TUTORIALS IN
OPERATIONS RESEARCH
INFORMS—New Orleans 2005

 | ISBN 0000-0000
DOI 10.1287/educ.1053.0000

53

Cutting Plane Methods and Subgradient Methods

John E. Mitchell
Mathematical Sciences, Rensselaer Polytechnic Institute, Troy NY 12180, USA. mitchj@rpi.edu

Abstract Interior point methods have proven very successful at solving linear programming problems. When an explicit linear programming formulation is either not available or is too large to employ directly, a column generation approach can be used. Examples of column generation approaches include cutting plane methods for integer programming and decomposition methods for many classes of optimization problems. We discuss the use of interior point methods in a column generation scheme.

Semidefinite programming relaxations of combinatorial optimization problems are often tighter than linear programming relaxations. We describe some research in using SDP relaxations to find exact solutions to combinatorial optimization problems. Semidefinite programs are expensive to solve directly, so we also consider cutting surface approaches to solving them.

Finally, we look at recent smoothing techniques for solving nonsmooth optimization problems using a subgradient approach; these methods have some links to cutting surface approaches.

Keywords interior point column generation, cutting planes, cutting surfaces, semidefinite programming, subgradients

1. Introduction

In the 25 years since the publication of Karmarkar's seminal paper [71], interior point algorithms have become the method of choice for large scale linear programming problems. They have also had a great impact beyond linear programming: the extension of the theory of interior point methods for linear programming to more general conic programming [109] has led to the explosion of interest in semidefinite programming and also second-order cone programming. Further, many packages for general nonlinear programming use interior point methods. For example, the package CVX [55] exploits interior point theory to solve convex optimization problems in an accessible MATLAB framework.

Many optimization problems require a possibly very large number of constraints or variables. For example, there has been great success solving integer programming problems in recent years with cutting plane and branch-and-cut methods, methods which work with linear programming relaxations of the integer program. In these methods, the number of potential cutting planes is typically at least exponential in the size of the problem, so writing out all of the constraints explicitly as a giant linear program is not a tractable approach, even with fast interior point LP solvers available. As another example, consider optimizing the Lagrangian dual function of an integer program or nonlinear program. The Lagrangian dual function is convex, and piecewise linear in certain cases such as integer programming, so it can in principle be converted into a linear program by using subgradients to the function at each point. But explicitly writing out this linear program is not a practical approach. As a third example, decomposition approaches are also not amenable to explicit formulation as linear programs. In all of these cases, an initial linear program is constructed to approximate the problem of interest, and this linear programming approximation is refined as the

algorithm proceeds. The refinement typically requires the addition of a set of constraints or a set of variables, and we refer to such methods as cutting plane or column generation approaches, depending on context.

In order to exploit the success of interior point methods for linear programming, it is desirable to use interior point methods in column generation algorithms. This is the topic of §2. Of course, the classical simplex method is used widely in column generation algorithms, because it is easy to warm start and to reoptimize when just a few columns are added at a time. We argue in §2 that interior point methods can compete with the simplex method when many constraints are added at once, or when it is necessary to stabilize the column generation process. Interior point column generation methods also have theoretical performance guarantees, unlike the corresponding simplex methods. It is also useful to combine interior point and simplex cutting plane methods: the interior point method is used in the initial stages, when stabilization is important and many variables may be added at a time, and simplex is used for the later stages once the process is reasonably close to optimality and just a few variables are added at a time.

Many combinatorial optimization problems have a semidefinite programming relaxation that is tighter than the linear programming relaxation (see, for example, [45, 82, 85]). The strength of the relaxation makes a cutting plane or branch-and-cut approach using the semidefinite programming relaxation attractive. Recently, there has been some computational interest in such approaches, and some of this work is discussed in §3.

Semidefinite programs can be solved by primal-dual interior point methods. However, these methods have notable linear algebra requirements, which makes them slow for larger scale problems. Alternative approaches based on decompositions or relaxations are being developed and these methods can be especially attractive when it is not necessary to solve the semidefinite program (SDP) to a high level of accuracy; such is the case when employing an SDP relaxation within a cutting plane approach. These decomposition approaches to solving semidefinite programs and other conic programs are the topic of §4.

Column generation methods can be used to optimize a convex nonsmooth function, with a piecewise linear underestimator to the function of interest refined as the algorithm proceeds. A subgradient algorithm to minimize such a function is analogous to a steepest descent approach for smooth functions. Recently, Nesterov [105] has proposed a smoothing approach for certain classes of nonsmooth functions, and then using a sophisticated subgradient algorithm to minimize the smoothed function. The optimal solution to the smoothed function is provably close to being an optimal solution to the nonsmooth function. This research is described in §5 and it can be used to develop an approach for semidefinite programming, along with other classes of problems.

We construct a single primal-dual pair of linear programming problems to illustrate both column generation and cutting plane methods:

$$\begin{array}{lll} \min_x & c^T x & \\ \text{subject to} & Ax = b & (P) \\ & x \geq 0 & \end{array} \qquad \begin{array}{lll} \max_{y,s} & b^T y & \\ \text{subject to} & A^T y + s = c & (D) \\ & s \geq 0. & \end{array}$$

The primal problem (P) is the model used in a column generation approach: additional variables are added to this problem. Conversely, the dual problem (D) is the model used in a cutting plane method, with additional constraints added as needed to (D). The parameter c and the variables x and s are all rational n-vectors, the parameter b and the variable y are rational m-vectors, and the rational matrix A is $m \times n$.

Interior point methods for linear programming can be motivated by using a barrier primal objective function of $c^T x - \mu \sum_{i=1}^{n} \ln x_i$, where μ is a positive parameter. The parameter μ is decreased as the algorithm iterates. The optimality conditions for the barrier problem consist of feasibility in (P) and (D) together with a shifted complementary slackness relation:

$$x_i s_i = \mu, \quad i = 1, \ldots, n. \tag{1}$$

For each $\mu > 0$ there is a unique solution to the barrier problem and the set of these solutions constitute the central path or central trajectory. Neighborhoods of the central path can be defined by looking at norms of the violation of (1). In practice, interior point methods work best if they are started close to the central trajectory and if the iterates remain in some neighborhood of the central trajectory.

Proximity to the central trajectory can also be measured using *potential functions:* the dual potential function is

$$\phi(s) := -\sum_{i=1}^{n} \ln\, s_i \tag{2}$$

and the primal-dual potential function is

$$\Phi_\rho(x,s) := \rho \ln\, x^T s - \sum_{i=1}^{n} \ln\,(x_i s_i), \tag{3}$$

where $\rho > n$ is a fixed parameter, typically equal to $n + \sqrt{n}$. Minimizing $\Phi_\rho(x,s)$ leads to an optimal solution to (P) and (D). If $\rho = n$ then the minimizers of $\Phi_\rho(x,s)$ are the points on the central trajectory. These potential functions are used in the complexity analysis of interior point column generation algorithms, as discussed in §2.3. Far more information on interior point methods for linear programming can be found in the books [126, 137, 138, 140].

Nesterov and Nemirovski [109] showed that interior point methods could also be used to solve more general conic programs, including semidefinite programs. The ideas of a central trajectory, neighborhoods, and potential functions carry over to the more general setting. Speed of convergence, in both theory and practice, depends on how closely centrality is observed. Renegar [125] gives a textbook exposition of interior point methods for conic programming.

There are several recent surveys that discuss topics of interest in this paper. Krishnan and Terlaky [80] consider semidefinite and interior point methods for solving combinatorial optimization problems. Goffin and Vial [48] and Mitchell [94] examine interior point cutting plane methods. Lübbecke and Desrosiers [88] describe column generation approaches, with a discussion of the importance of stabilization.

2. Interior point column generation for linear and integer programming

Column generation and cutting plane methods can be regarded as duals of each other. A column generation method is used when it is impractical to consider all the variables at once. For example, the variables may be generated only as needed, as in airline crew scheduling [10]: each variable corresponds to a pairing or schedule for a crew and there is a huge number of possible pairings, so useful pairings are constructed based on consideration of reduced costs.

Cutting plane methods are used when a relaxation of the problem is solved and this relaxation is gradually tightened through the addition of constraints, or cutting planes. For example, expressing the traveling salesman problem as an integer programming problem requires subtour elimination constraints; however, there is an exponential number of these constraints, so they are added only as needed [5, 6].

Problems (P) and (D) are approximations to the true problems we wish to solve, which are expressed as $(\hat{P})$ and $(\hat{D})$:

$$\begin{array}{lll} \min_{\hat{x}} & \hat{c}^T\hat{x} & \\ \text{subject to} & \hat{A}\hat{x} = b & (\hat{P}) \\ & \hat{x} \geq 0 & \end{array} \qquad \begin{array}{lll} \max_{y,\hat{s}} & b^T y & \\ \text{subject to} & \hat{A}^T y + \hat{s} = \hat{c} & (\hat{D}) \\ & \hat{s} \geq 0. & \end{array}$$

Here the parameter $\hat{c}$ and the variables $\hat{x}$ and $\hat{s}$ are all rational $\hat{n}$-vectors, the parameter b and the variable y are rational m-vectors, and the rational matrix $\hat{A}$ is $m \times \hat{n}$. The underlying assumption, and indeed rationale for using a cutting plane approach, is that we can work with problems (P) and (D) with $n << \hat{n}$. The problem (D) is a relaxation of $(\hat{D})$, since some of the constraints have been omitted. The problem (P) is a constrained version of $(\hat{P})$, since the missing variables are in effect forced to take the value zero.

If the optimal solution (x^*, y^*, s^*) to the primal-dual pair (P) and (D) is not optimal to the underlying problem then there must be additional dual constraints $A_0^T y \leq c_0$ that are violated by y^*. Here c_0 is a rational p-vector and the rational matrix A_0 is $m \times p$, with $p \geq 1$. The determination of a set of cutting planes may require solution of a subproblem; this is referred to as a call to an oracle. Adding these constraints to (D) results in the following primal-dual pair of linear programs:

$$\begin{array}{lll} \min_{x,x_0} & c^T x & \\ \text{subject to} & Ax + A_0 x_0 = b & (P_0) \\ & x, x_0 \geq 0 & \end{array} \qquad \begin{array}{lll} \max_{y,s,s_0} & b^T y & \\ \text{subject to} & A^T y + s = c & (D_0) \\ & A_0^T y + s_0 = c_0 & \\ & s, s_0 \geq 0 & \end{array}$$

where the variables x_0 and s_0 are rational p-vectors. This modified pair is then solved and the whole process repeated as necessary.

The identification of the cutting planes $A_0^T y \leq c_0$ does not require an optimal solution y^* to (D). An approximate solution may well be sufficient for the cutting plane generation routine. As we argue below in §2.1, an approximate solution may even result in stronger cutting planes. For example, Huisman *et al.* [68, page 255] state that

> "Bixby et al. [16] [and] Barnhart et al. [10] note that in case of alternative dual solutions, column generation algorithms seem to work better with dual variables produced by interior point methods than with dual variables computed with simplex algorithms. The latter give a vertex of the face of solutions whereas interior point algorithms give a point in the center of the face, providing a better representation of it."

The desired accuracy of the solution to the current relaxation can be adjusted as the algorithm proceeds. Typically, later LPs will be solved to a tighter tolerance than earlier ones. The conceptual framework of a cutting plane algorithm is given below:

> *0. Initialize:* Choose the initial parameters A, b, and c. Choose a tolerance ϵ for the overall algorithm. Choose a tolerance τ for the desired accuracy of the LP relaxations.
> *1. Approximately solve the LP:* Solve the current primal-dual linear programming pair to a relative accuracy of τ in the duality gap. Obtain a primal-dual pair $(\tilde{x}, \tilde{y}, \tilde{s})$.
> *2. Find cuts:* Determine if $\tilde{y}$ violates any additional dual constraints.
> *(a)* If no constraints are found and $\tau \leq \epsilon$ then STOP with a solution to the underlying problem.
> *(b)* If no constraints are found and $\tau > \epsilon$ then reduce τ and return to Step 1.
> *(c)* If constraints are found, add a subset of the constraints to the LP relaxation, modify τ if desired, and return to Step 1.
>
> Algorithm 1: Solving the LP pair $(\hat{P})$ and $(\hat{D})$.

