Interior Point Methods 25 Years Later

Gondzio, J 2012 'Interior Point Methods 25 Years Later' European Journal of Operational Research, vol 218, pp. 587-601.

Interior Point Methods 25 Years Later*

Jacek Gondzio†

School of Mathematics
and Maxwell Institute for Mathematical Sciences
The University of Edinburgh
Mayfield Road, Edinburgh EH9 3JZ
United Kingdom.

Technical Report ERGO-2011-003‡

February 23, 2011, revised September 8, 2011,
accepted for publication in **European Journal of Operational Research**

Abstract

Interior point methods for optimization have been around for more than 25 years now. Their presence has shaken up the field of optimization. Interior point methods for linear and (convex) quadratic programming display several features which make them particularly attractive for very large scale optimization. Among the most impressive of them are their low-degree polynomial worst-case complexity and an unrivalled ability to deliver optimal solutions in an almost constant number of iterations which depends very little, if at all, on the problem dimension. Interior point methods are competitive when dealing with small problems of dimensions below one million constraints and variables and are beyond competition when applied to large problems of dimensions going into millions of constraints and variables.

In this survey we will discuss several issues related to interior point methods including the proof of the worst-case complexity result, the reasons for their amazingly fast practical convergence and the features responsible for their ability to solve very large problems. The ever-growing sizes of optimization problems impose new requirements on optimization methods and software. In the final part of this paper we will therefore address a redesign of interior point methods to allow them to work in a matrix-free regime and to make them well-suited to solving even larger problems.

Keywords: Interior Point Methods, Linear Programming, Quadratic Programming, Worst-case Complexity Analysis, Implementation, Matrix-Free Methods.

*Supported by EPSRC grant EP/I017127/1.
†Email: J.Gondzio@ed.ac.uk, URL: http://maths.ed.ac.uk/~gondzio/
‡For other papers in this series see http://www.maths.ed.ac.uk/ERGO/

1 Introduction

Linear programs (LPs) have been at the centre of attention of the optimization field since the 1940s. For several decades the simplex algorithm [60, 23] was the only method available to solve this important class of optimization problems. Although in theory the simplex method is a non-polynomial algorithm (in the worst-case it might have to make a very large number of steps which depends exponentially on the problem dimension), in practice it has proved to be a very efficient and reliable method. By its design, the simplex method visits the vertices of the polytope defined by the constraints and, because their number may be astronomical, the method is exposed to a danger of having to visit many of them before reaching an optimal one. No polynomial simplex-type pivoting algorithm is known to date and it is hard to believe that one will ever be found although researchers have not lost hope, and continue their search for one [99].

The first polynomial algorithm for LP was developed by Khachiyan [66]. His ellipsoid algorithm constructs a series of ellipsoids inscribed into the feasible set. The centres of these ellipsoids form a sequence of points convergent to an optimal solution of the LP. The construction of ellipsoids provides a guarantee that steady progress towards optimality can be made from one iteration to another. The development of the ellipsoid method made a huge impact on the theory of linear programming but the method has never become a competitive alternative to the simplex method because the per-iteration cost of the linear algebra operations to update the ellipsoids is too high [41].

Karmarkar's projective LP algorithm [61] could be interpreted as a refinement of the ellipsoid method. Instead of inscribing an ellipsoid into the ill-conditioned corner of the feasible polytope, Karmarkar's algorithm employs projective geometry to transform this "corner" into a regular well-conditioned simplex polytope, inscribes a ball into it, and exploits the fact that optimization on a ball is a trivial operation. Additionally, Karmarkar's method uses a notion of a potential function (a sort of merit function) to guarantee a steady reduction of a distance to optimality at each iteration. Although a single iteration of Karmarkar's method is expensive (it requires a projection operation to be applied, and this operation changes at each iteration), optimality is reached after a relatively small number of iterations which makes the algorithm computationally attractive.

Karmarkar's proof of the worst-case complexity result was rather complicated. Its development was accompanied by claims of unprecedented efficiency for the new method, which managed to attract huge interest from the optimization community. The efforts of numerous researchers soon led to improvements and clarifications of the theory. Gill et al. [40] established an equivalence between Karmarkar's projective method and the projected Newton barrier method. This increased interest in the role of barrier functions in the theory of interior point methods and has drawn the community's attention to numerous advantageous features of *logarithmic barrier functions*. (Interestingly, the use of a logarithmic barrier method in the context of optimization had already been proposed in 1955 by Frisch [37] and studied extensively by Fiacco and McCormick [32] in the context of nonlinear optimization.)

It is broadly accepted today that an infeasible-primal-dual algorithm is the most efficient interior point method. A number of attractive features of this method follow from the fact that a logarithmic barrier method is applied to the primal and the dual problems at the same time. This was first suggested by Megiddo [77]. Independently, Kojima, Mizuno and Yoshise [69]

developed the theoretical background of this method and gave the first complexity results. Further progress was made by Kojima, Megiddo and Mizuno [68] who provided good theoretical results for the primal-dual algorithm with extra safe-guards and those could be translated into computational practice. The interpretation of interior point methods as algorithms which follow a path of centres (central trajectory) on their way to an optimal solution was gaining wide acceptance [47]. In the late 80's, Mehrotra and independently Lustig, Marsten, Shanno and their collaborators made impressive progress in the implementation of interior point methods and provided also a better understanding of the crucial role played by the logarithmic barrier functions in the theory [76]. By the early 90's sufficient evidence was already gathered to justify claims of the spectacular efficiency of IPMs for very large scale linear programming [78, 73, 74]. A new class of optimization methods able to compete with the simplex method was quickly gaining wide appreciation.

It is worth mentioning at this point that the presence of interior point methods have put considerable pressure on developers of commercial simplex implementations and have led to impressive developments of the simplex method over the last 25 years [13, 33, 54, 75, 98]. Both methods are widely used nowadays and continue to compete with each other. Although the large size of the problem generally seems to favour interior point methods, it is not always possible to predict the winner on a particular class of problems. For example, the sparsity structure of the problem determines the cost of linear algebra operations and therefore determines the efficiency of a given algorithm leading sometimes to astonishing results by significantly favouring one method over the other. The simplex method easily takes advantage of any hyper-sparsity in a problem [54] but its sequential nature makes it difficult to parallelise [53]. On the other hand, interior point methods are able to exploit any block-matrix structure in the linear algebra operations and therefore significant speed-ups can be achieved by massive parallelisation [44].

Having applied a *nonlinear* programming technique (based on the use of logarithmic barrier function) to solve the *linear* optimization problem was the key reason of IPMs success. Soon after the major role played by the logarithmic barrier function [40] had been understood, a similar methodology was applied to solve quadratic [103] and nonlinear optimization problems [104] and indeed, as it was nicely pointed out by Forsgren et al. [34] "an especially appealing aspect of the interior-point revolution is its spirit of unification, which has brought together areas of optimization that for many years were treated as firmly disjoint".

Nesterov and Nemirovskii [85] provided an insightful explanation why the logarithmic function is such a well-suited barrier function. Its advantage results from the *self-concordance* property which makes the logarithmic function particularly attractive to be applied in an optimization technique based on the Newton method. The theory of self-concodrant barriers [85] expanded further the area of IPM applications and covered a semidefinite programming and more generally a conic optimization which also includes another important class of the second-order cone programming. In this survey we will focus on the linear and convex quadratic programming problems, the classes of optimization problems which are by far the most frequently used in various real-life applications. The readers interested in nonlinear, semidefinite and second-order cone programming are referred to excellent surveys of Forsgren et al. [34], Vandenberghe and Boyd [101], and Lobo et al. [70], respectively.

In this paper we will (gently) guide the reader through major issues related to the fascinating theory and implementation of IPMs. The survey is organised as follows. In Section 2 we will introduce the quadratic optimization problem, define the notation used in the paper and discuss in detail an essential difference between the simplex and interior point method, namely the way in

which these methods deal with the complementarity condition. In Section 3 we will perform the worst-case analysis of a particular interior point algorithm for convex quadratic programming. We will analyse the feasible algorithm operating in a small neighbourhood of the central path induced by the 2-norm. Our proof will follow a scheme set for linear programming in an excellent textbook on IPMs by Wright [105] and will generalize it to the case of quadratic programming. In Section 4 we will guide the reader through the issues of implementation of IPMs and we will discuss several techniques which are responsible for the impressive practical efficiency of these methods. In Section 5 we will discuss the issues of linear algebra applied in IPMs including direct and iterative methods for symmetric systems of linear equations. In Section 6 we will discuss existing and potential techniques which may further accelerate the performance of interior point methods. In particular we will discuss the recently developed framework in which IPMs can work in a matrix-free regime, which makes them eligible for solving even larger scale problems of the future. We will demonstrate the approach when applied to two otherwise intractable classes of optimization problems: linear relaxations of notoriously difficult quadratic assignment problems and challenging linear programs arising in quantum mechanics. Finally, in Section 7 we will give our conclusions and comment on possible further developments of interior point methods.

2 Interior Point Methods: Background

We are concerned in this paper with the theory and implementation of interior point methods for solving linear and convex quadratic programming problems. We consider the following general primal-dual pair of convex quadratic programming (QP) problems

$$\begin{array}{lll} \text{Primal} & & \text{Dual} \\ \min & c^Tx + \frac{1}{2}x^TQx & \max \quad b^Ty - \frac{1}{2}x^TQx \\ \text{s.t.} & Ax = b, & \text{s.t.} \quad A^Ty + s - Qx = c, \\ & x \geq 0; & \quad\; y \text{ free}, \;\; s \geq 0, \end{array} \tag{1}$$

where $A \in \mathcal{R}^{m\times n}$ has full row rank $m \leq n$, $Q \in \mathcal{R}^{n\times n}$ is a positive semidefinite matrix, $x, s, c \in \mathcal{R}^n$ and $y, b \in \mathcal{R}^m$. In a special case of $Q = 0$ these problems become the primal-dual pair of linear programming (LP) problems.

Using Lagrangian duality theory [11] the first order optimality conditions for these problems can be written as:

$$\begin{array}{rcl} Ax & = & b, \\ A^Ty + s - Qx & = & c, \\ XSe & = & 0, \\ (x, s) & \geq & 0, \end{array} \tag{2}$$

where X and S are diagonal matrices in $\mathcal{R}^{n\times n}$ with elements of vectors x and s spread across the diagonal, respectively and $e \in \mathcal{R}^n$ is the vector of ones. The third equation $XSe = 0$, called the complementarity condition, can be rewritten as $x_js_j = 0$, $\forall j \in \{1, 2, \ldots, n\}$ and implies that at least one of the two variables x_j and s_j has to be zero at the optimum. The way in which complementarity condition is dealt with determines the type of optimization algorithms.

Active Set Methods and their prominent example, the simplex method for linear programming, make an intelligent guess that either $x_j = 0$ or $s_j = 0$. They choose a subset of indices

$j \in \mathcal{B} \subset \{1, 2, \ldots, n\}$ such that x_j is allowed to be nonzero and force the corresponding $s_j = 0$, while the remaining indices $j \in \mathcal{N} = \{1, 2, \ldots, n\} \setminus \mathcal{B}$ force $x_j = 0$ and allow s_j to take nonzero values.

Interior Point Methods perturb the complementarity condition and replace $x_j s_j = 0$ with $x_j s_j = \mu$, where the parameter μ is driven to zero. This removes the need to "guess" the partitioning into active and inactive inequality constraints: the algorithm forces a reduction of μ; the partition of vectors x and s into zero and nonzero elements is gradually revealed as the algorithm progresses.

The way of handling the complementarity condition has major consequences for the implementation of the method, its theoretical worst-case complexity and its practical efficiency.

The simplex method allows only one index to be swapped between $\mathcal{B}$ and $\mathcal{N}$ at each iteration. (In the more general context of active set methods the index of only one variable and/or active constraint can be exchanged at each iteration.) The linear algebra techniques used by the simplex method allow us to eliminate the non-basic part of the constraint matrix and to express all operations in terms of the basis matrix B, a submatrix of A induced by columns from set $\mathcal{B}$. Swapping one index between $\mathcal{B}$ and $\mathcal{N}$ at each iteration means that only one column of B needs to be replaced and this is exploited in a very efficient updating of the basis inverse representation. Consequently the cost of a single iteration in the simplex method is usually very low. However, the choice of which indices to swap relies on local information associated with the current $(\mathcal{B}, \mathcal{N})$ partition and one cannot control the number of such replacements required before the optimal partition is found. Klee and Minty [67] constructed a problem of dimension n, the solution of which requires 2^n iterations of the simplex method, which demonstrates that the simplex method is not a polynomial algorithm. It has to be said, however, that it is only a theoretical drawback and in practice it is rather exceptional for the simplex method to perform more than $m + n$ iterations on its way to an optimal solution [13, 33, 54, 75, 98].

