\mu, S_\mu)\}$ forms the *central path* that is a smooth analytical curve converging to an optimal solution (X^*, y^*, S^*), as $\mu \to 0$.

If we solve (11) by Newton's method, we get the following linearized system

$$\begin{aligned} \mathcal{A} \triangle X &= 0, \\ \mathcal{A}^T \triangle y + \triangle S &= 0, \\ \triangle XS + X \triangle S &= \mu I - XS. \end{aligned} \tag{12}$$

Since X and S are matrices, they do not always commute i.e., $XS \neq SX$. In fact, we have $m + n^2 + \frac{n(n+1)}{2}$ equations, but only $m + n(n+1)$ unknowns in (12), which constitutes an overdetermined system of linear equations. This is different from the LP case in Section 1, where X and S are diagonal matrices and hence commute. As a result, the solution ΔX may not be symmetric, and $X + \Delta X$ is not in the cone of symmetric positive semidefinite matrices $\mathcal{S}_+^n$. To ensure the symmetry of ΔX, Zhang [164] introduces the symmetrization operator

$$H_P(M) = \tfrac{1}{2}(PMP^{-1} + (PMP^{-1})^T), \tag{13}$$

where P is a given nonsingular matrix, and uses this to symmetrize the linearized complementary slackness conditions, i.e., we replace the last equation in (12) by

$$H_P(\Delta XS + X\Delta S + XS) = \mu I. \tag{14}$$

A family of directions arises for various choices of P, that vary with regard to their theoretical properties, and practical efficiency, and it is still unclear which is the best direction in the primal-dual class. The Nesterov-Todd (NT) [114] direction has the most appealing theoretical properties, and is shown to arise for a particular choice of $P = (X^{-\frac{1}{2}}(X^{\frac{1}{2}}SX^{\frac{1}{2}})^{-\frac{1}{2}}X^{\frac{1}{2}}S)^{\frac{1}{2}}$ in Todd et al. [149]. On the other hand, the H..K..M direction (proposed independently in Helmberg et al. [63], Kojima et al. [79], and Monteiro [104]) is very efficient in practice (Tütüncü et al. [153]), and also requires the least number of arithmetic operations per iteration. It arises for $P = S^{\frac{1}{2}}$, and a nice justification for this choice appears in Zhang [164]. However, since the NT direction employs a primal-dual scaling in P as opposed to a dual scaling in H..K..M, it is more efficient in solving difficult SDP problems. The H..K..M direction is also obtained in Helmberg et al. [63] by solving the Newton system (12) for ΔX, and then symmetrizing ΔX by replacing it with $\frac{1}{2}(\Delta X + \Delta X^T)$. Finally, a good survey of various search directions appears in Todd [148]. As in IPMs for LP in Section 1, we need to take damped Newton steps. Similarly we introduce a proximity measure $\delta(X, S, \mu)$ that measures the proximity of (X, y, S) to (X_μ, y_μ, S_μ) on the central path. We present the generic IPM for SDP. For simplicity, we shall consider the H..K..M direction using the original interpretation of Helmberg et al. [63].

Generic Primal-Dual IPM for SDP

Input: $\mathcal{A}, b, C$, a feasible starting point (X^0, y^0, S^0) also satisfying the interior point condition, i.e., $X^0 \succ 0$, $S^0 \succ 0$, $\mathcal{A}(X^0) = b$, and $\mathcal{A}^T y^0 + S^0 = C$. Further, we may assume without loss of generality that $X^0 S^0 = I$. Other parameters include a barrier parameter $\mu = 1$, a proximity threshold $\tau > 0$ such that $\delta(X^0, S^0, \mu) \leq \tau$, and an accuracy parameter $\epsilon > 0$.

1. Reduce the barrier parameter μ.
2. **If** $\delta(X, S, \mu) > \tau$ compute $(\Delta X, \Delta y, \Delta S)$ from (12) and replacing ΔX by $\frac{1}{2}(\Delta X + \Delta X^T)$.
3. Choose some $\alpha \in (0, 1]$ so that $(X + \alpha \Delta X), (S + \alpha \Delta S) \succ 0$, and proximity $\delta(X, S, \mu)$ is suitably reduced.
4. Set $(X, y, S) = (X + \alpha \Delta X, y + \alpha \Delta y, S + \alpha \Delta S)$.
5. If $X \bullet S \leq \epsilon$ then **stop**,
   else if $\delta(X, y, \mu) \leq \tau$ **goto** step 1,
   else goto step 2.

One can solve an SDP with rational data to within a tolerance ϵ in $O(\sqrt{n} \log(\frac{1}{\epsilon}))$ feasible iterations (see Todd [147] for more details). This is the best iteration complexity bound for SDP. Interestingly, this is the same bound as in the LP case.

We now examine the work involved in each iteration. The main computational task in each iteration is in solving the following normal system of linear equations.

$$\mathcal{A}(X \mathcal{A}^T(\Delta y) S^{-1}) \;=\; b \tag{15}$$

This system results from eliminating ΔS, and ΔX from (12). Let $M : \mathcal{R}^m \to \mathcal{R}^m$ be the linear operator given by $My = \mathcal{A}(X \mathcal{A}^T(y) S^{-1})$. The ith row of $M \Delta y$ is given by

$$A_i \bullet X \mathcal{A}^T(\Delta y) S^{-1} \;=\; \sum_{j=1}^{m} \Delta y_j \mathrm{Trace}(X A_i S^{-1} A_j)$$

Each entry of the matrix M thus has the form $M_{ij} = \mathrm{Trace}(X A_i S^{-1} A_j)$. This matrix is symmetric and positive definite, if we assume matrices A_i, $i = 1, \ldots, m$ are linearly independent in $\mathcal{S}^n$.

Solving for Δy requires $\frac{m^3}{3}$ arithmetic operations, when the Cholesky decomposition is used. Moreover, M has be to recomputed in each iteration. An efficient way to build one row of M is the following

1. Compute XA_iS^{-1} once in $O(n^3)$ time;

2. Determine the m single elements via $XA_iS^{-1} \bullet A_j$ in $O(mn^2)$ arithmetic operations.

In total the construction of M requires $O(mn^3 + m^2n^2)$ arithmetic operations, and this is the most expensive operation in each iteration. On the whole, an interior point method requires $O(m(n^3 + mn^2 + m^2)\sqrt{n}\log(\frac{1}{\epsilon}))$ arithmetic operations. For most of the combinatorial problems such as maxcut, the constraint matrices A_i have a rank one structure, and this reduces the computation of M to $O(mn^2 + m^2n)$ operations.

Excellent software based on primal-dual IPMs for SDP include CSDP by Borchers [22], SeDuMi by Sturm [144], and SDPT3 by Tütüncü et al. [153]. An independent benchmarking of various SDP software appears in Mittleman [102].

In many applications the constraint matrices A_i have a special structure. The dual slack matrix S inherits this sparsity structure, while the primal matrix X is usually dense regardless of the sparsity. Benson et al. [19] proposed a dual scaling algorithm that exploits the sparsity in the dual slack matrix. Also, Fukuda et al. [44] and Nakata et al. [109] employ ideas from the completion of positive semidefinite matrices (Grone et al. [54], Laurent [87]) to deal with dense X in a primal-dual IPM for SDP. Burer [23] on the other hand utilizes these ideas to develop a primal-dual IPM entirely within the space of partial positive semidefinite matrices.

However, in most approaches, the matrix M is dense, and the necessity to store and factorize this dense matrix M limits the applicability of IPMs to problems with around 3000 constraints on a well equipped work station.

One way to overcome the problem of having to store the matrix M via the use of an iterative scheme, which only accesses this matrix through matrix vector multiplications, is discussed in Toh & Kojima [151]. This approach is not entirely straightforward since the Schur matrix M becomes increasingly ill-conditioned as the iterates approach the boundary. Hence, there is a need for good pre-conditioners for the iterative method to converge quickly. Recently, Toh [150] has reported excellent computational results with a choice of a good preconditioner in solving the normal system of linear equations.

5 First order techniques for SDP

Interior point methods discussed in Section 4.2 are fairly limited in the size of problems they can handle. We discuss various first order techniques with a view of solving

large scale SDPs in this section. As opposed to primal-dual interior point methods, these methods are mostly dual-only, and in some cases primal methods. These methods exploit the structure prevalent in combinatorial optimization problems; they are applicable in solving only certain classes of SDPs. Unlike IPMs there is no proof of polynomial complexity, and moreover these methods are not recommended for those problems, where a high accuracy is desired. Nevertheless excellent computational results have been reported for problems that are inaccessible to IPMs due to demand for computer time and storage requirements. A nice overview of such methods appears in the recent survey by Monteiro [105]. In this section, we will focus on the first order techniques which are very efficient in practice.

The first method is the spectral bundle method due to Helmberg & Rendl [61]. The method is suitable for large m, and recent computational results are reported in Helmberg [59]. The method is first order, but a second order variant which converges globally and which enjoys asymptotically a quadratic rate of convergence was recently developed by Oustry [118].

The spectral bundle method works with the dual problem (SDD). Under an additional assumption that $\text{Trace}(X) = \beta$, for some constant $\beta \geq 0$, for all X in the primal feasible set, the method rewrites (SDD) as the following eigenvalue optimization problem.

$$\max \quad \beta\lambda_{\min}(C - \mathcal{A}^T\mathrm{y}) \quad + \quad b^T\mathrm{y}, \tag{16}$$

where $\lambda_{\min}(S)$ denotes the smallest eigenvalue of S. Problem (16) is a concave nonsmooth optimization problem, that is conveniently tackled by bundle methods for non-differentiable optimization. In the spectral bundle scheme the maximum eigenvalue is approximated by means of vectors in the subspace spanned by the bundle P which contains the important subgradient information. For simplicity we mention (see Krishnan & Mitchell [82] for a discussion) that this can be interpreted as solving the following problem in lieu of (16)

$$\max \quad \beta\lambda_{\min}(P^T(C - \mathcal{A}^T y)P) + b^T y, \tag{17}$$

whose dual is the following SDP

$$\begin{array}{lrcl} \min & (P^TCP) \bullet W & & \\ \text{s.t.} & (P^TA_iP) \bullet W & = & b_i, \quad i = 1, \ldots, m \\ & I \bullet W & = & \beta \\ & W & \succeq & 0. \end{array} \tag{18}$$

In the actual bundle method, instead of (17), we solve an SDP with a quadratic objective term; the quadratic term arises from the regularization term employed in the bundle method. For more details we refer the reader to Helmberg et al. [57, 61, 60]. In (18), we are approximately solving (SDP), by considering only a subset of the feasible X matrices. By keeping the number of columns r in P small, the resulting SDP can be solved quickly. The dimension of the subspace P is roughly bounded by the square root of number of constraints. This follows from a bound by Pataki [122] on the rank of extreme matrices in SDP. The optimum solution of (17) typically produces an indefinite dual slack matrix $S = (C - \mathcal{A}^T y)$. The negative eigenvalues and corresponding eigenvectors of S are used to update the subspace, P and the process is iterated. A recent primal active set approach for SDP which also deals with (18) has been recently developed by Krishnan et al. [83].