The point $(\tilde{x}, \tilde{y}, \tilde{s})$ provides a warm start after the linear programming problem is modified in Step 2(c). The simplex method can exploit this warm start by solving the modified problem using the dual simplex method. It is not so straightforward to exploit a warm start when using an interior point method, but various techniques have been suggested and they

do reduce the number of interior point iterations. These techniques are discussed in §2.2. Theoretical properties of interior point cutting plane methods are described in §2.3. The final subsection §2.4 discusses the related problem of warm starting when the constraint matrix A is modified.

2.1. Practical benefits of an interior point cutting plane algorithm

The dual simplex algorithm can exploit an optimal solution to (P) and (D) to find an optimal solution to (P_0) and (D_0) in just a few pivots when one or just a handful of constraints are added at a time. When large numbers of constraints are added at once, the dual simplex method is not so effective. The optimal solution to (P) and (D) returned by the simplex method is an extreme point of the feasible region, so the generated cutting planes may not cut off much more than this extreme point.

Thus there are two areas where it can be advantageous to use an interior point cutting plane method: when many cuts are added at once, and when the problem (D) has a large optimal face or a large set of near-optimal solutions. In the former situation, an interior point method can reoptimize more quickly than simplex, especially if the LP is reasonably large. In the latter situation, the cuts generated from an interior point can be stronger than those generated from a basic feasible solution.

One especially effective approach is to combine interior point and simplex cutting plane algorithms. In early iterations, many constraints are added at once and it is not necessary to solve the relaxations to high accuracy in order to find strong cuts. In later iterations, once the relaxations become good approximations to the underlying problem in the region of the optimal solution, only a few constraints are added at a time and a higher accuracy LP solution can lead to better cuts. Therefore, the first few stages can be solved using an interior point method and the later stages solved with the simplex method. Bixby et al [16] implemented such a method for solving very large scale linear programs arising from LP relaxations of airline crew scheduling problems. Columns were only included in the constrained version (P) as necessary. They showed that the combined approach outperformed either a pure interior point method or a pure simplex method. Mitchell and Borchers [97] solved linear ordering problems using a cutting plane approach. The linear ordering problem can be formulated as an integer programming problem. For their instances, a combined approach was up to 10 times faster than using either a pure simplex cutting plane approach or a pure interior point cutting plane approach.

Solving (P) and (D) using the simplex method leads to an extreme point solution of each problem. The dual solution y^* is used to generate new columns for (P). In situations where there are multiple optimal dual solutions or where there are many near-optimal dual extreme points, it may be restrictive to look at only one dual extreme point. It may be useful to look at a broader selection of dual points and hence obtain a richer set of new columns. The validity of this observation has been shown in a number of papers, as we discuss in the rest of this subsection. This phenomenon is illustrated in Figure 1.

Just using an extreme point solution leads to Kelley's cutting plane algorithm [72]. A lack of stability in this algorithm has been noted for a long time, with the dual solutions possibly moving dramatically after adding cutting planes. The method can also be very slow: Nemirovski and Yudin [104] constructed an example which requires $O((1/\epsilon)^{m/2})$ calls to the oracle to get within ϵ of optimality. Briant et al. [21] compared simplex cutting plane methods and bundle methods experimentally. In a bundle method, only "important" columns of A are kept in the problem and typically a quadratic proximity term is included in the dual objective in order to prevent the dual solution from changing drastically. This is an attempt to stabilize the sequence of iterates and [21] includes a survey of different stabilization techniques. The use of bundle methods for stabilization is discussed by Hiriart-Urruty and Lemarechal [66, 67]. An interior point cutting plane method uses central dual solutions so it

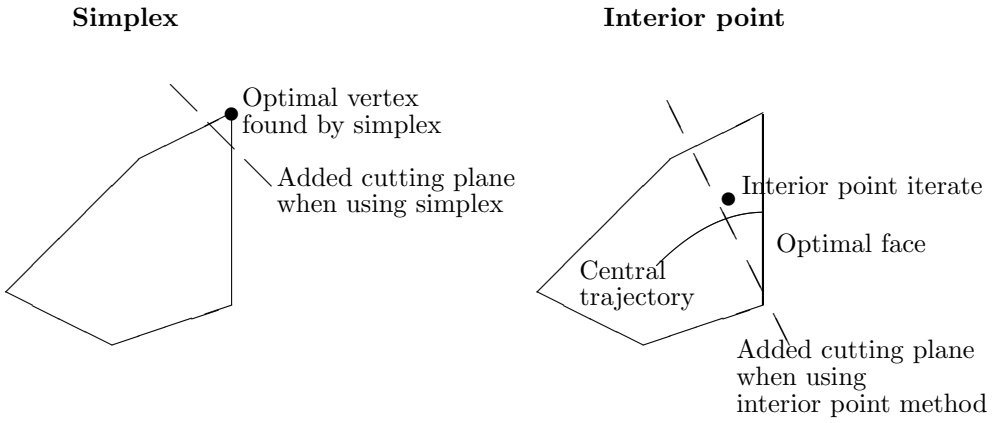

FIGURE 1. Comparing the strength of simplex and interior point cutting planes

can also be regarded as a stabilization technique. Briant et al. showed experimentally that the bundle method appears far more robust for larger instances than Kelley's method.

Goffin et al. used decomposition approaches to attack nonlinear multicommodity network flow problems [46] and stochastic programming problems [9]. These problems were then solved using an analytic center cutting plane method. More recently, Elhedhli and Goffin [36] solved bin packing and capacitated facility location problems using a branch-and-price approach. This method solves a Lagrangian dual problem at each node of the tree, using an analytic center cutting plane method. Interior point warm starting was also used when branching. The interior point method outperformed the contemporary version of CPLEX on the bin packing problems.

Fragnière at al. [40] formulated and solved large scale linear programs arising in stochastic programming and asset management. These problems have a structure that is amenable to decomposition. They used an interior point column generation algorithm to solve the linear programs and they were able to warm start successfully. Gondzio and Kouwenberg [53] extended this method to solve a linear program with over 24 million variables. They had to disable the warm start features in the code for this large instance, because of the memory requirements to store the warm start point. Nonetheless, they could solve the problem in a few hours on a multiprocessor computer. Colombo et al. [28] developed a warm start interior point cutting plane method to solve stochastic programming problems and showed that their method led to faster solution times than a direct method or a cold-started interior point method for a standard class of stochastic programming problems available online.

Gondzio and Vial [54] and Gondzio and Grothey [51] solved several different classes of block-angular linear programs using a primal-dual interior point cutting plane method with two warm starting approaches. Both warm starting approaches led to a significant reduction in the number of iterations and enabled the solution of large problems.

Pages et al. [116, 115] solved quadratic programming problems with an exponential number of linear constraints. These problems arise in power planning. They used a modification of a method in [51] to warm start an interior point column generation algorithm. Their method was able to solve problems that could not be solved directly.

Interior point cutting plane algorithms for integer programming problems were the subject of [98, 96, 92]. These papers discussed matching problems, linear ordering problems, and maxcut problems. The maxcut problems arose from Ising spin glass problems in statistical physics and were defined on a grid. The nodes correspond to spins that can be either up or down, and the edges to pairwise interactions between neighboring spins. The instances

solved in [92] were on 2-dimensional grids with edge interactions equal to ± 1. For these problems, the interior point cutting plane method was far faster than a comparable simplex cutting plane method [34]. The simplex cutting plane solver has been updated and can now be accessed online [83].

Mehrotra and Özevin [89] have proposed an interior point decomposition scheme for two-stage stochastic programming problems, where the underlying cones can be as general as SDP cones. This was extended by Chen and Mehrotra [26] to multistage stochastic convex optimization problems. Their decomposition scheme exploits self-concordance results.

Rousseau et al. [127, 128] have proposed an interior point stabilization approach. In this approach, the problems (P) and (D) are still solved using the simplex method, but multiple dual optimal solutions are determined and then a point in the convex hull of these solutions is chosen. This point is then used in the generation of columns. They solved vehicle routing problems with time windows using a column generation approach, with the columns corresponding to different possible routes. In their applications, not many cutting planes are generated at a time, so the linear programming problems do not change much, and so they did not report results with using an interior point method to solve the linear programming problems. They showed that the interior point stabilization approach worked very well, better than other types of dual stabilization for their instances of vehicle routing problems.

Much of the recent success with branch-and-cut algorithms for integer programming is due to the successful incorporation of classical Gomory cutting planes and lift-and-project cutting planes, along with other classes of cutting planes [17]. Mitchell [91] showed that Gomory cutting planes can even be derived at an interior primal-dual point. Lift-and-project cuts can also be derived when using an interior point method: these cuts are found by constructing and solving a linear programming problem, designed to determine a constraint that separates the convex hull of feasible points from the current iterate. There is no need for the current iterate to be an extreme point, so lift-and-project cuts can readily be found when using an interior point cutting plane method.

2.2. Warm starting

The linear programming pair (P) and (D) is approximately solved, resulting in a triple $(\tilde{x}, \tilde{y}, \tilde{s})$, and cutting planes are added, giving the modified pair (P_0) and (D_0). The triple is approximately centered in (P) and (D), which means that $\tilde{x}_i \tilde{s}_i \approx \mu$ for each component $i = 1, \ldots, n$, where $n\mu$ is the current duality gap. If the problem has not been modified too much then the point $(\tilde{x}, \tilde{y}, \tilde{s})$ should be somewhat close to a solution to the new problem, and it is desirable to try to exploit this warm start. In what follows, we assume the point $(\tilde{x}, \tilde{y}, \tilde{s})$ is feasible in (P) and (D), which implies that $x = \tilde{x}$, $x_0 = 0$ is feasible in (P_0), but $y = \tilde{y}$ is not feasible in (D_0). In order to use a primal-dual interior point method, we need points with x, x_0, s, and s_0 all strictly positive. Primal-dual interior point methods work best if the initial iterate in (P_0) and (D_0) is approximately centered.

If just one column is added to (P_0), so $p = 1$, then a strictly feasible solution to (P_0) can be regained by increasing x_0 and compensating by modifying the components of x. We let a_0 denote the single column of A_0. There are many possible ways to modify x and Mitchell and Todd [98] proposed one using the Dikin ellipsoid. In this approach, the components of x are rescaled and then the least squares solution is found. The resulting direction can be expressed as

$$\Delta x = -D^2 A^T (A D^2 A^T)^{-1} a_0, \tag{4}$$

where D is a diagonal scaling matrix, for example the primal variables.

The Dikin ellipsoid for the problem (P) at a strictly feasible point $\bar{x}$ is the set of points

$$E^P(\bar{x}) := \{x \in I\!R^n : (x - \bar{x})^T D^{-2} (x - \bar{x}) \leq 1\} \tag{5}$$

where again D is a diagonal scaling matrix. If D is taken to be the primal scaling matrix with $D_{ii} = \bar{x}_i$ for $i = 1, \ldots, n$ then E^P is contained within the positive orthant. The direction Δx is the optimal solution d for the problem

$$\begin{array}{ll} \max_{t,d} & t \\ \text{subject to} & Ad + ta_0 = 0 \\ & \bar{x} + d \in E^P(\bar{x}). \end{array} \tag{6}$$

Points in the Dikin ellipsoid have potential function value close to that of $\bar{x}$; potential functions can be used to measure centrality, and for more on potential functions see §2.3. Thus, the subproblem (6) balances the desire to move the incoming variable x_0 away from the boundary with the desire to keep x reasonably well centered. This method is the basis for most restart techniques proposed subsequently in the literature.