Interior point methods, on the other hand, force the convergence to optimality by gradually reducing the perturbation μ in the approximate complementarity condition and eventually driving it arbitrarily close to zero. Therefore, the decision which of the two variables x_j or s_j should be zero is delayed (but certainly not left to "guessing"). All variables (x, y, s) change at each iteration and the linear algebra operations which are required to update them have to involve the complete matrix A. This makes a single iteration of the interior point method significantly more expensive than that of the simplex method. The theory of IPMs focuses on clever ways of reducing μ to zero and gradually revealing the optimal partition. Normally, IPMs terminate when an ε-accurate solution ($\mu \leq \varepsilon$) is found. The best known to date IPM algorithm for linear and quadratic programming finds the ε-accurate solution of an LP or a convex QP problem in $\mathcal{O}(\sqrt{n} \ln(1/\varepsilon))$ iterations [90]. Accodring to the general theory [83, Ch. 4] the term $\sqrt{n}$ is the best one can expect for an IPM using a self-concordant barrier function such as the logarithmic barrier.

In practice IPMs perform much better than that and converge in a number of iterations which is almost a constant, independent of the problem dimension [22].

Interior point methods for optimization exploit the following three tools:

- they use logarithmic barrier functions to "replace" the inequality constraints,
- they apply duality theory to replacement problems involving barriers to derive the first order optimality conditions which take the form of a system of nonlinear equations, and

- they apply the Newton method to solve this system of nonlinear equations.

Rather than explicitly forcing a nonnegativity constraint $x_j \geq 0$ in the IPM framework a logarithmic barrier term $-\mu \ln x_j$ is added to the objective of the optimization problem. By replacing all nonnegativity constraints in the primal problem (1) we obtain the barrier QP subproblem

$$\min \;\; c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^{n} \ln x_j \quad \text{s.t.} \quad Ax = b, \tag{3}$$

The parameter μ controls the relation between the barrier term and the original QP objective. Clearly the smaller the barrier parameter μ the less important influence the barrier term becomes and therefore more attention is paid to the QP objective. The optimization starts with a large value of μ, which corresponds to ignoring the original objective and keeping the solution x in the interior of the positive orthant, that is, preventing any of the components x_j from approaching their boundary value of zero. Next the parameter μ is gradually reduced hence the logarithmic barrier is weakened allowing the algorithm to minimize the original objective and ultimately to approach an optimal solution in which some of the components x_j may be zero. Associated with (3) is the Lagrangian

$$\mathcal{L}(x, y) = c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^{n} \ln x_j - y^T (Ax - b) \tag{4}$$

and the first order optimality conditions become

$$\begin{array}{rclcl} \nabla_y \mathcal{L}(x, y) & = & Ax - b & = & 0, \\ \nabla_x \mathcal{L}(x, y) & = & c - A^T y - \mu X^{-1} e + Qx & = & 0. \end{array} \tag{5}$$

By introducing a new variable $s = \mu X^{-1} e$ (i.e. $XSe = \mu e$) the first order optimality conditions become

$$\begin{array}{rcl} Ax & = & b, \\ A^T y + s - Qx & = & c, \\ XSe & = & \mu e, \\ (x, s) & \geq & 0. \end{array} \tag{6}$$

It is worth comparing (2) and (6) and noticing that the only difference is a perturbation of the complementarity constraint. A consequence of the presence of the logarithmic barrier is that all complementarity products take the same value μ. It is easy to show (we will do it in Lemma 3.1 below) that if (x, y, s) is primal and dual feasible then the duality gap in the barrier subproblem is equal to the complementarity gap $x^T s = n\mu$, which immediately reveals that the barrier term μ controls the distance to optimality.

The theory of IPMs focuses on a choice of a suitable sequence $\{\mu^k\}$. Large values of μ at the beginning of the optimization process promote *centrality*, that is, prevent components of x and s from getting too small. The reduction of μ moves the balance from centrality to optimality and eventually allows the algorithm to approach an optimal solution which may be (and in the LP case it always is) at the boundary of the feasible region. For any $\mu > 0$, system (6) has a unique solution $(x(\mu), y(\mu), s(\mu))$, $x(\mu) > 0$, $s(\mu) > 0$ which is called a μ-centre. A family of these solutions for all positive values of μ determines a (continuous) path $\{(x(\mu), y(\mu), s(\mu)) : \mu > 0\}$ which is called the primal-dual *central path* or *central trajectory*.

The first two equations in (6) which correspond to primal and dual feasibility conditions in (1) are linear and only the third equation which corresponds to the perturbed complementarity condition is (mildly) nonlinear. Interior point algorithms do not have to solve this system of nonlinear equations to a high degree of accuracy. Recall that (6) is only an approximation of (2) corresponding to a specific choice of the barrier parameter μ and the barrier term will have to be reduced to force convergence. IPMs apply the Newton method to solve (6). To be more precise, IPMs compute the Newton direction and make *one* step in this direction before reducing the barrier parameter μ. The Newton direction $(\Delta x, \Delta y, \Delta s)$ is computed by solving the following system of linear equations

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I_n \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} = \begin{bmatrix} b - Ax \\ c + Qx - A^T y - s \\ \sigma\mu e - XSe \end{bmatrix}, \tag{7}$$

where I_n denotes the identity matrix of dimension n.

It is worth having a closer look at a particular form of the third row of the Jacobian matrix on the left hand side of the Newton system (7). Due to "symmetry" of the perturbed complementarity condition $XSe = \mu e$ this equation becomes

$$S\Delta x + X\Delta s = \xi_\mu.$$

To guarantee progress in reducing the complementarity gap $x^T s$, interior point algorithms attempt to reduce the barrier term from its current value μ to a new one $\sigma\mu$ where $\sigma \in (0,1)$ and therefore by using $e^T e = n$ and $x^T s = n\mu$ the third equation in (7) implies

$$s^T\Delta x + x^T\Delta s = \sigma\mu e^T e - x^T s = (\sigma - 1)x^T s$$

and provides the necessary term to estimate the progress of the algorithm. Indeed, when a damped Newton step is made in the direction $(\Delta x, \Delta y, \Delta s)$, with stepsize α possibly smaller that 1, the complementarity gap at the new point $(\bar{x}, \bar{y}, \bar{s}) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s)$ becomes

$$\begin{aligned} \bar{x}^T\bar{s} &= (x + \alpha\Delta x)^T(s + \alpha\Delta s) \\ &= x^T s + \alpha(s^T\Delta x + x^T\Delta s) + \alpha^2\Delta x^T\Delta s \\ &= (1 - \alpha(1-\sigma))x^T s + \alpha^2\Delta x^T\Delta s \end{aligned} \tag{8}$$

and if the second order term $\Delta x^T\Delta s$ can be kept small the new gap $\bar{x}^T\bar{s}$ is smaller than the old one $x^T s$. It is easier to derive the polynomial worst-case complexity result for the *feasible* interior point algorithm. Therefore we will assume that all primal-dual iterates belong to $\mathcal{F}^0 = \{(x, y, s) \,|\, Ax = b,\ A^T y + s - Qx = c,\ (x, s) > 0\}$. The proof of the complexity result requires swift control of the complementarity product $x^T s$ and keeping the second order error term $\Delta x^T\Delta s$ small. This is achieved by containing the iterates to a small vicinity of the central path, namely by guaranteeing that the error in equation $XSe = \mu e$ is small. The error is measured by some norm of the residual of this equation $\|XSe - \mu e\|$. We follow [105] and use the Euclidean norm in this context hence the following neighbourhood of the central path is defined:

$$N_2(\theta) = \{(x, y, s) \in \mathcal{F}^0 \,|\, \|XSe - \mu e\| \le \theta\mu\}, \tag{9}$$

where $\theta \in (0, 1)$ is an arbitrary parameter and the barrier term is equal to the average complementarity product $\mu = x^T s/n$. Such a definition of the neighbourhood of the central path implies that the size of the neighbourhood is decreasing as long as the central path approaches

an optimal solution (as long as μ approaches zero). The need for the iterates to stay in the vicinity of the central path is a source of an alternative name for IPMs as path-following methods: indeed, these algorithms follow the central path on their way to optimality [47].

We summarize this section by giving a general algorithm of the infeasible path-following method for convex quadratic programming.

INFEASIBLE PATH-FOLLOWING METHOD FOR QUADRATIC PROGRAMMING

Parameters

$\alpha_0 = 0.99$ a fraction-to-the-boundary stepsize factor;
$\sigma \in (0,1)$ barrier reduction parameter;
ε_p, ε_d, ε_o primal feasibility, dual feasibility and optimality tolerances:
IPM stops when $\frac{\|\xi_p^k\|}{1+\|b\|} \leq \varepsilon_p$, $\frac{\|\xi_d^k\|}{1+\|c\|} \leq \varepsilon_d$ and $\frac{(x^k)^T s^k/n}{1+|c^T x^k + 1/2(x^k)^T Q x^k|} \leq \varepsilon_o$.

Initialize IPM

iteration counter $k = 0$;
primal-dual point $x^0 > 0$, $y^0 = 0$, $s^0 > 0$;
barrier parameter $\mu^0 = (x^0)^T s^0/n$;
primal and dual infeasibilities $\xi_p^0 = b - Ax^0$ and $\xi_d^0 = c - A^T y^0 - s^0 + Qx^0$.

Interior Point Method

while ($\frac{\|\xi_p^k\|}{1+\|b\|} > \varepsilon_p$ or $\frac{\|\xi_d^k\|}{1+\|c\|} > \varepsilon_d$ or $\frac{(x^k)^T s^k/n}{1+|c^T x^k + 1/2(x^k)^T Q x^k|} > \varepsilon_o$) **do**

Update (reduce) the barrier $\mu^{k+1} = \sigma \mu^k$;

Solve the KKT system (7): find the primal-dual Newton direction $(\Delta x, \Delta y, \Delta s)$.

Find $\alpha_P = \max\{\alpha : x^k + \alpha \Delta x \geq 0\}$ and $\alpha_D = \max\{\alpha : s^k + \alpha \Delta s \geq 0\}$;
Set $\alpha_P := \alpha_0 \alpha_P$ and $\alpha_D := \alpha_0 \alpha_D$;
Make step
$x^{k+1} = x^k + \alpha_P \Delta x$;
$y^{k+1} = y^k + \alpha_D \Delta y$;
$s^{k+1} = s^k + \alpha_D \Delta s$.
Compute the infeasibilities: $\xi_p^{k+1} = b - Ax^{k+1}$ and $\xi_d^{k+1} = c - A^T y^{k+1} - s^{k+1} + Qx^{k+1}$;
Update the iteration counter: $k := k + 1$.

end-while

In the next section we will prove that interior point algorithm operating in the $N_2(\theta)$ neighbourhood applied to a convex QP converges to an ε-accurate solution in $\mathcal{O}(\sqrt{n}\ln(1/\varepsilon))$ iterations. This variant is known in the literature as the short-step path-following algorithm.

3 Polynomial Complexity Result

Our proof will follow a general scheme set in an excellent textbook on IPMs by Wright [105]. In particular, our proofs of Lemmas 3.3, 3.5, 3.6 and Theorem 3.1 are taken from Lemmas 5.3, 5.4 and Theorem 3.2 in [105]. However, we deal with an IPM for quadratic programming and this requires care to be taken of some extra terms which were not present in Wright's proof for the linear programming case. We will extensively exploit the fact that all iterates belong to the $N_2(\theta)$ neighbourhood (9) of the central path. Therefore all iterates are strictly primal-dual feasible which implies that the right hand side vector in the linear system defining the Newton direction (7) takes the form $(0, 0, \sigma\mu e - XSe)$.

Slow but systematic reduction of the complementarity gap (and duality gap) is achieved by imposing a decrease of the barrier term in each iteration. A marginal per-iteration reduction of μ is enough to deliver a polynomial complexity for the algorithm. We ask for $\mu^{k+1} = \sigma\mu^k$, where $\sigma = 1 - \beta/\sqrt{n}$ for some $\beta \in (0, 1)$ hence σ is barely smaller than 1. We will make an exception in a small paragraph below and use the letter e to denote the Euler number (the base of natural logarithm). Everywhere else in the paper e will always denote a vector of ones of appropriate dimension. Using the observation that if a strictly positive sequence $\{u_n\}$ converges to zero then the following sequence $\{(1 + u_n)^{1/u_n}\}$ converges to the Euler number e, we deduce that after a number of iterations of order $\sqrt{n}$ the achieved reduction of complementarity gap becomes

$$\frac{\mu^k}{\mu^0} = (1 - \beta/\sqrt{n})^{\sqrt{n}} \approx e^{-\beta},$$

and after $C \cdot \sqrt{n}$ iterations, the reduction achieves $e^{-C\beta}$. Therefore, for a sufficiently large constant C the complementarity gap can become arbitrarily small. In other words, after a number of iterations proportional to $\sqrt{n}$ the algorithm gets arbitrarily close to a solution. In terms of complexity theory the algorithm converges in $\mathcal{O}(\sqrt{n})$ iterations.

First, we need to prove a series of technical results.