Another variation of the low rank factorization idea mentioned above has been pursued by Burer & Monteiro [24]. They consider factorizations $X = RR^T$, where $R \in \mathcal{R}^{n \times r}$, and instead of (SDP) they solve the following formulation for R

$$\begin{array}{lll} \min & C \bullet (RR^T) & \\ \text{s.t.} & \mathcal{A}(RR^T) & = \quad b. \end{array}$$

This is a non-convex optimization problem that is solved using a modified version of the augmented Lagrangian method. The authors claim via extensive computational experiments that the method converges to the exact optimum value of (SDP), while a recent proof of convergence for a variant of this approach appears in Burer & Monteiro [25]. As a particular case of this approach, Burer et al. have employed rank two relaxations of maximum cut [28], and maximum stable set [29] problems with considerable computational success. The rank two relaxation is in fact an exact formulation of the maximum stable set problem.

We now turn to the method due to Burer et al. [26]. This method complements the bundle approach discussed previously; it recasts the dual SDP as a non-convex but smooth unconstrained problem. The method operates on the following pair of SDPs.

$$\begin{array}{lrcl} \max & C \bullet X & & \\ \text{s.t.} & \text{diag}(X) & = & d, \\ & \mathcal{A}(X) & = & b, \\ & X \succeq 0, & & \end{array} \tag{19}$$

with dual

$$\begin{array}{rcl} \min & d^T z + b^T y & \\ \mathcal{A}^T y + \mathrm{Diag}(z) - S & = & C, \\ S & \succeq & 0. \end{array} \tag{20}$$

Burer et al. consider only strictly feasible solutions of (20), i.e., $S = (\mathcal{A}^T y + \mathrm{Diag}(z) - C) \succ 0$. Consider now a Cholesky factorization of

$$S = (\mathrm{Diag}(v) + L_0)(\mathrm{Diag}(v) + L_0)^T, \tag{21}$$

where $v \in \mathcal{R}^n_{++}$, and L_0 is a strictly lower triangular matrix. In (21), there are $\frac{n(n+1)}{2}$ equations, and $m + n + \frac{n(n+1)}{2}$ variables. So one can use the equations to write $\frac{n(n+1)}{2}$ variables, namely z and L_0, in terms of the other variables v and y. Thus one can transform (20) into the following equivalent nonlinear programming problem

$$\begin{array}{lrcl} \inf & d^T z(v,y) + b^T y & & \\ \text{s.t.} & v & > & 0, \end{array} \tag{22}$$

where $z(v, y)$ indicates that z has been written in terms of v and y using (21). We note that the nonlinearity in (20) has been shifted from the constraints to the objective function, i.e., in the term $z(v, y)$ in (22). The latter problem does not attain its optimal solution, however we can use its intermediate solutions to approach the solution of (20) for a given $\epsilon > 0$. Moreover, the function $z(v, y)$ is a smooth analytic function. The authors then use a log-barrier term introducing the $v > 0$ constraint into the objective function, and suggest a potential reduction algorithm to solve (22); thus their approach amounts to reducing SDP to a non-convex, but smooth unconstrained problem. The main computational task is the computation of the gradient, and Burer et al. [27] develop formulas that exploit the sparsity of the problem data. Although the objective function is non-convex, the authors prove global convergence of their method, and have obtained excellent computational results on large scale problems.

Other approaches include Benson & Vanderbei [18], a dual Lagrangian approach due to Fukuda et al. [45], and PENNON by Kocvara & Stingl [78] that can also handle nonlinear semidefinite programs. A variant of the bundle method has also been applied to the Quadratic Assignment Problem (QAP) by Rendl and Sotirov [135]; their bounds are the strongest currently available for the QAP and this is one of the largest SDPs solved to date.

6 Branch and cut SDP based approaches

We discuss an SDP based branch and cut approach in this section that is designed to solving combinatorial optimization problems to optimality via a series of SDP relaxations of the underlying problem. Our particular emphasis is on the maxcut problem.

A branch and cut approach combines the advantages of cutting plane, and branch and bound methods. In a pure branch and bound approach the relaxation is improved by dividing the problem into two subproblems, where one of the variables is restricted to taking certain values. The subproblems form a tree known as the branch and bound tree, rooted at the initial relaxation.

In a branch and cut approach cutting planes are added to the subproblems in the branch and bound tree, improving these relaxations until it appears that no progress can be made. Once this is the case, we resort to branching again. We do not discuss branch and cut LP approaches in this survey, but rather refer the reader to the survey by Mitchell et al. [98].

Consider now the maxcut problem. As discussed in Section 1, for $S \subseteq V$ with cut $\delta(S)$, the maxcut problem (MC) can be written as

$$\max_{S \subset V} \sum_{\{i,j\} \in \delta(S)} w_{ij}. \tag{MC}$$

Without loss of generality, we can assume that our graph is complete. In order to model an arbitrary graph in this manner, define $w_{ij} = 0$, $\{i,j\} \notin E$. Finally, let $A = (w_{ij})$ be the weighted adjacency matrix of the graph.

We consider an SDP relaxation of the maxcut problem in this section. The maxcut problem can be formulated as the following integer program (23) in the x variables, where $x_i = 1$ if vertex $i \in S$, and -1 if $i \in V \backslash S$

$$\max_{x \in \{-1,1\}^n} \sum_{i,j=1}^{n} w_{ij} \frac{1 - x_i x_j}{4}. \tag{23}$$

A factor of $\frac{1}{2}$ accounts the fact that each edge is considered twice. Moreover, the expression $\frac{(1-x_i x_j)}{2}$ is 0 if $x_i = x_j$, i.e., if i and j are in the same set, and 1 if $x_i = -x_j$. Thus $\frac{(1-x_i x_j)}{2}$ yields the *incidence vector* of a cut associated with a cut vector x, evaluating to 1 if and only if edge $\{i,j\}$ is in the cut. Exploiting the fact that $x_i^2 = 1$, we have

$$\begin{aligned} \tfrac{1}{4} \sum_{i,j=1}^{n} w_{ij}(1 - x_i x_j) &= \tfrac{1}{4} \sum_{i=1}^{n} (\sum_{j=1}^{n} w_{ij} x_i^2 - \sum_{j=1}^{n} w_{ij} x_i x_j) \\ &= \tfrac{1}{4} x^T (\mathrm{Diag}(Ae) - A) x. \end{aligned} \tag{24}$$

The matrix $L = \mathrm{Diag}(Ae) - A$ is called the *Laplacian* matrix of the graph G. Letting $C = \frac{1}{4}L$, we find that the maxcut problem can be interpreted as a special case of the following more general $\{+1, -1\}$ integer programming problem

$$\max_{x \in \{-1,1\}^n} \quad x^T C x. \tag{25}$$

We are now ready to derive a semidefinite programming relaxation for the maxcut problem. First note that $x^T C x = \mathrm{Trace}(C x x^T)$. Now consider $X = xx^T$, i.e., $X_{ij} = x_i x_j$. Since $x \in \{-1, 1\}^n$, the matrix X is positive semidefinite, and its diagonal entries are equal to one. Thus (25) is equivalent to the following problem

$$\begin{array}{lrcl} \max & C \bullet X & & \\ \text{s.t.} & \mathrm{diag}(X) & = & e, \\ & X & \succeq & 0, \\ & \mathrm{rank}(X) & = & 1. \end{array} \tag{26}$$

The rank restriction is a non-convex constraint. To get a convex problem one drops the rank one restriction, and arrives at the following semidefinite programming relaxation of the maxcut problem

$$\begin{array}{lrcl} \max & C \bullet X & & \\ \text{s.t.} & \mathrm{diag}(X) & = & e, \\ & X & \succeq & 0, \end{array} \tag{27}$$

and its dual

$$\begin{array}{lrcl} \min & e^T y & & \\ \text{s.t.} & \mathrm{Diag}(y) - S & = & C, \\ & S & \succeq & 0. \end{array} \tag{28}$$

Lemarechal & Oustry [90] and Poljak et al. [125] derive the SDP relaxation (27) by taking the dual of the Lagrangian dual of (23), which incidentally is (28). We will refer to the feasible region of (27) as the elliptope. A point that must be emphasized is that the elliptope is no longer a polytope. Thus (27) is actually a non-polyhedral relaxation of the maxcut problem.

These semidefinite programs satisfy strong duality, since $X = I$ is strictly feasible in the primal problem, and we can generate a strictly feasible dual solution by assigning y an arbitrary positive value. In fact, setting $y_i = 1 + \sum_{j=1}^{n} |C_{ij}|$ and $S = \mathrm{Diag}(y) - C$ should suffice.

We can improve the relaxation (27) using the following linear inequalities.

1. **The odd cycle inequalities:**

$$X(\mathcal{C}\backslash\mathcal{F}) - X(\mathcal{F}) \leq |\mathcal{C}| - 2 \quad \text{for each cycle } \mathcal{C}, \mathcal{F} \subset \mathcal{C}, |\mathcal{F}| \text{ odd} \tag{29}$$

These include among others the triangle inequalities. They provide a complete description of the cut polytope for graphs not contractible to K_5 (Barahona [16] and Seymour [140]). Although there are an exponential number of linear constraints in (29), Barahona & Mahjoub [17] (also Grötschel et al. [55]) describe a polynomial time separation oracle for these inequalities, that involves solving n shortest path problems on an auxiliary graph with twice the number of nodes, and four times the number of edges. Thus it is possible to find the most violated odd cycle inequality in polynomial time.