One simple method proposed in [98] to find a strictly feasible solution to (P_0) when $p > 1$ is to sum the columns A_0 into a column a_0, then use (6) to determine a direction d, and set the direction for each component of x_0 equal to the optimal value of t. This method has been generalized subsequently to take a_0 to be any positive combination of the columns of A_0; see, for example, [49, 50, 35]. This choice can be further refined by considering potential functions.

The loss of feasibility makes it considerably harder to find a good interior point warm start in the dual problem (D_0) than in the primal problem (P_0). One common method is to determine a feasible solution by exploiting the structure of the problem. For example, earlier dual iterates can be stored and it may be that one of those is feasible. Alternatively, in the case of cutting plane methods for integer programming problems, a feasible restart point can be found by taking a convex combination of integer feasible points [92]. Given a point $\check{y}$ in the interior, a minimum ratio test can be used to find the convex combination of $\bar{y}$ and $\check{y}$ that is on the boundary of the feasible region of (D_0), and then a line search can be used to find an interior feasible point that works well with the primal feasible point.

Gondzio [49] proposed restarting from an approximate analytic center that has been calculated on the way to an approximate solution to (P) and (D). In particular, if the tolerance τ is, say, 10^{-6}, once the solution is within 10^{-3} of optimality an approximate analytic center meeting this accuracy requirement is calculated. Even if this stored approximate analytic center violates the new constraints, he found that feasibility could usually be recovered quickly by using a target following approach [69, 126].

An added constraint that is satisfied at equality by $\bar{y}$ is called a *central cut*. If the constraint is violated by $\bar{y}$ then it is a *deep cut*. If the cut is satisfied strictly by $\bar{y}$ then it is a *shallow cut*. In practice, the added cuts are deep. Most of the theoretical convergence results discussed in §2.3 shift the constraint so that the current point $\bar{y}$ is feasible, either strictly interior or on the boundary. When $p = 1$ and the point is on the boundary, a direction analogous to (4) can be used to find a strictly feasible point. This direction can be motivated by considering a subproblem of maximizing the increase in the slack s_0 while keeping y in the Dikin ellipsoid given by

$$E^D(\bar{y}, \bar{s}) := \{y \in I\!R^m : A^T y + s = c, (s - \bar{s})^T D^2 (s - \bar{s}) \leq 1\}, \tag{7}$$

where D is an appropriate scaling matrix. Hence, the subproblem is to solve

$$\begin{array}{ll} \max_{d,t} & a_0^T d \\ \text{subject to} & A^T d + t = 0 \\ & \bar{y} + d \in E^D(\bar{y}, \bar{s}) \end{array} \tag{8}$$

where $t \in I\!R^n$ and $d \in I\!R^m$. This gives the direction

$$\Delta y = (AD^2 A^T)^{-1} a_0, \tag{9}$$

after scaling. This direction is illustrated in Figure 2.

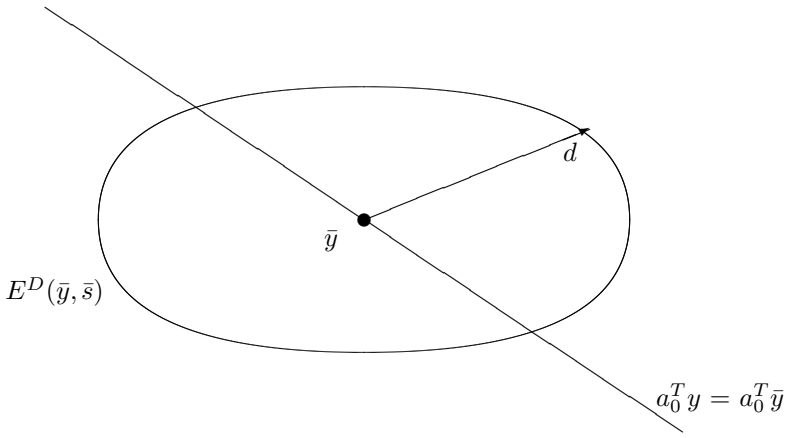

FIGURE 2. The dual restart direction is found by moving as far off the added constraint as possible while staying within the Dikin ellipsoid so as to maintain centrality in the original constraints.

When the set of added constraints determines a cone, it is harder to find an initial strictly feasible dual solution, even if the constraints are central. In this case, one remedy is to minimize an appropriate penalty function of the new dual slack variables, subject to remaining in the dual ellipsoid given by the original set of constraints [47]. The direction finding subproblem has the form

$$\begin{array}{ll} \min_{d,t,t_0} & f_0(t_0) \\ \text{subject to} & A^T d + t = 0 \\ & A_0^T d + t_0 = 0 \\ & \bar{s} + t \in E^D(\bar{y}, \bar{s}) \end{array} \tag{10}$$

where t_0 is the vector of slacks in the added dual constraints and $f_0(t_0)$ is the penalty function. This situation is illustrated in Figure 3. Ramaswamy and Mitchell [120] proposed a selective orthonormalization procedure to make it straightforward to find a new dual feasible solution. In this approach, some of the constraints are weakened if necessary, to ensure that the inner products between the normals of all the added constraints are nonnegative in a rescaled space. In this situation, a strictly interior dual feasible point can be found by taking a positive sum of the scaled normals of the added constraints.

Finally in this section we discuss shifted barrier methods. These methods were proposed by Polyak [119] and investigated further by Freund [41] and Mitchell [90]. Renegar's method [124] can also be regarded as a shifted barrier approach. They have also recently been implemented by Engau et al. [37, 38]. In this approach, the nonnegativity requirements are relaxed, or shifted, and then these requirements are gradually tightened as the algorithm proceeds. In this way, centrality is achieved very easily, at a cost of possible infeasibility. Feasibility is gradually recovered as the barriers are returned to their original positions.

2.3. Theoretical convergence

In the worst case, a cutting plane algorithm that uses the simplex algorithm to solve the LP relaxations can require time that is exponential in m, even if violated cutting planes can be determined in polynomial time. One of the consequences of the theoretical development of the ellipsoid algorithm was a proof that if violated constraints can be found in polynomial

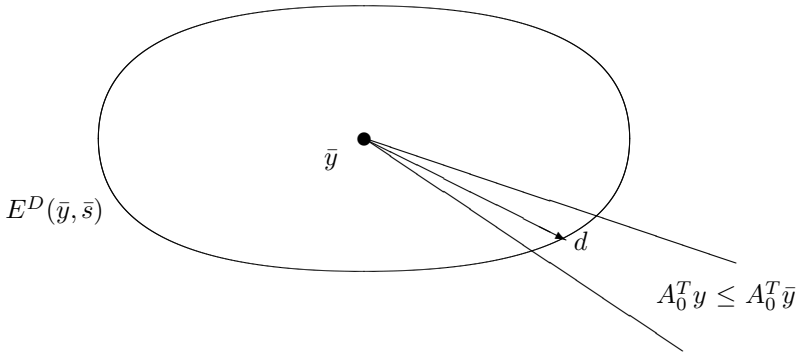

FIGURE 3. The dual restart direction is found by minimizing the penalty function of the new slack variables while staying within the Dikin ellipsoid so as to maintain centrality in the original constraints.

time then the problems $(\hat{P})$ can $(\hat{D})$ can be solved in time polynomial in m [73, 18, 57]. Thus, the separation problem and the optimization problem are polynomially equivalent.

Different interior point cutting plane algorithms have been proposed. Simpler interior point cutting plane algorithms have been shown to converge in fully polynomial time; that is, they get within ε of optimality in time polynomial in $1/\varepsilon$ and m. These methods are surveyed by Goffin and Vial [48]. More complicated interior point cutting plane algorithms have been shown to be polynomial, requiring time polynomial in both m and $\ln(\frac{1}{\varepsilon})$; see the survey paper by Mitchell [94].

Convergence results for interior point cutting plane methods are typically stated in terms of solving the convex feasibility problem to a tolerance ε. Thus, instead of solving problem $(\hat{D})$, we solve the problem

Convex feasibility problem: Given a convex set $C \subseteq \mathbb{R}^m$, either prove C is empty or find a point $y \in \mathbb{R}^m$ such that the Euclidean distance from y to C is no greater than ε.

A polyhedral outer approximation to C is constructed and refined and a trial point $\bar{y}$ is chosen that is an approximate center of this outer approximation. These algorithms can be outlined as follows:

0. Initialize: Choose a polyhedral outer approximation $\{y \in \mathbb{R}^m : A^T y \le c\}$ to C and a tolerance ε.
1. Find an approximate center of the outer approximation. Obtain a point $\tilde{y}$.
2. Check feasibility: If $\tilde{y}$ is within ε of C, STOP with feasibility.
3. Check infeasibility: If sufficiently many iterations have occurred, STOP with the conclusion that C does not contain a ball of radius ε.
4. Modify the polyhedral description: Add one or more constraints to the polyhedral outer approximation, or drop a constraint.
5. Loop: Return to Step 1.

Algorithm 2: Solving the convex feasibility problem

Note that this framework is general enough to include the ellipsoid algorithm: we can regard the ellipsoid as a further outer approximation to the polyhedral outer approximation, and then the "approximate center" in Step 1 is the center of the ellipsoid.

Goffin and Vial [47] showed that such an algorithm converges in a fully polynomial number of outer iterations when the "approximate center" is an approximate analytic center and no constraints are dropped.

Theorem 1. *[47] If C contains a ball of radius ε then an interior point variant of Algorithm 2 stops with feasibility after adding no more than $O(\frac{m^2p^2}{\varepsilon^2})$ cutting planes, where at most p cutting planes are added in each call to Step 4, constraints are never dropped, and an approximate analytic center is found in Step 1. Further, each call to Step 1 requires at most $O(p \ln p)$ Newton steps.*

The proof of this result examines a dual potential function $-\sum_{i=1}^{n} \ln s_i$. An upper bound on this dual potential function can be constructed from the requirement that C contain a ball of radius ε. A lower bound can be obtained from the required change in the dual potential function when a constraint is added through the current approximate analytic center. The upper bound increases as constraints are added, but it increases less quickly than the lower bound; hence, an upper bound can be placed on the number of iterations.

Atkinson and Vaidya [8] developed a version of Algorithm 1 that both adds constraints and drops unimportant constraints. They also used the approximate analytic center. They shifted their cuts to make them weak so that a new approximate analytic center could be recovered in $O(1)$ Newton steps. They were able to prove that their algorithm requires polynomial time.

Theorem 2. *[8] If C contains a ball of radius ε then an interior point variant of Algorithm 2 stops with feasibility after adding no more than $O(m \ln(\frac{1}{\varepsilon})^2)$ iterations, when extra conditions are used to determine constraints to add and drop, and an approximate analytic center is found in Step 1.*

Also of interest are *volumetric center* cutting plane algorithms, first proposed by Vaidya [136] and subsequently refined by Anstreicher [1, 2]. If y is strictly feasible in the current outer approximation to C with slacks s, then the volumetric barrier function is given as

$$V(y) := \frac{1}{2} \ln \det(AS^{-2}A^T). \tag{11}$$

The matrix $AS^{-2}A^T$ is the Hessian of the dual potential function at y. The volumetric center is the point that maximizes this function.

Theorem 3. *[136] If C contains a ball of radius ε then an interior point variant of Algorithm 2 stops in $O(m \ln(\frac{1}{\varepsilon}))$ calls to the oracle and either $O(m \ln(\frac{1}{\varepsilon}))$ or $O(m^{1.5} \ln(\frac{1}{\varepsilon}))$ approximate Newton steps, depending on the choice of parameters, if the center is an approximate volumetric center, and a single cut is added or dropped at each call to Step 4.*

Like Theorem 1, the results in Theorems 2 and 3 follow from developing upper and lower bounds on an appropriate potential function. Both bounds grow as cuts are added, but the lower bound grows more quickly.