Lemma 3.1 *If $(x, y, s) \in N_2(\theta)$ then the duality gap is equal to the complementarity gap, i.e.*

$$c^T x + \frac{1}{2}x^T Q x - (b^T y - \frac{1}{2}x^T Q x) = x^T s. \tag{10}$$

Proof. Any point in $N_2(\theta)$ is primal and dual feasible hence by substituting $Qx = A^T y + s - c$ and using $Ax = b$ we get

$$c^T x + \frac{1}{2}x^T Q x - (b^T y - \frac{1}{2}x^T Q x) = c^T x - b^T y + x^T Q x = x^T s.$$

□

Lemma 3.2 *The Newton direction $(\Delta x, \Delta y, \Delta s)$ defined by the equation system*

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I_n \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e - XSe \end{bmatrix}, \tag{11}$$

satisfies:

$$\Delta x^T \Delta s = \Delta x^T Q \Delta x \geq 0. \tag{12}$$

Proof: From the first two equations in (11) we get $A\Delta x = 0$ and $\Delta s = Q\Delta x - A^T\Delta y$ hence

$$\Delta x^T \Delta s = \Delta x^T (Q\Delta x - A^T \Delta y) = \Delta x^T Q \Delta x \geq 0,$$

where the last inequality follows from positive definiteness of Q. □

The Newton method uses a local linearization of the complementarity condition. When a step in the Newton direction $(\Delta x, \Delta y, \Delta s)$ of (11) is made, the error in the approximation of complementarity products is determined by the second-order term which is a product of Δx and Δs. A series of technical results in Lemmas 3.3–3.6 will provide a bound on $\|\Delta X \Delta S e\|$.

Lemma 3.3 *Let u and v be any two vectors in $\mathcal{R}^n$ such that $u^T v \geq 0$. Then*

$$\|UVe\| \leq 2^{-3/2}\|u+v\|^2,$$

where $U = diag\{u_1, \cdots, u_n\}$, $V = diag\{v_1, \cdots, v_n\}$.

Proof: Let us partition all products $u_j v_j$ into positive and negative ones: $\mathcal{P} = \{j \,|\, u_j v_j \geq 0\}$ and $\mathcal{M} = \{j \,|\, u_j v_j < 0\}$ and observe that

$$0 \leq u^T v = \sum_{j\in\mathcal{P}} u_j v_j + \sum_{j\in\mathcal{M}} u_j v_j = \sum_{j\in\mathcal{P}} |u_j v_j| - \sum_{j\in\mathcal{M}} |u_j v_j|.$$

We can now write

$$\begin{aligned}
\|UVe\| &= (\|[u_j v_j]_{j\in\mathcal{P}}\|^2 + \|[u_j v_j]_{j\in\mathcal{M}}\|^2)^{1/2} \\
&\leq (\|[u_j v_j]_{j\in\mathcal{P}}\|_1^2 + \|[u_j v_j]_{j\in\mathcal{M}}\|_1^2)^{1/2} \\
&\leq (2\|[u_j v_j]_{j\in\mathcal{P}}\|_1^2)^{1/2} \\
&\leq \sqrt{2}\|[\frac{1}{4}(u_j+v_j)^2]_{j\in\mathcal{P}}\|_1 \\
&= 2^{-3/2}\sum_{j\in\mathcal{P}}(u_j+v_j)^2 \\
&\leq 2^{-3/2}\sum_{j=1}^{n}(u_j+v_j)^2 \\
&= 2^{-3/2}\|u+v\|^2,
\end{aligned}$$

which completes the proof. □

Lemma 3.4 *If $(x,y,s) \in N_2(\theta)$ for some $\theta \in (0,1)$, then $(1-\theta)\mu \leq x_j s_j \leq (1+\theta)\mu \;\forall j$. In other words, $\min_{j\in\{1,2,\ldots,n\}} x_j s_j \geq (1-\theta)\mu$ and $\max_{j\in\{1,2,\ldots,n\}} x_j s_j \leq (1+\theta)\mu$.*

Proof: Since $\|x\|_\infty \leq \|x\|$, from the definition of the neighbourhood (9) we conclude

$$\|XSe - \mu e\|_\infty \leq \|XSe - \mu e\| \leq \theta\mu.$$

Hence

$$|x_j s_j - \mu| \leq \theta\mu \quad \forall j,$$

from which the result easily follows. □

Lemma 3.5 *If $(x, y, s) \in N_2(\theta)$ for some $\theta \in (0, 1)$, then*

$$\|XSe - \sigma\mu e\|^2 \leq \theta^2\mu^2 + (1-\sigma)^2\mu^2 n.$$

Proof: Observe first that $e^T(XSe - \mu e) = x^T s - \mu e^T e = n\mu - n\mu = 0$. Therefore

$$\begin{aligned} \|XSe - \sigma\mu e\|^2 &= \|(XSe - \mu e) + (1-\sigma)\mu e\|^2 \\ &= \|XSe - \mu e\|^2 + 2(1-\sigma)\mu e^T(XSe - \mu e) + (1-\sigma)^2\mu^2 e^T e \\ &\leq \theta^2\mu^2 + (1-\sigma)^2\mu^2 n. \end{aligned}$$

□

Lemma 3.6 *If $(x, y, s) \in N_2(\theta)$ for some $\theta \in (0, 1)$, then*

$$\|\Delta X \Delta S e\| \leq \frac{\theta^2 + n(1-\sigma)^2}{2^{3/2}(1-\theta)}\mu.$$

Proof: The third equation in the Newton system has the following form

$$S\Delta x + X\Delta s = -XSe + \sigma\mu e.$$

Having multiplied it with $(XS)^{-1/2}$, we obtain

$$X^{-1/2}S^{1/2}\Delta x + X^{1/2}S^{-1/2}\Delta s = (XS)^{-1/2}(-XSe + \sigma\mu e).$$

By applying Lemma 3.3 for $u = X^{-1/2}S^{1/2}\Delta x$ and $v = X^{1/2}S^{-1/2}\Delta s$ (with $u^T v \geq 0$ from Lemma 3.2) we get

$$\begin{aligned} \|\Delta X \Delta S e\| &= \|(X^{-1/2}S^{1/2}\Delta X)(X^{1/2}S^{-1/2}\Delta S)e\| \\ &\leq 2^{-3/2}\|X^{-1/2}S^{1/2}\Delta x + X^{1/2}S^{-1/2}\Delta s\|^2 \\ &= 2^{-3/2}\|X^{-1/2}S^{-1/2}(-XSe + \sigma\mu e)\|^2 \\ &= 2^{-3/2}\sum_{j=1}^{n}\frac{(-x_j s_j + \sigma\mu)^2}{x_j s_j} \\ &\leq 2^{-3/2}\frac{\|XSe - \sigma\mu e\|^2}{\min_j x_j s_j} \\ &\leq \frac{\theta^2 + n(1-\sigma)^2}{2^{3/2}(1-\theta)}\mu, \end{aligned}$$

where the expressions in the numerator and the denominator of the last inequality come from Lemmas 3.5 and 3.4, respectively. □

Recall that the required reduction of the complementarity gap satisfies $\sigma = 1 - \beta/\sqrt{n}$. Therefore $n(1-\sigma)^2 = \beta^2$ and by a choice of sufficiently small constants, for example, $\theta = 0.1$ and $\beta = 0.1$ we can easily guarantee that

$$\frac{\theta^2 + n(1-\sigma)^2}{2^{3/2}(1-\theta)} \leq \frac{1}{2}\sigma\theta,$$

hence the bound obtained in Lemma 3.6 may be simplified to

$$\|\Delta X \Delta S e\| \leq \frac{1}{2}\sigma\theta\mu. \tag{13}$$

The technical results proved so far will be used to show that if the current iterate belongs to the $N_2(\theta)$ neighbourhood of the central path and $\sigma = 1 - \beta/\sqrt{n}$ (with $\theta = 0.1$ and $\beta = 0.1$) then a full Newton step is feasible and the new iterate $(\bar{x}, \bar{y}, \bar{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$ also belongs to the $N_2(\theta)$ neighbourhood of the central path. Actually, we will show an even stronger result that for any step $\alpha \in (0, 1]$ in the Newton direction the new iterate

$$(x(\alpha), y(\alpha), s(\alpha)) = (x, y, s) + \alpha(\Delta x, \Delta y, \Delta s) \tag{14}$$

is primal-dual feasible and belongs to the $N_2(\theta)$ neighbourhood.

Consider the third equation in the Newton system (11). The summation of its components gives

$$e^T(S\Delta x + X\Delta s) = s^T\Delta x + x^T\Delta s = -e^T XSe + \sigma\mu e^T e = -x^T s \cdot (1 - \sigma).$$

Using (8), observe that when a step α is made in the Newton direction the new complementarity gap becomes $x(\alpha)^T s(\alpha) = (1-\alpha(1-\sigma))x^T s + \alpha^2 \Delta x^T \Delta s$ and therefore the corresponding average complementarity gap is

$$\mu(\alpha) = x(\alpha)^T s(\alpha)/n = (1 - \alpha(1 - \sigma))\mu + \alpha^2 \Delta x^T \Delta s/n. \tag{15}$$

Writing the third equation in the Newton system for a single component $j \in \{1, 2, \ldots, n\}$ gives $s_j\Delta x_j + x_j\Delta s_j = \sigma\mu - x_j s_j$ hence

$$\begin{aligned} x_j(\alpha)s_j(\alpha) - \mu(\alpha) &= (x_j + \alpha\Delta x_j)(s_j + \alpha\Delta s_j) - \mu(\alpha) \\ &= x_j s_j + \alpha(s_j\Delta x_j + x_j\Delta s_j) + \alpha^2\Delta x_j\Delta s_j - \mu(\alpha) \\ &= (1-\alpha)x_j s_j + \alpha\sigma\mu + \alpha^2\Delta x_j\Delta s_j - (1-\alpha)\mu - \alpha\sigma\mu - \alpha^2\Delta x^T\Delta s/n \\ &= (1-\alpha)(x_j s_j - \mu) + \alpha^2(\Delta x_j\Delta s_j - \Delta x^T\Delta s/n). \end{aligned}$$

Consequently, the proximity measure for point $(x(\alpha), y(\alpha), s(\alpha))$ becomes

$$\|X(\alpha)S(\alpha)e - \mu(\alpha)e\| \leq (1-\alpha)\|XSe - \mu e\| + \alpha^2\|\Delta X\Delta S e - (\Delta x^T\Delta s/n)e\|. \tag{16}$$

Lemma 3.7 *Let (x, y, s) be the current iterate in $N_2(\theta)$ neighbourhood and $(\Delta x, \Delta y, \Delta s)$ be the Newton direction which solves equation system (11). Set $\theta = \beta = 0.1$ and define the new iterate (14) after a step α in this direction and the new average complementarity gap (15). For any $\alpha \in (0, 1]$*

$$\|X(\alpha)S(\alpha)e - \mu(\alpha)e\| \quad \leq \quad \theta\mu(\alpha). \tag{17}$$

Proof: Expanding the square and using $e^T e = n$ and $(\Delta X\Delta S e)^T e = \Delta x^T\Delta s$ we get

$$\|\Delta X\Delta S e - (\Delta x^T\Delta s/n)e\|^2 = \|\Delta X\Delta S e\|^2 - \frac{1}{n}(\Delta x^T\Delta s)^2 \leq \|\Delta X\Delta S e\|^2. \tag{18}$$

Next, using (16), the definition of $N_2(\theta)$, and inequalities (18) and (13) we write

$$\|X(\alpha)S(\alpha)e - \mu(\alpha)e\| \;\leq\; (1-\alpha)\theta\mu + \alpha^2\|\Delta X\Delta S e\| \;\leq\; (1-\alpha)\theta\mu + \frac{1}{2}\alpha^2\sigma\theta\mu.$$

Since $\alpha \in (0,1]$ implies $\frac{1}{2}\alpha^2 \leq \alpha$ and Lemma 3.2 guarantees that $\Delta x^T \Delta s \geq 0$ we further write

$$\begin{aligned} \|X(\alpha)S(\alpha)e - \mu(\alpha)e\| &\leq (1-\alpha)\theta\mu + \frac{1}{2}\alpha^2\sigma\theta\mu \\ &\leq \theta(1-\alpha(1-\sigma))\mu \\ &\leq \theta[(1-\alpha(1-\sigma))\mu + \alpha^2\Delta x^T\Delta s/n] \\ &= \theta\mu(\alpha), \end{aligned}$$

which completes the proof. □

It is easy to prove that the Newton step in a feasible algorithm preserves the feasibility of primal and dual equality constraints; hence Lemma 3.7 guarantees that for any $\alpha \in (0,1]$ the new iterate (14) belongs to the $N_2(\theta)$ neighbourhood of the central path as well. We will assume that the maximum stepsize $\alpha = 1$ is chosen and full Newton step is made. Therefore the new iterate is defined as $(\bar{x}, \bar{y}, \bar{s}) = (x, y, s) + (\Delta x, \Delta y, \Delta s)$. Setting $\alpha = 1$ in (15) gives

$$\bar{\mu} = \sigma\mu + \Delta x^T \Delta s/n.$$

Using Cauchy-Schwartz inequality and inequality (13) we derive the following bound

$$\Delta x^T \Delta s = (\Delta X \Delta S e)^T e \leq \|\Delta X \Delta S e\|\|e\| \leq \frac{1}{2}\theta\sigma\mu\sqrt{n}$$

hence

$$\bar{\mu} = \sigma\mu + \Delta x^T\Delta s/n \leq \sigma\mu + \frac{1}{2\sqrt{n}}\theta\sigma\mu \leq (1 + \frac{\theta}{2\sqrt{n}})\sigma\mu.$$

With our choice of $\sigma = 1 - \beta/\sqrt{n}$ and $\theta = \beta = 0.1$ we deduce that $(1 + \frac{\theta}{2\sqrt{n}})\sigma \leq \bar{\sigma} = 1 - \frac{\beta}{2\sqrt{n}}$ and

$$\bar{\mu} \leq (1 - \frac{\beta}{2\sqrt{n}})\mu = \bar{\sigma}\mu. \tag{19}$$

We are now ready to prove the complexity result.