2. **The hypermetric inequalities:**
These are inequalities of the form (30)

$$aa^T \bullet X \geq 1, \quad \text{where } a \in \mathcal{Z}^n, \quad \sum_{i=1}^{n} a_i \text{ odd} \quad \text{and} \min\{(a^T x)^2 : x \in \{-1,1\}^n\} = 1. \tag{30}$$

For instance, the triangle inequality $X_{ij} + X_{ik} + X_{jk} \geq -1$ can be written as a hypermetric inequality by letting a to be the incidence vector of the triangle (i, j, k). On the other hand the other inequality $X_{ij} - X_{ik} - X_{jk} \geq -1$ can be written in a similar way, except that $a_k = -1$. Although there are a countably infinite number of them, these inequalities also form a polytope known as the hypermetric polytope (Deza & Laurent [40]). The problem of checking violated hypermetric inequalities is NP-hard (Avis [12] and Avis & Grishukhin [13]). However, Helmberg & Rendl [62] describe simple heuristics to detect violated hypermetric inequalities.

We sketch a conceptual SDP cutting plane approach for the maxcut problem in this section.

An SDP cutting plane approach for maxcut

1. **Initialize.** Start with (27) as the initial SDP relaxation.

2. **Solve the current SDP relaxation.** Use a primal-dual IPM as discussed in Section 4.2. This gives an upper bound on the optimal value of the maxcut problem.

3. **Separation.** Check for violated odd cycle inequalities. Sort the resulting violated inequalities, and add a subset of the most violated constraints to the relaxation.
   If no violated odd cycle inequalities are found **goto** step 5.

4. **Primal heuristic.** Use the Goemans-Williamson [47] randomized rounding procedure (discussed in Section 7) to find a good incidence cut vector. This is a lower bound on the optimal value.

5. **Check for termination. If** the difference between the upper bound and the value of the best cut is small, **then** stop.
   If no odd cycle inequalities were found in step 3 **then goto** step 4.
   Else goto step 2.

6. **Branching.** Resort to branch and bound as discussed in Section 6.1.

The choice of a good SDP branch and cut approach hinges on the following:

1. **Choice of a good initial relaxation:** The choice of a good initial relaxation is important, and provides a tight upper bound on the maxcut value. The SDP relaxation (27) is an excellent choice; it is provably tight in most cases. Better initial relaxations (see Anjos & Wolkowicz [5, 6], Lasserre [85], and Laurent [88]) do exist, but they are more expensive to solve. In contrast the polyhedral cutting plane approaches rely on poor LP relaxations, the ratio of whose bounds to the maxcut optimal value can be as high as 2 (see Poljak & Tuza [126]).

2. **Generating good lower bounds:** The Goemans-Williamson rounding procedure in Step 4 is an algorithm for generating incidence cut vectors, that provide good lower bounds. We will see in Section 7 that this procedure is instrumental in developing a 0.878 approximation algorithm for the maxcut problem.

3. **Choice of good cutting planes:** It is important to use good cutting planes that are facets of the maxcut polytope, and use heuristics for finding such constraints quickly. In the above cutting plane approach for instance we might first check for violated triangle inequalities by complete enumeration, and use the Barahona-Mahjoub separation oracle when we run out of triangle inequalities (Mitchell [95]).

4. **Choice of the branching rule:** Typically we may have to resort to branch and bound in Step 6. It is important to choose a good branching rule to keep

the size of the branch and bound tree small. We present a short discussion on branch and bound in an SDP branch and cut framework in Section 6.1.

5. **Restarting after branching or the addition of cutting planes:** One of the major shortcomings of an SDP branch and cut approach, where a primal-dual IPM is employed in solving the SDP relaxations is the issue of restarting the SDP relaxations after the addition of cutting planes. Although some warm start strategies do exist for the maxcut problem (Mitchell [97]), they are prohibitively expensive. We will discuss some of these strategies in Section 6.2. There do exist simplex-like analogues for SDP (Pataki [123, 124] and Krishnan et al. [83]), and dual simplex variants of these schemes could conceivably be used to re-optimize the SDP relaxations after the addition of cutting planes.

6.1 Branch and bound in the SDP context

We provide a short overview on branch and bound within the SDP context in this section. Some excellent references for branch and bound within the SDP context of the maxcut problem are Helmberg & Rendl [62], and Mitchell [97].

Consider $X = V^T V$, with $V = (v_1, \ldots, v_n)$. We want to branch based on the values of $X_{ij} = (v_i^T v_j)$. Typically this is the most fractional variable, i.e., the X_{ij} closest to zero. The branching scheme is based on whether vertices i and j should be on the same side of the cut or on opposite sides. With this branching rule X_{ki} and X_{kj} are also then constrained to be either the same or different, $\forall k = \{1, \ldots, n\} \backslash \{i, j\}$. This means that the problem can be replaced by an equivalent semidefinite program of dimension one less. Without loss of generality let us assume that we are branching on whether vertices $n-1$ and n are on the same or opposite sides. Let we write the Laplacian matrix L in (27) as

$$L \;=\; \begin{bmatrix} \bar{L} & p_1 & p_2 \\ p_1^T & \alpha & \beta \\ p_2^T & \beta & \gamma \end{bmatrix}.$$

Here $\bar{L} \in \mathcal{S}^{n-2}$, $p_1, p_2 \in \mathcal{R}^{n-2}$ and α, β, and $\gamma \in \mathcal{R}$. The SDP relaxation that corresponds to putting both $n-1$ and n on the same side is

$$\begin{array}{lll} \max & \frac{1}{4} \begin{bmatrix} \bar{L} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{bmatrix} \bullet X & \\ \text{s.t.} & \operatorname{diag}(X) = e, & \\ & X \succeq 0, & \end{array} \tag{31}$$

with dual

$$\begin{array}{lrcl} \min & e^T y & & \\ \text{s.t.} & \mathrm{Diag}(y) - S & = & \frac{1}{4}\begin{bmatrix} \bar{L} & p_1 + p_2 \\ p_1^T + p_2^T & \alpha + 2\beta + \gamma \end{bmatrix}, \\ & S & \succeq & 0. \end{array} \tag{32}$$

Note that $X, S \in \mathcal{S}^{n-1}$, and $y \in \mathcal{R}^{n-1}$, i.e., not only do we have a semidefinite program of dimension one less, but the number of constraints in (31) has dropped by one as well. This is because performing the same transformation (as the Laplacian) on the nth coefficient matrix $e_n e_n^T$ leaves it as $e_{n-1}e_{n-1}^T$, which is in fact the $(n-1)$th coefficient matrix.

On the other hand, putting $n-1$ and n on opposite sides, we get a similar SDP relaxation, with the Laplacian matrix now being $\frac{1}{4}\begin{bmatrix} \bar{L} & p_1 - p_2 \\ p_1^T - p_2^T & \alpha - 2\beta + \gamma \end{bmatrix}$.

It is desirable that we use the solution of the parent node, in this case the solution of (27), to speed up the solution of the child (31). As we mentioned previously, this is a major issue in the SDP, since there is no analogue to the dual simplex method, unlike the LP case for re-optimization. More details on this can be found in Mitchell [97].

Another important issue is determining good bounds for each of the subproblems, so that some of these subproblems in the branch and bound tree could be *fathomed*, i.e., not explicitly solved. In the LP approach, we can use reduced costs to estimate these bounds, and hence fix some of the variables without having to solve both subproblems. In the SDP case things are not so easy, since the constraints $-1 \leq X_{ij} \leq 1$ are not explicitly present in the SDP relaxation (they are implied through the $\mathrm{diag}(X) = e$ and $X \succeq 0$ constraints). Thus, the dual variables corresponding to these constraints are not directly available. Helmberg [58] describes a number of approaches to fix variables in semidefinite relaxations.

6.2 Warm start strategies for the maxcut problem

In cutting plane algorithms it is of fundamental importance that re-optimization is carried out in reasonable time after the addition of cutting planes. Since the cutting planes cut off the optimal solution X^{prev} to the previous relaxation, we need to generate a new strictly feasible point X^{start} for restarting the method.

We first discuss two strategies of restarting the primal problem since this is the more difficult problem.

1. **Backtracking along iterates:**
   This idea is originally due to Mitchell & Borchers [99] for the LP. The idea is to store all the previous iterates on the central path, during the course of solving the original SDP relaxation (27), and restart from the last iterate that is strictly feasible with respect to the new inequalities. Also, this point is hopefully close to the new central path, and the interior point algorithm will work better if this is the case.

2. **Backtracking towards the analytic center:**
   This was employed in Helmberg & Rendl [62]. The idea is to backtrack towards I along a straight line between the last iterate X^{prev} and I. Thus we choose $X^{\text{start}} = (\lambda X^{\text{prev}} + (1-\lambda)I)$ for some $\lambda \in [0, 1)$. Since the identity matrix I is the analytic center of the feasible region of (27), it is guaranteed that the procedure will terminate with a strictly feasible primal iterate.

Restarting the dual which has additional variables corresponding to the number of cutting planes in the primal is relatively straightforward, since we can get into the dual SDP cone $S \succeq 0$, by assigning arbitrarily large values to the first n components of y (that originally appear in $\text{Diag}(y)$).

7 Approximation algorithms for combinatorial optimization

One of the most important applications of SDP is in developing approximation algorithms for various combinatorial optimization problems. The euphoria began with the Goemans & Williamson approximation algorithm [47] for the maxcut problem, and the technique has since been applied to a variety of other problems. For some of these problems such as MAX 3SAT the SDP relaxation (Karloff & Zwick [72]) provides the tightest approximation algorithm possible unless $P = NP$.