2.4. Warm starting when the constraint matrix is modified

Warm starting is not only needed when constraints are added to the problem. For example, in finding the efficient frontier of a portfolio optimization problem, a sequence of closely related quadratic programs is solved. The only difference between these quadratic programs is in the objective function. Thus, the solution process used for one quadratic program should give a warm start for a nearby quadratic program.

Yildirim and Wright [141] proposed methods for regaining starting points that are within neighborhoods of the central trajectory. Their methods require backing up along the central path for the unmodified problem, in order to make the complementarity products $x_i s_i$ large enough. Then after the problem is modified, the point can be modified to ensure that the new iterate is still reasonably well centered. John and Yildirim [70] implemented techniques for warm starting, showing that these techniques could dramatically reduce the required number of Newton steps.

The method proposed in [49] was extended by Gondzio and Vial [54] to solve linear programs where only the objective function was changed. They observed that the warm starting technique required only about one third as many interior point iterations as using a cold start.

Gondzio and Grothey [52] developed an unblocking scheme for warm starting an interior point approach to linear and convex quadratic programs. They looked at problems where the data of the problem are modified, but the problem size is not changed. This approach involves taking the Newton restart direction, determining which components are leading to a short steplength, and trying to modify the direction using sensitivity analysis.

Benson and Shanno [13, 14] have investigated interior point warm starting techniques for nonlinear programming problems. They use regularization and relaxation techniques to aid in warm starting the algorithm.

3. Tightening SDP relaxations of combinatorial optimization problems

Semidefinite programming formulations provide strong relaxations for many combinatorial optimization problems. Typically these relaxations are stronger than LP relaxations. The Lovasz θ number [85] relaxation for maximum stable set, node packing and maximum clique problems was the first illustration of the power of an SDP relaxation. For these problems, there is an upper bound available from looking at an appropriate coloring. The strength of the SDP relaxation is illustrated by the fact that it gives a bound that is at least as good as that obtained from the coloring upper bound. If the graph is perfect [15, 27] then the value of the SDP relaxation gives the optimal value of both the stable set problem and the coloring problem. In general, there is no polynomial time approximation algorithm for the stable set problem, unless P=NP [7]. Perhaps the best known SDP relaxation is Goemans and Williamson's celebrated method for the MaxCut problem [45], which guarantees a solution with value at least 0.878 of the optimal value. Relaxations of the traveling salesman problem are discussed by de Klerk et al. [33]. Laurent and Rendl [82] give an excellent survey of SDP relaxations of integer programs; this paper also includes discussion of the relationship between lifting methods for integer programming constraints and relaxations arising from semidefinite programming. For more details on these relationships, see Goemans and Tunçel [44]. Structure-exploiting techniques that are applicable when solving SDP relaxations of combinatorial optimization problems are surveyed by de Klerk [32].

In principle, an SDP relaxation can be exploited in the same way as an LP relaxation to solve the underlying combinatorial optimization problem. In particular, these relaxations can be embedded within branch-and-cut procedures. The strength of the SDP relaxations should lead to a considerable reduction in the number of relaxations that must be solved when compared to the use of LP relaxations. The principal practical difficulty with this approach is that the SDP relaxations are far harder to solve than LP relaxations. Thus, there is a tradeoff between the number of relaxations and the difficulty of the relaxations.

A semidefinite programming problem and its dual can be written as follows:

$$\begin{array}{lll} \min_X & C \bullet X & \\ \text{subject to} & A_i \bullet X = b_i \ \forall i & (SDP) \\ & X \succeq 0 & \end{array} \qquad \begin{array}{lll} \max_{y,S} & b^T y & \\ \text{subject to} & \sum_{i=1}^m y_i A_i + S = C & (SDD) \\ & S \succeq 0. & \end{array}$$

Here, there are m matrices A_i that are each $n \times n$, X and C are also $n \times n$ matrices, y and b are m vectors, $\bullet$ denotes the Frobenius inner product between two matrices, and $X \succeq 0$ indicates that X must be symmetric and positive semidefinite. Many of these relaxations give exact representations of the underlying combinatorial optimization problem if additional conditions are imposed on the rank of X. For example, the SDP relaxations of node packing [85] and MaxCut [45] each relax the requirement that the rank of X should be one. In such a situation, if the optimal solution to the relaxation has the correct rank then this solution solves the underlying problem.

Helmberg and Rendl [64] experimented with using cutting planes and branch-and-bound in an SDP approach. They solved quadratic 0-1 problems, which are equivalent to MaxCut problems. The cutting planes have the form

$$\pm X_{ij} \pm X_{ik} \pm X_{jk} \geq -1 \tag{12}$$

for any three distinct indices i, j, and k, where an odd number of terms have positive sign. For smaller instances ($n \leq 50$), they found that it was unnecessary to branch. Other cutting planes can be derived based on the fact that a cycle and a cut must intersect in an even number of edges.

The branching routine in [64] was designed so as to reduce the dimension of the matrix X at each branching step. Mitchell [93] showed how the solution process can be warm-started after branching, with a new interior point recovered efficiently. Helmberg [59] developed a technique for exploiting an analogue of reduced costs to fix variables. Helmberg [60, 62] strengthened the computational ability of SDP branch-and-cut algorithms by exploiting the spectral bundle method [65] to solve the SDP relaxations, which allowed the relaxations to be solved approximately and the process to be warm-started effectively. Krishnan and Mitchell [79] developed an SDP branch-and-cut-and-price algorithm to solve MaxCut problems, where the SDP relaxations were solved approximately using an LP cutting plane approach.

Cutting plane or branch-and-cut algorithms with SDP relaxations have been implemented for several classes of problems in recent years. Problems that have been solved include maxcut and equipartition problems [39, 121, 122], stable set problems [58], minimum k-partition problems, possibly with restrictions on the sizes of the clusters [84, 43], and quadratic linear ordering problems [22].

Anstreicher et al. [4] solved quadratic assignment problems (QAP) using a branch-and-bound method on a computational grid. There exist LP and SDP relaxations for the QAP (see, for example, Rendl and Sotirov [123]), but the LP relaxation is too weak for hard instances and the SDP relaxation is too hard to solve repeatedly. Therefore, they used a convex quadratic programming bound based on a projected eigenvalue bound [3]. This gives a better bound than the LP relaxation and at less cost than determining the SDP bound. They were able to solve instances of the QAP that had been open problems for 30 years. Peng et al. [99, 118] developed novel SDP relaxations of the QAP that can be solved effectively and that are stronger than the quadratic programming bound of [4]. Their approach uses a matrix splitting approach, driven by the eigenvalue spectrum of one of the matrices in the problem definition. They also show how their relaxation can be enhanced through the use of cutting planes, and their relaxation may well be effective in a branch-and-cut algorithm.

Theoretical results for interior point cutting plane methods for semidefinite programming problems were derived by Sun et al. [133, 134]. Their algorithm solves a convex feasibility problem, where the initial relaxation is defined by a semidefiniteness constraint, and linear constraints are added as cutting planes. They showed that their algorithm converges in a number of iterations that is polynomial in m, ϵ, and the maximum number of constraints added at each iteration, under various assumptions. Generalizations of this result are considered in §4.

4. Interior point cutting surface methods

A primal-dual interior point algorithm can be used to solve (SDP) and (SDD), and several versions of such an algorithm have been implemented [20, 132, 135, 139]. These methods enjoy excellent theoretical convergence properties and are surveyed in [100] and elsewhere. However, they are computationally expensive; for example, problems where the dimension of X is a few thousand require several hours even on a machine with four processors [20]. The development of algorithms for solving large-scale semidefinite programming problems is an active area of research. Various approaches are discussed in, for example, [25, 24, 23, 75, 142], and some of these methods are surveyed in [100, 80].

In this section, we discuss an alternative class of approaches where relaxed versions of (SDD) are constructed and the relaxations tightened through the addition of cutting surfaces. Given an $n \times p$ matrix P, a primal-dual pair of SDPs that are a constrained version of (SDP) and a relaxed version of (SDD) can be written as

$$\begin{array}{lll} \min_V & C \bullet PVP^T & \\ \text{subject to} & A_i \bullet PVP^T = b_i \ \forall i & (SDPP) \\ & V \succeq 0 & \end{array} \qquad \begin{array}{lll} \max_{y,S} & b^T y & \\ \text{subject to} & \sum_{i=1}^m y_i A_i + S = C & (SDDP) \\ & P^T S P \succeq 0, & \end{array}$$

where V is a $p \times p$ variable matrix [78]. If $\text{rank}(P) = n$ then this pair of problems is equivalent to (SDP) and (SDD). This formulation can be extended further by relaxing the dual semidefiniteness constraint and imposing corresponding restrictions on the primal matrix V. For example, requiring that V be diagonal results in a dual problem where the semidefiniteness constraints just requires the diagonal entries of $P^T S P$ to be nonnegative [76, 77]. Similarly, it is possible to require that V have a block-diagonal structure, which leads to a dual requirement that only the corresponding blocks of $P^T S P$ be positive semidefinite [110].

In order to simplify the presentation, we make two assumptions about the problem (SDP).

Assumption 1. *The matrices A_i, $i = 1, \ldots, m$, are linearly independent in the space of symmetric $n \times n$ matrices.*

Assumption 2. *Both (SDP) and (SDD) have strictly feasible solutions, with X and S positive definite.*

One consequence of these assumptions is that the optimal values of (SDP) and (SDD) agree.

A solution (y, S) to $(SDDP)$ is optimal for (SDD) if $S \succeq 0$. If this is not the case then S has at least one eigenvector u with negative eigenvalue. Let $\lambda_{\min}(S)$ denote the minimum eigenvalue of S. Requiring that S be positive semidefinite is equivalent to imposing the convex nonsmooth constraint that $\lambda_{\min}(S) \geq 0$. For a given matrix $\bar{S}$, let $d(\bar{S})$ denote the degree of the minimum eigenvalue of $\bar{S}$, and let $u_1(\bar{S}), \ldots, u_{d(\bar{S})}(\bar{S})$ be an eigenbasis for the corresponding eigenspace. The subdifferential of the convex function $-\lambda_{\min}(S)$ at $\bar{S}$ is then the following convex set of symmetric positive semidefinite matrices:

$$\partial(-\lambda_{\min}(\bar{S})) = \{\sum_{i=1}^{d(\bar{S})} \tau_i u_i u_i^T : \tau_i \geq 0, i = 1, \ldots, d(\bar{S}), \sum_{i=1}^{d(\bar{S})} \tau_i = 1\}.$$

Thus, adding the constraint $u_i^T S u_i \geq 0$ to $(SDDP)$ corresponds to adding a subgradient inequality. If we let U denote the $n \times d(\bar{S})$ matrix whose columns are the vectors $u_1, \ldots, u_{d(\bar{S})}$, then the whole subdifferential can be imposed through the addition of the constraint

$$U^T S U \succeq 0. \tag{13}$$

This constraint is valid for any matrix U, of course. In practice, it is not necessary that the columns of U all correspond to the same eigenvalue, so a tolerance can be used to select

an appropriate set of columns, with each of the columns corresponding to some negative eigenvalue of $\bar{S}$. These columns can be added to P, with an appropriate modification of the structure of the dual semidefiniteness constraint in $(SDDP)$. If U has just one column then (13) is a linear constraint.

