A regularized simplex method

Csaba I. Fábián[*†] Krisztián Eretnek[‡] Olga Papp[*†]

Abstract

In case of a special problem class, the simplex method can be implemented as a cutting-plane method that approximates a polyhedral convex objective function. In this paper we consider a regularized version of this cutting-plane method, and interpret the resulting procedure as a regularized simplex method. (Regularization is performed in the dual space and only affects the process through the pricing mechanism. Hence the resulting method moves among basic solutions.) We present favorable test results with this regularized simplex method.

For general linear programming problems, we propose a Newton-type approach which requires the solution of a sequence of special problems.

Keywords. Linear programming, convex programming ; simplex method, cutting-plane methods, regularization.

1 Introduction

The simplex method can be interpreted as a cutting-plane method that *approximates the feasible polyhedron* of the dual problem, in case a special simplex pricing rule is used. This aspect is discussed in [9]. The cutting-plane pricing rule gives priority to column vectors that have already been basic. This is just the opposite of the pricing rule of Pan [20]. The latter rule proves rather effective in practice, see [19] and [4] for test results and insights. It turns out that for many industrially relevant problems, variables frequently re-enter the basis after relatively few simplex iterations. – In this light, the cutting-plane rule seems quite unreasonable for a general pricing rule. No wonder that it spoiled the efficacy of the simplex method in the computational study of [9].

However, there is a class of linear programming problems, in whose case the simplex method with the above-mentioned pricing rule can be interpreted as a cutting-plane method that *approximates the objective function* in a certain convex polyhedral optimization problem. Namely, this class consists of the duals of *ball-fitting problems* – the latter type of problem is about finding the largest ball that fits into a given polyhedron. When applied to problems of this class, the cutting-plane rule did not spoil the simplex method in the computational study of [9]. (To be quite precise, the cutting-plane pricing rule is just a preference scheme that can be combined with either the Dantzig or the steepest-edge rule. In [9], two pricing rules were compared: the Dantzig rule on the one hand, and a combination of the cutting-plane and the Dantzig rules on the other hand. The latter proved competitive with the former.)

Moreover, we found that fitting a ball into a polyhedron is significantly easier than optimizing a linear function over the same polyhedron. – This observation is in accord with common experience: the simplex method is very flexible in handling feasibility issues, and this flexibility is exploited in finding a central point of a polyhedron.

[*]Department of Informatics, Faculty of Mechanical Engineering and Automation, Kecskemét College, 10 Izsáki út, 6000 Kecskemét, Hungary. E-mail: fabian.csaba@gamf.kefo.hu.

[†]Doctoral School in Mathematics, Eötvös Loránd University, Budapest, Hungary

[‡]Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary

In the present paper we propose a regularized version of the simplex method. As mentioned above, a ball-fitting problem can be formulated as the minimization of a polyhedral convex function, and the simplex method can be implemented as a cutting-plane method that approximates this objective function. The latter method enables regularization. From the primal perspective, the regularized procedure is interpreted as a regularized simplex method. Regularization is performed in the dual space and only affects the process through the pricing mechanism. Hence the resulting method moves among basic solutions. The prospect of such regularized solution was observed in [7], and a detailed analysis is put forward in the present paper.

The paper is organized as follows. In section 2 we apply the level method of Lemaréchal, Nemirovskii, and Nesterov [11] to minimize the convex polyhedral objective function of a ball-fitting problem, and interpret this procedure as a regularized simplex method. In Section 3 we present and compare test results obtained by a basic implementation of the simplex method on the one hand, and a level-regularized version on the other hand.

In Section 4 we propose a Newton-type approach for the solution of general linear programming problems. This requires the solution of a sequence of ball-fitting problems. – As observed above, such problems are much more easily solved than general linear programming problems. Moreover, level regularization can be applied to the ball-fitting problems. – The efficiency of this approach can be enhanced by using approximate solutions of the ball-fitting problems. We adapt an approximation scheme of Lemaréchal, Nemirovskii, and Nesterov [11] to the present problem.

Section 5 contains conclusions, and places the proposed methods in context. We mention that the proposed regularization approach bears an analogy to known simplex techniques. We also mention that regularized cutting-plane methods have been successfully applied before to certain large-scale convex polyhedral problems.

1.1 Notation and clarification of terms

Let us consider the primal-dual pair of problems

$$\text{(1.P)}\quad \begin{array}{l} \max\ \boldsymbol{c}^T\boldsymbol{x} \\ A\boldsymbol{x} \le \boldsymbol{b} \\ \boldsymbol{x} \ge \boldsymbol{0}, \end{array} \qquad \Bigg| \qquad \text{(1.D)}\quad \begin{array}{l} \min\ \boldsymbol{b}^T\boldsymbol{y} \\ A^T\boldsymbol{y} \ge \boldsymbol{c} \\ \boldsymbol{y} \ge \boldsymbol{0}, \end{array} \tag{1}$$

where A is an $m \times n$ matrix, and the vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ have appropriate dimensions. We assume that both problems in (1) are feasible (and hence both have a finite optimum). We will solve these problems by the simplex and the dual simplex method, respectively. To this end, we formulate them with equality constraints, by introducing slack and surplus variables:

$$\text{(2.P)}\quad \begin{array}{l} \max\ \boldsymbol{c}^T\boldsymbol{x} \\ A\boldsymbol{x} + I\boldsymbol{u} = \boldsymbol{b} \\ (\boldsymbol{x}, \boldsymbol{u}) \ge \boldsymbol{0}, \end{array} \qquad \Bigg| \qquad \text{(2.D)}\quad \begin{array}{l} \min\ \boldsymbol{y}^T\boldsymbol{b} \\ \boldsymbol{y}^T A - \boldsymbol{v}^T I = \boldsymbol{c}^T \\ (\boldsymbol{y}, \boldsymbol{v}) \ge \boldsymbol{0}, \end{array} \tag{2}$$

where I denotes identity matrix of appropriate dimension ($m \times m$ in the primal problem and $n \times n$ in the dual problem).

A basis $\mathcal{B}$ of the primal problem is an appropriate subset of the columns of the matrix (A, I). A basis is said to be feasible if the corresponding basic solution is non-negative.

Let B denote the basis matrix corresponding to the basis $\mathcal{B}$. Given a vector $\boldsymbol{z} \in \mathbb{R}^{n+m}$, let $\boldsymbol{z}_\mathcal{B} \in \mathbb{R}^m$ denote the sub-vector containing the basic positions of $\boldsymbol{z}$. Specifically, let $(\boldsymbol{c}, \boldsymbol{0})_\mathcal{B}$ denote the basic part of the objective vector.

Let $\boldsymbol{y}$ denote the shadow-price vector (a.k.a. dual vector) belonging to basis $\mathcal{B}$. This is determined by the equation $\boldsymbol{y}^T B = (\boldsymbol{c}, \boldsymbol{0})_\mathcal{B}^T$. The reduced-cost vector belonging to basis $\mathcal{B}$ is $\left(A^T\boldsymbol{y} - \boldsymbol{c},\ \boldsymbol{y}\right) \in \mathbb{R}^{n+m}$.