We discuss the GW algorithm in detail below. The algorithm works with the SDP relaxation (27) for the maxcut problem we introduced in Section 6. We outline the main steps in the algorithm as follows:

The Goemans-Williamson (GW) approximation algorithm for maxcut

1. Solve the SDP relaxation (27) to get a primal matrix X.

2. Compute $V = (v_1, \ldots, v_n)$ such that $X = V^T V$. This can be done either by computing the Cholesky factorization of X, or by computing its spectral decomposition $X = P\Lambda P^T$, with $V = \sqrt{\Lambda} P^T$.

3. Randomly partition the unit sphere in $\mathcal{R}^n$ into two half spheres H_1 and H_2 (the boundary in between can be on either side), and form the bipartition consisting of $V_1 = \{i : v_i \in H_1\}$ and $V_2 = \{i : v_i \in H_2\}$. The partitioning is carried out in practice by generating a random vector r on the unit sphere, and assigning i to V_1 if $v_i^T r \geq 0$, and V_2 otherwise. In practice, one may repeat this procedure more than once, and pick the best cut obtained.

Hereafter, we refer to Step 3 as the *GW rounding procedure*. It is important to note that Step 3 gives a lower bound on the optimal maxcut solution, while the SDP relaxation in Step 1 gives an upper bound. The entire algorithm can be derandomized as described in Mahajan & Hariharan [94].

A few notes on the GW rounding procedure: For any factorization of $X = V^T V$ in Step 2, the columns of V yield vectors v_i, $i = 1, \ldots, n$. Since we have $\mathrm{diag}(X) = e$, each vector v_i is of unit length, i.e., $||v_i|| = 1$. Associating a vector v_i with node i, we may interpret v_i as the relaxation of $x_i \in \{-1, 1\}$ to the n dimensional unit sphere. Thus we are essentially solving

$$\begin{array}{ll} \max & \displaystyle\sum_{i,j=1}^{n} \frac{L_{ij}}{4} v_i^T v_j \\ \text{s.t.} & ||v_i|| = 1 \quad \forall i = 1, \ldots, n, \\ & v_i \in \mathcal{R}^n. \end{array} \tag{33}$$

This vector formulation provides a way to interpret the solution to the maxcut SDP. Since v_i and v_j are unit vectors, $v_i^T v_j$ is the cosine of the angle between these vectors. If all the edge weights w_{ij} are nonnegative, the off diagonal entries of the Laplacian matrix are negative. Thus, if the angle between the vectors is large, we should separate the corresponding vertices, if it is small we put them in the same set (since this would improve the objective function in the vector formulation). In order to avoid conflicts, Goemans & Williamson [47] consider the random hyperplane technique mentioned in Step 3. This step is in accord with our earlier intuition, since vectors with a large angle between them are more likely to be separated, since the hyperplane can end up between them.

The hyperplane with normal r in Step 3 of the algorithm divides the unit circle into two halfspheres, and an edge $\{i, j\}$ belongs to the cut $\delta(S)$ if and only if the

vectors v_i and v_j do not belong to the same half-sphere. The probability that an edge $\{i,j\}$ belongs to $\delta(S)$ is equal to $\frac{\arccos(v_i^T v_j)}{\pi}$, and the expected weight $E(w(S))$ of the cut $\delta(S)$ is

$$\begin{aligned} E(w(S)) &= \sum_{i,j=1}^{n} \frac{L_{ij}}{4} \frac{\arccos(v_i^T v_j)}{\pi} \\ &= \sum_{i,j=1}^{n} \frac{L_{ij}}{4} \frac{1 - v_i^T v_j}{2} \frac{2}{\pi} \frac{\arccos(v_i^T v_j)}{1 - v_i^T v_j} \\ &\geq 0.878 \times (\text{objective value of relaxation (33)}) \\ &\geq 0.878 \times (\text{optimal maxcut value}). \end{aligned}$$

The second to last inequality holds if we assume that all the edge weights are nonnegative, and from the observation that

$$\min_{-1 \leq x \leq 1} \frac{2}{\pi} \frac{\arccos(x)}{1-x} \geq 0.878.$$

The last inequality from the fact that the objective value of relaxation (33) provides an upper bound on the maxcut solution. Hence, we have an 0.878 approximation algorithm for the maxcut problem, when all the edge weights are nonnegative. On the negative side Håstad [56] showed that it is NP-hard to approximate the maxcut problem to within a factor of 0.9412.

For the general case where $L \succeq 0$, Nesterov [111] showed that the GW rounding procedure gives an $\frac{2}{\pi}$ approximation algorithm for the maxcut problem.

Interestingly, although, the additional inequalities such as triangle inequalities (mentioned with regard to the metric polytope) improve the SDP relaxation, they do not necessarily give better approximation algorithms. On the negative side Karloff [71] exhibited a set of graphs for which the optimal solution of relaxation (27) satisfies all the triangle inequalities as well, so after the GW rounding procedure we are still left with a 0.878 approximation algorithm.

Goemans & Williamson [47] show that the randomized rounding procedure performs well if the ratio of the weight of the edges in the cut, to those in the graph is more than 85%. If this is not true, then it pays to introduce more randomness in the rounding procedure. Zwick [165] considers the randomized rounding as applied to $(\gamma I + (1-\gamma)X)$ rather than X, for some appropriate $\gamma \in [0,1]$.

There have been several extensions of SDP, and the randomized rounding technique to other combinatorial optimization problems. These include quadratic programming (Nesterov [111], Ye [161]), maximum bisection (Frieze & Jerrum [43], Ye [162]), max k-cut problem (Frieze & Jerrum [43], De Klerk et al. [36], Goemans

& Williamson [48]), graph coloring (Karger et al. [70]), vertex cover (Kleinberg & Goemans [77]), maximum satisfiability problem (Goemans & Williamson [47] and more recently in De Klerk & Van Maaren [38], De Klerk et al. [39], and Anjos [4]) with extensions to Max 2SAT (Feige & Goemans [42]) and Max 3SAT (Karloff & Zwick [72]), and finally the maximum directed cut problem (Goemans & Williamson [47], Feige & Goemans [42]). A nice survey on the techniques employed in designing approximation algorithms for these problems is Laurent & Rendl [89], while a good overview of the techniques for satisfiability, graph coloring, and max k-cut appears in the recent monograph by De Klerk [33].

8 Convex approximations of integer programming

The results in this section are based on recent results by Nesterov [112], Lasserre [84, 85], Parrilo [121] and De Klerk et al. [35, 21]. A nice survey of these methods also appears in Laurent & Rendl [89].

8.1 Semidefinite approximations of polynomial programming

Consider the following polynomial programming problem

$$\begin{array}{lll} \min & g_0(x) & \\ \text{s.t.} & g_k(x) & \geq \quad 0, \quad k = 1, \ldots, m, \end{array} \tag{34}$$

where $g_k(x)$, $k = 0, \ldots, m$ are polynomials in $x = (x_1, \ldots, x_n)$. This is a general problem which encompasses $\{0, 1\}$ integer programming problems, since the condition $x_i \in \{0, 1\}$ can be expressed as the polynomial equation $x_i^2 - x_i = 0$. The importance of (34) is that, under some technical assumptions, this problem can be approximated by a sequence of semidefinite programs. This result, due to Lasserre [84], relies on the fact that certain nonnegative polynomials can be expressed as sums of squares (SOS) [1] of other polynomials. Also, see Nesterov [112], Parrilo [121], and Shor [142] for using SOS representations of polynomials for approximating (34).

We give a brief overview of some of the main ideas underlying this approach. For ease of exposition we shall confine our attention to the unconstrained problem

$$g^* \quad = \quad \min \quad g(x) \quad \text{s.t. } x \in \mathcal{R}^n, \tag{35}$$

[1] This is not to be confused with *specially ordered sets* commonly used in integer programming.

where without loss of generality we assume $g(x)$ is a polynomial of even degree $2d$. Let

$$\left[1, x_1, x_2, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, x_2x_3, \ldots, x_n^2, \ldots, x_1^{2d}, \ldots, x_n^{2d}\right]$$

be a basis for $g(x)$. Let

$$S_{2d} \;=\; \{\alpha \in \mathcal{Z}_+^n : \textstyle\sum_i \alpha_i \leq 2d\},$$

and let $s(2d) = |S_{2d}|$. The above basis can then be conveniently represented as $\{x^\alpha\}$, $\alpha \in S_{sd}$. We write $g(x) = \sum_{\alpha \in S_{2d}} \gamma_\alpha x^\alpha$, with $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, where $\gamma = \{\gamma_\alpha\} \in \mathcal{R}^{s(2d)}$ is the coefficient vector of $g(x)$ in the basis. Then problem (35) can also be written as

$$g^* \;=\; \max \;\; \{\lambda \;\; \text{s.t.} \;\; g(x) - \lambda \geq 0, \;\; \forall x \in \mathcal{R}^n\}. \tag{36}$$

This problem encompasses integer and non-convex optimization problems, and consequently is NP hard. However, lower bounds on g^* can be obtained by considering sufficient conditions for the polynomial $g(x) - \lambda \geq 0$ on $\mathcal{R}^n$. One such requirement is that $g(x) - \lambda$ be expressible as a sum of squares of polynomials, i.e., have an SOS representation. Thus,

$$g^* \;\geq\; \max \;\; \{\lambda \;\; \text{s.t.} \;\; g(x) - \lambda \text{ has an SOS representation}\}. \tag{37}$$

Problem (37) can be expressed as a semidefinite program. To see this, let $z = \{x^\alpha\}$ with $\alpha \in S_d$ be the basis vector consisting of all monomials of degree $\leq d$. Then one can easily verify that $g(x)$ has an SOS representation if and only if $g(x) = z^T X z$ for some positive semidefinite matrix X. For $\gamma \in S_{2d}$, let

$$B_\gamma \;=\; \sum_{\alpha,\beta \in S_d, \alpha+\beta=\gamma} E_{\alpha,\beta},$$

where $E_{\alpha,\beta}$ is the elementary matrix with all zero entries except entries 1 at positions (α,β) and (β,α). Using this we have:

$$\begin{aligned} z^T X z \;&=\; \sum_{\alpha,\beta \in S_d} X_{\alpha,\beta} x^{\alpha+\beta}, \\ &=\; \sum_{\gamma \in S_{2d}} x^\gamma \sum_{\alpha,\beta \in S_d, \alpha+\beta=\gamma} X_{\alpha,\beta}, \\ &=\; \sum_{\gamma \in S_{2d}} x^\gamma (B_\gamma \bullet X). \end{aligned}$$