Different algorithms can be defined by using different methods to update P and by choosing varying relaxations of the dual semidefiniteness constraint. This leads to a class of algorithms of the following form:

> *0. Initialize:* Choose the initial matrix P. Choose a tolerance ϵ for the overall algorithm. Choose a tolerance τ for the desired accuracy of the SDP relaxations. Choose a structure for the relaxation of the dual semidefiniteness constraint.
> *1. Approximately solve the SDP relaxation:* Solve $(SDPP)$ and $(SDDP)$ to a relative accuracy of τ in the duality gap. Obtain a primal-dual pair $(\tilde{X}, \tilde{y}, \tilde{S})$.
> *2. Modify P:* Check whether $\tilde{S} \succeq 0$.
> *(a)* If $\tilde{S} \succeq 0$ and $\tau \leq \epsilon$ then STOP with a solution to the underlying problem.
> *(b)* If $\tilde{S} \succeq 0$ and $\tau > \epsilon$ then reduce τ and return to Step 1.
> *(c)* If $\tilde{S}$ has one or more eigenvectors with negative eigenvalue, add a subset of the eigenvectors to P, drop or aggregate some of the columns of P if desired, modify τ if desired, choose a structure for the updated dual semidefiniteness constraint, and return to Step 1.
>
> Algorithm 3: Cutting surface algorithms for (SDP) and (SDD).

Krishnan and Mitchell [77] proposed a linear programming version of Algorithm 3. They only required the diagonal elements of $P^T SP$ to be nonnegative, which is equivalent to the linear constraints

$$\sum_{i=1}^{m} (p_j^T A_i p_j)\, y_i \leq p_j^T C p_j \tag{14}$$

for each column p_j of P. The linear programming dual of the corresponding $(SDDP)$ gives problem $(SDPP)$, with V required to be diagonal. This means that the primal matrix $X = PVP^T = \sum_j v_j p_j p_j^T$; that is, X is a nonnegative sum of outer products of the columns of P. Several columns can be added to P at once, if S has multiple negative eigenvalues, with each of these columns leading to an additional linear constraint. Because this is a linear programming problem, the techniques discussed in §2.2 can be used to speed up solution. In practice, this algorithm needs to add a large number of columns to P. It can get close to optimality in a reasonable amount of time, but it is difficult to get close to optimality in time competitive with primal-dual interior point methods for the smaller problems in SDPLIB [19]. The algorithm is more competitive for larger problems when m is not too large compared to n. For another version of a similar LP approach to semidefinite programs, see Sherali and Fraticelli [129].

4.1. Theoretical convergence results

It is common in the framework of Algorithm 3 that the matrix S has multiple negative eigenvalues. Oskoorouchi and Goffin [110] proposed adding a low-dimensional semidefinite constraint to $(SDDP)$ corresponding to the set of corresponding eigenspaces. In particular, if the columns of the matrix $\bar{P}$ give a linearly independent set in the union of the eigenspaces of S with negative eigenvalue, a constraint $\bar{P}^T S \bar{P} \succeq 0$ can be added to $(SDDP)$. This results in extra flexibility in X, with the addition of a term $\bar{P}\bar{V}\bar{P}^T$, with $\bar{V} \succeq 0$. This added constraint is a cutting surface and is stronger than adding linear constraints of the form (14).

The cutting surface has the form (13) in terms of the dual slack variables, and the following equivalent form as a constraint on the dual variables y:

$$\sum_{i=1}^{m}(P^T A_i P)y_i \preceq P^T C P. \tag{15}$$

Thus, the problems $(SDPP)$ and $(SDDP)$ are semidefinite programs, with block-diagonal restrictions on V and on the positive semidefiniteness of $P^T S P$. The original semidefiniteness constraint $S \succeq 0$ has been approximated by a number of lower dimensional semidefiniteness constraints. The algorithm can be initialized either using linear bound constraints on y or by imposing a second order cone constraint on y restricting its Euclidean norm. The resulting algorithm was shown to be fully polynomial. Computational results are described in [112].

A cutting surface method with q added cuts of the form (14) and r added cuts of the form (15) leads to versions of $(SDPP)$ and $(SDDP)$ of the following form:

$$\begin{array}{lrlll}
\min_{X,V^k,v_j} & C \bullet X & & \max_{y,S} & b^T y \\
\text{subject to} & A_i \bullet X = b_i & \forall i & \text{subject to} & \sum_{i=1}^m y_i A_i + S = C \\
& \sum_{j=1}^q v_j p_j p_j^T + \sum_{k=1}^r P_k V^k P_k^T = X & & & P_k^T S P_k \succeq 0 \ \ \forall k \\
& V^k \succeq 0 & \forall k & & p_j^T S p_j \geq 0 \ \ \forall j \\
& v_j \geq 0 & \forall j & &
\end{array}$$

One aim is to make this problem easier to solve than the original (SDP) by keeping the sizes of the blocks V^k small. The linear constraints can be regarded as SDP blocks of size 1.

Computationally, second order cone (SOC) constraints are a lot easier to use than semidefiniteness constraints. Therefore, Oskoorouchi and Goffin [111] proposed restricting the size of the blocks to be no larger than 2. A semidefiniteness constraint of size 2 is equivalent to an SOC constraint. Oskoorouchi and Mitchell [113] extended the algorithm to allow the simultaneous addition of multiple SOC constraints. If the number of negative eigenvalues of S is large, then the semidefiniteness constraint $\bar{P}^T S \bar{P} \succeq 0$ is approximated by several SOC constraints. Thus, $(SDPP)$ and $(SDDP)$ are a primal-dual pair of second order cone programs. Again, the resulting algorithm was shown to be fully polynomial. Computational experiments with this algorithm can be found in [113]. The algorithm works very well when m is not large compared to n.

The theoretical convergence results in [110, 111] are for the convex feasibility problem when a separation oracle returns a cutting surface corresponding to a positive semidefiniteness constraint. Basescu and Mitchell [11] generalized the algorithms in [76, 77, 110, 111] by allowing the cutting surfaces to correspond to general self-scaled cones. They were able to show that the resulting algorithm was fully polynomial, with the general complexity result reducing to the ones obtained in [110, 111] when the cutting surfaces are restricted to be semidefinite cones or second order cones.

The theoretical algorithm in [110] is warm-started by using a method analogous to (10). in particular, the cutting surface is shifted so that it is central, and then the potential function of the added constraint is minimized, subject to remaining in an appropriate Dikin ellipsoid. The primal problem can also be restarted in a similar manner. It was shown in [110] that a new approximate analytic center could then be recovered in $O(p)$ Newton steps, where p is the number of columns in $\bar{P}$. This method was extended in [113, 11], with analogous complexity results. Practical warm starting procedures are described in [112, 113].

All of these convergence results include a condition number in the complexity. It is needed in order to get a tractable upper bound on the dual potential function. The upper bound used in the convergence analysis uses the assumption that the feasible region contains a ball of radius ϵ if it is nonempty. The condition number relates the dual potential function based on the slack variables in an added dual constraint to the norm of the dual variables y. This vector of slack variables is required to be in a cone K_0, expressed as $s \succeq_{K_0} 0$. A cutting

surface can be denoted by $\mathcal{A}_0^* y \preceq_{K_0} c_0$, where $\mathcal{A}_0^* y$ is an appropriate linear function of y. If K_0 is a set of positive semidefinite matrices then the corresponding dual potential function for a matrix S of dual slack variables is $\phi(S) = -\ln\det(S)$. Each dual potential function $f^*_{K_0}(s)$ has a parameter ϑ_f which relates $f^*_{K_0}(s)$ to $f^*_{K_0}(\alpha s)$ for positive scalars α, with $f^*_{K_0}(\alpha s) = f^*_{K_0}(s) - \vartheta_f \ln t$. In particular, $\vartheta_f = p$ for either a cone K of positive semidefinite matrices of size $p \times p$, or $K = I\!R^p_+$. If we assume the feasible region contains a ball of radius ϵ centered at an iterate $\bar{y}$, it follows that $c_0 - \mathcal{A}_0^*(\bar{y} + \epsilon d) \succeq_{K_0} 0$ for any vector d with $||d|| = 1$. Hence,

$$f^*_{K_0}(c_0 - \mathcal{A}_0^*\bar{y}) \leq f^*_{K_0}(\epsilon \mathcal{A}_0^* d) = f^*_{K_0}(\mathcal{A}_0^* d) - \vartheta_{f_{K_0}} \ln \epsilon,$$

since the potential function is logarithmically homogeneous, so this gives an upper bound. A condition number μ_{K_0} can be defined as

$$\ln \mu_{K_0} := \inf\{f^*_K(\mathcal{A}_0^* d) : \mathcal{A}_0^* d \succ_{K_0} 0,\ ||d|| = 1\}.$$

The smaller the value of μ_{K_0}, the better the upper bound on $f^*_{K_0}(c - \mathcal{A}^*\bar{y})$. For an example of an SDP cutting surface with a poor condition number, consider the following:

Example 1. Let

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix},\ A_2 = \begin{bmatrix} 0 & 1 \\ 1 & \delta \end{bmatrix},\ A_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix},\ A_4 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad \text{and } \mathcal{A}^*(d) = \sum_{i=1}^{4} d_i A_i.$$

If $||d|| = 1$ then $\det(\mathcal{A}^*(d))$ is at most $O(\delta)$. Thus, for small δ and for any choice of direction, it is necessary to take a step of length $O(1/\delta)$ in order to get a negative potential function value.

Mitchell and Basescu [95] removed this condition number by allowing the constraint to be weakened using a selective orthonormalization procedure, an extension of the LP procedure in [120].

4.2. A Lagrangian dual function and eigenvalue minimization

The problem (SDD) can be derived through consideration of Lagrangian duality. Rather than deriving this in the general case, we consider imposing an additional assumption. The Lagrangian dual problem can then be stated naturally as an eigenvalue optimization problem.

If the primal problem (SDP) has a bounded feasible region then it can be rescaled and a constraint $I \bullet X = 1$ imposed, possibly after adding a slack variable [60]. This constraint is already included in the Lovasz θ relaxation of the stable set problem, and most SDP relaxations of combinatorial optimization problem in the literature already imply $I \bullet X = a$ for some appropriate constant a. We make the following assumption:

Assumption 3. *Every feasible solution to (SDP) satisfies $I \bullet X = 1$.*

Under Assumption 3, a Lagrangian relaxation of (SDP) can be formulated as

$$\begin{array}{rll} \Theta(y) := \min_X & b^T y + (C - \sum_{i=1}^m y_i A_i) \bullet X & \\ \text{subject to} & I \bullet X = 1 & (SDPLR(y)) \\ & X \succeq 0. & \end{array}$$

Let $\lambda_{\min}(C - \sum_{i=1}^m y_i A_i)$ denote the minimum eigenvalue of $C - \sum_{i=1}^m y_i A_i$, which may be negative, and let u^1 be a corresponding eigenvector. The optimal solution to $(SDPLR(y))$ is to take $X = uu^T$, with value $\Theta(y) = b^T y + \lambda_{\min}(C - \sum_{i=1}^m y_i A_i)$. Thus, the Lagrangian dual problem can be stated as the unconstrained optimization problem

$$\max_y\ b^T y + \lambda_{\min}(C - \sum_{i=1}^m y_i A_i). \qquad (SDPLD)$$

Under Assumptions 1 and 2, the optimal values of (SDP), (SDD), and ($SDPLD$) all agree. Since Lagrangian dual functions are concave, the problem ($SDPLD$) can be solved using a cutting surface approach. This is the approach followed by Sivaramakrishnan et al. [131].

4.3. Developing a practical algorithm

Computational results are contained in the previously mentioned [112, 113, 131], along with discussion of practical techniques. In this subsection, we look at some of the practical issues.

A primal feasible point is immediately available by setting the added block of variables V_k equal to zero. A strictly feasible warm-start point can be found using Dikin ellipsoids, as discussed in §4.1. Oskoorouchi et al. [112, 113] used a primal interior point method. Such a method converges to an approximate analytic center, at which point a dual feasible solution to ($SDDP$) can be determined.