2 A ball-fitting problem

Let us consider the primal-dual pair of problems (1), and assume we want to find a 'central' point in the feasible domain of Problem (1.D). Constraints in (1.D) have the form $\boldsymbol{a}_j^T \boldsymbol{y} \geq c_j$ $(j = 1, \ldots, n)$. Given a tolerance $\zeta > 0$, a vector $\boldsymbol{y}$ satisfies the jth constraint with this tolerance if $\boldsymbol{a}_j^T \boldsymbol{y} + \zeta \geq c_j$ holds. Suppose we want to find a point $\boldsymbol{y} \geq \boldsymbol{0}$ that satisfies each constraint with minimum tolerance. This is described by the following problem:

$$\begin{array}{l} \min\ \zeta \\ A^T\boldsymbol{y} + \zeta\boldsymbol{1} \geq \boldsymbol{c} \\ \boldsymbol{y} \geq \boldsymbol{0},\ \zeta \in \mathbb{R}, \end{array} \tag{3}$$

where $\boldsymbol{1} = (1, \ldots, 1)^T$ denotes a vector of appropriate dimension. A negative tolerance means that the current solution satisfies a constraint set 'uniformly' stricter than that of (1.D). – If (1.D) is feasible and the Euclidean norms of the columns $\boldsymbol{a}_j$ are each 1, then (3) finds a maximum-radius ball that fits into the polyhedron $\{\boldsymbol{y} \,|\, A^T\boldsymbol{y} \geq \boldsymbol{c}\}$, such that the center of the ball falls into the positive orthant. Problem (3) is always feasible as ζ may take positive values. To ensure a bounded objective, the constraint $\boldsymbol{1}^T\boldsymbol{y} \leq d$ was added with a large constant d. The center-finding problem is formulated as (4.D) that makes a primal-dual pair with (4.P) :

$$\text{(4.P)} \quad \begin{array}{l} \max\ \boldsymbol{c}^T\boldsymbol{x} - d\xi \\ A\boldsymbol{x} - \xi\boldsymbol{1} \leq \boldsymbol{0} \\ \boldsymbol{1}^T\boldsymbol{x} = 1 \\ \boldsymbol{x} \geq \boldsymbol{0},\ \xi \geq 0, \end{array} \qquad \Bigg| \qquad \text{(4.D)} \quad \begin{array}{l} \min\ \zeta \\ A^T\boldsymbol{y} + \zeta\boldsymbol{1} \geq \boldsymbol{c} \\ -\boldsymbol{1}^T\boldsymbol{y} \geq -d \\ \boldsymbol{y} \geq \boldsymbol{0},\ \zeta \in \mathbb{R}, \end{array} \tag{4}$$

where d is a constant.

Problem (4.D) can be formulated as minimization of a polyhedral convex function over a simplex, formally

$$\min_{\boldsymbol{y} \in U} \varphi(\boldsymbol{y}) \quad \text{where} \quad \varphi(\boldsymbol{y}) := \max_{1 \leq j \leq n} \left\{ c_j - \boldsymbol{a}_j^T\boldsymbol{y} \right\}, \quad U := \left\{ \boldsymbol{y} \in \mathbb{R}^m \,\middle|\, \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{1}^T\boldsymbol{y} \leq d \right\}. \tag{5}$$

Problem (5) has a finite optimum, and clearly $(\boldsymbol{y}^\star, \zeta^\star)$ is an optimal solution of Problem (4.D) if and only if $\boldsymbol{y}^\star$ is an optimal solution, and $\zeta^\star$ is the optimal objective value, of Problem (5).

We solve Problem (5) by a cutting-plane method that builds the model function

$$\varphi_{\mathcal{J}}(\boldsymbol{y}) := \max_{j \in \mathcal{J}} \left\{ c_j - \boldsymbol{a}_j^T\boldsymbol{y} \right\}, \tag{6}$$

where $\mathcal{J} \subseteq \{1, \ldots, n\}$ denotes the set of cuts found so far. In this paper we assume that each occurring model problem has a finite optimum. This is achieved by a broad enough initial set of cuts. Algorithm 1, below, describes a cutting-plane method to the present problem. The algorithm is structured in a manner that will later accommodate a regularization mechanism. We also use some extra notation: ϕ^{best} denotes the best function value known, and ϕ_{bound} denotes the minimum of the model function. Obviously ϕ^{best} is an upper bound and ϕ_{bound} is a lower bound for the minimum of Problem (5). The aim is to decrease the gap $\phi^{best} - \phi_{bound}$.

Algorithm 1 *A cutting plane method for Problem (5).*

1.0 Initialization.

Set stopping tolerance $\epsilon > 0$.

Let $\mathcal{J} \subset \{1, \ldots, n\}$ be such that the minimum of $\varphi_{\mathcal{J}}$ over U is finite.

Let $\boldsymbol{y}^\star$ be a minimizer of the model function $\varphi_{\mathcal{J}}$ over U, and let $\phi_{bound} := \varphi_{\mathcal{J}}(\boldsymbol{y}^\star)$.

Let the initial iterate be $\boldsymbol{y}^\circ := \boldsymbol{y}^\star$.

Let $\phi^{best} := +\infty$.

1.1 Objective function evaluation and update of the best solution.
If $\varphi(\boldsymbol{y}^\circ) < \phi^{best}$ then let $\boldsymbol{y}^{best} := \boldsymbol{y}^\circ$, $\phi^{best} := \varphi(\boldsymbol{y}^\circ)$.

1.2 Near-optimality check.
If $\phi^{best} - \phi_{bound} \le \epsilon$ then stop; $\boldsymbol{y}^{best}$ is an ϵ-optimal feasible solution of the original Problem (5).

1.3 Update and minimization of the model function.
Let $\jmath^\circ \in \{1, \dots, n\}$ be such that $l(\boldsymbol{y}) = c_{\jmath^\circ} - \boldsymbol{a}_{\jmath^\circ}^T \boldsymbol{y}$ is a support function to $\varphi(\boldsymbol{y})$ at $\boldsymbol{y}^\circ$, and let $\mathcal{J} := \mathcal{J} \cup \{\jmath^\circ\}$.

Let $\boldsymbol{y}^\star$ denote a minimizer of the updated function $\varphi_{\mathcal{J}}$ over U, and let $\phi_{bound} := \varphi_{\mathcal{J}}(\boldsymbol{y}^\star)$.

1.4 Setting new iterate.
Let $\boldsymbol{y}^\circ := \boldsymbol{y}^\star$.

Repeat from step *1.1*.

The criterion of step *1.2* clearly implies that $\boldsymbol{y}^{best}$ is an ϵ-optimal solution of problem (5) because we have

$$\min_{\boldsymbol{y} \in U} \varphi(\boldsymbol{y}) \;\ge\; \min_{\boldsymbol{y} \in U} \varphi_{\mathcal{J}}(\boldsymbol{y}) \;\ge\; \phi_{bound} \;\ge\; \varphi(\boldsymbol{y}^{best}) - \epsilon.$$

The model problem can be formulated as (7.D), below; a relaxation of (4.D). This is a linear programming problem that makes a primal-dual pair with (7.P).

$$\text{(7.P)} \quad \begin{array}{l} \max\ \boldsymbol{c}_{\mathcal{J}}^T \boldsymbol{x}_{\mathcal{J}} - d\xi \\ A_{\mathcal{J}} \boldsymbol{x}_{\mathcal{J}} - \xi \mathbf{1} \le \mathbf{0} \\ \mathbf{1}^T \boldsymbol{x}_{\mathcal{J}} = 1 \\ \boldsymbol{x}_{\mathcal{J}} \ge \mathbf{0},\ \xi \ge 0, \end{array} \qquad \Bigg| \qquad \text{(7.D)} \quad \begin{array}{l} \min\ \zeta \\ A_{\mathcal{J}}^T \boldsymbol{y} + \zeta \mathbf{1} \ge \boldsymbol{c}_{\mathcal{J}} \\ -\mathbf{1}^T \boldsymbol{y} \ge -d \\ \boldsymbol{y} \ge \mathbf{0},\ \zeta \in \mathbb{R}, \end{array} \qquad (7)$$

where $A_{\mathcal{J}}$ denotes the sub-matrix of A containing the columns $\boldsymbol{a}_j$ $(j \in \mathcal{J})$. Similarly, $\boldsymbol{x}_{\mathcal{J}}$ and $\boldsymbol{c}_{\mathcal{J}}$ denote respective sub-vectors of $\boldsymbol{x}$ and $\boldsymbol{c}$, containing positions $j \in \mathcal{J}$.