Assuming the constant term g_0 in the polynomial $g(x)$ is zero, and comparing coefficients in $g(x) - \lambda = \sum_{\gamma \in S_{2d}} x^\gamma (B_\gamma \bullet X)$ for $\gamma = 0$, we have $\lambda = -B_0 \bullet X$. Hence, one

can equivalently write (37) as the following SDP

$$\begin{array}{lrcl} \max & -B_0 \bullet X & & \\ \text{s.t.} & B_\gamma \bullet X & = & g_\gamma, \quad \gamma \in S_{2d} \backslash \{0\}, \\ & X & \succeq & 0, \end{array} \tag{38}$$

with dual

$$\begin{array}{ll} \min & \sum_{\alpha \in S_{2d}} g_\alpha y_\alpha \\ \text{s.t.} & \sum_{\alpha \in S_{2d}} B_\alpha y_\alpha \succeq 0. \end{array} \tag{39}$$

The dual (39) has an equivalent interpretation in the theory of moments, and forms the basis for the original approach of Lasserre [84]. Another advantage of this dual approach of Lasserre [84], over the primal approach of Parrilo [121], is that it also yields certificates ensuring that an optimal solution is attained in the series of relaxations, and also gives a mechanism for extracting these solutions (see Henrion & Lasserre [66]).

In general for a polynomial with even degree $2d$ in n variables, the SDP (38) has $\binom{n+2d}{2d}$ constraints, where X is a matrix in $\mathcal{S}^{\binom{n+d}{d}}$. The lower bound from (38) is equal to g^* if the polynomial $g(x) - \lambda$ has an SOS representation; this is true for $n = 1$, but not in general if $n \geq 2$. In such cases, one can estimate g^* asymptotically by a sequence of SDPs, if one assumes that an upper bound R is known a priori on the norm of a global minimizer x of $g(x)$ (see Lasserre [84]), by using a theorem of Putinar [130] for SOS representations of the positive polynomial $g(x) - \lambda + \epsilon$ on the set $\{x : ||x|| \leq R\}$. This gives a sequence of SDP approximations, whose objective values asymptotically converge to g^*. A similar approach has been adopted by Lasserre [84] for the constrained case (34).

In the $\{0, 1\}$ case, when the constraints $x_i^2 - x_i = 0$ are part of the polynomials in the constraint set, Lasserre [85] shows there is finite convergence in n steps. Laurent [86] shows that the Lassere approach is actually a strengthened version of the Sherali & Adams [141] lift and project procedure, and since the latter scheme converges in at most n steps so does the above approach. Other lift and project methods include Lovász & Schrijver [92], and Balas et al. [15] in the context of estimating the convex hull of the feasible set of $\{0, 1\}$ programming problems, and the successive convex approximations to non-convex sets introduced in Kojima & Tuncel [80]. We also refer the reader to Laurent [86], and the recent survey by Laurent & Rendl [89] for a comparison of these various approaches. Finally, MATLAB code based on the above approach have been developed by Prajna et al. [129], and Henrion & Lasserre [65].

8.2 Copositive formulations of IP and SDP approximations of copositive programs

As another instance of convex approximations to integer programming, we consider the problem of finding the stability number of a graph. This problem can be expressed as a copositive program (see Quist et al. [131] and Bomze et al. [20]), that is a convex optimization problem. Recently, De Klerk & Pasechnik [35] apply the technique of approximating the copositive cone through a series of semidefinite approximations introduced by Parrilo [121], and use this to estimate the stability number of the graph to any degree of accuracy. We present a brief overview of their approach in this section.

The stability number of a graph $G = (V, E)$, denoted by $\alpha(G)$, can be expressed as the solution to the following copositive programming problem (due to Quist et al. [131], and is based on an earlier representation of $\alpha(G)$ due to Motzkin & Strauss [106]) that amounts to minimizing a particular quadratic function over the simplex.

$$\begin{array}{lrcl} \min & \lambda & & \\ \text{s.t.} & \lambda I + yA - S & = & ee^T, \\ & S & \in & \mathcal{C}_n, \end{array} \tag{40}$$

with dual

$$\begin{array}{lrcl} \max & ee^T \bullet X & & \\ \text{s.t.} & I \bullet X & = & 1, \\ & A \bullet X & = & 0, \\ & X & \succeq & 0, \end{array} \tag{41}$$

where $\lambda, y \in \mathcal{R}$, e is the all-ones vector, A is the adjacency matrix of the graph $G = (V, E)$, and $\mathcal{C}_n = \{X \in \mathcal{S}^n : d^T X d \geq 0, \forall d \geq 0\}$ is the set of $n \times n$ symmetric copositive matrices. The problem (40) is not solvable in polynomial time since the decision problem whether a matrix is copositive or not is NP-hard (Murthy & Kabadi [108]). In fact, De Klerk and Pasechnik [35] show that the equality constraints in (41) can be combined together as $(A+I) \bullet X = 1$. Thus, we can drop the additional variable y in (40), and rewrite the slack matrix as $S = \lambda(I + A) - ee^T$.

A sufficient condition for a matrix M to be copositive is $M \succeq 0$. In fact, setting $S \succeq 0$ in (40) gives a constrained version of the following SDP representing the Lovász

theta function (see Lovász [91] and Grötschel et al. [55]).

$$\begin{array}{lrcl} \min & \lambda & & \\ \text{s.t.} & \lambda I + \sum_{\{i,j\}\in E} y_{ij}E_{ij} - S & = & ee^T, \\ & S & \succeq & 0, \end{array} \tag{42}$$

where $E_{ij} \in \mathcal{S}^n$ is the elementary matrix with all zero entries, except entries 1 in positions (i,j) and (j,i), corresponding to edge $\{i,j\}$ in the graph. In the search for stronger sufficient conditions for copositivity, Parrilo [121, 120] proposes approximating the copositive cone using SOS representations of polynomials. To see this, note that a matrix $M \in \mathcal{C}_n$ if and only if the polynomial

$$g_M(x) \;=\; \sum_{i,j=1}^{n} M_{ij}x_i^2x_j^2$$

is nonnegative on $\mathcal{R}^n$. Therefore, a sufficient condition for M to be copositive is that $g_M(x)$ has an SOS representation, or more generally the polynomial $g_M(x)(\sum_{i=1}^{n} x_i^2)^r$ has an SOS representation for some integer $r \geq 0$. In fact a theorem due to Polya suggests that M is copositive, then $g_M(x)(\sum_{i=1}^{n} x_i^2)^r$ has an SOS representation for some r. An upper bound on r is given by Powers & Reznick [128].

If we define $\mathcal{K}_n^r$ to be the set of matrices in $\mathcal{S}^n$ for which $g_M(x)(\sum_{i=1}^{n} x_i^2)^r$ has an SOS representation, we have the following hierarchy of approximations to $\mathcal{C}_n$.

$$\begin{array}{rcccl} \mathcal{S}_+^n & \subseteq & \mathcal{K}_n^0 & \subseteq & \cdots \\ & \subseteq & \mathcal{K}_n^r & = & \mathcal{C}_n. \end{array} \tag{43}$$

For each $\mathcal{K}_n^r$, one can define the parameter

$$\gamma^r(G) \;=\; \min \;\; \lambda \;\; \text{s.t.} \;\; \lambda I + yA - ee^T \in \mathcal{K}_n^r, \tag{44}$$

where $\gamma^r(G) = \alpha(G)$ for some r. It was remarked in the previous section that the SOS requirement on a polynomial can be written as a semidefinite program, and so (44) represents a hierarchy of semidefinite programs, whose objective values eventually converge to the stability number of the graph. Parrilo [121] gives explicit SDP representations for $\mathcal{K}_n^r$, $r = 0, 1$. For instance $S \in \mathcal{K}_n^0$, if and only if $S = P + N$, for $P \succeq 0$, and $N \geq 0$. For the stable set problem, this first lifting gives the Schrijver formulation (see Schrijver [139]) of the Lovász theta function. In particular using Powers & Reznick's [128] estimate of r, De Klerk & Pasechnik show that

$$\alpha(G) \;=\; \lfloor \gamma^r(G) \rfloor, \;\; \text{if } r \geq \alpha^2(G).$$

We refer the reader to De Klerk & Pasechnik [35] for more details.

One obtaines the same result by applying the hierarchy of SDP approximations due to Lasserre (discussed in Section 8.1) on the original Motzkin & Strauss [106] formulation for the maximum stable set problem. In fact, De Klerk et al. [34] have shown that the polynomial progamming approach of Section 8.1 are in fact equivalent for the problem of minimizing a quadratic function over the simplex (standard quadratic programming problem).

Recently, Bomze & De Klerk [21] developed the first polynomial time approximation scheme (PTAS) for the standard quadratic programming problem, by applying a similar technique of LP and SDP approximations to the copositive cone. A good account also appears in the recent survey by De Klerk [33].

As of now, copositive programming has only been applied to the standard quadratic programming problem (De Klerk [32]). It is therefore interesting to speculate on other classes of problems that can be modelled as copositive programs.

9 Conclusions

We have presented an overview of some of the most recent developments in IPMs for solving various combinatorial optimization problems. IPMs are adapted in a number of ways to solving the underlying discrete problem; directly via a potential reduction approach in Section 2, in conjunction with an oracle in a cutting plane approach in Section 3, or applied to SDP relaxations or other convex reformulations of these problems as discussed in Sections 6 and 8. SDP is a major tool in continuous approaches to combinatorial problems, and IPMs of Section 4 can also be used in conjunction with ingenious randomized rounding schemes to generate solutions for various combinatorial optimization problems with provable performance guarantees. This was the topic of Section 7.