Theorem 3.1 *Given $\epsilon > 0$, suppose that a feasible starting point $(x^0, y^0, s^0) \in N_2(0.1)$ satisfies $(x^0)^T s^0 = n\mu^0$, where $\mu^0 \leq 1/\epsilon^\kappa$, for some positive constant κ. Then there exists an index L with $L = \mathcal{O}(\sqrt{n}\ln(1/\epsilon))$ such that $\mu^l \leq \epsilon, \quad \forall l \geq L$.*

Proof: Let l be the iteration index. We have shown that $\mu^{l+1} \leq \bar{\sigma}\mu^l$. Taking logarithms of both sides of this inequality we obtain $\ln\mu^{l+1} \leq \ln\bar{\sigma} + \ln\mu^l$. By repeatedly applying this formula for $l = 0, 1, 2, ...$ and using $\mu^0 \leq 1/\epsilon^\kappa$, we get

$$\ln\mu^l \leq l\ln\bar{\sigma} + \ln\mu^0 \leq l\ln(1 - \beta/(2\sqrt{n})) + \kappa\ln(1/\epsilon).$$

From a property of logarithmic function we have $\ln(1 - \beta/(2\sqrt{n})) \leq -\beta/(2\sqrt{n})$. Thus

$$\ln\mu^l \leq l(-\beta/(2\sqrt{n})) + \kappa\ln(1/\epsilon).$$

To satisfy $\mu^l \leq \epsilon$, we need:

$$l(-\beta/(2\sqrt{n})) + \kappa\ln(1/\epsilon) \leq \ln\epsilon.$$

This inequality holds for any $l \geq L$, where

$$L = 2\frac{\kappa+1}{\beta} \cdot \sqrt{n} \cdot \ln(1/\epsilon). \tag{20}$$

□

This excellent worst-case complexity result of IPM for linear and quadratic programming is unequalled in the field of optimization. Two aspects of it are worth giving particular attention. Firstly, the number of iterations is bounded by the square root of the problem dimension n and this needs to be compared with a number of iterations that the simplex method might have to perform that is possibly exponential in n. The computational experience of [22] shows the much better practical iteration complexity achieved by interior point methods: it displays a logarithmic dependence on the problem dimension. Secondly, the complexity result (20) displays merely a logarithmic dependence $\mathcal{O}(\ln(1/\epsilon))$ on the required termination accuracy ϵ. There has been recently growing interest in gradient methods [84] which can only ensure $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}(1/\epsilon^2)$ terms in their worst-case complexity results. Although they display fast initial progress to optimality they become slow if a high accuracy of solution (small ϵ) is requested. Indeed, the terms $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}(1/\epsilon^2)$ in the complexity result seem acceptable when a rough approximation of solution is sought with $\epsilon = 10^{-1}$ or $\epsilon = 10^{-2}$ but they become a liability when more accurate solutions are needed. The superiority of IPMs becomes obvious when more accurate solutions are requested.

4 From Theory to Implementation of IPMs

In order to keep the complexity proof easy (and accessible to undergraduate students of optimization) we have made several simplifying assumptions. Many of them may be dropped without noticeably penalising the worst-case complexity result, however, unfortunately they make the proofs more tedious. Modern implementations of IPMs further relax quite a few of these unnecessary restrictions in the quest to make the algorithms as efficient as possible when used in practice.

Feasible vs Infeasible Methods. In practice there is no need to impose feasibility of all iterates and force the algorithm to stay in the primal-dual feasible set $\mathcal{F}^0$. For infeasible points all three terms in the right-hand side of the equation defining Newton step (7) are nonzero. This does not add any extra cost when solving the equation. By exploiting the decomposition of the right-hand side

$$(\xi_p, \xi_d, \xi_\mu) = (\xi_p, 0, 0) + (0, \xi_d, 0) + (0, 0, \xi_\mu)$$

the direction $\Delta = (\Delta x, \Delta y, \Delta s)$ may be decomposed accordingly into $\Delta = \Delta_p + \Delta_d + \Delta_\mu$, where Δ_p and Δ_d are terms responsible for restoring primal and dual feasibility, respectively and Δ_μ is the "optimizing" term. In the feasible algorithm $\Delta_p = \Delta_d = 0$ and $\Delta = \Delta_\mu$ is determined by equation (11).

It is easy to show that if a step α is made in the Newton direction in the infeasible algorithm then primal and dual infeasibilities are reduced $1 - \alpha$ times. Indeed, the first and second equations in (6) are linear hence fast progress in reducing the residuals in these equations is expected. In particular, if the full step is feasible and $\alpha = 1$ then all infeasibilities are absorbed in one step.

In practice it is common [22] that primal and dual feasibility is reached before the optimality criteria is met. However, working with an infeasible IPM usually requires damping the Newton step and accepting stepsizes α which are significantly smaller than 1 (cf. Wright [105, Ch. 6]).

Aggressive Optimization. The theory of the short-step feasible algorithm allows only a small reduction in the complementarity gap at each iteration with a factor $\sigma = 1 - \beta/\sqrt{n}$ which is very close to 1. Practical algorithms allow σ to be any number from the interval $(0, 1]$ and indeed, the author's experience [22] shows that the average reduction of the complementarity gap achieved in each IPM iteration $\sigma_{average}$ is usually in the interval $(0.1, 0.5)$. Using small values of σ such as $\sigma \in (0.01, 0.1)$ sets up aggressive targets on reducing the complementarity gap (and optimality gap) and usually causes the Newton direction to point towards infeasible solutions. To preserve the nonnegativity of the new iterates x and s the Newton step needs to be damped in such case. A fraction-to-the-boundary rule is used to preserve the strict positivity of x and s. This is achieved by using a factor α_0 close to but strictly smaller than 1 to reduce the stepsizes in the primal and dual spaces.

Centrality. The theory requires the algorithm to keep all iterates in the (very small) $N_2(\theta)$ neighbourhood of the central path. Practical algorithms use a symmetric neighbourhood $N_S(\gamma) = \{(x, y, s) \in \mathcal{F}^0 \,|\, \gamma\mu \leq x_j s_j \leq 1/\gamma\mu, \forall j\}$, where $\gamma \in (0, 1)$ or a so-called infinity neighbourhood $N_{-\infty}(\gamma) = \{(x, y, s) \in \mathcal{F}^0 \,|\, \gamma\mu \leq x_j s_j, \forall j\}$ in which only too small complementarity products are forbidden. These wider neighbourhoods leave the algorithm with more room to manoeuver. However, it is no longer possible to prove that full Newton step keeps the iterates in such neighbourhoods and therefore Newton step needs to be damped (cf. Wright [105, Ch. 5]). Still the price of having to accept the reduced stepsizes is worth paying because the algorithms which use $N_S(\gamma)$ or $N_{-\infty}(\gamma)$ neighbourhoods perform very well in practice.

Predictor-Corrector and Centrality Corrector Techniques. "Centrality" is understood as keeping well-balanced the complementarity products of all primal-dual iterates, and it plays an essential role in both the theory and implementation of IPMs. The theoretical developments in the previous section focused on ensuring that $x_j s_j \approx \mu$, $\forall j \in \{1, 2, \ldots, n\}$. The target-following approach developed by Jansen et al. [57] provides theoretical background to a more relaxed treatment of centrality. In the analysis presented in Section 3 the uniform distribution of products $x_j s_j$ was achieved by containing (x, y, s) within the $N_2(\theta)$ neighbourhood which required special care in the proof to be devoted to the second-order term $\Delta X \Delta S e$ (Lemma 3.6 and inequality (13)). The implementation of IPM very cleverly exploits this error term.

Mehrotra's predictor-corrector technique [78, 73] splits the computation of the Newton direction when solving system (7) into two steps. In the first one, the right-hand side ignores centrality and sets $\sigma = 0$ which corresponds to attempting to reach complementarity and optimality in just one shot. The corresponding term of direction $\Delta^{pred} = (\Delta x, \Delta y, \Delta s)$ focuses on optimization but neglects centrality and it is used to predict how much progress in reducing the complementarity gap and infeasibilities may be achieved. This term of direction corresponds to the right-hand side $(\xi_p, \xi_d, -XSe)$ in (7) and is called the *predictor*. If a full step in this direction was made then the new complementarity product would be

$$(X + \Delta X)(S + \Delta S)e = XSe + (S\Delta X + X\Delta S)e + \Delta X \Delta S e = \Delta X \Delta S e$$

and obviously could be very different from the required value $\sigma\mu e$. Therefore, the *corrector* term Δ^{corr} is computed by solving an equation similar to (11), but with the new right-hand side $(0, 0, \sigma\mu e - \Delta X^{pred} \Delta S^{pred} e)$ and eventually the predictor and corrector terms are combined into

a Newton direction

$$\Delta = \Delta^{pred} + \Delta^{corr}.$$

Centrality correctors [42, 22] use targets which are less aggressive (but hopefully easier to reach). Rather than attempting to take all complementarity products towards the same value $\mu^{new} = \sigma\mu^{old}$, centrality correctors aim at bringing them all to the interval $[\gamma\mu^{new}, 1/\gamma\mu^{new}]$ corresponding to the symmetric neighbourhood $N_S(\gamma)$ of the central path. Technically this is done by computing a corrector term which solves an equation like (11), but for the right-hand side equal to $(0, 0, t - \Delta X^{pred}\Delta S^{pred}e)$, where t is the projection of the hypothetical complementarity product $(X + \Delta X^{pred})(S + \Delta S^{pred})e$ onto the hypercube $[\gamma\mu^{new}e, 1/\gamma\mu^{new}e]$. Unlike Mehrotra's predictor-corrector technique which allows only one corrector term to be computed, centrality correctors technique can be applied recursively, that is, a direction $\Delta = \Delta^{pred} + \Delta^{corr}$ may become a new predictor term and the process of computing the corrector may be repeated. Extensive numerical tests [42, 22] provided evidence that the use of multiple centrality correctors leads to significant reduction of the number of IPM iterations to reach optimality. Indeed, the method is widely used by academic IPM solvers such as BPMPD, HOPDM, OOPS, OOQP, PCx, as well as commercial IPM solvers such as Cplex, Mosek and Xpress. (It has also been used with success in the context of semidefinite programming with IPMs [52].)

IPMs are sensitive to the choice of an initial point. A general advice is to use a point which is sufficiently far away from the boundary of the positive orthant, that is, for which x and s are "not too small" [4, 38, 78]. For more recent attempts to design a good starting point and the review of existing alternative approaches, the reader should consult D'Apuzzo et al. [25] and the references therein.

Finally, the practical efficiency of IPMs strongly depends on the linear algebra techniques which are employed to solve the Newton equation system: either (11) if we use a feasible algorithm of theoretical interest or (7) if we use a practical infeasible algorithm. Multiple linear systems with different right-hand sides may have to be solved when predictor-corrector technique [78, 73] or centrality correctors technique [42, 22] are used. The linear algebra aspects of IPM implementation are addressed in the next section.