Since (7.D) has an optimal solution, it follows from LP duality that (7.P) has an optimal solution. Clearly $(\boldsymbol{y}^\star, \zeta^\star)$ is an optimal solution of the dual problem (7.D) if and only if $\boldsymbol{y}^\star$ is a minimizer of the model function over the simplex U, and $\zeta^\star$ is the optimal objective value.

We will now specialize Algorithm 1. The special Algorithm 2, below, will be interpreted as a simplex method. Let us formulate the pair of problems (4) with equality constraints by introducing slack variables in the primal problem and surplus variables in the dual problem. In the primal problem we add m slack columns. Together with the ξ-column, that makes $m+1$ artificial columns.

Algorithm 2 *A special simplex method for Problem (4.P).*

2.0 Initialization.
Set stopping tolerance $\epsilon > 0$.

Let $\mathcal{B} \subset \{1, \dots, n+1+m\}$ be a feasible basis for the primal Problem (4.P) such that ξ is a basic variable.
Let $\mathcal{J} \subset \{1, \dots, n\}$ consist of the x-columns belonging to $\mathcal{B}$, and let $(\boldsymbol{y}^\star, \zeta^\star)$ be the dual vector corresponding to $\mathcal{B}$.

Let the initial dual vector be $(\boldsymbol{y}^\circ := \boldsymbol{y}^\star,\ \zeta^\circ := \zeta^\star)$.

Let $\phi^{best} := +\infty$ and $\phi_{bound} := \zeta^\star$.

2.1 Pricing and update of the best dual vector.
Let $z^\circ := \max\limits_{j \in \{1,\dots,n\} \setminus \mathcal{J}} \left\{ c_j - (\boldsymbol{y}^\circ, \zeta^\circ)^T (\boldsymbol{a}_j, 1) \right\}$.

If $\zeta^\circ + [z^\circ]_+ < \phi^{best}$ then let $\left(\boldsymbol{y}^{best} := \boldsymbol{y}^\circ,\ \phi^{best} := \zeta^\circ + [z^\circ]_+ \right)$.

2.2 Near-optimality check.

If $\phi^{best} - \phi_{bound} \leq \epsilon$ then stop; $\mathcal{B}$ is a feasible ϵ-optimal basis to Problem (4.P) – the evidence for ϵ-optimality being the dual feasible vector $(\boldsymbol{y}^{best}, \phi^{best})$.

2.3 Update and solution of the model problem.

Let $\mathcal{J} := \mathcal{J} \cup \{\jmath^\circ\}$ with some $\jmath^\circ \in \arg \max_{j \in \{1,\dots,n\}\setminus\mathcal{J}} \left\{ c_j - (\boldsymbol{y}^\circ, \zeta^\circ)^T(\boldsymbol{a}_j, 1) \right\}$.

Let's solve the updated model problem (7.P) by the simplex method starting from the current basis $\mathcal{B}$.

Let $\mathcal{B}$ denote an optimal basis, and let $(\boldsymbol{y}^\star, \zeta^\star)$ denote the corresponding the dual vector.
Let $\phi_{bound} := \zeta^\star$.

2.4 Setting new dual vector.

Let $(\boldsymbol{y}^\circ := \boldsymbol{y}^\star,\ \zeta^\circ := \zeta^\star)$.

Repeat from step *2.1.*

Algorithm 2 is indeed a specific version of Algorithm 1. This is quite obvious, only the pricing step *2.1* needs explanation. By the definition of z°, we have $\zeta^\circ + z^\circ = \max_{j \in \{1,\dots,n\}\setminus\mathcal{J}} \left\{ c_j - \boldsymbol{a}_j^T \boldsymbol{y} \right\}$. It follows that $\varphi(\boldsymbol{y}^\circ) = \max\{\ \zeta^\circ + z^\circ,\ \varphi_{\mathcal{J}}(\boldsymbol{y}^\circ)\ \}$. Taking into account $\zeta^\circ = \varphi_{\mathcal{J}}(\boldsymbol{y}^\circ)$, we get $\varphi(\boldsymbol{y}^\circ) = \zeta^\circ + [z^\circ]_+$.

Algorithm 2 is a special simplex method applied to Problem (4.P) – the arguments of [9] apply to these forms of the methods as well. The stopping criterion in step *2.2* somewhat differs from the usual simplex stopping criterion. Actually, it is more easily satisfied, because the usual criterion $z^\circ \leq \epsilon$ implies $\phi^{best} - \phi_{bound} \leq \epsilon$. (This follows from $\zeta^\circ + [z^\circ]_+ \geq \phi^{best}$ and $\phi_{bound} = \zeta^\circ$.)

In step *2.3* a feasible ϵ-optimal basis means that the corresponding basic solution is a feasible ϵ-optimal solution of Problem (4.P). Since $\mathcal{B}$ is feasible basis of the model Problem (7.P), it is also a feasible basis of the Problem (4.P). Due to duality, the objective value belonging to the corresponding basic solution is ϕ_{bound}. On the other hand, $(\boldsymbol{y}^{best}, \phi^{best})$ is a feasible vector of the dual Problem (4.D), with the objective value ϕ^{best}. This is obviously an upper bound for the primal objective.

2.1 Regularization

We propose regularizing the above cutting-plane method by the level method of Lemaréchal, Nemirovskii, and Nesterov [11]. This is a special bundle-type method that uses level sets of the model function for regularization. In contrast with other bundle-type methods that need continual tuning of the parameters, the level method uses a single fixed parameter. This parameter, denoted by λ, is used in determining level sets of the model function, and needs to be fixed at a value $0 < \lambda < 1$.

First we describe the level method as applied to the problem $\min_{\boldsymbol{y} \in U}\ \varphi(\boldsymbol{y})$ – that is the polyhedral Problem (5). In order to regularize Algorithm 1, step 4 is modified as follows:

1.4′ Setting new iterate.

Let $\Pi(\boldsymbol{y}^\circ)$ denote the optimal solution of the convex quadratic programming problem

$$\min \ \| \boldsymbol{y} - \boldsymbol{y}^\circ \|^2 \quad \text{such that} \quad \boldsymbol{y} \in U, \quad \varphi_{\mathcal{J}}(\boldsymbol{y}) \leq (1-\lambda)\phi_{bound} + \lambda \phi^{best}.$$

Let the new iterate be $\boldsymbol{y}^\circ := \Pi(\boldsymbol{y}^\circ)$.

Repeat from step *1.1.*

In words, the next iterate is the 'projection' $\Pi(\boldsymbol{y}^\circ)$ of the current iterate $\boldsymbol{y}^\circ$ to a specific level set of the model function.

Remark 3 *Concerning theoretical efficiency of the level method, the following estimate is proven in [11]: To obtain a gap smaller than ϵ, it suffices to perform $\kappa\,(DL/\epsilon)^2$ iterations, where D is the diameter of the feasible polyhedron, L is a Lipschitz constant of the objective function, and κ is a constant that depends only on the parameter of the algorithm.*

However, the level method performs much better in practice than the above estimate implies. Nemirovski in [16], Chapter 5.3.2 observes the following experimental fact: when solving a problem with m variables, every m steps add a new accurate digit in our estimate of the optimum. This observation is confirmed by the experiments reported in [10], where the level method was applied in a decomposition scheme for the solution of two-stage stochastic programming problems.

Remark 4 *In the present formulation, we added an extra constraint in the ball-fitting problem (4.D) so as to make the feasible domain bounded. The level method can be implemented for problems with unbounded domain. (A set of initial cuts is then needed to make the master objective bounded from below, just like with a pure cutting-plane method.) The constant D (diameter of the feasible polyhedron) is never actually used in course of the level method; it is used only in the convergence proof and in the theoretical efficiency estimate cited above.*

Let us apply this regularization to the specific cutting-plane Algorithm 2. Step 4 is modified as follows:

2.4′ Setting new dual vector.