We conclude with a summary of some of the important issues, and open problems in the topics discussed:

1. The interior point cutting plane methods of Section 3, especially ACCPM, and its variants have been applied to solve a variety of convex optimization problems with some degree of practical success. It is interesting to speculate whether ACCPM is indeed a polynomial time solution procedure for the convex feasibility problem. The volumetric center IPM on the other hand has the best complexity among cutting plane methods which is provably optimal, and has rendered the

classical ellipsoid algorithm obsolete. Recent work by Anstreicher [8] has considerably improved the constants involved in the analysis of the algorithm, and it would be interesting to consider practical implementations of this algorithm in the near future.

2. The primal-dual IPMs described in Section 4.2 are indeed the algorithms of choice for SDP; however as of now they are fairly limited in the size of problems they can handle in computational practice. The ability of future IPMs to handle large SDPs will depend to a great extent on the design of good pre-conditioners (Toh [150], Toh & Kojima [151]), that are required in an iterative method to solve the normal system of equations. On the other hand, the first order approaches discussed in Section 5 exploit the structure in the underlying SDP problem, and are consequently able to solve larger problems; albeit to a limited accuracy.

3. On the theoretical side, the complexity of the semidefinite feasibility problem (SDFP) discussed in Section 4.1 is still an open problem.

4. There have been several applications of SDP to hard discrete optimization problems as discussed in Section 7 of this survey. However, to the best of our knowledge, there have been relatively few applications of second order cone programming (SOCP) in combinatorial optimization. In this regard we note the work of Kim & Kojima [76] and Muramatsu & Suzuki [107]. An open question is whether one could develop good approximation algorithms for combinatorial optimization using SOCP relaxations of the underlying problem, since the SOCP can be solved more quickly than SDP using IPMs.

5. An important issue in the branch and cut approaches discussed in Section 6 is that of restarting the new relaxation with a strictly interior point after branching, or the addition of cutting planes. In this regard, it is interesting to consider dual analogues of the primal active set approaches investigated in Krishnan et al. [83], which conceivably (like the dual simplex method for LP) could be employed for re-optimization.

6. One of the major applications of the SDP is its use in developing approximation algorithms for various combinatorial optimization problems as discussed in Section 7. In many cases, such as the MAX 3 SAT problem, the SDP in conjunction with rounding schemes provides the tightest possible approximation algorithms

for these problems unless P = NP. Recently, there has been renewed interest in SDP approximations to polynomial and copositive programming, which are provably exact in the limit. We discussed some of these ideas in Section 8. Although, there are a variety of problems that can be modelled as polynomial programs, the situation with respect to copositive programming is far less clear. In this regard it is interesting to speculate on the classes of problems, that can be written as copositive programs.

10 Acknowledgements

The authors would like to thank an anonymous referee whose comments greatly improved the presentation of the paper.

References

[1] F. Alizadeh and D. Goldfarb, *Second-order cone programming*, Mathematical Programming, 95(2003), pp. 3-51.

[2] F. Alizadeh, J.P.A. Haeberly, and M.L. Overton, *Primal-dual interior-point methods for semidefinite programming: Convergence rates, stability and numerical results*, SIAM Journal on Optimization, 8(1998), pp. 746-768.

[3] E.D. Andersen, J. Gondzio, Cs. Mészáros, and X. Xu, *Implementation of interior point methods for large scale linear programming*, In Interior Point Methods for Linear Programming: T. Terlaky (editor), Kluwer Academic Publishers, The Netherlands, (1996), pp. 189-252.

[4] M.F. Anjos, *An improved semidefinite programming relaxation for the satisfiability problem*, to appear in Mathematical Programming 2004.

[5] M.F. Anjos and H. Wolkowicz, *Geometry of semidefinite max-cut relaxations via matrix ranks*, Journal of Combinatorial Optimization, 6(2002), pp. 237-270.

[6] M.F. Anjos and H. Wolkowicz, *Strengthened semidefinite relaxations via a second lifting for the maxcut problem*, Discrete Applied Mathematics, 119(2002), pp. 79-106.

[7] K.M. Anstreicher, *On Vaidya's volumetric ctting plane method for convex programming*, Mathematics of Operations Research, 22(1997), pp. 63-89.

[8] K.M. Anstreicher, *Towards a practical volumetric cutting plane method for convex programming*, SIAM Journal on Optimization, 9(1999), pp. 190-206.

[9] K.M. Anstreicher, *The volumetric barrier for semidefinite programming*, Mathematics of Operations Research, 25(2000), pp. 365-380.

[10] S. Arora and C. Lund, *Hardness of Approximations*, In D. Hochbaum (editor), Approximation Algorithms for NP-hard Problems, PWS Publishing, 1996, (appeared as Chapter 10 in the volume).

[11] D.S. Atkinson and P.M. Vaidya, *A cutting plane algorithm for convex programming that uses analytic centers*, Mathematical Programming, 69(1995), pp. 1-43.

[12] D. Avis, *On the complexity of testing hypermetric, negative type, k-gonal and gap inequalities*, to appear in *Lecture Notes in Computer Science*, 2004.

[13] D. Avis and V.P. Grishukhin, *A bound on the k-gonality of facets of the hypermetric cone and related complexity problems*, Computational Geometry: Theory & Applications, 2(1993), pp. 241-254.

[14] O. Bahn, O. Du Merle, J.L. Goffin, and J.P. Vial, *Solving nonlinear multicommodity network flow problems by the analytic center cutting plane method*, Mathematical Programming, 76(1997), pp. 45-73.

[15] E. Balas, S. Ceria, and G. Cornuejols, *A lift and project cutting plane algorithm for mixed 0-1 programs*, Mathematical Programming, 58(1993), pp. 295-324.

[16] F. Barahona, *The maxcut problem in graphs not contractible to K_5*, Operations Research Letters, 2(1983), pp. 107-111.

[17] F. Barahona and A.R. Mahjoub, *On the cut polytope*, Mathematical Programming, 44(1986), pp. 157-173.

[18] H.Y. Benson and R.J. Vanderbei, *Solving problems with semidefinite and related constraints using interior-point methods for nonlinear programming*, Mathematical Programming, 95(2003), pp. 279-302.

[19] S.J. Benson, Y. Ye, and X. Zhang, *Solving large-scale semidefinite programs for combinatorial optimization*, SIAM Journal on Optimization, 10 (2000), pp. 443-461.

[20] I. Bomze, M. Dur, E. De Klerk, C. Roos, A.J. Quist, and T. Terlaky, *On copositive programming and standard quadratic optimization problems*, Journal of Global Optimization, 18(2000), pp. 301-320.

[21] I. Bomze and E. De Klerk, *Solving standard quadratic optimization problems via linear, semidefinite and copositive programming*, Journal of Global Optimization, 24(2002), pp. 163-185.

[22] B. Borchers, *CSDP: A C library for semidefinite programming*, Optimization Methods and Software, 11(1999), pp. 613-623.

[23] S. Burer, *Semidefinite programming in the space of partial positive semidefinite matrices*, SIAM Journal on Optimization, 14(2003), pp. 139-172.

[24] S. Burer and R.D.C. Monteiro, *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Mathematical Programming, 95(2003), pp. 329-357.

[25] S. Burer and R.D.C. Monteiro, *Local minima and convergence in low-rank semidefinite programming*, Technical Report, Department of Management Sciences, University of Iowa, September 2003.

[26] S. Burer, R.D.C. Monteiro, and Y. Zhang, *Solving a class of semidefinite programs via nonlinear programming*, Mathematical Programming, 93(2002), pp. 97-102.

[27] S. Burer, R.D.C. Monteiro, and Y. Zhang, *A computational study of a gradient based log-barrier algorithm for a class of large-scale SDPs*, Mathematical Programming, 95(2003), pp. 359-379.

[28] 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(2002), pp. 503-521.

[29] S. Burer, R.D.C. Monteiro, and Y. Zhang, *Maximum stable set formulations and heuristics based on continuous optimization*, Mathematical Programming, 94(2002), pp. 137-166.

[30] V. Chvátal, *Linear Programming*, W.H. Freeman and Company, 1983.

[31] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Trust-Region Methods*, MPS-SIAM Series on Optimization, 2000.

[32] E. De Klerk, personal communication, 2003.

[33] E. De Klerk, *Aspects of Semidefinite Programming: Interior Point Algorithms and Selected Applications*, Applied Optimization Series, Vol. 65, Kluwer Academic Publishers, May 2002.

[34] E. De Klerk, M. Laurent, and P. Parrilo, *On the equivalence of algebraic approaches to the minimization of forms on the simplex*, Preprint, CWI, Amsterdam, October 2003.

[35] E. De Klerk and D. Pasechnik, *Approximating the stability number of graph via copositive programming*, SIAM Journal on Optimization, 12(2002), pp. 875-892.

[36] E. De Klerk, D.V. Pasechnik, and J.P. Warners, *Approximate graph coloring and Max-k-cut algorithms based on the ζ function*, to appear in Journal of Combinatorial Optimization, 2004.

[37] E. De Klerk, C. Roos, and T. Terlaky, *Infeasible-start semidefinite programming algorithms via self-dual embeddings*, Fields Institute Communications, 18(1998), pp. 215-236.

[38] E. De Klerk and H. Van Maaren, *On semiefinite programming relaxations of 2+p-SAT*, Annals of Mathematics of Artificial Intelligence, 37(2003), pp. 285-305.

[39] E. De Klerk, J. Warners, and H. Van Maaren, *Relaxations of the satisfiability problem using semidefinite programming*, Journal of Automated Reasoning, 24(2000), pp. 37-65.

[40] M.M. Deza and M. Laurent, *Geometry of Cuts and Metrics*, Springer-Verlag, Berlin, 1997.

[41] S. Elhedhli and J.L. Goffin, *The integration of an interior-point cutting plane method within a branch-and-price algorithm*, to appear in Mathematical Programming 2004.

[42] U. Feige and M. Goemans, *Approximating the value of two prover proof systems withe applications to MAX-2SAT and MAX-DICUT*, In Proceedings of the 3rd Isreal Symposium on Theory of Computing and Systems, ACM, NY, pp. 182-189, 1995.