5 Linear Algebra of IPMs

The Newton equation system—whether (11) if we use a feasible algorithm or (7) if we use an infeasible algorithm—involves matrices A and Q and two diagonal matrices X and S which change at every iteration. The whole system has dimension $(2n + m) \times (2n + m)$ and because of the presence of several zero blocks and diagonal matrices, it displays an interesting and exploitable sparsity pattern. Additionally, if an optimization problem is large then the matrices A and Q themselves are expected to be sparse. The system (7) needs to be solved for at least one but usually (when correctors are used) for several different values of right-hand side vectors. It is a common approach to eliminate $\Delta s = X^{-1}(\xi_\mu - S\Delta x)$ from (7) and get the following symmetric but indefinite augmented system

$$\begin{bmatrix} -Q - \Theta^{-1} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f \\ d \end{bmatrix} = \begin{bmatrix} \xi_d - X^{-1}\xi_\mu \\ \xi_p \end{bmatrix}, \tag{21}$$

where $\Theta = XS^{-1}$. If the elimination process is continued and $\Delta x = (Q+\Theta^{-1})^{-1}(A^T\Delta y - f)$ is substituted into the second equation in (21) then the following symmetric and positive definite normal equations system is obtained

$$(A(Q+\Theta^{-1})^{-1}A^T)\Delta y \;=\; g = A(Q+\Theta^{-1})^{-1}f + d. \tag{22}$$

The inversion of $(Q+\Theta^{-1})$ in this formulation is a highly questionable operation because it is well-known [29] that an inverse of a sparse matrix may be dense. When linear optimization problems ($Q=0$) or separable quadratic optimization problems in which Q is a diagonal matrix are solved the operation $(Q+\Theta^{-1})^{-1}$ produces a diagonal matrix and the normal equations system (22) is usually the preferable (and default) option. Whenever matrix Q has a more complicated structure, the augmented system (21) is the best option. It is worth mentioning that even in the case of $Q=0$ certain sparsity patterns of the LP constraint matrix such as the presence of dense columns may adversely affect the normal equations approach and make it inefficient [5, 29]. After all, even in the case when the (1,1) block of the augmented system is diagonal, (22) is obtained from (21) by employing a particular pivot order, namely by pivoting on the (1,1) block first. In general, using the augmented system formulation (21) allows for more freedom in the choice of pivot order and it is a safer option.

Using interior point methods to solve nonlinear optimization problems requires dealing with systems similar to (21) except that matrices A and Q depend on the current primal-dual point (x,y,s) and therefore change at every iteration. This again strongly advocates for the use of formulation (21) which offers more flexibility in choosing the pivot order and accommodating varying sparsity patterns of A and Q. For linear and quadratic problems, Mészáros [80] performed an interesting analysis and developed a heuristic to identify which of the two alternative approaches (21) or (22) is likely to produce a more sparse symmetric decomposition [81].

Normal equations formulation (22) has an important advantage: its matrix is symmetric and positive definite hence eligible for Cholesky decomposition [29]. The matrix in the augmented system formulation (21) is indefinite hence its symmetric decomposition LDL^T may require 2×2 pivots in D, see [15]. More importantly, when indefinite matrices are factored it is in general impossible to separate the sparsity analysis from the numerical phase and this substantially increases the cost of decomposition [29]. Vanderbei [102] observed that by adding a diagonal term to the (2,2) block, the matrix appearing in (21) can be transformed to a quasidefinite one. Quasidefinite matrices are strongly factorisable, that is, a decomposition LDL^T with diagonal D exists for any symmetric row and column permutation applied to the matrix. Consequently, there is no need to use 2×2 pivots and the sparsity analysis can be separated from the numerical decomposition.

To produce a diagonal term in the (2,2) block Vanderbei added artificial variables to all constraints [102]. Saunders and Tomlin [93, 94] achieved a similar result by adding Tikhonov-type regularization terms to the original quadratic problem

$$\begin{aligned} \min\;\; c^Tx + \frac{1}{2}x^TQx + \frac{\rho^2}{2}x^Tx + \frac{1}{2}p^Tp & \\ \text{s.t.}\;\; Ax + \delta p = b, & \\ x \geq 0,\; p \text{ free}; & \end{aligned} \tag{23}$$

where ρ and δ are (small) positive terms and $p \in \mathcal{R}^m$ is a free variable. The corresponding augmented system for (23) has the following form

$$\begin{bmatrix} -(Q+\Theta^{-1}+\rho^2 I) & A^T \\ A & \delta^2 I \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} f' \\ d' \end{bmatrix}, \tag{24}$$

where $f^{'} \in \mathcal{R}^n$, $d^{'} \in \mathcal{R}^m$ are appropriately computed right-hand-side vectors. The presence of diagonal terms in (1,1) and (2,2) blocks guarantees the boundedness of condition number of the matrix involved.

The primal-dual regularization approach proposed in [3] replaces the primal-dual pair (1) by two problems: an approximate primal and an approximate dual:

$$\begin{array}{lll} \min & c^Tx+\frac{1}{2}x^TQx+\frac{1}{2}(x-x_0)^TR_p(x-x_0) & \\ \text{s.t.} & Ax=b, & \\ & x\geq 0; & \end{array} \qquad \begin{array}{lll} \max & b^Ty-\frac{1}{2}x^TQx-\frac{1}{2}(y-y_0)^TR_d(y-y_0) & \\ \text{s.t.} & A^Ty+s-Qx=c, & \\ & y \text{ free}, \;\; s\geq 0, & \end{array} \tag{25}$$

where primal and dual positive definite diagonal regularization matrices $R_p \in \mathcal{R}^{n\times n}$ and $R_d \in \mathcal{R}^{m\times m}$ and primal and dual reference points in proximal terms $x_0 \in \mathcal{R}^n$ and $y_0 \in \mathcal{R}^m$, respectively can be chosen dynamically. The resulting regularized augmented system has the following form

$$\left[\begin{array}{cc} -(Q+\Theta^{-1}+R_p) & A^T \\ A & R_d \end{array} \right] \left[\begin{array}{c} \Delta x \\ \Delta y \end{array} \right] = \left[\begin{array}{c} f^{''} \\ d^{''} \end{array} \right], \tag{26}$$

where $f^{''} \in \mathcal{R}^n$, $d^{''} \in \mathcal{R}^m$ are appropriately computed right-hand-side vectors. The terms R_p and R_d are chosen dynamically during the factorization and can be kept small except for pivots which are dangerously close to zero and potentially unstable. Therefore, the perturbation introduced using primal-dual regularization [3] is usually much smaller than the uniform diagonal matrices $\rho^2 I$ and $\delta^2 I$ added in Saunders' approach [93]. Computational evidence was provided in [3] that primal-dual regularization cures instability without adversely affecting the fast convergence of the primal-dual method. A further insight into the primal-dual regularization method is provided in a recent paper of Friedlander and Orban [36].

Under the condition that 2×2 pivots are never required, the computation of a symmetric factorization of the augmented system matrix (21) can benefit from all advantageous techniques known for the computation of a sparse Cholesky factorization of the positive definite matrix [29]. All state of the art implementations of interior point methods developed in the 90's used direct methods [4, 74]. To achieve maximum efficiency the implementations of sparsity exploiting symmetric factorization of matrix (21) or (22) make use of numerous advanced features of modern computers. For example, they exploit loop unrolling, adjust the sizes of supernodes to maximize the usage of available cache memory and, most recently, exploit new features offered by multi-core processors. All these improvements play very important roles in the efficiency of codes. They do not rely on *algorithmic* features of interior point methods but exploit implementation tricks and are therefore beyond the scope of this paper. We refer the reader to Mészáros [79] for an example of such a study. In the coming years we may expect further specializations which will employ graphical processor units to perform very fast numerical computations in the linear algebra kernel.

Cholesky decomposition and the symmetric decomposition LDL^T which involves only 1×1 pivots (but does not require to use 2×2 pivots) can be relatively easily parallelised because the sparsity structure analysis is performed only once to determine pivot order and the numerical phase of the decomposition uses static data structures. To better exploit modern computer architecture the numerical operations are performed on small blocks which fit into cache memory. Viewed from this perspective, the symmetric factorization becomes a sequence of block-matrix operations. It is therefore natural to expect that if matrices A and Q are block-structured then their block-sparsity pattern may be exploited to determine a block-pivot order in which the eliminations should be performed. Several specialized implementations of IPMs for particular

block-structured matrices have been developed in the 90's, for example [12, 21, 49, 56, 58, 96, 100] to mention a few. All these papers describe specialized algorithms, each exploiting *one* particular structure of the constraint matrix. Some of these developments [49, 58, 96] also present dedicated parallel implementations.

In a series of papers Gondzio, Grothey and Sarkissian [44, 45, 46] have demonstrated that using modern object-oriented programming techniques it is possible to exploit *any* block matrix structure in the implementation of interior point method. The resulting software, Object-Oriented Parallel Solver, OOPS[1] has been applied to solve various optimization problems arising in finance, telecommunications, data mining, electricity transmission and utility distribution applications. Very large problems with sizes reaching 10^9 variables [44] can be handled by IPMs if structure is exploited in the linear algebra kernel.

A closer look at the matrix $\Theta = XS^{-1}$ which appears in the augmented system (21) and normal equation system (22) reveals that the elements of this matrix display an unwelcome feature. Indeed, when IPM approaches the optimal solution, the primal variables and dual slacks converge to their optimal values and (for a linear program) converge to a strongly complementary pair [105, p. 27]:

$$\begin{array}{llllll} x_j \to x_j^* > 0 & \text{and} & s_j \to s_j^* = 0, & \text{for} & j \in \mathcal{B}, \\ x_j \to x_j^* = 0 & \text{and} & s_j \to s_j^* > 0, & \text{for} & j \in \mathcal{N}. \end{array} \tag{27}$$

Consequently, the elements $\theta_j, j \in \mathcal{B}$ are very large and display $\mathcal{O}(\mu^{-1})$ dependence, while the elements $\theta_j, j \in \mathcal{N}$ are very small and display $\mathcal{O}(\mu)$ dependence. This has disastrous consequences for the conditioning of matrices in (21) and (22) and makes the solution of these systems challenging. This feature of linear systems arising in IPMs is very clearly seen in the case of linear programming, when $Q = 0$. The situation gets more complicated for quadratic problems because there may exist pairs which are not strictly complementary. This further complicates the partitioning (27) and adds a third cluster of elements in Θ which are $\mathcal{O}(1)$.

Interestingly, the ill-conditioning of matrices turns out to be rather benign as long as *feasible* problems are solved. This is a consequence of a particular structure of the right-hand side vector in (21). This surprising result goes back to Dikin [27] (see also Stewart [97]) and guarantees that for feasible LPs the solution of (22) is bounded irrespective of the spread of Θ. Computational experience in the 90's [4] confirmed that *direct* methods provide sufficiently accurate solutions for IPMs to converge quickly regardless the ill-conditioning of (21).

The situation is completely different when *iterative* approaches such as Krylov-subspace methods [65] are applied to (21) or (22). The ill-conditioning of Θ and the resulting ill-conditioning of matrices involved makes these systems intractable. Iterative methods are powerful when applied to well-conditioned systems or when suitable preconditioners are available for ill-conditioned systems. For a long time there was a consensus among the interior point community that there were no obvious preconditioners for linear systems arising in IPM algorithms and therefore the application of iterative methods was not really seriously considered. For at least two reasons this view started to change in late 90's. Firstly, the problems got larger and direct methods (however successful) started to show their limitations. Occasionally, even for very sparse problems with matrices A and Q having merely a few nonzero elements per column the factorizations of (21) and (22) display very significant fill-in and become prohibitively expensive. Secondly, a better understanding of the conditioning of matrices arising in IPMs encouraged researchers to look into a design of special preconditioners.

[1] `http://www.maths.ed.ac.uk/~gondzio/parallel/solver.html`

The first successful preconditioners were developed for specially structured problems arising in network optimization [91] (see also [59, 35] for more recent developments). They exploited the particularities of node-arc incidence matrices and an ability to represent simplex bases by means of spanning trees. A class of preconditioners applicable in a general LP context was analysed by Gill et al. [39], Oliveira [87] and Oliveira and Sorensen [88]. Their preconditioners use the partitioning (27) to determine a splitting of elements of Θ into two sets of "small" and "large" ones and employ it to guess a basis matrix B which then becomes the key object in the definition of the preconditioner $P = EE^T$ for the augmented system matrix H in (21). Suppose a basic-nonbasic partition of $A = [B|N]$ deduced from a $(\mathcal{B},\mathcal{N})$ partition of the set $\{1, 2, ..., n\}$ is available, and B is a nonsingular matrix. The symmetric application of the preconditioner [88] to the augmented system matrix from the left and from the right gives:

$$E^{-1}HE^{-T} = \left[\begin{array}{cc|c} \Theta_B^{1/2} & & \Theta_B^{-1/2}B^{-1} \\ & \Theta_N^{1/2} & \\ \hline \Theta_B^{1/2} & & \end{array}\right] \left[\begin{array}{cc|c} -\Theta_B^{-1} & & B^T \\ & -\Theta_N^{-1} & N^T \\ \hline B & N & 0 \end{array}\right] \left[\begin{array}{cc|c} \Theta_B^{1/2} & & \Theta_B^{1/2} \\ & \Theta_N^{1/2} & \\ \hline B^{-T}\Theta_B^{-1/2} & & \end{array}\right]$$
$$= \left[\begin{array}{cc|c} I_m & W^T & \\ W & -I_{n-m} & \\ \hline & & -I_m \end{array}\right],$$

where $W = \Theta_N^{1/2}N^TB^{-T}\Theta_B^{-1/2}$. Sufficiently close to an optimal solution (when $\mu \to 0$) we have $\Theta_B^{-1} = \mathcal{O}(\mu) \to 0$ and $\Theta_N = \mathcal{O}(\mu) \to 0$ giving $W = \Theta_N^{1/2}N^TB^{-T}\Theta_B^{-1/2} \approx 0$ hence the preconditioned matrix is strongly dominated by its diagonal. The preconditioner requires B to be nonsingular and relies on the assumption that such B can be found by considering a subset of columns of A which correspond to the largest elements of Θ. This task is actually much more difficult than it seems because the partitioning (27) does not imply that the subset of columns of A which correspond to $\mathcal{B}$ defines a full row rank matrix. Therefore an identification of a suitable basis matrix B is a nontrivial problem. Several techniques have been developed to deal with it [2, 14, 88]. The preconditioners which try to guess an optimal basis work well only in later iterations of the interior point method, that is, when the iterates approach the optimal partition (27). Different preconditioners are needed in earlier IPM iterations [14].