Let $\Pi(\boldsymbol{y}^\circ)$ denote the optimal solution of the following convex quadratic programming problem:

$$\min \parallel \boldsymbol{y} - \boldsymbol{y}^\circ \parallel^2 \quad \text{such that} \quad \boldsymbol{y} \in U, \quad A_{\mathcal{J}}^T \boldsymbol{y} + \zeta \mathbf{1} \geq \boldsymbol{c}_{\mathcal{J}} \tag{8}$$

with $\zeta = (1-\lambda)\phi_{bound} + \lambda\phi^{best}$.

Let $\boldsymbol{y}^\circ := \Pi(\boldsymbol{y}^\circ)$ and with this updated vector, let $\zeta^\circ := \varphi_{\mathcal{J}}(\boldsymbol{y}^\circ)$.

Let the new dual vector be $(\boldsymbol{y}^\circ, \zeta^\circ)$.

Repeat from step *2.1.*

Like the simplex method, this method generates basic feasible solutions of Problem (4.P). Regularization is performed in the dual space and only affects the process through the pricing mechanism.

Remark 5 *As observed in [9], the whole Algorithm 2 can be viewed as a single simplex method.*

Selecting a support function in step 3 of Algorithm 1 naturally translates to selecting the column vector having the most negative reduced cost in step 3 of Algorithm 2. This is known as Dantzig pricing rule in the terminology of simplex implementation techniques.

If moreover we use the Dantzig pricing rule in course of the solution of the updated model problem (7.P), then Algorithm 2 will become a simplex method using a special pricing rule that is the combination of the cutting-plane preference scheme and the Dantzig rule.

Remark 6 *We need not insist on applying exact support functions in a cutting-plane framework. We can as well work with approximate support functions (i.e., epsilon-subgradients). Selecting an appropriate epsilon-subgradient in step 3 of the cutting-plane algorithm 1 will yield the steepest-edge pricing rule applied to the selection of the incoming column index $\jmath^\circ$ in step 3 of the simplex algorithm 2.*

If moreover we use the steepest-edge pricing rule in course of the solution of the updated model problem (7.P), then Algorithm 2 will become a simplex method using a special pricing rule that is the combination of the cutting-plane preference scheme and the steepest-edge rule.

Remark 7 *We can take advantage of the efficacy of the steepest-edge pricing rule in the minimization of any convex function, even if the minimization problem at hand is not a ball-fitting problem. As a specific area, let us mention the two-stage stochastic programming problem that can be formulated as the minimization of a polyhedral convex function, called the expected recourse function. Cutting-plane methods are a widely used means of minimizing the expected recourse function. In [6] we show that aggregated cuts are generally preferred to disaggregated cuts by the steepest-edge selection rule.*

3 A computational study

In this section we present and compare test results obtained by a basic implementation of the simplex method on the one hand, and a level-regularized version on the other hand.

3.1 Implementation issues

We implemented a basic but usable form of the simplex method: a revised method with the product form of the inverse. It is an extended version of the solver that was used in [9].

In our standard simplex implementation, we use Dantzig pricing. In the regularized simplex method, we implemented the level method working with exact support functions. (This results a combination of the cutting-plane pricing rule and the Dantzig rule, as mentioned in Remark 6.) The special quadratic subproblems of the level framework were solved by the CPLEX dual optimizer which proved to be better on our test problems than the barrier optimizer. We used CLPEX Concert Technology through the C++ API. The level parameter λ was set to 0.5.

The test runs were performed on a Dell Vostro laptop with Intel Core i7 CPU, 2GHz, and 4GB RAM. The operating system was Windows 7. Run times were measured by the standard C 'clock' function. We executed our tests on a standalone computer, with no other user process running. Each run was performed ten times, and the run times reported in this study are the averages of the respective 10 run times measured.

3.2 Test problems

We considered 59 test problems from the NETLIB collection (the smaller ones). These are listed in Table 1. We solved these NETLIB problems with maximization scope. Since the default scope in NETLIB is minimization, these problems were converted to maximization problems by taking the negatives of the objective vectors.

Given a NETLIB problem, we constructed two ball-fitting problems that we call *type-A* and *type-B* variants of the NETLIB problem. The type-A variant was constructed by taking the matrix A in (4.P) to be the matrix of the NETLIB problem, and taking the objective vector $\boldsymbol{c}$ in (4.P) to be the negative of objective vector of the NETLIB problem. The type-B variant was constructed by taking the matrix A in (4.P) to be the negative of matrix of the NETLIB problem, and taking the objective vector $\boldsymbol{c}$ in (4.P) to be the negative of objective vector of the NETLIB problem. When constructing the ball-fitting variants, we used only the matrices and the objective vectors of the NETLIB problems. We did not scale the columns of the matrices. In each test problem, the constant d was set individually, such that increasing d would not change the optimal basis of the test problem.

Each test problem was solved by our standard simplex implementation and also by the level-regularized method.

3.3 Test results

Tables 2 and 3 show test results of the standard simplex and the level-regularized implementation. We can see iteration numbers and total run time for the standard simplex method and numbers of iterations, numbers of level steps, and total run times for the level-regularized method. We included both iteration numbers and level steps because they may not be equal. – On the one hand, we may need more then one simplex iteration in step 3 of the regularized version of Algorithm 2 to find an optimal solution for the current model problem. On the other hand, $j^\circ \in \mathcal{J}$ may occur in this step as well, and in this case no simplex iteration is needed. (The latter case may occur because our objective function is polyhedral.) – Run times of the level-regularized method of course also include the time of building the quadratic subproblems and solving them.

The regularized method proves effective in our computational study. The standard simplex method could solve $118 - 7 = 111$ problems, and the level-regularized method could solve $118 - 3 = 115$ problems. There were 110 problems solved by both methods.

problem name	rows	columns
ADLITTLE	56	97
AFIRO	27	32
AGG	488	163
AGG2	516	302
AGG3	516	302
BANDM	305	472
BEACONFD	173	262
BLEND	74	83
BNL1	643	1175
BOEING1	351	384
BOEING2	166	143
BORE3D	233	315
BRANDY	220	249
CAPRI	271	353
DEGEN2	444	534
E226	223	282
ETAMACRO	400	688
FFFFF800	524	854
FINNIS	497	614
FIT1D	24	1026
FIT2D	25	10500
FORPLAN	161	421
GFRD-PNC	616	1092
GROW15	300	645
GROW22	440	946
GROW7	140	301
ISRAEL	174	142
KB2	43	41
LOTFI	153	308
PEROLD	625	1376
PILOT4	410	1000
RECIPELP	91	180
SC105	105	103
SC205	205	203
SC50A	50	48
SC50B	50	48
SCAGR25	471	500
SCAGR7	129	140
SCFXM1	330	457
SCFXM2	660	914
SCORPION	388	358
SCRS8	490	1169
SCTAP1	300	480
SCSD1	77	760
SCSD6	147	1350
SEBA	515	1028
SHARE1B	117	225
SHARE2B	96	79
SHELL	536	1775
SHIP04L	402	2118
SHIP04S	402	1458
STAIR	356	467
STANDATA	359	1075
STANDGUB	361	1184
STANDMPS	467	1075
STOCFOR1	117	111
TUFF	333	587
VTP-BASE	198	203
WOOD1P	244	2594

Table 1: The NETLIB test problems used in this study

Considering the 110 problems solved by both methods, the total iteration counts were: $22,414$ when the standard simplex method was used, and $8,681$ when the regularized simplex method was used. The total run times were: 41.8 seconds when the standard simplex method was used, and 41.3 when the regularized simplex method was used. Hence the ratios between iteration counts and running times were 39% and 99%, respectively.

Regularization helped most in the solution of the problems that were the most difficult ones for our pure simplex implementation.