Upper bounds are available from any X that is feasible in (SDP). Hence, any feasible solution to ($SDPP$) will give an upper bound, since ($SDPP$) is a constrained version of (SDP). One advantage of Assumption 3 and the resulting formulation ($SDPLD$) is that it is straightforward to find a lower bound on the optimal value. In particular, because of the presence of the implicit primal constraint $I \bullet X = 1$, the identity matrix is a combination of the constraint matrices A_i. We write this explicitly as

$$\sum_{i=1}^{m} \beta_i A_i = I$$

for appropriate multipliers β. Given a dual vector $\bar{y}$, the vector

$$\tilde{y}_i = \bar{y}_i + \beta_i \lambda_{\min}(C - \sum_{i=1}^{m} \bar{y}_i A_i) \tag{16}$$

is feasible in (SDD). In particular, each eigenvalue of $C - \sum_{i=1}^{m} \tilde{y}_i A_i$ is $-\lambda_{\min}(C - \sum_{i=1}^{m} \bar{y}_i A_i)$ larger than the corresponding eigenvalue of $C - \sum_{i=1}^{m} \bar{y}_i A_i$, so $C - \sum_{i=1}^{m} \tilde{y}_i A_i$ is positive semidefinite. Hence, we obtain a lower bound of

$$LB = b^T \bar{y} + b^T \beta \; \lambda_{\min}(C - \sum_{i=1}^{m} \bar{y}_i A_i).$$

Note that $-\beta$ is a ray in (SDD), so we must have $b^T \beta \geq 0$ if (SDP) is feasible.

Because it is easy to find a feasible solution to ($SDDP$) under the condition of Assumption 3, it is also easy to warm-start the dual problem after adding cutting surfaces at a point $\bar{y}$. The modification given in (16) gives a feasible solution, and a strictly feasible dual solution can be found by making a slightly larger change to $\bar{y}$. Because of the availability of a dual warm start point, Sivaramakrishnan et al. [131] used the primal-dual interior point implementation SDPT3 [135] to solve ($SDPP$) and ($SDDP$). This implementation also allows control of the optimality tolerance; in addition the Dikin ellipsoid direction finding subproblem can also be solved using SDPT3.

Sivaramakrishnan [130] developed a parallel decomposition approach for block-angular semidefinite programming problems. The primal matrix in these problems is comprised of r smaller diagonal blocks, each with its own set of linear constraints. Further, there are linear constraints linking the blocks together. Let n_i denote the dimension of the ith diagonal block, let m denote the number of linking constraints, and let m_i denote the number of linear constraints on the ith block. The primal and dual problems have the form

$$\begin{array}{lll} \min_{X_i} & \sum_{i=1}^{r} C_i \bullet X_i & \max_{y,w_i} \quad b^T y + \sum_{i=1}^{r} d_i^T w_i \\ \text{subject to} & \sum_{i=1}^{r} \mathcal{A}_i(X_i) = b & \text{subject to } \mathcal{A}_i^T y + \mathcal{B}_i^T w_i \preceq C_i,\ i = 1, \ldots, r \\ & \mathcal{B}_i(X_i) = d_i,\ i = 1, \ldots, r & \\ & X_i \succeq 0, \ \ i = 1, \ldots, r & \end{array}$$

where b and y are m-vectors, d_i and w_i are m_i-vectors and C_i is an $n_i \times n_i$ matrix for $i = 1, \ldots, r$, $\mathcal{A}_i(X_i)$ denotes an m-vector whose jth component is $A_{ij} \bullet X_i$ with A_{ij} an $n_i \times n_i$ matrix, $\mathcal{A}_i^T y = \sum_j y_j A_{ij}$, and the linear operators $\mathcal{B}_i(X_i)$ and $\mathcal{B}_i^T w_i$ are defined similarly. A Lagrangian dual function $\Theta(y)$ can be constructed as

$$\Theta(y) \;=\; b^T y + \sum_{i=1}^{r} \min\{(C_i - \mathcal{A}_i^T y) \bullet X : \mathcal{A}_i(X_i) = b,\, X_i \succeq 0\} \tag{17}$$

$$=: b^T y + \sum_{i=1}^{r} \Theta_i(y). \tag{18}$$

Maximizing this function gives the optimal value of the original problem, under Assumptions 1 and 2. Sivaramakrishan constructs piecewise linear approximations to each $\Theta_i(y)$ by solving the r disaggregated semidefinite programming subproblems

$$\Theta_i(y) = \min\{(C_i - \mathcal{A}_i^T y) \bullet X : \mathcal{A}_i(X_i) = b,\, X_i \succeq 0\}$$

for different choices of y arising as solutions of a master problem. These subproblems are solved in parallel. The master problem is solved using a stabilized column generation procedure. The solutions X_i to the subproblems lead to the subgradients $-\mathcal{A}_i(X_i)$ of $\Theta_i(y)$.

The decomposition algorithm in [130] is applied to general semidefinite programs by exploiting chordal extensions [32]. In this approach, a graph is constructed for problem (SDP) with nodes corresponding to the rows of the matrix X. There is an edge between nodes j and k if there is a nonzero entry in position (j, k) in one of the data matrices C or $\{A_i : i = 1, \ldots, m\}$. If the graph is chordal then the matrix can be decomposed into blocks corresponding to the maximal cliques in this graph. If each of the blocks is positive semidefinite then appropriate entries for the remainder of X can be determined to ensure that the whole matrix X is positive semidefinite [56]. This strong result can be used as the basis for a decomposition algorithm: smaller semidefinite programs are constructed for each of the maximal cliques, and linking constraints are imposed to ensure that the solutions to the subproblems agree with one another. If the graph is not chordal then a chordal extension can be constructed: additional edges are added until the graph is chordal [42]. The decomposition approach can then be used.

Impressive computational results are contained in [130] for SDP relaxations of large combinatorial optimization problems. In some of these problems, n is as large as 14000. The memory requirements of this algorithm are considerably smaller than for primal-dual interior point methods, so the algorithm is able to find solutions to 2 or 3 digits of accuracy for problems that cannot be solved by other approaches.

One useful technique for speeding up cutting plane and cutting surface methods for semidefinite programs is to aggregate unimportant constraints [74, 60, 65, 76, 77, 131, 130]. For any symmetric positive semidefinite matrix W, the constraint

$$W \bullet (C - \mathcal{A}^T y) \geq 0$$

is valid. Initially, W can be taken to be the zero matrix. When it is determined that a constraint $P_k^T(C - \mathcal{A}^T y)P_k \succeq 0$ is no longer important, it could be dropped in order to shrink the size of the problem. Alternatively, it can be merged into W, so W is updated as

$$W \leftarrow W + \alpha P_k P_k^T \tag{19}$$

for some positive scalar α. One advantage of this approach for handling unimportant constraints is that it makes it easier to restart in the primal problem: if instead the dual constraint is dropped then primal variables are dropped, making the current solution infeasible. With an aggregation of the dual constraints, the primal variables can also be aggregated, leading to retention of primal feasibility.

4.4. The spectral bundle method

The spectral bundle method was introduced in [65] and developed further in [63, 60, 61]. It was developed for semidefinite programs satisfying Assumptions 1, 2, and 3. It determines a solution to $(SDPLD)$ by developing a semidefinite programming approximation to the Lagrangian dual function $\Theta(y)$ given in $(SDPLR(y))$. Computational results with the SBmethod implementation of this algorithm are very impressive [61], and a variant of it has been used in deriving the computational results in, for example, [39, 121, 122, 58, 84, 22]. Further computational results are contained in [102] and a parallel implementation is presented in [101]. For a second-order spectral bundle method, see [114].

The spectral bundle method constructs problems of the form $(SDPP)$ and $(SDDP)$, with V and P^TSP general symmetric positive semidefinite matrices. The number of columns in P is controlled: once this number reaches a threshold, the addition of extra columns results in old columns being merged into an aggregated matrix, as in (19). A proximal bundle term is added to the dual relaxation, so it has the form

$$\begin{array}{lll} \max & b^Ty + \frac{u}{2}||y-\bar{y}||^2 & \\ \text{subject to} & \mathcal{A}^Ty + S = C & (QSDDP) \\ & P^TSP \succeq 0 & \\ & W \bullet S \geq 0 & \end{array}$$

where $\bar{y}$ is the current iterate and u is a positive parameter that is adjusted dynamically based on the progress of the algorithm. This semidefinite program with a quadratic objective can be solved efficiently using a primal-dual interior point method. In a bundle method, null steps are taken if progress is insufficient. In such a step, the matrices P and W are updated but $\bar{y}$ is not changed. Progress can be insufficient because $(QSDDP)$ is an approximation to (SDD) and it may not be accurate enough. The value of a possible new dual point can be measured using the expression for $\Theta(y)$ contained in the objective function of $(SDPLD)$, namely $b^Ty + \lambda_{\min}(C - \sum_{i=1}^m y_iA_i)$. If the solution to $(QSDDP)$ results in a value of $\Theta(y)$ that is not close to the value predicted by the cutting surface approximation then a null step is taken; otherwise, a serious step is taken and $\bar{y}$ is updated along with P and W.

Helmberg [60] proves that the spectral bundle method converges under various assumptions. The algorithm has been extended to situations where the dual variables have bounds [63]. It was shown in Pataki [117] that there is an optimal solution to (SDP) with rank no greater than $O(m^{0.5})$, which gives a theoretical justification to bounding the number of columns in P. The flexibility provided by the aggregate matrix W enables the algorithm to approach an optimal solution even when the number of columns in P is restricted to be smaller than the bound in [117].

5. Recent subgradient methods

The interior point cutting plane and surface methods discussed in sections 2 and 4 are not the only methods for solving convex conic programming problems. For problems where the computational requirements of these algorithms are daunting, subgradient approaches can be used. There has been renewed interest in these approaches recently, spurred in part by Nesterov's smoothing technique [105]. The article [108] gives a survey of these results, and discusses their relationship to other subgradient techniques. Subgradient methods can be used to solve nonsmooth convex optimization problems of the form

$$\min_x \{f(x) : x \in Q\}$$

where Q is assumed to be a bounded closed convex set and $f(x)$ is a continuous convex function on Q. Given a feasible iterate $\bar{x}$, a subgradient ξ of $f(.)$ at $\bar{x}$ is determined and the iterate is updated to $\bar{x} - \alpha\xi$ for some appropriate steplength τ. Such a method is known

to converge under certain conditions on τ (see [12, 67], for example). The work required at each iteration of a subgradient scheme can be broken into two parts: (i) the determination of a function value and subgradient, which can be considered to be provided by an oracle, and (ii) the update of the iterate. The advantage of a subgradient approach lies in the low cost of (ii). The disadvantage is that the number of iterations may be large.

The complexity of the subgradient algorithms discussed in this section is measured in terms of ε, which is the desired absolute accuracy in objective function value. The complexity estimates are bounds on the number of iterations of the subgradient scheme. The work per iteration depends on the dimension of the variables. Nemirovski and Yudin [104] showed that every subgradient scheme that makes no assumptions about the structure of the function f needs at least $O(1/\varepsilon^2)$ iterations in the worst case.

When the function $f(x)$ is smooth and the set Q has an appropriate structure, subgradient methods have a better worst case complexity. This provides the motivation for the smoothing technique: develop a smooth approximation to a nonsmooth function and then apply an optimal subgradient approach to the smooth function. It is necessary to exploit the structure of the nonsmooth function in order to construct such an approximation, so this smoothing technique cannot be used when the nonsmooth function is just given by an oracle. As long as sufficient information is available about the function, and as long as it satisfies certain assumptions, a smooth approximation can be constructed.