Oliveira and Sorensen [88] have shown that for every preconditioner of the normal equations, an equivalent preconditioner can be derived for the augmented system. However, the converse statement is not true. This is a strong argument for developing preconditioners for the augmented system. However, the matrix in (21) is indefinite and therefore the conjugate gradient algorithm [55] cannot be applied to this equation system. The developments of Lukšan and Vlček [72] and Rozložník and Simoncini [92] have provided conditions under which the conjugate gradient algorithm may be employed to solve symmetric indefinite systems such as (21).

Symmetric indefinite systems similar to (21) appear commonly in the context of partial differential equations as the *saddle point problem*. A vast literature exists which addresses iterative solution methods for such systems and the design of specialized preconditioners for them. The reader interested in such approaches should consult an excellent survey of Benzi, Golub and Liesen [8]. A plethora of preconditioners for symmetric indefinite systems arising in optimization have been proposed in the last decade [9, 10, 20, 28, 30, 64, 95]. Many of them belong to a broad class of constraint preconditioners [28].

Independently of these efforts to develop preconditioners for a general augmented system (21), several other clever preconditioners have been proposed recently for different specially structured

problems. Castro [18] designed a power-series preconditioner for the Schur complement matrix arising in the solution of multicommodity network flow problems and recently Castro and Cuesta [19] extended this approach to any linear or quadratic problem in which the constraint matrix has primal block-angular structure.

The use of iterative methods to solve systems such as (21) or (22) opens new questions regarding the theory of interior point methods. The key feature of iterative methods is a gradual (sometimes slow) reduction of the error in the equation system and this is very different from the case when direct methods are used and very accurate solutions are available. It is advantageous to interrupt iterative process early, obtain only a rough approximation of the solution, and cut off the tailing effect whenever possible. This opens a question how *inexact* solutions of Newton system may influence the convergence and the complexity estimates of the interior point method. The problem can be viewed and analysed in the general context of the inexact Newton method [26]. The inexact solution of (7) satisfies

$$\begin{bmatrix} A & 0 & 0 \\ -Q & A^T & I_n \\ S & 0 & X \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta s \end{bmatrix} = \begin{bmatrix} \xi_p \\ \xi_d \\ \xi_\mu \end{bmatrix} + \begin{bmatrix} r_p \\ r_d \\ r_\mu \end{bmatrix} \tag{28}$$

and admits errors r_p, r_d and r_μ in all three equations. Such analyses were performed for example by Bellavia [7], Mizuno and Jarre [82] and Cafieri et al. [17].

There are advantages in studying the behaviour of inexact Newton method together with the design of a particular preconditioner for (21) because this allows us to achieve a better understanding of the influence of errors on the performance of interior point methods and ultimately a tighter control of these errors. Recall that (7) is a linearization of (6) and the first two equations of (6) are linear. The matrix in the Newton equation system (28) contains several zero blocks and three diagonal matrices I_n, X and S, which allows for different manipulations of the error and absorbing some components of it at the expense of increasing the others. For example, it is easy to absorb any error r_d in (28) into Δs by setting $\Delta\bar{s} = \Delta s + r_d$ causing possibly an increase of the error in the third equation: $S\Delta x + X\Delta\bar{s} = \xi_\mu + r_\mu + Xr_d$. Such tricks have been used in the analysis of two particular preconditioners [1, 71].

Although so much effort has gone recently into the design of preconditioners for IPMs, in the author's opinion, we still lack an ultimate preconditioner. Indeed, the efficiency of iterative methods depends on the quality of the preconditioner and it is difficult to design a general-purpose preconditioner. In consequence, iterative methods used in IPMs are capricious and depend on fine-tuning. There is increased research activity in this area and it is natural to expect that it will produce new interesting developments. It is encouraging that recently there has been a noticeable shift of interest from direct to iterative methods, see the survey of D'Apuzzo et al. [24].

The current state of the art is to employ direct methods to solve (21) (or (22) for LPs and separable QPs) as long as problems are not too large and there is enough memory to store sparse symmetric factorization. The increasing size of problems to be solved will move the balance towards iterative methods. The only way to solve truly large problems with IPMs is to use an appropriately preconditioned iterative method. We will discuss one such attempt in the next section.

6 Matrix-Free Interior Point Method

Huge optimization problems may defy any algorithm simply because of the memory requirements. On the other hand, such problems do not always have to be solved to a high degree of accuracy. Therefore, fast algorithms are needed which could provide at least a rough approximation to the solution of the problem. This justifies the increased interest in gradient methods. Their worst-case complexity bounds display $\mathcal{O}(1/\epsilon)$ or $\mathcal{O}(1/\epsilon^2)$ dependence on the required accuracy ϵ [84] which is significantly worse than the $\mathcal{O}(\ln(1/\epsilon))$ dependence of interior point methods. However, gradient methods enjoy a low per-iteration cost and can solve some very large problems to one- or two-digit accuracy.

Below we briefly review a recent development of Gondzio [43] to redesign interior point methods to allow the solution of very large LPs and QPs with IPMs. In this new approach an inexpensive iterative method is used to solve the Newton equations (21) or (22) only approximately. Neither the Newton equation system nor the preconditioner for it is formulated explicitly. Matrices A and Q are used as operators only to deliver the results of matrix-vector multiplications Ax, A^Ty and Qx. This *matrix-free* feature of the approach is essential when one considers problems which may be too large to store, or when the problem data is not explicitly available.

Pivoting on the (1,1) block in the regularized augmented system (26) produces the following regularized normal equation matrix

$$G_R = A(Q + \Theta^{-1} + R_p)^{-1}A^T + R_d, \tag{29}$$

in which the presence of the nonzero primal regularization R_p guarantees an upper bound on the largest eigenvalue of G_R and the presence of the nonzero dual regularization R_d guarantees that the spectrum of G_R is bounded away from zero. Consequently, for appropriately chosen regularizations R_p and R_d the condition number of G_R is bounded, regardless of the conditioning of Θ. The proposed preconditioner $P \in \mathcal{R}^{m\times m}$ attempts to identify the largest eigenvalues of G_R and guarantee that $\kappa(P^{-1}G_R) \ll \kappa(G_R)$. A partial Cholesky decomposition of this regularized normal equation matrix is computed using complete pivoting, that is, choosing the largest (diagonal) pivots from G_R:

$$G_R = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}, \tag{30}$$

where $L = \begin{bmatrix} L_{11} \\ L_{21} \end{bmatrix}$ is a trapezoidal matrix which contains the first k columns of Cholesky factor of G_R, $D_L \in \mathcal{R}^{k\times k}$ is a diagonal matrix formed by the k largest pivots of G_R and $S \in \mathcal{R}^{(m-k)\times(m-k)}$ is the Schur complement obtained after eliminating k pivots. The trapezoidal matrix L is split into triangular matrix $L_{11} \in \mathcal{R}^{k\times k}$ and rectangular matrix $L_{21} \in \mathcal{R}^{(m-k)\times k}$ which contains the remaining part of columns of the partial Cholesky factor. With k equal to m we would obtain a complete Cholesky decomposition of $G_R = LD_LL^T$. However, we use a small number k ($k \ll m$) and interrupt the decomposition to determine only a partial Cholesky factorization.

The preconditioner for (29) is obtained by replacing the Schur complement matrix S in (30) with its diagonal D_S:

$$P = \begin{bmatrix} L_{11} & \\ L_{21} & I \end{bmatrix} \begin{bmatrix} D_L & \\ & D_S \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ & I \end{bmatrix}. \tag{31}$$

The preconditioner is determined without calculating off-diagonal entries of S. Only the diagonal and selected columns of the Schur complements are computed and this can be done using an *implicit* process in which neither G_R nor its Schur complements need to be fully formulated. The complete pivoting guarantees that the entries of diagonal matrices D_L and $D_S = diag(S)$ satisfy the following inequalities

$$\underbrace{d_1 \geq d_2 \geq \cdots \geq d_k}_{D_L} \;\geq\; \underbrace{d_{k+1} \geq d_{k+2} \geq \cdots \geq d_m}_{D_S}. \tag{32}$$

The greedy heuristic which places the largest pivots of G_R in D_L reduces the trace of the Schur complement at the fastest possible rate and it is expected to capture the cluster of the largest eigenvalues of G_R.

If nonseparable quadratic problems are solved and Q is a non-diagonal matrix then the reduction from the augmented system to normal equations should not be made. Instead, a preconditioner for the augmented system is constructed. First, all off-diagonal elements of Q are dropped and the matrix $\bar{Q} = \text{diag}\{Q\} + \Theta^{-1} + R_p$ is computed and used to determine (implicitly) $\bar{G}_R = A\bar{Q}^{-1}A^T + R_d$. Next, its partial Cholesky decomposition is computed

$$\bar{G}_R = \begin{bmatrix} \bar{L}_{11} & \\ \bar{L}_{21} & I \end{bmatrix} \begin{bmatrix} \bar{D}_L & \\ & \bar{S} \end{bmatrix} \begin{bmatrix} \bar{L}_{11}^T & \bar{L}_{21}^T \\ & I \end{bmatrix} \tag{33}$$

and again the matrix $\bar{S}$ is replaced with its diagonal to define an approximation of $\bar{G}_R$

$$\bar{P} = \begin{bmatrix} \bar{L}_{11} & \\ \bar{L}_{21} & I \end{bmatrix} \begin{bmatrix} \bar{D}_L & \\ & \bar{D}_S \end{bmatrix} \begin{bmatrix} \bar{L}_{11}^T & \bar{L}_{21}^T \\ & I \end{bmatrix} = \bar{L}\bar{D}\bar{L}^T \tag{34}$$

and, finally, the preconditioner for the augmented system (26) is determined

$$P_{aug} = \begin{bmatrix} I & \\ -A\bar{Q}^{-1} & \bar{L} \end{bmatrix} \begin{bmatrix} -\bar{Q} & \\ & \bar{D} \end{bmatrix} \begin{bmatrix} I & -\bar{Q}^{-1}A^T \\ & \bar{L}^T \end{bmatrix}. \tag{35}$$

In the construction and in the application of both preconditioners (31) and (35), matrices A and Q are used implicitly to fit into the matrix-free regime.

Spectral analysis of the preconditioner (31) was performed in [43] and the preliminary computational results confirmed that the matrix-free approach is promising when applied to certain classes of difficult problems. Below we will demonstrate how the method performs when applied to two classes of problems which defy any existing optimization software: LP relaxations of quadratic assignment problems and a special class of linear programs arising in quantum mechanics. All problems used in our experiments are available on request from the author. Computational experiments were performed on Dell Precision M60 laptop computer with a 2GHz (single core) processor and 2GB of RAM running Linux. The matrix-free version of HOPDM [43] was compiled with the GNU Fortran compiler `g77` with optimization option `-O2`. The solver terminates when the infeasibilities in the primal and dual spaces drop below $\varepsilon_p = \varepsilon_d = 10^{-3}$ and the relative duality gap drops below $\varepsilon_0 = 10^{-4}$. All CPU times reported in tables below are in seconds unless specifically stated otherwise.

It is worth adding that it is possible to run the matrix-free method with a more demanding stopping criteria that is with the smaller feasibility and optimality tolerances. However, this would require allowing it to use an increased rank of the partial Cholesky factorization to produce a better preconditioner and allowing it to perform more conjugate gradient iterations to reduce the error in the inexact Newton directions.

6.1 LP relaxations of quadratic assignment problems

The quadratic assignment problem (QAP) belongs to a class of facility location problems. A set of n facilities and n locations is considered. For each pair of locations, two parameters are determined: a *flow*, the amount of goods to be shipped between these two facilities, and a *distance* between them. The quadratic assignment problem is to assign all facilities to different locations so that the sum of distances multiplied by the corresponding flows is minimized.

Problem	Dimensions			Cholesky nonzeros	
	rows	columns	nonzeros	Normal Eqns	Aug System
qap12	3192	8856	38304	2135388	1969957
qap15	6330	22275	94950	8191638	7374972
nug12	3192	8856	38304	2789960	1969957
nug15	6330	22275	94950	11047639	7374972

Table 1: Symmetric LDL^T factors in small QAP problems.

Problem	Cplex 11.0.1				Mtx-Free HOPDM			
	Simplex		Barrier		rank=200		rank=500	
	iters	time	iters	time	iters	time	iters	time
qap12	83236	174.61	16	10.51	8	**2.37**	8	15.59
qap15	348190	2369.98	15	65.75	8	**6.13**	8	25.58
nug12	96148	187.02	13	9.86	7	**2.06**	7	14.53
nug15	387873	2451.40	16	71.33	7	**9.78**	7	33.73

Table 2: Solution statistics for small QAP problems.