Remark 8 *Our present results confirm the observation concerning the practical efficiency of the level method, namely, that every m level steps add a new accurate digit in our estimate of the optimum, when solving a problem with m variables. – Details of this observation are mentioned in Remark 3.*

When solving our test problems with the level-regularized method, the gap $\phi^{best} - \phi_{bound}$ halved generally in less than $m/5$ iterations. (This happened so in 94% of all the problems solved by the regularized method. In the remaining 6% of the problems, the worst case was $0.71\,m$ iterations.)

problem name	standard simplex		level-regularized simplex		
	iterations	run time (millisec)	iterations	level steps	total run time (millisec)
ADLITTLE	24	1	25	18	32
AFIRO	22	1	24	24	8
AGG	213	303	278	86	456
AGG2	-	-	-	-	-
AGG3	-	-	574	136	1434
BANDM	426	658	-	-	-
BEACONFD	128	30	6	13	13
BLEND	73	2	62	67	49
BNL1	36	117	35	33	265
BOEING1	67	60	67	76	175
BOEING2	20	5	21	17	15
BORE3D	2	4	2	1	2
BRANDY	1	2	1	0	1
CAPRI	36	22	21	24	39
DEGEN2	725	2622	634	244	2986
E226	202	126	211	105	268
ETAMACRO	4	7	3	1	8
FFFFF800	167	490	47	48	245
FINNIS	2	7	2	1	10
FIT1D	68	15	24	18	18
FIT2D	54	123	49	33	185
FORPLAN	88	33	42	44	56
GFRD-PNC	18	60	11	6	61
GROW15	11	13	14	10	31
GROW22	13	28	12	9	47
GROW7	8	4	9	7	12
ISRAEL	-	-	279	81	239
KB2	6	1	10	16	7
LOTFI	41	8	11	16	14
PEROLD	-	-	19	22	166
PILOT4	11	16	15	17	69
RECIPELP	8377	1892	48	51	27
SC105	143	18	216	101	114
SC205	312	199	328	170	409
SC50A	71	4	73	43	26
SC50B	64	2	97	49	26
SCAGR25	23	33	15	12	47
SCAGR7	3	1	10	7	7
SCFXM1	5	5	3	1	8
SCFXM2	7	28	4	1	24
SCORPION	13	11	9	6	16
SCRS8	48	100	48	49	258
SCTAP1	196	180	139	131	324
SCSD1	88	24	50	63	66
SCSD6	9	12	8	6	16
SEBA	3	9	4	2	19
SHARE1B	261	61	204	70	93
SHARE2B	109	9	87	65	61
SHELL	594	4005	540	595	6811
SHIP04L	58	189	29	39	252
SHIP04S	32	67	27	37	176
STAIR	126	173	58	69	158
STANDATA	59	109	7	4	25
STANDGUB	24	47	3	1	11
STANDMPS	211	669	9	26	99
STOCFOR1	74	9	58	69	48
TUFF	136	184	48	37	109
VTP-BASE	-	-	27	20	23
WOOD1P	187	476	661	200	3794

Table 2: Results for type-A problems

problem name	standard simplex		level-regularized simplex		
	iterations	run time (millisec)	iterations	level steps	total run time (millisec)
ADLITTLE	6	1	4	1	2
AFIRO	8	1	9	11	7
AGG	9	2	12	18	33
AGG2	9	16	9	13	34
AGG3	9	14	9	13	36
BANDM	19	16	7	5	13
BEACONFD	1	1	1	0	1
BLEND	21	1	21	14	8
BNL1	9	29	12	19	125
BOEING1	41	36	6	21	31
BOEING2	43	5	22	26	20
BORE3D	316	177	63	81	114
BRANDY	60	25	100	44	81
CAPRI	21	12	7	7	12
DEGEN2	10	10	5	2	15
E226	4	4	7	3	5
ETAMACRO	28	39	27	33	97
FFFFF800	948	7884	4	14	49
FINNIS	16	24	23	19	83
FIT1D	3	2	9	4	7
FIT2D	3	7	4	1	14
FORPLAN	64	24	58	28	47
GFRD-PNC	5	18	5	3	34
GROW15	812	1478	320	205	899
GROW22	2192	8032	543	362	3251
GROW7	244	92	253	99	189
ISRAEL	-	-	3	1	4
KB2	81	4	71	28	15
LOTFI	9	2	12	24	17
PEROLD	177	847	175	106	1371
PILOT4	-	-	-	-	-
RECIPELP	4	1	5	3	2
SC105	24	1	15	21	12
SC205	14	9	13	12	15
SC50A	19	1	18	16	8
SC50B	9	1	7	6	5
SCAGR25	2099	4725	588	244	1699
SCAGR7	88	9	142	75	77
SCFXM1	3	5	7	4	13
SCFXM2	6	22	6	4	39
SCORPION	3	5	25	17	45
SCRS8	3	12	2	1	14
SCTAP1	4	5	8	3	7
SCSD1	23	7	49	76	78
SCSD6	9	7	8	6	16
SEBA	516	1880	509	510	5889
SHARE1B	24	5	25	20	17
SHARE2B	63	5	37	28	15
SHELL	546	2672	519	525	5579
SHIP04L	47	154	46	51	359
SHIP04S	50	120	44	53	263
STAIR	47	45	63	57	158
STANDATA	2	7	3	11	31
STANDGUB	29	60	25	24	108
STANDMPS	2	7	3	11	34
STOCFOR1	52	7	30	41	28
TUFF	2	4	6	19	34
VTP-BASE	43	12	13	28	28
WOOD1P	241	615	248	248	2008

Table 3: Results for type-B problems

3.4 Remarks and possible enhancements

In the present implementation we never discarded cuts from the cutting-plane model of the objective function. In other words, the set $\mathcal{J}$ is monotone increasing. An effective bundle reduction strategy is proposed in [11]: All the non-binding cutting planes are eliminated after *critical* iterations. An iteration is called critical if it substantially decreases the gap. (The exact definition is the following: The first iteration is labelled critical. Let Δ denote the gap after the latest critical iteration. If the present iteration decreases the gap below $(1-\lambda)\Delta$, then it is also labelled critical.) The theoretical efficiency estimate of the level method is preserved by this reduction strategy. Concerning practical behavior, the numbers of the iterations are reported to have increased by at most 15 %.

One may expect the above bundle-reduction strategy to effectively control the sizes of the quadratic subproblems, taking into account the observation concerning the practical efficiency of the level method, mentioned in Remark 3, and its confirmation by our present test results in Remark 8.

In our implementation, the convex quadratic programming problems were solved exactly. One naturally expects that approximate solutions should be sufficient. A theoretical convergence proof of the level method using approximate solutions of the quadratic problems is presented in [21], though without practical considerations or numerical experiments.

4 On the solution of general linear programming problems

In this section we consider a general linear programming problem of the form (1). We propose approximately solving the general problem by solving a sequence of ball-fitting problems. As observed in [9], finding a central point in the feasible polyhedron of (1.D) is much easier than solving this dual problem. – Moreover, regularization can be applied to the ball-fitting problem. – Using the definition of φ in (5), this observation can be put in the following form: Minimizing $\varphi(\boldsymbol{y})$ under the constraints $\boldsymbol{y} \geq \mathbf{0},\ \boldsymbol{b}^T\boldsymbol{y} \leq d$ with fixed $d \in \mathrm{I\!R}$ is easier than minimizing $\boldsymbol{b}^T\boldsymbol{y}$ under the constraints $\boldsymbol{y} \geq \mathbf{0},\ \varphi(\boldsymbol{y}) \leq 0$. A natural idea is exploring the efficient frontier using a Newton method. The efficiency of this approach can be enhanced by using approximate solutions of the ball-fitting problems. In this section we adapt an approximation scheme of Lemaréchal, Nemirovskii, and Nesterov [11] to the present problem.