Nesterov [105] shows that for certain classes of problems a smooth approximation $f_\mu(x)$ can be constructed where the smooth function is Lipschitz continuous with constant L_μ. Solving the smooth function with a subgradient scheme can be accomplished in $O(\sqrt{L_\mu/\varepsilon})$ steps. This approximation is such that $L_\mu = O(1/\varepsilon)$, so the overall complexity becomes $O(1/\varepsilon)$ subgradient steps. Each subgradient step requires solving two quadratic programs, where the quadratic term is a norm in one problem and a simple strongly convex function of Q in the other, and it is assumed that the structure of Q makes these subproblems simple to solve. For example, if Q is a box of the form $\{x \in I\!R^n : l \le x \le u\}$ then the quadratic programs can be constructed to be separable. The subproblems are also easy to solve if $Q = \{x \in I\!R^n : ||x - a||_2^2 \le r\}$ for a vector $a \in I\!R^n$ and a positive scalar r. In one quadratic program, the linear term is a subgradient approximation to $f_\mu(x)$; in the other, it is a cutting plane model of $f_\mu(x)$ constructed from all of the earlier subgradients, with more recent subgradients weighted more heavily. The new iterate is chosen to be a convex combination of the solutions to these two subproblems.

Classes of problems that have been attacked with smoothing techniques are certain Lagrangian dual problems [105, 106], variational inequalities [103], semidefinite programming [107, 87, 31, 29, 30], and conic programming [81, 86]. For the Lagrangian dual problems, a strictly convex function is added to the primal problem which has the effect of smoothing the dual. The smoothed function $f_\mu(X)$ for a semidefinite program is the function

$$f_\mu(X) = \mu \ln \sum_{i=1}^{n} e^{\lambda_i(X)/\mu}$$

where $\{\lambda_i(X) : i = 1, \ldots, n\}$ denote the eigenvalues of X. Calculating the gradient of $f_\mu(X)$ requires calculating a matrix exponential function. D'Aspremont [29] showed that this can be approximated using just a few of the eigenvalues of the matrix, which makes the algorithm more practical. There are relationships between the smoothing method for SDP and the methods considered in §4.

6. Summary

Interior point methods provide an excellent choice for stabilizing a column generation approach. The strength of interior point methods for linear programming means that these column generation approaches scale well, with theoretical polynomial or fully polynomial

convergence depending on the variant. In practice, combining interior point and simplex column generation methods has proven especially effective, with the interior point methods used early on and the simplex method used later. In this way, the speed of an interior point method for large scale LPs and its stabilization properties while getting to the location of the optimal solution can be combined with the simplex method's reoptimization speed when just a few constraints are added. The development of methods that automatically combine the two linear programming approaches would be useful, and theoretical development of a more intuitive polynomial interior point cutting plane algorithm is desirable.

Research in cutting surface and subgradient methods for quickly finding good solutions to semidefinite programming problems is an active area. The links between the various approaches in §4 and §5 can be developed further. Complexity results for cutting surface methods have been developed, but can probably be improved further. These methods have been developed for conic problems where the cones are self-dual, and it would be interesting to extend the methods and results to more general classes of conic optimization problems.

References

[1] K. M. Anstreicher. On Vaidya's volumetric cutting plane method for convex programming. *Mathematics of Operations Research*, 22(1):63–89, 1997.

[2] K. M. Anstreicher. Towards a practical volumetric cutting plane method for convex programming. *SIAM Journal on Optimization*, 9(1):190–206, 1999.

[3] K. M. Anstreicher and N. W. Brixius. A new bound for the quadratic assignment problem based on convex quadratic programming. *Mathematical Programming*, 89(3):341–357, 2001.

[4] K. M. Anstreicher, N. W. Brixius, J.-P. Goux, and J. Linderoth. Solving large quadratic assignment problems on computational grids. *Mathematical Programming*, 91(3):563–588, 2002.

[5] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Implementing the Dantzig-Fulkerson-Johnson algorithm for large traveling salesman problems. *Mathematical Programming*, 97(1–2):91–153, 2003.

[6] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. *The traveling salesman problem: a computational study*. Princeton University Press, Princeton, NJ, 2006.

[7] S. Arora and C. Lund. Hardness of approximations. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, chapter 10. PWS, 1995.

[8] D. S. Atkinson and P. M. Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69:1–43, 1995.

[9] O. Bahn, O. Du Merle, J. L. Goffin, and J. P. Vial. A cutting plane method from analytic centers for stochastic programming. *Mathematical Programming*, 69:45–73, 1995.

[10] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

[11] V. L. Basescu and J. E. Mitchell. An analytic center cutting plane approach for conic programming. *Mathematics of Operations Research*, 33(3):529–551, 2008.

[12] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. John Wiley, New York, second edition, 1993.

[13] H. Y. Benson and D. F. Shanno. An exact primal-dual penalty method approach to warmstarting interior-point methods for linear programming. *Computational Optimization and Applications*, 38(3):371–399, 2007.

[14] H. Y. Benson and D. F. Shanno. Interior point methods for nonconvex nonlinear programming: Regularization and warmstarts. *Computational Optimization and Applications*, 40(2):143–189, 2008.

[15] C. Berge. *The Theory of Graphs and its Applications*. John Wiley and Sons, New York, 1962.

[16] R. E. Bixby, J. W. Gregory, I. J. Lustig, R. E. Marsten, and D. F. Shanno. Very large-scale linear programming: a case study in combining interior point and simplex methods. *Operations Research*, 40:885–897, 1992.

[17] R. E. Bixby and E. Rothberg. Progress in computational mixed integer programming — a look back from the other side of the tipping point. *Annals of Operations Research*, 149:37–41, 2007.

[18] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: a survey. *Operations Research*, 29(6):1039–1091, 1981.

[19] B. Borchers. SDPLIB 1.2, a library of semidefinite programming test problems. *Optimization Methods and Software*, 11:683–690, 1999.

[20] B. Borchers and J. G. Young. Implementation of a primal-dual method for SDP on a shared memory parallel architecture. *Computational Optimization and Applications*, 37(3):355–369, 2007.

[21] O. Briant, C. Lemaréchal, P. Meurdosoif, , S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113(2):299–344, 2008.

[22] C. Buchheim, A. Wiegele, and L. Zheng. Exact algorithms for the quadratic linear ordering problem. Technical Report zaik2007-560, Zentrum für Angewandte Informatik Köln, 2007. to appear in *INFORMS Journal on Computing*.

[23] S. Burer and R. D. C. Monteiro. A computational study of a gradient-based log-barrier algorithm for a class of large-scale SDPs. *Mathematical Programming*, 95(2):359–379, 2003.

[24] S. Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003.

[25] S. Burer, R. D. C. Monteiro, and Y. Zhang. Interior-point algorithms for semidefinite programming based on a nonlinear programming formulation. *Computational Optimization and Applications*, 22(1):49–79, 2002.

[26] M. Chen and S. Mehrotra. Self-concordant tree and decomposition based interior point methods for stochastic convex optimization problem. Technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208, May 2007.

[27] M. Chudnovsky, N. Robertson, P. D. Seymour, and R. Thomas. Progress on perfect graphs. *Mathematical Programming*, 97(1–2):405–422, 2003.

[28] M. Colombo, J. Gondzio, and A. Grothey. A warm-start approach for large-scale stochastic linear programs. Technical report, School of Mathematics, The University of Edinburgh, Scotland, August 2006.

[29] A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.

[30] A. d'Aspremont. Smooth semidefinite optimization. *Optima*, 78:9–11, 2008.

[31] A. d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. G. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49(3):434–448, 2007.

[32] E. de Klerk. Exploiting special structure in semidefinite programming: a survey of theory and applications. Technical report, Department of Econometrics and OR, Tilburg University, The Netherlands, January 2009.

[33] E. de Klerk, D. V. Pasechnik, and R. Sotirov. On semidefinite programming relaxations of the traveling salesman problem. *SIAM Journal on Optimization*, 19(4):1559–1573, 2008.

[34] C. De Simone, M. Diehl, M. Jünger, P. Mutzel, G. Reinelt, and G. Rinaldi. Exact ground states of two-dimensional $\pm J$ Ising spin glasses. *Journal of Statistical Physics*, 84:1363–1371, 1996.

[35] O. du Merle, J.-L. Goffin, and J.-P. Vial. On improvements to the analytic center cutting plane method. *Computational Optimization and Applications*, 11(1):37–52, 1998.

[36] S. Elhedhli and J.-L. Goffin. The integration of an interior-point cutting plane method within a branch-and-price algorithm. *Mathematical Programming*, 100(2):267–294, 2004.

[37] A. Engau, M. F. Anjos, and A. Vannelli. On interior point warmstarts for linear and combinatorial optimization. Technical report, Department of Management Sciences, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada, 2008.

[38] A. Engau, M. F. Anjos, and A. Vannelli. A primal-dual slack approach to warmstarting interior-point methods for linear programming. Technical report, Department of Management Sciences, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada, 2008.

[39] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. *Mathematical Programming*, 105(2–3):451–469, 2006.

[40] E. Fragniere, J. Gondzio, and J.-P. Vial. Building and solving large-scale stochastic programs on an affordable distributed computing system. *Annals of Operations Research*, 99:167–187, 2000.

[41] R. M. Freund. Theoretical efficiency of a shifted-barrier-function algorithm for linear programming. *Linear Algebra and its Applications*, 152:19–42, 1991.

[42] 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(3):647–674, 2001.

[43] B. Ghaddar, M. F. Anjos, and F. Liers. A branch-and-cut algorithm based on semidefinite programming for the minimum k-partition problem. Technical report, Department of Management Sciences, University of Waterloo, Waterloo, Ontario, N2L 3G1 Canada, 2007.

[44] M. X. Goemans and L. Tunçel. When does the positive semidefiniteness constraint help in lifting procedures? *Mathematics of Operations Research*, 26(4):796–815, 2001.

[45] Michel X. Goemans and David P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

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

[47] J.-L. Goffin and J.-P. Vial. Multiple cuts in the analytic center cutting plane method. *SIAM Journal on Optimization*, 11(1):266–288, 2001.

[48] J.-L. Goffin and J.-P. Vial. Convex nondifferentiable optimization: a survey focussed on the analytic center cutting plane method. *Optimization Methods and Software*, 17(5):805–867, 2002.

[49] J. Gondzio. Warm start of the primal-dual method applied in the cutting plane scheme. *Mathematical Programming*, 83:125–143, 1998.

[50] 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 Operational Research*, 94:206–211, 1996.

[51] J. Gondzio and A. Grothey. Re-optimization with the primal-dual interior point method. *SIAM Journal on Optimization*, 13(3):842–864, 2003.

[52] J. Gondzio and A. Grothey. A new unblocking technique to warmstart interior point methods based on sensitivity analysis. *SIAM Journal on Optimization*, 19(3):1184–1210, 2008.

[53] J. Gondzio and R. Kouwenberg. High-performance computing for asset-liability management. *Operations Research*, 49(6):879–891, 2001.

[54] J. Gondzio and J.-P. Vial. Warm start and ε-subgradients in a cutting plane scheme for block-angular linear programs. *Computational Optimization and Applications*, 14(1):17–36, 1999.

[55] M. C. Grant and S. P. Boyd. CVX users' guide. Technical report, Information Systems Laboratory, Department of Electrical Engineering, Stanford University, February 2009.

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

[57] M. Grötschel, L. Lovasz, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[58] G. Gruber and F. Rendl. Computational experience with stable set relaxations. *SIAM Journal on Optimization*, 13(4):1014–1028, 2003.

[59] C. Helmberg. Fixing variables in semidefinite relaxations. *SIAM Journal on Matrix Analysis and Applications*, 21(3):952–969, 2000.

[60] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report ZR-00-34, TU Berlin, Konrad-Zuse-Zentrum, Berlin, October 2000. Habilitationsschrift.

[61] C. Helmberg. Numerical evaluation of SBmethod. *Mathematical Programming*, 95(2):381–406, 2003.