QAP is one of the great challenges of combinatorial optimization. It is known to be NP-complete and therefore considered to be very difficult. A plethora of approaches (heuristics) have been developed to tackle this problem [89]. QAP can be formulated as a quadratic integer optimization problem which allows us to study its different polyhedral approximations. The examples considered in this section are LP relaxations of problems described by Nugent et al. [86]. These instances can be found at the QAPLIB website [16].

The difficulty of LP relaxations of QAPs originates from a high connectivity (and regularity) of the sparsity pattern in the constraint matrix A. The statistics of symmetric LDL^T decompositions of the normal equations (22) and augmented system (21) reported in Table 1 display an unusually high density of factors for these small linear optimization problems.

Matrix-free method in which the normal equations are solved by the conjugate gradient method preconditioned with the partial Cholesky matrix of rank 200 or 500 was applied to solve these problems. The results collected in Table 2 show that using the preconditioner with rank $k = 200$, the matrix-free method was significantly faster than the Cplex Simplex and Cplex Barrier solvers and could solve these problems in a few seconds of CPU time. We have used a default setting of Cplex parameters hence it solved the problems to a higher accuracy than the matrix-free HOPDM. Using less demanding accuracy requirements would not change much in the simplex runs but it could possibly save two or three last IPM iterations of the Cplex Barrier and therefore improve slightly Cplex Barrier CPU times.

Problem	Dimensions			Cholesky nonzeros
	rows	columns	nonzeros	(Cplex 11.0.1)
nug20	15240	72600	304800	38×10^6
nug30	52260	379350	1567800	459×10^6

Table 3: Symmetric LDL^T factors in larger QAP problems.

Problem	Cplex 11.0.1				Mtx-Free HOPDM			
	Simplex		Barrier		rank=200		rank=500	
	iters	time	iters	time	iters	time	iters	time
nug20	2855100	79451.00	18	1034.26	6	**35.19**	5	122.08
nug30	?	>28 days	-	OoM	5	**1271.97**	5	4465.60

Table 4: Solution statistics for larger QAP problems.

In our second experiment, larger instances of QAP problems were considered. The statistics of these problems and of their solution are reported in Tables 3 and 4, respectively. Although for today's standards these problems are small to medium scale, they challenge standard LP approaches. Nug20 could still be solved by the Cplex Simplex in about 22 hours of computations and in about 18 minutes by the Cplex Barrier solver. However, the larger nug30 defied both Cplex solvers. The Simplex run was interrupted after 28 days of computations still far away from optimality and making very slow progress towards it. Cplex Barrier was unable to fit the Cholesky matrix into memory (Out of Memory, OoM). The matrix-free approach with a partial Cholesky preconditioner of rank 200 was able to solve these two problems in 35 seconds and 21 minutes, respectively.

6.2 LP problems arising in quantum mechanics

The paper of Einstein, Podolsky and Rosen [31] stated an open question which is known as the EPR paradox (Einstein-Podolsky-Rosen paradox). Following Wikipedia: "[EPR paradox] refutes the dichotomy that *either* the measurement of a physical quantity in one system must affect the measurement of a physical quantity in another, spatially separate, system *or* the description of reality given by a wave function must be incomplete."

Quantum mechanics predicts that the measurements performed on spatially separated parts of quantum systems may instantaneously influence each other. The phenomenon is known as *quantum entanglement*. A possible way to resolve the EPR paradox is to introduce *hidden variables* and allow the Heisenberg uncertainty principle to act on these variables. Bell [6] showed that the EPR thought experiment of quantum mechanics predicts much stronger statistical correlations between the measurement results performed on different axes than the theory of hidden variables. He expressed the differences in a form of inequalities (known as Bell inequalities) which can be experimentally detected/verified. Numerous experiments have been constructed to test Bell's inequalities on different quantum systems and they essentially confirmed the predictions of quantum mechanics [48].

The assumption of a local hidden variable model leads to a number of equations constraining

the relationship between quantum-mechanically-predicted probabilities of experiment results and the model's local probability distribution. However, for certain quantum entangled states these constraints cannot be satisfied. Żukowski and Kaszlikowski [106] introduced a visibility of the state and used it to measure the resistance of a state to satisfy the local hidden variable model's constraints. A non-visible state always satisfies all local hidden variable model's constraints, while a fully visible state might violate them. A boundary visibility, below which the constraints are satisfied and above which they are violated, is called the critical visibility. Finding this critical visibility can be cast as an LP problem where the above constraints form the polytope and the visibility is the cost function to be maximized.

Such an approach was used for example to show that non-classicality of two-quNit correlations is higher than for two qubits, and that it increases with N, the dimension of the Hilbert space [62, 63]. Gruca et al. [51] have applied this LP-based approach to analyse several known experimental results and demonstrated that it is possible to skip direct analysis of Bell inequalities altogether in the process. The examples considered in [51] require optimization of critical visibility which is a continuous but non-differentiable function of the observables measured in the Bell test experiment [50]. Since each computation of this function's value requires a solution of a non-trivial linear optimization problem the overall computations may sometimes require weeks of CPU time. Clearly, the ability to solve LP subproblems efficiently becomes crucial.

Similar to QAP examples, the LP problems modelling quantum physics are very sparse but they are difficult for IPMs because they produce almost completely dense Cholesky factors as a consequence of very high connectivity in the sparsity pattern of the constraint matrix. The statistics of symmetric LDL^T decompositions of normal equations (22) are reported in Table 5 and flag the expected difficulty of IPM. Factorizations of the augmented system (21) do not offer any computational advantage.

Problem	Dimensions			Cholesky
	rows	columns	nonzeros	nonzeros
1kx1k0	1025	1025	34817	0.5×10^6
4kx4k0	4097	4097	270337	8×10^6
16kx16k0	16385	16385	2129921	128×10^6
64kx64k0	65537	65537	16908289	2048×10^6

Table 5: Symmetric LDL^T factors of quantum physics problems.

Problem	Cplex 11.0.1				Mtx-Free HOPDM			
	Simplex		Barrier		rank=100		rank=200	
	iters	time	iters	time	iters	time	iters	time
1kx1k0	929	**0.38**	7	0.82	6	0.55	6	1.09
4kx4k0	5418	11.17	20	89.19	6	**8.47**	6	14.45
16kx16k0	62772	924.94	10	2350.93	7	63.99	5	**59.94**
64kx64k0	2578265	111.5 h	-	OoM	8	**520.33**	9	917.87

Table 6: Solution statistics for quantum physics problems.

The problems are also difficult for the simplex method as can be seen in the results reported in Table 6. Their difficulty grows rapidly with increasing dimensions. The smaller examples are

solved efficiently by the Cplex Simplex but the CPU time to solve the largest one exceeds 111 hours. Cplex Barrier delivers the solution of three smaller examples but due to dense factors it is less efficient than the simplex method. Cplex Barrier runs out of memory (OoM) for the largest example. The matrix-free approach is run with partial preconditioners of rank 100 and 200 and solves all instances in competitive time.

We wrap up this section with a few general comments regarding the efficiency of interior point methods and their matrix-free variant. For the two particular classes of problems discussed in this section standard IPMs struggle. However, under the condition that there is enough memory to store Cholesky factors they can deliver solutions and, except for the very small problems, usually do it faster than the simplex method. The reader should not generalise these observations to any class of LPs. The simplex method and the interior point method are two powerful algorithms which compete with each other. However, there exist classes of problems which can be dealt with easily as well as classes of problem which challenge (and possibly defy) any of these methods.

The matrix-free variant of the interior point method [43] with a partial Cholesky preconditioner was successful when applied to solve the two classes of problems presented in this section. However, the reader should be cautious of raising their expectations that the approach will be equally effective when applied to *any* very large LP or QP. Being an iterative approach, the matrix-free variant of IPM relies on the quality of the preconditioner and therefore it is exposed to the usual difficulties of finding a well-suited preconditioner for a given class of problems. There is certainly a need for other preconditioners able to work in matrix-free regime and the author expects increased research activity in this area in the near future.

7 Conclusions

We have addressed in this paper several of the most important issues of interior point methods. Since their developments started by the seminal paper of Karmarkar [61] IPMs have changed the landscape of optimization. They have become a very competitive alternative to the simplex method when linear programming problems are solved. Unlike the simplex method, IPMs provide a guarantee to solve optimization problems in $\mathcal{O}(\sqrt{n}\ln(1/\varepsilon))$ iterations and, in practice, display an unequalled convergence in merely a few iterations almost independent of the problem size.

We have presented an easy to follow proof of polynomial complexity of IPM applied to convex quadratic optimization problems and have discussed several issues of their implementation. Interior point methods owe their reliability to the use of *direct* methods of linear algebra. However, direct factorization techniques also impose a limitation on the methods' ability to solve very large optimization problems. *Iterative* methods of linear algebra should offer the way forward. Indeed, there has been increased research activity in the last decade to develop specialized preconditioners able to improve the conditioning of linear systems arising in IPMs. Several interesting preconditioners have already been proposed and we expect many inspiring approaches are still to be developed.

Acknowledgement
The author is grateful to Jacek Gruca for providing challenging linear optimization problems arising in quantum physics and to Julian Hall, Kristian Woodsend and Pavel Zhlobich for read-

ing earlier versions of this paper and useful suggestions which led to an improvement of the presentation. The author is also grateful to the anonymous referees whose comments helped to improve the paper.

References

[1] G. Al-Jeiroudi and J. Gondzio, *Convergence analysis of inexact infeasible interior point method for linear optimization*, Journal of Optimization Theory and Applications, 141 (2009), pp. 231–247.

[2] G. Al-Jeiroudi, J. Gondzio, and J. Hall, *Preconditioning indefinite systems in interior point methods for large scale linear optimization*, Optimization Methods and Software, 23 (2008), pp. 345–363.

[3] A. Altman and J. Gondzio, *Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization*, Optimization Methods and Software, 11-12 (1999), pp. 275–302.

[4] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, *Implementation of interior point methods for large scale linear programming*, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189–252.

[5] M. Arioli, I. S. Duff, and P. P. M. de Rijk, *On the augmented system approach to sparse least-squares problems*, Numerische Mathematik, 55 (1989), pp. 667–684.

[6] J. S. Bell, *On the Einstein-Podolsky-Rosen paradox*, Physics, 1 (1964), pp. 195–200.

[7] S. Bellavia, *An inexact interior point method*, Journal of Optimization Theory and Applications, 96 (1998), pp. 109–121.

[8] M. Benzi, G. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[9] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli, *Inexact constraint preconditioners for linear system arising in interior point methods*, Computational Optimization and Applications, 36 (2007), pp. 137–147.

[10] L. Bergamaschi, J. Gondzio, and G. Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications, 28 (2004), pp. 149–171.

[11] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Massachusetts, 1995.

[12] J. R. Birge and L. Qi, *Computing block-angular Karmarkar projections with applications to stochastic programming*, Management Science, 34 (1988), pp. 1472–1479.

[13] R. E. Bixby, *Progress in linear programming*, ORSA Journal on Computing, 6 (1994), pp. 15–22.

[14] S. Bocanegra, F. Campos, and A. Oliveira, *Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods*, Computational Optimization and Applications, 36 (2007), pp. 149–164.

[15] J. R. Bunch and B. N. Parlett, *Direct methods for solving symemtric indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 8 (1971), pp. 639–655.

[16] R. E. Burkard, S. E. Karisch, and F. Rendl, *QAPLIB–a quadratic assignment problem library*, Journal of Global Optimization, 10 (1997), pp. 391–403.

[17] S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, and G. Toraldo, *Convergence analysis of an inexact potential reduction method for convex quadratic programming*, Journal of Optimization Theory and Applications, 135 (2007), pp. 355–366.

[18] J. Castro, *A specialized interior-point algorithm for multicommodity network flows*, SIAM Journal on Optimization, 10 (2000), pp. 852–877.

[19] J. Castro and J. Cuesta, *Quadratic regularizations in an interior point method for primal block-angular problems*, Mathematical Programming, (2010). Published online 19 February 2010, DOI: 10.1007/s10107-010-0341-2.

[20] J. Chai and K. Toh, *Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming*, Computational Optimization and Applications, 36 (2007), pp. 221–247.

[21] I. C. Choi and D. Goldfarb, *Exploiting special structure in a primal-dual path following algorithm*, Mathematical Programming, 58 (1993), pp. 33–52.

[22] M. Colombo and J. Gondzio, *Further development of multiple centrality correctors for interior point methods*, Computational Optimization and Applications, 41 (2008), pp. 277–305.

[23] G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, 1963.

[24] M. D'Apuzzo, V. De Simone, and D. di Serafino, *On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods*, Computational Optimization and Applications, 45 (2010), pp. 283–310.