The above mentioned efficient frontier is characterized by Problem (9.D), where $d \in \mathrm{I\!R}$ is a parameter. Indeed, for an optimal solution $(\boldsymbol{y}_\star, \zeta_\star)$ of this problem, we always have $\zeta_\star = [\varphi(\boldsymbol{y}_\star)]_+$. We have put (9.D) in 'dual form', and its primal pair is (9.P).

$$
\text{(9.P)}\quad
\begin{array}{l}
\max\ \boldsymbol{c}^T\boldsymbol{x} - d\xi \\
A\boldsymbol{x} - \xi\boldsymbol{b} \leq \mathbf{0} \\
\mathbf{1}^T\boldsymbol{x} \leq 1 \\
\boldsymbol{x} \geq \mathbf{0},\ \xi \geq 0,
\end{array}
\qquad\Bigg|\qquad
\text{(9.D)}\quad
\begin{array}{l}
\min\ \zeta \\
A^T\boldsymbol{y} + \zeta\mathbf{1} \geq \boldsymbol{c} \\
-\boldsymbol{b}^T\boldsymbol{y} \geq -d \\
\boldsymbol{y} \geq \mathbf{0},\ \zeta \geq 0.
\end{array}
\qquad (9)
$$

As before, we assume that both problems in (1) are feasible. Let $\boldsymbol{x}^\star$ and $\boldsymbol{y}^\star$ denote respective optimal solutions of the primal and the dual problem. Using these, let $\boldsymbol{x}_\star := \boldsymbol{x}^\star,\ \xi_\star := 1$ if $\mathbf{1}^T\boldsymbol{x}^\star \leq 1$, and $\boldsymbol{x}_\star := \left(\mathbf{1}^T\boldsymbol{x}^\star\right)^{-1}\boldsymbol{x}^\star,\ \xi_\star := \left(\mathbf{1}^T\boldsymbol{x}^\star\right)^{-1}$ otherwise. Clearly $(\boldsymbol{x}_\star, \xi_\star)$ is a feasible solution of Problem (9.P) for any value of the parameter d. On the other hand, let $\mathcal{D}$ denote the set of the parameter values that make Problem (9.D) feasible. It is easily seen that $\mathcal{D} = [0, +\infty)$ if $\boldsymbol{b} \geq \mathbf{0}$, and $\mathcal{D} = (-\infty, +\infty)$ if $\boldsymbol{b}$ has negative components. Let $f(d)$ $(d \in \mathcal{D})$ denote the common optimum of the pair of problems (9). Clearly f is a monotone decreasing, piecewise linear convex function.

Let $d^\star := \boldsymbol{c}^T\boldsymbol{x}^\star = \boldsymbol{b}^T\boldsymbol{y}^\star$ denote the common optimum of the pair of problems (1). Let us consider the pair of problems $(9:\ d = d^\star)$. We have seen that $(\boldsymbol{x}_\star, \xi_\star)$ is a feasible solution of the primal problem, and it is easy to check that the corresponding objective value is 0. On the other hand, we have $\varphi(\boldsymbol{y}^\star) \leq 0,\ \boldsymbol{y}^\star \geq \mathbf{0}$ due to the feasibility of $\boldsymbol{y}^\star$ in Problem (1.D). Hence $\left(\boldsymbol{y}_\star = \boldsymbol{y}^\star, \zeta_\star = [\varphi(\boldsymbol{y}_\star)]_+ = 0\right)$ is a feasible solution of the dual problem, and the corresponding objective value is 0. It follows that $f(d^\star) = 0$.

Let $d_\circ \in \mathcal{D}$ be such that $d_\circ < d^\star$. Let $(\boldsymbol{y}_\circ, \zeta_\circ)$ denote an optimal solution of the problem $(9.\text{D} : d = d_\circ)$. We must have $\zeta_\circ > 0$. (Otherwise $\boldsymbol{y}_\circ$ would be a feasible solution of (1.D) with an objective value $< d^\star$.)

Hence $f(d_\circ) = \zeta_\circ > 0$. Let $(\boldsymbol{x}_\circ, \xi_\circ)$ be an optimal solution of the problem (9.P : $d = d_\circ$). By definition, we have $f(d_\circ) = \boldsymbol{c}^T\boldsymbol{x}_\circ - d_\circ\xi_\circ$, and hence the linear function $l_\circ(d) := \boldsymbol{c}^T\boldsymbol{x}_\circ - d\xi_\circ$ is a support function to $f(d)$ at $d = d_\circ$. The steepness of $l_\circ$ being $-\xi_\circ$, we get $-\xi_\circ < 0$ from $d_\circ < d^\star$, $f(d_\circ) > 0$, $f(d^\star) = 0$ and the convexity of f. Hence $\xi_\circ^{-1}\boldsymbol{x}_\circ$ is a feasible solution of problem (1.P). Considering the objective value, we have $\boldsymbol{c}^T\left(\xi_\circ^{-1}\boldsymbol{x}_\circ\right) > d_\circ$ from $\boldsymbol{c}^T\boldsymbol{x}_\circ - d_\circ\xi_\circ = f(d_\circ) > 0$.

A near-optimality criterion. Let $d_\circ$ be an approximate solution of $f(d) = 0$. To be specific, let

$$d_\circ \in \mathcal{D},\ d_\circ < d^\star \quad \text{such that} \quad \rho \geq f(d_\circ) \tag{10}$$

holds with a given tolerance $\rho > 0$. Let $(\boldsymbol{x}_\circ, \xi_\circ)$ and $(\boldsymbol{y}_\circ, \zeta_\circ)$ be respective optimal solutions of the problems (9 : $d = d_\circ$). As shown above, we have $\xi_\circ > 0$, and $\xi_\circ^{-1}\boldsymbol{x}_\circ$ is a feasible solution of problem (1.P), with the objective value $\boldsymbol{c}^T\left(\xi_\circ^{-1}\boldsymbol{x}_\circ\right) > d_\circ$.

Considering the dual objective value, we have $\zeta_\circ = f(d_\circ) \leq \rho$. Hence $(\boldsymbol{y}_\circ, \rho)$ is a feasible solution of Problem (9.D : $d = d_\circ$). It follows that $\varphi(\boldsymbol{y}_\circ) \leq \rho$ and $\boldsymbol{b}^T\boldsymbol{y}_\circ \leq d_\circ$. In other words, $\boldsymbol{y}_\circ$ is a ρ-feasible solution of Problem (1.D), and the corresponding objective value is not larger that $d_\circ$.

Summing up the above, $\xi_\circ^{-1}\boldsymbol{x}_\circ$ is a feasible solution of (1.P) and $\boldsymbol{y}_\circ$ is a ρ-feasible solution of Problem (1.D) such that $\boldsymbol{c}^T\left(\xi_\circ^{-1}\boldsymbol{x}^\circ\right) \geq \boldsymbol{b}^T\boldsymbol{y}_\circ$. This situation is analogous to the traditional stopping criterion of the simplex method, ρ representing the usual pricing tolerance. As the simplex stopping rule is reliable in practice, we are going to accept the above situation as a criterion for near-optimality. – On the practical appropriateness of the simplex stopping rule, and in general, on the use of tolerances in simplex implementations, see, e.g., [18], [14], [13]. Moreover, scaling is discussed in [5] as a point of reference for absolute tolerances.

The practical suitability of the simplex stopping rule indicates a special characteristic of the function $f(d)$. Let $d^\rho \in \mathcal{D}$ be such that $f(d^\rho) = \rho$, and let us consider the linear function $\ell(d) := \frac{\rho}{d^\star - d^\rho}(d^\star - d)$. Of course $\ell(d^\rho) = f(d^\rho)$ and $\ell(d^\star) = f(d^\star)$, hence from the convexity of f, we have

$$f(d) \geq \ell(d) \quad (d \in \mathcal{D},\ d \leq d^\rho). \tag{11}$$

In simplex technical terms, the absolute value $|\ell'|$ of the steepness is the ratio between the pricing tolerance ρ and the near-optimality measure $d^\star - d^\rho$. As we assume that the traditional simplex stopping rule is usable in case of Problem (1.P), we conclude $|\ell'| \gg 0$.