[43] A. Frieze and M.R. Jerrum, *Improved approximation algorithms for Max k-cut and Max Bisection*, Algorithmica 18(1997), pp. 61-77.

[44] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion I: General framework*, SIAM Journal on Optimization, 11(2000), pp. 647-674.

[45] M. Fukuda, M. Kojima, and M. Shida, *Lagrangian dual interior-point methods for semidefinite programs*, SIAM Journal on Optimization, 12(2002), pp. 1007-1031.

[46] M.R. Garey and D.S. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness*, San Francisco, California: W.H. Freeman & Company, 1979.

[47] M. Goemans and D.P. Williamson, *Improved approximation algorithms for max cut and satisfiability problems using semidefinite programming*, J. A.C.M 42 (1995), pp. 1115-1145.

[48] M. Goemans and D.P. Williamson, *Approximation algorithms for MAX-3-CUT and other problems via complex semidefinite programming*, In Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, ACM, NY, pp. 443-452, 2001.

[49] J.L. Goffin, J. Gondzio, R. Sarkissian, and J.P. Vial, *Solving nonlinear multicommodity flow problems by the analytic center cutting plane method*, Mathematical Programming, 76(1997), pp. 131-154.

[50] J.L. Goffin, Z.Q. Luo, and Y. Ye, *Complexity analysis of an interior point cutting plane method for convex feasibility problems*, SIAM Journal on Optimization, 6(1996), pp. 638-652.

[51] J.L. Goffin and J.P. Vial, *Convex nondifferentiable optimization: a survey focused on the analytic center cutting plane method*, Optimization Methods and Software, 17(2002), pp. 805-867.

[52] J.L. Goffin and J.P. Vial, *Multiple cuts in the analytic center cutting plane method*, SIAM Journal on Optimization, 11(2000), pp. 266-288.

[53] J. Gondzio, O. du Merle, R. Sarkissian, and J.P. Vial, *ACCPM - A library for convex optimization based on an analytic center cutting plane method*, European Journal of Operations Research, 94(1996), pp. 206-211.

[54] B. Grone, C.R. Johnson, E. Marques de Sa, and H. Wolkowicz, *Positive definite completions of partial Hermitian matrices*, Linear Algebra and its Applications, 58(1984), pp. 109-124.

[55] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Springer Verlag, 1993.

[56] J. Håstad, *Some optimal inapproximability results*, In Proc. of the 29th ACM Symposium on Theory and Computing, (1997), pp. 1-10.

[57] C. Helmberg, *Semidefinite Programming for Combinatorial Optimization*, Habilitation Thesis, ZIB-Report ZR-00-34, Konrad-Zuse-Zentrum Berlin, October 2000.

[58] C. Helmberg, *Fixing variables in semidefinite relaxations*, SIAM Journal on Matrix Analysis and Applications, 21(2000), pp. 952-969.

[59] C. Helmberg, *Numerical evaluation of SBmethod*, Mathematical Programming, 95(2003), pp. 381-406.

[60] C. Helmberg and F. Oustry, *Bundle methods to minimize the maximum eigenvalue function*. In Handbook of Semidefinite Programming: H. Wolkowicz, R. Saigal, and L. Vandenberghe (editors), Kluwer Academic Publishers, (2000), pp. 307-337.

[61] C. Helmberg and F. Rendl, *A spectral bundle method for semidefinite programming*, SIAM Journal on Optimization, 10(2000), pp. 673-696.

[62] C. Helmberg and F. Rendl, *Solving quadratic (0,1) problems by semidefinite programs and cutting planes*, Mathematical Programming, 82(1998), pp. 291-315.

[63] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, *An interior point method for semidefinite programming*, SIAM Journal on Optimization, 6(1996), pp. 673-696.

[64] C. Helmberg, *Semidefinite Programming webpage*, *http://www-user.tu-chemnitz.de/∼helmberg/semidef.html*

[65] D. Henrion and J. Lasserre, *Gloptipoly: Global optimization over polynomials with MATLAB and SeDuMi*, ACM Transactions on Mathematical Software, 29(2003), pp. 165-194.

[66] D. Henrion and J. Lasserre, *Detecting global optimality and extracting solutions in Globtipoly*, Technical Report LAAS-CNRS, October 2003.

[67] Interior-Point Methods Online, *http://www-unix.mcs.anl.gov/otc/InteriorPoint*

[68] A.P. Kamath, N. Karmarkar, K.G. Ramakrishnan, and M.G.C. Resende, *Computational experience with an interior point algorithm on the satisfiability problem*, Annals of OR., 25(1990), pp. 43-58.

[69] A.P. Kamath, N. Karmarkar, K.G. Ramakrishnan, and M.G.C. Resende, *A continuous approach to inductive inference*, Mathematical Programming 57(1992), pp. 215-238.

[70] D. Karger, R. Motwani, and M. Sudan, *Approximate graph coloring by semidefinite programming*, Journal of A.C.M., 45(1998), pp. 246-265.

[71] H. Karloff, *How good is the Goemans-Williamson MAX CUT algorithm?*, SIAM Journal on Computing, 29(1999), pp. 336-350.

[72] H. Karloff and U. Zwick, *A 7/8 approximation algorithm for MAX 3SAT?*, In Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Science Press, Los Alamitos, CA, pp. 406-415, 1997.

[73] N. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica, 4(1984), pp. 373-395.

[74] N. Karmarkar, *An interior point approach to NP-complete problems*, Contemporary Mathematics, 114(1990), pp. 297-308.

[75] N. Karmarkar, M.G.C. Resende, and K.G. Ramakrishnan, *An interior point approach to solve computationally difficult set covering problems*, Mathematical Programming, 52(1991), pp. 597-618.

[76] S. Kim and M. Kojima, *Second order cone programming relaxations of nonconvex quadratic optimization problems*, Optimization Methods & Software, 15(2001), pp. 201-224.

[77] J. Kleinberg and M. Goemans, *The Lovász theta function and a semidefinite programming relaxation of vertex cover*, SIAM Journal on Discrete Mathematics, 11(1998), pp. 196-204.

[78] M. Kocvara and M. Stingl, *PENNON: A code for convex nonlinear and semidefinite programming*, Optimization Methods & Software, 18(2003), pp. 317-333.

[79] M. Kojima, S. Shindoh, and S. Hara, *Interior-point methods for the monotone linear complementarity problem in symmetric matrices*, SIAM Journal on Optimization, 7(1997), pp. 86-125.

[80] M. Kojima and L. Tuncel, *Cones of matrices and successive convex relaxations of nonconvex sets*, SIAM Journal on Optimization, 10(2000), pp. 750-778.

[81] K. Krishnan and J.E. Mitchell, *Properties of a cutting plane method for semidefinite programming*, Technical Report, Department of Computational & Applied Mathematics, Rice University, May 2003.

[82] K. Krishnan and J.E. Mitchell, *An unifying survey of exisiting cutting plane methods for semidefinite programming*, AdvOL-Report No. 2004/1, Advanced Optimization Laboratory, McMaster University, December 2003 (to appear in Optimization Methods & Software).

[83] K. Krishnan, G. Pataki, and Y. Zhang, *A non-polyhedral primal active set approach to semidefinite programming*, Technical Report, Dept. of Computational & Applied Mathematics, Rice University, forthcoming.

[84] J.B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11(2001), pp. 796-817.

[85] J.B. Lasserre, *An explicit exact SDP relaxation for nonlinear 0-1 programs*, SIAM Journal on Optimization, 12(2002), pp. 756-769.

[86] M. Laurent, *A comparison of the Sherali-Adams, Lovász-Schrijver and Lasserre relaxations for 0-1 programming*, Mathematics of Operations Research, 28(2003), pp. 470-496.

[87] M. Laurent, *A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems*, In Topics in Semidefinite and Interior Point Methods, Volume 18 of The Fields Institute for Research in Mathematical Sciences, Communications Series, Providence, Rhode Island, 1998, AMS.

[88] M. Laurent, *Semidefinite relaxations for max-cut*, In M. Grötschel (editor), The Sharpest Cut, Festschrift in honor of M. Padberg's 60th birthday, pp. 291-327, MPS-SIAM, 2004.

[89] M. Laurent and F. Rendl, *Semidefinite programming and integer programming*, Technical Report PNA-R0210, CWI, Amsterdam, April 2003.

[90] C. Lemaréchal and F. Oustry, *Semidefinite relaxations and Lagrangian duality with applications to combinatorial optimization*, RR-3710, INRIA Rhone-Alpes, June 1999.

[91] L. Lovász, *On the Shannon capacity of a graph*, IEEE Transactions on Information Theory, 25(1979), pp. 1-7.

[92] L. Lovász and A. Schrijver, *Cones of matrices and set functions and 0-1 optimization*, SIAM Journal on Optimization, 1(1991), pp. 166-190.

[93] Z.Q. Luo and J. Sun, *An analytic center based column generation method for convex quadratic feasibility problems*, SIAM Journal on Optimization, 9(1998), pp. 217-235.

[94] S. Mahajan and R. Hariharan, *Derandomizing semidefinite programming based approximation algorithms*, SIAM Journal on Computing, 28(1999), pp. 1641-1663.

[95] J.E. Mitchell,*Computational Experience with an Interior Point Cutting Plane Algorithm*, SIAM Journal on Optimization, 10(2000), pp. 1212-1227.

[96] J.E. Mitchell, *Polynomial interior point cutting plane methods*, Optimization Methods & Software, 18(2003), pp. 507-534.

[97] J.E. Mitchell, *Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems*, Journal of Combinatorial Optimization, 5(2001), pp. 151-166.

[98] J.E. Mitchell, P. Pardalos, and M.G.C. Resende, *Interior point methods for combinatorial ptimization*, In Handbook of Combinatorial Optimization, Vol. 1 (1998), Kluwer Academic Publishers, pp. 189-297.

[99] J.E. Mitchell and B. Borchers, *Solving real-world linear ordering problems using a primal-dual interior point cutting plane method*, Annals of Operations Research, 62(1996), pp. 253-276.

[100] J.E. Mitchell and S. Ramaswamy, *A long step cutting plane algorithm for linear and convex programming*, Annals of OR, 99(2000), pp. 95-122.