[25] ——, *Starting-point strategies for an infeasible potential reduction method*, Optimization Letters, 4 (2010), pp. 131–146.

[26] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis, 19 (1982), pp. 400–408.

[27] I. I. Dikin, *On the speed of an iterative process*, Upravlaemye Sistemy, 12 (1974), pp. 54–60.

[28] H. Dollar, N. Gould, W. Schilders, and A. Wathen, *Using constraint preconditioners with regularized saddle-point problems*, Computational Optimization and Applications, 36 (2007), pp. 249–270.

[29] I. S. Duff, A. M. Erisman, and J. K. Reid, *Direct methods for sparse matrices*, Oxford University Press, New York, 1987.

[30] C. Durazzi and V. Ruggiero, *Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems*, Numerical Linear Algebra with Applications, 10 (2003), pp. 673–688.

[31] A. Einstein, B. Podolsky, and N. Rosen, *Can quantum-mechanical description of physical reality be considered complete?*, Physical Review, 47 (1935), pp. 777–780.

[32] A. V. Fiacco and G. P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, 1968.

[33] J. J. H. Forrest and D. Goldfarb, *Steepest-edge simplex algorithms for linear programming*, Mathematical Programming, 57 (1992), pp. 341–374.

[34] A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Review, 44 (2002), pp. 525–597.

[35] A. Frangioni and C. Gentile, *Prim-based support-graph preconditioners for min-cost flow problems*, Computational Optimization and Applications, 36 (2007), pp. 271–287.

[36] M. Friedlander and D. Orban, *A primal-dual regularized interior-point method for convex quadratic programming*, Tech. Rep. G-2010-47, GERAD, University of Montreal, Canada, 2010.

[37] K. R. Frisch, *The logarithmic potential method of convex programming*, tech. rep., University Institute of Economics, Oslo, Norway, 1955.

[38] E. M. Gertz and S. J. Wright, *Object-oriented software for quadratic programming*, ACM Transactions on Mathematical Software, 29 (2003), pp. 58–81.

[39] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Preconditioners for indefinite systems arising in optimization*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 292–311.

[40] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright, *On the projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method*, Mathematical Programming, 36 (1986), pp. 183–209.

[41] D. Goldfarb and M. J. Todd, *Linear programming*, in Optimization, G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, eds., North Holland, Amsterdam, 1989, pp. 73–170.

[42] J. Gondzio, *Multiple centrality corrections in a primal-dual method for linear programming*, Computational Optimization and Applications, 6 (1996), pp. 137–156.

[43] ——, *Matrix-free interior point method*, Computational Optimization and Applications, (2010). Published online October 14, 2010: DOI 10.1007/s10589-010-9361-3.

[44] J. Gondzio and A. Grothey, *Direct solution of linear systems of size* 10^9 *arising in optimization with interior point methods*, in Parallel Processing and Applied Mathematics, R. Wyrzykowski, J. Dongarra, N. Meyer, and J. Wasniewski, eds., vol. 3911 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2006, pp. 513–525.

[45] ——, *Solving nonlinear portfolio optimization problems with the primal-dual interior point method*, European Journal of Operational Research, 181 (2007), pp. 1019–1029.

[46] J. Gondzio and R. Sarkissian, *Parallel interior point solver for structured linear programs*, Mathematical Programming, 96 (2003), pp. 561–584.

[47] C. C. Gonzaga, *Path–following methods for linear programming*, SIAM Review, 34 (1992), pp. 167–224.

[48] S. Grőblacher, T. Paterek, R. Kaltenbaek, C. Brukner, M. Żukowski, M. Aspelmeyer, and A. Zeilinger, *An experimental test of non-local realism*, Nature, 446 (2007), pp. 871–875.

[49] M. D. Grigoriadis and L. G. Khachiyan, *An interior point method for bordered block-diagonal linear programs*, SIAM Journal on Optimization, 6 (1996), pp. 913–932.

[50] J. Gruca, *Nonclassical properties of multiqubit quantum states applicable in quantum information science*, Master's thesis, University of Gdansk, 2008.

[51] J. Gruca, W. Laskowski, M. Żukowski, N. Kiesel, W. Wieczorek, C. Schmid, and H. Weinfurter, *Nonclassicality thresholds for multiqubit states: Numerical analysis*, Physical Review A, 82 (2010), p. 012118.

[52] J.-P. Haeberly, M. Nayakkankuppam, and M. Overton, *Extending Mehrotra and Gondzio higher order methods to mixed semidefinite-quadratic-linear programming*, Optimization Methods and Software, 11 (1999), pp. 67–90.

[53] J. A. J. Hall, *Towards a practical parallelisation of the simplex method*, Computational Management Science, 7 (2010), pp. 139–170.

[54] J. A. J. Hall and K. I. M. McKinnon, *Hyper-sparsity in the revised simplex method and how to exploit it*, Computational Optimization and Applications, 32 (2005), pp. 259–283.

[55] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bur. Stand, 49 (1952), pp. 409–436.

[56] J. K. Hurd and F. M. Murphy, *Exploiting special structure in a primal-dual interior point methods*, ORSA Journal on Computing, 4 (1992), pp. 39–44.

[57] B. Jansen, C. Roos, T. Terlaky, and J.-P. Vial, *Primal–dual target following algorithms for linear programming*, Annals of Operations Research, 62 (1996), pp. 197–231. K. Anstreicher and R. Freund (eds.).

[58] E. R. Jessup, D. Yang, and S. A. Zenios, *Parallel factorization of structured matrices arising in stochastic programming*, SIAM Journal on Optimization, 4 (1994), pp. 833–846.

[59] J. J. Júdice, J. Patricio, L. F. Portugal, M. G. C. Resende, and G. Veiga, *A study of preconditioners for network interior point methods*, Computational Optimization and Applications, 24 (2003), pp. 5–35.

[60] L. V. Kantorovich, *The Best Use of Economic Resources*, Harvard University Press, Harvard, 1965.

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

[62] D. Kaszlikowski, P. Gnacinski, M. Żukowski, W. Miklaszewski, and A. Zeilinger, *Violations of local realism by two entangled qunits are stronger than for two qubits*, Physics Review Letters, 85 (2000), p. 4418.

[63] D. Kaszlikowski, M. Żukowski, and P. Gnacinski, *A note on bound entanglement and local realism*, Physical Review A, 65 (2002), p. 032107.

[64] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300–1317.

[65] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, vol. 16 of Frontiers in Applied Mathematics, SIAM, Philadelphia, 1995.

[66] L. G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Mathematics Doklady, 20 (1979), pp. 191–194.

[67] V. Klee and G. Minty, *How good is the simplex algorithm*, in Inequalities-III, O. Shisha, ed., Academic Press, 1972, pp. 159–175.

[68] M. Kojima, N. Megiddo, and S. Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Mathematical Programming, 61 (1993), pp. 263–280.

[69] M. Kojima, S. Mizuno, and A. Yoshise, *A primal-dual interior point algorithm for linear proramming*, in Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods, N. Megiddo, ed., Springer Verlag, Berlin, Germany, 1989, pp. 29–47.

[70] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, *Applications of second-order cone programming*, Linear Algebra and its Applications, 284 (1998), pp. 193–228.

[71] Z. Lu, R. D. S. Monteiro, and J. W. O'Neal, *An iterative solver-based infeasible primal-dual path-following algorithm for convex quadratic programming*, SIAM Journal on Optimization, 17 (2006), pp. 287–310.

[72] L. Lukšan and J. Vlček, *Indefinitely preconditioned inexact Newton method for large sparse equality constrained nonlinear programming problems*, Numerical Linear Algebra with Applications, 5 (1998), pp. 219–247.

[73] I. J. Lustig, R. E. Marsten, and D. F. Shanno, *On implementing Mehrotra's predictor-corrector interior point method for linear programming*, SIAM Journal on Optimization, 2 (1992), pp. 435–449.

[74] ——, *Interior point methods for linear programming: computational state of the art*, ORSA Journal on Computing, 6 (1994), pp. 1–14.

[75] I. Maros, *Computational Techniques of the Simplex Method*, Kluwer Academic Publishers, Boston, 1st ed., 2003.

[76] R. E. Marsten, R. Subramanian, M. J. Saltzman, I. J. Lustig, and D. F. Shanno, *Interior point methods for linear programming: Just call Newton, Lagrange, and Fiacco and McCormick!*, Interfaces, 20 (1990), pp. 105–116.

[77] N. Megiddo, *Pathways to the optimal set in linear programming*, in Progress in Mathematical Programming: Interior-Point Algorithms and Related Methods, N. Megiddo, ed., Springer Verlag, Berlin, Germany, 1989.

[78] S. Mehrotra, *On the implementation of a primal–dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601.

[79] C. Mészáros, *On the performance of the Cholesky factorization in interior point methods on Pentium 4 processors*, Central European Journal of Operations Research, 13 (2005), pp. 289–298.

[80] ——, *Sparsity in convex quadratic programming with interior point methods*, Optimization Methods and Software, 21 (2006), pp. 733–745.

[81] ——, *Detecting "dense" columns in interior point methods for linear programming*, Computational Optimization and Applications, 36 (2007), pp. 309–320.

[82] S. Mizuno and F. Jarre, *Global and polynomial-time convergence of an infeasible-interior point algorithm using inexact computation*, Mathematical Programming, 84 (1999), pp. 105–122.

[83] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, Kluwer, Boston, 2004.

[84] ——, *Smooth minimization of non-smooth functions*, Mathematical Programming, 103 (2005), pp. 127–152.

[85] Y. Nesterov and A. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming: Theory and Applications*, SIAM, Philadelphia, 1994.

[86] C. E. Nugent, T. E. Vollman, and J. Ruml, *An experimental comparison of techniques for assignment of facilities to locations*, Operations Research, 16 (1968), pp. 150–173.

[87] A. R. L. Oliveira, *A New Class of Preconditioners for Large-Scale Linear Systems from Interior Point Methods for Linear Programming*, PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005, 1997.

[88] A. R. L. Oliveira and D. C. Sorensen, *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications, 394 (2005), pp. 1–24.

[89] P. M. Pardalos, F. Rendl, and H. Wolkowicz, *The quadratic assignment problem: a survey of recent developments*, in Quadratic Assignment and Related Problems, P. M. Pardalos and H. Wolkowicz, eds., DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 1994, pp. 1–42.

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

[91] M. G. C. Resende and G. Veiga, *An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks*, SIAM Journal on Optimization, 3 (1993), pp. 516–537.

[92] M. Rozložník and V. Simoncini, *Krylov subspace methods for saddle point problems with indefinite preconditioning*, SIAM Journal of Matrix Analysis and Applications, 24 (2002), pp. 368–391.

[93] M. Saunders, *Cholesky-based methods for sparse least squares: the benefits of regularization*, in Linear and Nonlinear Conjugate Gradient-Related Methods, L. Adams and L. Nazareth, eds., SIAM, Philadelphia, USA, 1996, pp. 92–100.

[94] M. Saunders and J. A. Tomlin, *Solving regularized linear programs using barrier methods and KKT systems*, Tech. Rep. SOL 96-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305, USA, December 1996.

[95] O. Schenk, A. Wächter, and M. Weiser, *Inertia revealing preconditioning for large-scale nonconvex constrained optimization*, SIAM Journal on Scientific Computing, 31 (2008), pp. 939–960.

[96] G. Schultz and R. R. Meyer, *An interior point method for block angular optimization*, SIAM Journal on Optimization, 1 (1991), pp. 583–602.

[97] G. W. Stewart, *On scaled projections and pseudoinverses*, Linear Algebra and its Applications, 112 (1989), pp. 189–193.

[98] U. H. Suhl and L. M. Suhl, *Computing sparse LU factorizations for large-scale linear programming bases*, ORSA Journal on Computing, 2 (1990), pp. 325–335.

[99] T. Terlaky, *Criss-cross pivoting rules*, in Encyclopedia of Optimization, C. Floudas and P. Pardalos, eds., Kluwer Academic Publishers, 2009, pp. 584–590.

[100] M. J. Todd, *Exploiting special structure in Karmarkar's linear programming algorithm*, Mathematical Programming, 41 (1988), pp. 81–103.

[101] L. Vandenberghe and S. Boyd, *Semidefinite programming*, SIAM Review, 38 (1996), pp. 49–95.

[102] R. J. Vanderbei, *Symmetric quasidefinite matrices*, SIAM Journal on Optimization, 5 (1995), pp. 100–113.

[103] ——, *LOQO: An interior point code for quadratic programming*, Optimization Methods and Software, 11-12 (1999), pp. 451–484.

[104] R. J. Vanderbei and D. F. Shanno, *An interior-point algorithm for nonconvex nonlinear programming*, Computational Optimization and Applications, 13 (1999), pp. 231–252.

[105] S. J. Wright, *Primal-Dual Interior-Point Methods*, SIAM, Philadelphia, 1997.

[106] M. Żukowski and D. Kaszlikowski, *Critical visibility for N-particle Greenberger-Horne-Zeilinger correlations to violate local realism*, Physical Review A, 56 (1997), p. R1685.