4.1 A Newton method-based approximation scheme

The Newton method is a natural means of finding $d_\circ$ such that the near-optimality criterion (10) holds. A starting iterate $d_1 \in \mathcal{D},\ d_1 < d^\star$ is usually easy to find. (E.g., setting d_1 to be the objective value belonging to a feasible solution of (1.P) is often a good choice. Alternatively, $d_1 = 0$ is suitable in many cases.) Given an iterate $d_k \in \mathcal{D},\ d_k < d^\star$, Problem (9.P : $d = d_k$) can be solved by the regularized simplex method described in Section 2. – Though the present problem slightly differs from the one considered in Section 2, the level regularization can still be applied: Problem (9.D) can still be formulated as the minimization of the convex function over a polyhedron. Though this polyhedron may be unbounded, the level method can be implemented, and efficiency can be expected, on the basis of Remarks 4 and 3.

In order to further enhance efficiency, we are going to adapt the approximation scheme of the constrained Newton method of Lemaréchal, Nemirovskii, and Nesterov [11] to the present problem. The idea is that we need not construct exact supporting linear functions to f. Given iterate d_k, it suffices to find a linear function $l_k(d)$ satisfying

$$f(d) \geq l_k(d)\ (d \in \mathcal{D}) \quad \text{and} \quad l_k(d_k) \geq \ \beta\, f(d_k), \tag{12}$$

where β is a constant, $1/2 < \beta < 1$.

In this scheme, we are going to apply Algorithm 2 as a subroutine, for the approximate solution of problems of type (9).

An approximate Newton step. Given iterate $d_k \in \mathcal{D},\ d_k < d^\star$, let us approximately solve Problem (9.P: $d = d_k$) using Algorithm 2. It generates a sequence of basic feasible solutions with increasing objective values. The current objective value is denoted by ϕ_{bound}. Moreover, a dual feasible vector $\left(\boldsymbol{y}^{best}, \phi^{best}\right)$ is maintained that provides an upper bound ϕ^{best} to the optimum. In order to terminate early with an approximate solution, let us modify the near-optimality check in step 2 of Algorithm 2 as follows:

2.2′ Return check.

(i) If $\phi^{best} \le \rho$ then return with a near-optimal solution to Problem (1.P).

(ii) If $\phi_{bound} \ge \beta\phi^{best}$ then return with the next iterate d_{k+1}.

We are going to justify the above return criteria.

The optimum of Problem (9.P: $d = d_k$) is $f(d_k)$ and ϕ^{best} is an upper bound to it. Hence criterion *(i)* entails $f(d_k) \le \rho$, which in turn means that d_k satisfies the near-optimality criterion (10). Using the current basic feasible solution $(\boldsymbol{x}_\circ, \xi_\circ)$ of Problem (9.P: $d = d_k$), we can construct a near-optimal feasible solution $\xi_\circ^{-1}\boldsymbol{x}_\circ$ of Problem (1.P).

Let us assume now that the procedure has not stopped, but *(ii)* holds. Let $(\boldsymbol{x}_k, \xi_k)$ denote the current feasible basic solution of Problem (9.P : $d = d_k$), and let $l_k(d) :=\ \boldsymbol{c}^T\boldsymbol{x}_k - d\,\xi_k$. Clearly $l_k(d_k) = \phi_{bound}$ and $l_k(d) \le f(d)$ $(d \in \mathcal{D})$. From *(ii)*, we get $l_k(d_k) \ge \beta\phi^{best} \ge \beta f(d_k)$. Hence the linear function l_k satifies (12). The next iterate will be the solution of $l_k(d) = 0$, i.e., $d_{k+1} := \boldsymbol{c}^T\left(\xi_k^{-1}\boldsymbol{x}_k\right)$.

Observation 9 *The above subroutine returns after a relatively few iterations.*

Indeed, let us assume that neither *(i)* nor *(ii)* holds. Then we have

$$\phi^{best} - \phi_{bound}\ >\ \phi^{best} - \beta\phi^{best}\ =\ (1-\beta)\phi^{best}\ >\ (1-\beta)\rho.$$

But Remark 3 states that the gap $\phi^{best} - \phi_{bound}$ decreases at a brisk rate.

Convergence of the approximation scheme. Let the iterates $d_1, \ldots, d_{k+1}$ and the linear functions $l_1, \ldots, l_{k+1}$ be as defined above. We assume that the procedure did not stop before step $(k+1)$. Then the function values $l_1(d_1), \ldots, l_k(d_k)$ are all positive, and the derivatives $l_1', \ldots, l_k'$ are all negative.

Since $l_k(d_{k+1}) = l_k(d_k) + (d_{k+1} - d_k)\, l_k'$ and $l_k(d_{k+1}) = 0$ by definition, it follows that $d_{k+1} - d_k = \frac{l_k(d_k)}{|l_k'|}$. Using this, we get

$$l_{k+1}(d_k)\ =\ l_{k+1}(d_{k+1}) + (d_k - d_{k+1})\, l_{k+1}'\ =\ l_{k+1}(d_{k+1}) + \frac{l_k(d_k)}{|l_k'|}\,|l_{k+1}'|.$$

Hence

$$\frac{l_{k+1}(d_k)}{l_k(d_k)} = \frac{l_{k+1}(d_{k+1})}{l_k(d_k)} + \frac{|l_{k+1}'|}{|l_k'|}. \tag{13}$$

From (12), we have $l_k(d_k) \ge \beta\, f(d_k)$ and $f(d_k) \ge l_{k+1}(d_k)$, hence $\frac{1}{\beta} \ge \frac{l_{k+1}(d_k)}{l_k(d_k)}$. Substituting this into (13), we get

$$\frac{1}{\beta}\ \ge\ \frac{l_{k+1}(d_{k+1})}{l_k(d_k)} + \frac{|l_{k+1}'|}{|l_k'|}\ \ge\ 2\sqrt{\frac{l_{k+1}(d_{k+1})\,|l_{k+1}'|}{l_k(d_k)\,|l_k'|}}.$$

(This is the well known inequality between means.) It follows that

$$\frac{1}{4\beta^2}\, l_k(d_k)\,|l_k'|\ \ge\ l_{k+1}(d_{k+1})\,|l_{k+1}'|.$$

Let $\gamma := \frac{1}{4\beta^2} < 1$. By induction, we get

$$\gamma^k\, l_1(d_1)\,|l_1'|\ \ge\ l_{k+1}(d_{k+1})\,|l_{k+1}'|. \tag{14}$$

Let us assume that d_{k+1} does not satisfy the near-optimality criterion (10). Then we have $d_{k+1} < d^\rho$, and hence from (12) and (11) follows $l_{k+1}(d_{k+1}) \geq \beta\, f(d_{k+1}) \geq \beta\, \ell(d_{k+1})$. As moreover we have $l_{k+1}(d^\star) \leq f(d^\star) = 0 = \ell(d^\star)$, it follows that $|l'_{k+1}| \geq \beta\, |\ell'|$. We had concluded $|\ell'| \gg 0$ from the applicability of the traditional simplex stopping rule. Hence (14) assures linear convergence of the function values $l_{k+1}(d_{k+1})$, provided the traditional simplex stopping rule is usable in the case of Problem (1.P). Linear convergence of the values $f(d_{k+1})$ follows from (12).