[62] C. Helmberg. A cutting plane algorithm for large scale semidefinite relaxations. In M. Grötschel, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*, pages 233–256. SIAM, 2004.

[63] C. Helmberg and K. Kiwiel. A spectral bundle method with bounds. *Mathematical Programming*, 93(2):173–194, 2002.

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

[65] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[66] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Springer-Verlag, Berlin, 1993.

[67] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Springer-Verlag, Berlin, 1993.

[68] D. Huisman, R. Jans, M. Peeters, and A. P. M. Wagelmans. Combining column generation and Lagrangian relaxation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, GERAD 25th Anniversary Series, chapter 9, pages 247–270. Springer, Berlin, 2005.

[69] B. Jansen, C. Roos, T. Terlaky, and J.-P. Vial. Primal-dual target following algorithms for linear programming. *Annals of Operations Research*, 62:197–231, 1996.

[70] E. John and E. A. Yıldırım. Implementation of warm-start strategies in interior-point methods for linear programming in fixed dimension. *Computational Optimization and Applications*, 41(2):151–183, 2007.

[71] N. K. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[72] J. E. Kelley. The cutting plane method for solving convex programs. *Journal of the SIAM*, 8:703–712, 1960.

[73] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademiia Nauk SSSR*, 224:1093–1096, 1979. English Translation: *Soviet Mathematics Doklady*, Volume 20, pp. 191–194.

[74] K. C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. *Mathematical Programming*, 27:320–341, 1983.

[75] M. Kocvara and M. Stingl. On the solution of large-scale SDP problems by the barrier method using iterative solvers. *Mathematical Programming*, 109(2–3):413–444, 2007.

[76] K. Krishnan. *Linear programming approaches to semidefinite programming problems*. PhD thesis, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, July 2002.

[77] K. Krishnan and J. E. Mitchell. Semi-infinite linear programming approaches to semidefinite programming (SDP) problems. In P. M. Pardalos and H. Wolkowicz, editors, *Novel approaches to hard discrete optimization problems*, volume 37 of *Fields Institute Communications Series*, pages 123–142. AMS, 2003.

[78] K. Krishnan and J. E. Mitchell. A unifying framework for several cutting plane methods for semidefinite programming. *Optimization Methods and Software*, 21(1):57–74, 2006.

[79] K. Krishnan and J.E. Mitchell. A semidefinite programming based polyhedral cut-and-price approach for the maxcut problem. *Computational Optimization and Applications*, 33(1):51–71, 2006.

[80] K. Krishnan and T. Terlaky. Interior point and semidefinite approaches in combinatorial optimization. In D. Avis, A. Hertz, and O. Marcotte, editors, *Graph Theory and Combinatorial Optimization*, GERAD 25th anniversary volume, chapter 5. Springer, 2005.

[81] G. Lan, Z. Lu, and R. D. C. Monteiro. Primal-dual first-order methods with $o(1/\epsilon)$ iteration-complexity for cone programming. Technical report, School of ISyE, Georgia Tech, December 2006.

[82] M. Laurent and F. Rendl. Semidefinite programming and integer programming. In K. Aardal, G. Nemhauser, and R. Weismantel, editors, *Handbook on Discrete Optimization*, volume 12 of *Handbooks in Operations Research and Management Sciences*, pages 393–514. Elsevier, Amsterdam, The Netherlands, 2005.

[83] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi. Computing exact ground states of hard Ising spin glass problems by branch-and-cut. In A. Hartmann and H. Rieger, editors, *New Optimization Algorithms in Physics*, pages 47–68. John Wiley, 2004.

[84] A. Lisser and F. Rendl. Graph partitioning using linear and semidefinite programming. *Mathematical Programming*, 95(1):91–101, 2003.

[85] L. Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, 1979.

[86] Z. Lu and R. D. C. Monteiro. Iteration complexity of first-order penalty methods for convex programming. Technical report, School of ISyE, Georgia Tech, July 2008.

[87] Z. Lu, A. Newmirovski, and R. D. C. Monteiro. Large-scale semidefinite programming via a saddle point mirror-prox algorithm. *Mathematical Programming*, 109(2):211–237, 2007.

[88] M. E. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023, 2005.

[89] S. Mehrotra and M. Özevin. On the implementation of interior point decomposition algorithms for two-stage stochastic conic problems. Technical Report 2005–04, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Illinois 60208, October 2005.

[90] J. E. Mitchell. An interior point column generation method for linear programming using shifted barriers. *SIAM Journal on Optimization*, 4:423–440, May 1994.

[91] J. E. Mitchell. Fixing variables and generating classical cutting planes when using an interior point branch and cut method to solve integer programming problems. *European Journal of Operational Research*, 97:139–148, 1997.

[92] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM Journal on Optimization*, 10(4):1212–1227, 2000.

[93] J. E. Mitchell. Restarting after branching in the SDP approach to MAX-CUT and similar combinatorial optimization problems. *Journal of Combinatorial Optimization*, 5(2):151–166, 2001.

[94] J. E. Mitchell. Polynomial interior point cutting plane methods. *Optimization Methods and Software*, 18(5):507–534, 2003.

[95] J. E. Mitchell and V. L. Basescu. Selective Gram-Schmidt orthonormalization for conic cutting surface algorithms. *Mathematical Methods of Operations Research*, 67(1):91–115, 2008.

[96] 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*, 62:253–276, 1996.

[97] J. E. Mitchell and B. Borchers. Solving linear ordering problems with a combined interior point/simplex cutting plane algorithm. In H. L. Frenk *et al.*, editor, *High Performance Optimization*, chapter 14, pages 349–366. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[98] J. E. Mitchell and M. J. Todd. Solving combinatorial optimization problems using Karmarkar's algorithm. *Mathematical Programming*, 56(3):245–284, 1992.

[99] H. D. Mittelmann and J. Peng. Estimating bounds for quadratic assignment problems associated with Hamming and Manhattan distance matrices based on semidefinite programming. Technical report, Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, July 2008.

[100] R. D. C. Monteiro. First- and second-order methods for semidefinite programming. *Mathematical Programming*, 97(1–2):209–244, 2003.

[101] M. V. Nayakkankuppam. *Optimization over symmetric cones*. PhD thesis, Department of Computer Science, New York University, September 1999.

[102] M. V. Nayakkankuppam. Solving large-scale semidefinite programs in parallel. *Mathematical Programming*, 109(2-3):477–504, 2007.

[103] A. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

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

[105] Y. E. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[106] Y. E. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2006.

[107] Y. E. Nesterov. Smoothing technique and its applications in semidefinite programming. *Mathematical Programming*, 110(2):245–259, 2007.

[108] Y. E. Nesterov. How to advance in structural convex optimization. *Optima*, 78:2–5, 2008.

[109] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms*. SIAM Publications. SIAM, Philadelphia, USA, 1993.

[110] M. R. Oskoorouchi and J.-L. Goffin. The analytic center cutting plane method with semidefinite cuts. *SIAM Journal on Optimization*, 13(4):1029–1053, 2003.

[111] M. R. Oskoorouchi and J.-L. Goffin. An interior point cutting plane method for the convex feasibility problem with second-order cone inequalities. *Mathematics of Operations Research*, 30(1):127–149, 2005.

[112] M. R. Oskoorouchi and J.-L. Goffin. A matrix generation approach for eigenvalue optimization. *Mathematical Programming*, 109(1):155–179, 2007.

[113] M. R. Oskoorouchi and J. E. Mitchell. A second-order cone cutting surface method: complexity and application. Technical report, College of Business Administration, California State University San Marcos, California, May 2005. to appear in *Computational Optimization and Applications*.

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

[115] A. Pages, J. Gondzio, and N. Nabona. Warmstarting for interior point methods applied to the long-term power planning problem. *European Journal of Operational Research*, 197(1):112–125, 2009.

[116] Adela Pages Bernaus. *Column-generation and interior point methods applied to the long-term electric power-planning problem*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, 2007.

[117] G. Pataki. On the rank of extreme matrices in semidefinite programs and the multiplicity of optimal eigenvalues. *Mathematics of Operations Research*, 23(2):339–358, 1998.

[118] J. Peng, H. D. Mittelmann, and X. Li. New convex relaxations for quadratic assignment problems. Technical report, Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana-Champaign, October 2008.

[119] R. Polyak. Modified barrier functions (theory and methods). *Mathematical Programming*, 54:177–222, 1992.

[120] S. Ramaswamy and J. E. Mitchell. Using selective orthonormalization to update the analytic center after the addition of multiple cuts. *Journal of Optimization Theory and Applications*, 125(2):431–451, 2005.

[121] F. Rendl, G. Rinaldi, and A. Wiegele. A branch and bound algorithm for max-cut based on combining semidefinite and polyhedral relaxations. In *Integer Programming and Combinatorial Optimization, 12th International Conference, IPCO 2007, Ithaca, NY*, 2007.

[122] F. Rendl, G. Rinaldi, and A. Wiegele. Solving max-cut to optimality by intersecting semidefinite and polyhedral relaxations. Technical report, University of Klagenfurt, 2007. accepted for publication in *Mathematical Programming*.

[123] F. Rendl and R. Sotirov. Bounds for the quadratic assignment problem using the bundle method. *Mathematical Programming*, 109(2–3):505–524, 2007.

[124] J. Renegar. A polynomial-time algorithm based on Newton's method for linear programming. *Mathematical Programming*, 40:59–93, 1988.

[125] J. Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. Number 1 in MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2001.

[126] C. Roos, T. Terlaky, and J.-Ph. Vial. *Interior Point Methods for Linear Optimization*. Springer, 2005.

[127] L.-M. Rousseau. Stabilization issues for constraint programming based column generation. In J.-C. Régin and M. Rueher, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. First International Conference, CPAIOR 2004, Nice, France, April 20-22, 2004*, volume 3011 of *Lecture Notes in Computer Science*, pages 402–408. Springer, 2004.

[128] L.-M. Rousseau, M. Gendreau, and D. Feillet. Interior point stabilization for column generation. *Operations Research Letters*, 35(5):660–668, 2007.

[129] H. D. Sherali and B. M. P. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22(1–4):233–261, 2002.

[130] K. Krishnan Sivaramakrishnan. A parallel interior point decomposition algorithm for block angular semidefinite programs. Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, August 2007. accepted for publication in *Computational Optimization and Applications*.

[131] K. Krishnan Sivaramakrishnan, G. Plaza, and T. Terlaky. A conic interior point decomposition approach for large scale semidefinite programming. Technical report, Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, December 2005.

[132] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[133] J. Sun, K. C. Toh, and G. Y. Zhao. An analytic center cutting plane method for semidefinite feasibility problems. *Mathematics of Operations Research*, 27:332–346, 2002.

[134] K. C. Toh, G. Zhao, and J. Sun. A multiple-cut analytic center cutting plane method for semidefinite feasibility problems. *SIAM Journal on Optimization*, 12(4):1126–1146, 2002.

[135] R. H. Tütüncü, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using SDPT3. *Mathematical Programming*, 95(2):189–217, 2003.

[136] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73(3):291–341, 1996.

[137] R. J. Vanderbei. *Linear Programming: Foundations and Extensions*. Kluwer Academic Publishers, Boston, 1996. Second Edition: 2001.

[138] S. Wright. *Primal-dual interior point methods*. SIAM, Philadelphia, 1996.

[139] M. Yamashita, K. Fujisawa, and M. Kojima. SDPARA: Semidefinite programming algorithm parallel version. *Parallel Computing*, 29(8):1053–1067, 2003.

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

[141] E. A. Yildirim and S. J. Wright. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization*, 12(3):692–714, 2002.

[142] X.-Y. Zhao, D. Sun, and K.-C. Toh. A Newton-CG augmented Lagrangian method for semidefinite programming. Technical report, Department of Mathematics, University of Singapore, March 2008.