[101] J.E. Mitchell and M.J. Todd, *Solving combinatorial optimization problems using Karmarkar's algorithm*, Mathematical Programming, 56(1992), pp. 245-284.

[102] H.D. Mittleman, *An independent benchmarking of SDP and SOCP software*, Mathematical Programming, 95(2003), pp. 407-430.

[103] F.S. Mokhtarian and J.L. Goffin, *A nonlinear analytic center cutting plane method for a class of convex programming problems*, SIAM Journal on Optimization, 8(1998), pp. 1108-1131.

[104] R.D.C Monteiro, *Primal-dual path following algorithms for semidefinite programming*, SIAM Journal on Optimization, 7(1997), pp. 663-678.

[105] R.D.C. Monteiro, *First and second order methods for semidefinite programming*, Mathematical Programming, 97(2003), pp. 209-244.

[106] T.S. Motzkin and E.G. Strauss, *Maxima for graphs and a new proof of a theorem of Turan*, Canadian Journal of Mathematics, 17(1965), pp. 533-540.

[107] M. Muramatsu and T. Suzuki, *A new second-order cone programming relaxation for maxcut problems*, to appear in *Journal of Operations Research of Japan*, 2002.

[108] K.G. Murthy and S.N. Kabadi, *Some NP-complete problems in quadratic and linear programming*, Mathematical Programming, 39(1987), pp. 117-129.

[109] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, *Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results*, Mathematical Programming, 95(2003), pp. 303-327.

[110] A.S. Nemirovskii and D.B. Yudin, *Problem Complexity and Method Efficiency in Optimization*, John Wiley, 1983.

[111] Y.E. Nesterov, *Semidefinite relaxation and nonconvex quadratic optimization*, Optimization Methods and Software, 9(1998), pp. 141-160.

[112] Y.E. Nesterov, *Squared functional systems and optimization problems*, In J.B.G. Frenk, C. Roos, T. Terlaky and S. Zhang (editors), High Performance Optimization, pp. 405-440, Kluwer Academic Publishers, 2000.

[113] Y.E. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, SIAM Studies in Applied Mathematics, PA, 1993.

[114] Y.E. Nesterov and M.J. Todd, *Primal dual interior point methods for self-scaled cones*, SIAM Journal on Optimization, 8(1998), pp. 324-364.

[115] Optimization Online, *http://www.optimization-online.org*

[116] M. Oskoorouchi and J.L. Goffin, *The analytic center cutting plane method with semidefinite cuts*, SIAM Journal on Optimization, 13(2003), pp. 1029-1053.

[117] M. Oskoorouchi and J.L. Goffin, *An interior point cutting plane method for convex feasibility problems with second order cone inequalities*, Technical Report, College of Business Administration, California State University, San Marcos, April 2003.

[118] F. Oustry, *A second order bundle method to minimize the maximum eigenvalue function*, Mathematical Programming, 89(2000), pp. 1-33.

[119] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice Hall, 1982.

[120] P.A. Parrilo, *Structured Semidefinite Programs and Semialgebraic methods in Robustness and Optimization*, Ph.D. Thesis, California Institute of Technology, May 2000.

[121] P.A. Parrilo, *Semidefinite programming relxations for semialgebraic problems*, Mathematical Programming, 96(2003), pp. 293-320.

[122] G. Pataki, *On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues*, Mathematics of Operations Research, 23(1998), pp. 339-358.

[123] G. Pataki, *Cone-LP's and semidefinite programs: geometry and simplex type method*, In Proceedings of the 5th IPCO Conference, Lecture Notes in Computer Science, 1084(1996), pp. 162-174.

[124] G. Pataki, *Cone Programming and Nonsmooth Optimization: Geometry and Algorithms*, Ph.D. Thesis, Graduate School of Industrial Administration, Carnegie Mellon University, November 1996.

[125] S. Poljak, F. Rendl, and H. Wolkowicz, *A recipe for semidefinite relxations for* $\{0,1\}$*-quadratic programming*, Journal of Global Optimization, 7(1995), pp. 51-73.

[126] S. Poljak and Z. Tuza, *The expected relative error of the polyhedral approximation of the maxcut problem*, Operations Research Letters, 16(1994), pp. 191-198.

[127] L. Porkoláb and L. Khachiyan, *On the complexity of semidefinite programs*, Journal of Global Optimization, 10(1997), pp. 351-365.

[128] V. Powers and B. Reznick, *A new bound for Polya's theorem with applications to polynomials positive on polyhedra*, Journal of Pure and Applied Algebra, 164(2001), pp. 221-229.

[129] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*, Available at *http://www.cds.caltech.edu/sostools*, 2002.

[130] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana University Mathematics Journal, 42(1993), pp. 969-984.

[131] A.J. Quist, E. De Klerk, C. Roos, and T. Terlaky, *Copositive relaxations for general quadratic programming*, Optimization Methods and Software, 9(1998), pp. 185-209.

[132] M. Ramana, *An algorithmic analysis of multiquadratic and semidefinite programming problems*, Ph.D. Thesis, The John Hopkins University, October 1993.

[133] M. Ramana, *An exact duality theory for semidefinite programming and its complexity implications*, Mathematical Programming, 77(1997), pp. 129-162.

[134] M. Ramana, L. Tuncel, and H. Wolkowicz, *Strong duality for semidefinite programming*, SIAM Journal on Optimization, 7(1997), pp. 641-662.

[135] F. Rendl and R. Sotirov, *Bounds for the quadratic assignment problem using the bundle method*, Technical Report, University of Klagenfurt, Universitaetsstrasse 65-67, Austria, August 2003.

[136] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, MPS-SIAM Series on Optimization, 2001.

[137] C. Roos, T. Terlaky, and J.P. Vial, *Theory and Algorithms for Linear Optimization: An Interior Point Approach*, John Wiley & Sons, Chichester, England, 1997.

[138] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley-Interscience, New York, 1986.

[139] A. Schrijver, *A comparison of the Delsarte and Lovász bounds*, IEEE Transactions on Information Theory, 25(1979), pp. 425-429.

[140] P.D. Seymour, *Matroids and multicommodity flows*, European Journal of Combinatorics, 2(1981), pp. 257-290.

[141] H.D. Sherali and W.P. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, SIAM Journal on Discrete Mathematics, 3(1990), pp. 411-430.

[142] N. Shor, *Nondifferentiable Optimization and Polynomial Problems*, Kluwer Academic Publishers, 1998.

[143] J.F. Sturm, *Primal-Dual Interior Point Approach to Semidefinite Programming*, Tinbergen Institute Research Series, vol 156, Thesis Publishers, Amsterdam, The Netherlands, 1997; also appeared in *High performance optimization*, Frenk et al. editors, Kluwer Academic Publishers, 2000.

[144] J.F. Sturm, *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11-12(1999), pp. 625-653.

[145] J. Sun, K.C. Toh, and G.Y. Zhao, *An analytic center cutting plane method for the semidefinite feasibility problem*, Mathematics of Operations Research, 27(2002), pp. 332-346.

[146] T. Terlaky, *Interior Point Methods of Mathematical Programming*, Kluwer Academic Publishers, 1996.

[147] M.J. Todd, *Semidefinite optimization*, Acta Numerica, 10 (2001), pp. 515-560.

[148] M.J. Todd, *A study of search directions in interior point methods for semidefinite programming*, Optimization Methods and Software, 12(1999), pp. 1-46.

[149] M.J. Todd, K.C. Toh, and R.H. Tütüncü, *On the Nesterov-Todd direction in semidefinite programming*, SIAM Journal on Optimization, 8(1998), pp. 769-796.

[150] K.C. Toh, *Solving large scale semidefinite programs via an iterative solver on the augmented system*, SIAM Journal on Optimization, 14(2003), pp. 670-698.

[151] K.C. Toh and M. Kojima, *Solving some large scale semidefinite programs via the conjugate residual method*, SIAM Journal on Optimization, 12(2002), pp. 669-691.

[152] K.C. Toh, G.Y. Zhao, and J. Sun, *A multiple-cut analytic center cutting plane method for semidefinite feasibility problems*, SIAM Journal on Optimization, 12(2002), pp. 1026-1046.

[153] R.H. Tütüncü, K.C. Toh, and M.J. Todd, *Solving semidefinite-quadratic-linear programs using SDPT3*, Mathematical Programming, 95(2003), pp. 189-217.

[154] P.M. Vaidya, *A new algorithm for minimizing convex functions over convex sets*, Mathematical Programming, 73(1996), pp. 291-341.

[155] L. Vandenberghe and S. Boyd, *Semidefinite Programming*, SIAM Review, 38(1996), pp. 49-95.

[156] S.A. Vavasis, *Nonlinear Optimization*, Oxford Science Publications, New York, 1991.

[157] J.P. Warners, B. Jansen, C. Roos, and T. Terlaky, *A potential reduction approach to the frequency assignment problem*, Discrete Applied Mathematics, 78(1997), pp. 252-282.

[158] J.P. Warners, B. Jansen, C. Roos, and T. Terlaky, *Potential reduction approaches for structured combinatorial optimization problems*, Operations Research Letters, 21(1997), pp. 55-65.

[159] S.J. Wright, *Primal-dual interior point methods*, SIAM, Philadelphia, 1997.

[160] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook on Semidefinite Programming*, Kluwer Academic Publishers, 2000.

[161] Y. Ye, *Approximating quadratic programming with bound and quadratic constraints*, Mathematical Programming, 84(1999), pp. 219-226.

[162] Y. Ye, *A .699 approximation algorithm for max bisection*, Mathematical Programming, 90(2001), pp. 101-111.

[163] Y. Ye, *Interior Point Algorithms: Theory and Analysis*, John Wiley & Sons, New York, 1997.

[164] Y. Zhang, *On extending some primal-dual interior point algorithms from linear programming to semidefinite programming*, SIAM Journal on Optimization, 8(1998), pp. 365-386.

[165] U. Zwick, *Outward rotations: A tool for rounding solutions of semidefinite programming relaxations, with applications to maxcut and other problems*, In Proc. of 31st STOC, (1999), pp. 496-505.