Remark 10 *This Newton method-based approximation scheme can be considered as a regularized simplex method applied to the Problem (9.P). The parameter d is tuned in accordance with the progress of the procedure. Let $\mathcal{B}_k$ denote the basis at which the solution of Problem (9.P : $d = d_k$) was terminated. The solution of the next Problem (9.P : $d = d_{k+1}$) can of course be started from $\mathcal{B}_k$.*

There is an analogy between the pair of problems (9) and the self-dual form treated by Dantzig's self-dual parametric simplex method described in [1]. But the solution method we propose is different; it is a Newton-type method in contrast to Dantzig's parametric approach.

5 Conclusion and discussion

We proposed a regularized version of the simplex method.

First we considered a special problem class, consisting of the duals of ball-fitting problems – the latter type of problem is about finding the largest ball that fits into a given polyhedron. For problems of this class, the simplex method can be implemented as a cutting-plane method that approximates a convex polyhedral objective function. We applied the level method of Lemaréchal, Nemirovskii, and Nesterov [11] to minimize the convex polyhedral objective function.

From the primal perspective, this procedure is interpreted as a regularized simplex method. Regularization is performed in the dual space and only affects the process through the pricing mechanism. Hence the resulting method moves among basic solutions. – This method bears an analogy to the gravitational method of Murty [15] and its modifications, an overview of whom can be found in [26].

This regularization proved effective in our computational study. Considering aggregated data of our test problems, the total iteration count decreased by 60%. The solution time did not decrease due to the need of solving quadratic subproblems in the regularized simplex method. However, ours is but a basic implementation. Possible enhancements are discussed in Section 3.4.

Our present results summed up in Remark 8 show that the practical efficiency estimate of the level method applies also in case of the special polyhedral objectives of the ball-fitting problems. We think it important in view of further applications of this regularized simplex method.

For the solution of general linear programming problems, we propose a Newton-type approach which requires the solution of a sequence of ball-fitting problems. – As observed in the Introduction, such problems are much more easily solved than general linear programming problems. Moreover, level regularization can be applied to the ball-fitting problems. – The efficiency of this approach is enhanced by using approximate solutions of the ball-fitting problems. We adapted an approximation scheme and convergence proof of Lemaréchal, Nemirovskii, and Nesterov [11] to the present problem. In the adaptation of the convergence proof, we exploit a well-known and widely accepted observation concerning practical simplex implementations. The resulting proof states linear convergence that is interesting even from a practical point of view.

Let us mention that regularization and trust region methods have been successfully applied before to large-scale convex polyhedral problems. An especially fruitful field is stochastic programming, where such polyhedral objectives are obtained as expectations. See [22], [23], [12], [10], [27], [17], [24], [6] for regularization and trust region methods for the solution of two-stage linear stochastic programming problems; and [8], [25], [6] for the solution of a special risk-averse problem.

A different approach for a theoretical convergence proof was proposed by Michael Dempster. In a former paper [3], Dempster and Merkovsky present a geometrically convergent cutting-plane method for the minimization of a continuously differentiable, strictly convex function. In course of a discussion on applying a regularized cutting-plane method to a special risk-averse problem, Dempster [2] conjectured that the convergence proof of [3] could be extended to that method. We believe that further extension to the regularized solution of ball-fitting problems would also be possible. (This would result a theoretical bound stronger than that mentioned in Remark 3.)

References

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

[2] Dempster, M.A.H. (2007). *Private communication.*

[3] Dempster, M.A.H. and R.R. Merkovsky (1995). A practical geometrically convergent cutting plane algorithm. *SIAM Journal on Numerical Analysis* **32**, 631-644.

[4] Elble, J.M. (2010). *Computational Experience with Linear Optimization and Related Problems.* PhD dissertation, advisor: N. Sahinidis. The University of Illinois at Urbana-Champaign, Illinois, USA.

[5] Elble, J.M. and N.V. Sahinidis (2012). Scaling linear optimization problems prior to application of the simplex method. *Computational Optimization and Applications* **52**, 345-371.

[6] Fábián, C.I. (2012). Computational aspects of risk-averse optimisation in two-stage stochastic models. *Optimization Online,* August 2012.

[7] Fábián, C.I., K. Eretnek, and O. Papp (2012). Towards a regularised simplex method. In: *Applied Mathematical Optimization and Modelling*. Extended Abstracts of the APMOD 2012 Conference (L. Suhl, G. Mitra, C. Lucas, A. Koberstein, and L. Beckmann, editors), 3-9. Volume 8 of the DSOR Contributions to Information Sytems. DS&OR Lab, University of Paderborn, Germany.

[8] Fábián, C.I., G. Mitra, D. Roman, and V. Zverovich. (2011). An enhanced model for portfolio choice with SSD criteria: a constructive approach. *Quantitative Finance* **11**, 1525-1534.

[9] Fábián, C.I., O. Papp, and K. Eretnek (2011). Implementing the simplex method as a cutting-plane method. *Optimization Online,* July 2011.

[10] Fábián, C.I. and Z. Szőke (2007). Solving two-stage stochastic programming problems with level decomposition. *Computational Management Science* **4**, 313-353.

[11] Lemaréchal, C., A. Nemirovskii, and Yu. Nesterov (1995). New variants of bundle methods. *Mathematical Programming* **69**, 111-147.

[12] Linderoth, J. and S. Wright (2003). Decomposition algorithms for stochastic programming on a computational grid. *Computational Optimization and Applications* **24**, 207-250.

[13] Maros, I. (2003). *Computational Techniques of the Simplex Method.* International Series in Operations Research & Management Science. Kluwer Academic Publishers, Boston.

[14] Murtagh, B.A. (1981). *Advanced Linear Programming: Computation and Practice.* McGraw-Hill, New York.

[15] Murty, K.G. (1986). The gravitational method of linear programming. *Technical Report* No. 8619, Department of Industrial and Operations Engineering, The University of Michigan, Ann Arbor, Michigan, USA.

[16] Nemirovski, A. (2005). *Lectures in modern convex optimization.* ISYE, Georgia Institute of Technology.

[17] Oliveira, W. C. Sagastizábal, and S. Scheimberg. (2011). Inexact bundle methods for two-stage stochastic programming. *SIAM Journal on Optimization* **21**, 517-544.

[18] Orchard-Hays, W. (1968). *Advanced Linear-Programming Computing Techniques.* McGraw-Hill, New York.

[19] Pan, P.-Q. (2007). Nested pricing for the simplex algorithm: An empirical evaluation. *Optimization Online,* March 2007.

[20] Pan, P.-Q. (2008). Efficient nested pricing in the simplex algorithm. *Operations Research Letters* **36**, 309-313.

[21] Richtárik, P. (2012). Approximate level method for nonsmooth convex minimization. *Journal of Optimization Theory and Applications* **152**, 334-350.

[22] Ruszczyński, A. (1986). A regularized decomposition method for minimizing a sum of polyhedral functions. *Mathematical Programming* **35**, 309-333.

[23] Ruszczyński, A. and A. Świętanowski (1997). Accelerating the Regularized Decomposition Method for two-stage stochastic linear problems. *European Journal of Operational Research* **101**, 328-342.

[24] Sagastizábal, C. and M. Solodov (2012). Solving generation expansion problems with environmental constraints by a bundle method. *Computational Management Science* **9**, 163-182.

[25] Sun, H., H. Xu, R. Meskarian, and Y. Wang. (2012). Exact penalization, level function method and modified cutting-plane method for stochastic programs with second order stochastic dominance constraints. *Optimization Online,* February 2012.

[26] Terlaky, T. and S. Zhang (1993). Pivot rules for linear programming: a survey on recent theoretical developments. *Annals of Operations Research* **46**, 203-233.

[27] Zverovich, V., C.I. Fábián, F. Ellison, and G. Mitra (2012). A computational study of a solver system for processing two-stage stochastic linear programming problems. *Mathematical Programming Computation* **4**, 211-238.