Implementing the simplex method as a cutting-plane method

Csaba I. Fábián*† Olga Papp*† Krisztián Eretnek†

Abstract

We show that the simplex method can be interpreted as a cutting-plane method, assumed that a special pricing rule is used. This approach is motivated by the recent success of the cutting-plane method in the solution of special stochastic programming problems.

We compare the classic Dantzig pricing rule and the rule that derives from the cutting-plane approach. We focus on the special linear programming problem of finding the largest ball that fits into a given polyhedron.

Keywords. Simplex method, cutting-plane method, linear programming, convex programming, stochastic programming.

1 Introduction

Due to their relative simplicity and practical effectiveness, simplex-type and cutting-plane methods have been intriguing researchers since the invention of these methods.

The simplex method was developed by Dantzig in 1947 and published in 1951 [3]. It was the first effective method for the solution of linear programming problems, and to this day it is considered an effective practical method, the method of choice for many problem types. A practical state-of-art summary of simplex computational techniques with many experimental observations can be found in Maros's book [22]. On the other hand, worst-case complexity of the simplex method is exponential, see [13]. The gap between practical effectiveness and theoretical complexity was bridged by Borgwardt [1] and Spielman and Teng [30].

Cutting-plane methods for convex problems were proposed by Kelley [12], and Cheney and Goldstein [2] in 1959-1960. These methods are considered fairly efficient for quite general problem classes. However, according to our knowledge, efficiency estimates only exist for the continuously differentiable, strictly convex case. An overview can be found in Dempster and Merkovsky [5], where a geometrically convergent version is also presented.

In recent years, cutting-plane approaches proved remarkably effective for certain special stochastic programming problems; see [14] for problems with Integrated Chance Constraints, [16] for minimising Conditional Value at Risk, and [7], [28], [18] for problems involving Second-order Stochastic Dominance. Each of these stochastic programming problems can be formulated as linear programming problems having excessively many constraints. A cutting-plane method for the primal problem can be viewed as a special simplex method for the dual problem.

In this paper we examine the special pricing rule that derives from the cutting-plane approach. In section 2 we consider a general linear programming problem, and the corresponding cutting-plane method that approximates the feasible polyhedron. In section 3 we consider the special linear programming problem of finding the largest ball that fits into a given polyhedron. This type of problem can be formulated as minimisation of a polyhedral convex function, and the corresponding cutting-plane method approximates this objective function.

*Institute of Informatics, Kecskemét College, 10 Izsáki út, 6000 Kecskemét, Hungary. E-mail: fabian.csaba@gamf.kefo.hu.

†Eötvös Loránd University, Budapest, Hungary

A computational study is presented in section 4. We compare the classic Dantzig pricing rule and the cutting-plane pricing rule on general linear programming problems, and also on ball-fitting problems. The results show characteristic differences. Conclusions are drawn in section 5.

Some of the proofs presented in sections 2 and 3 exploit the theoretical equivalence of the primal and dual simplex methods. To put this equivalence more precisely, a dual simplex method applied to the dual problem can be interpreted as a simplex method applied to the primal problem.

1.1 Notes on the equivalence of the primal and the dual simplex methods

The dual simplex method was developed by Lemke in 1954 [17]. According to our knowledge, the equivalence of the simplex method and the dual simplex method was first noticed by Prékopa in the nineteen sixties [27], inspired by the essays [15]. Prékopa presented the equivalence in his 1967 - 1969 linear programming courses at the Eötvös Loránd University of Budapest. This topic was later left out of the curriculum, but some of the students noticed the equivalence by themselves: Terlaky [31] in the nineteen seventies, and one of the authors of this paper in the nineteen eighties.

Concerning written publications, Prékopa [25] treated the simplex method and the dual simplex method in a unified form, as row and column transformations, respectively, on an appropriate square matrix. Equivalence of simplex and dual simplex steps is stated in Prékopa [26] in the following form: primal transformation formulas result the same tableau that we obtain when we first carry out the dual transformation formulas and then take the negative transpose of the tableau. Padberg [24] observes the equivalence of the primal and the dual simplex methods with the remark that the numerical behaviour of the two methods may be very different. Vanderbei [33] states the equivalence in the following form: there is a negative transpose relationship between the primal and the dual problem, provided that the same set of pivots that were applied to the primal problem are also applied to the dual.

In the appendix we also give a brief proof to the primal-dual equivalence. (We think the simplicity of this proof justifies its inclusion.) This proof is based on a special relationship between primal and dual bases. This relationship was also observed in former works: an asymmetric form is described in Kall and Mayer [11], and a symmetric form in [9].

1.2 Notation and clarification of terms

Let us consider the primal-dual pair of problems

$$\begin{array}{ll|ll} & A\boldsymbol{x} \le \boldsymbol{b} & & A^T\boldsymbol{y} \ge \boldsymbol{c} \\ (1.\mathrm{P}) & \boldsymbol{x} \ge \boldsymbol{0} & (1.\mathrm{D}) & \boldsymbol{y} \ge \boldsymbol{0} \\ & \max\ \boldsymbol{c}^T\boldsymbol{x}, & & \min\ \boldsymbol{b}^T\boldsymbol{y}, \end{array} \qquad (1)$$

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:

$$\begin{array}{ll|ll} & A\boldsymbol{x} + I\boldsymbol{u} = \boldsymbol{b} & & \boldsymbol{y}^TA - \boldsymbol{v}^TI = \boldsymbol{c}^T \\ (2.\mathrm{P}) & (\boldsymbol{x}, \boldsymbol{u}) \ge \boldsymbol{0} & (2.\mathrm{D}) & (\boldsymbol{y}, \boldsymbol{v}) \ge \boldsymbol{0} \\ & \max\ \boldsymbol{c}^T\boldsymbol{x}, & & \min\ \boldsymbol{y}^T\boldsymbol{b}, \end{array} \qquad (2)$$

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

Notation for the primal 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 $\left(A^T\boldsymbol{y} - \boldsymbol{c},\, \boldsymbol{y} \right) \in \mathbb{R}^{n+m}$.

Notation for the dual problem. A basis $\mathcal{C}$ of the dual problem is an appropriate subset of the rows of the matrix $\begin{pmatrix} A \\ -I \end{pmatrix}$.

Let $\boldsymbol{x}$ denote the shadow-price vector belonging to basis $\mathcal{C}$. This is determined by the equation $C\boldsymbol{x} = (\boldsymbol{b}, \boldsymbol{0})_{\mathcal{C}}$, where C is the basis matrix, and the right-hand side expression denotes the basic part of the vector $(\boldsymbol{b}, \boldsymbol{0})$.

The reduced-cost vector belonging to basis $\mathcal{C}$ is $(\, \boldsymbol{b} - A\boldsymbol{x},\, \boldsymbol{x}\,) \in \mathbb{R}^{m+n}$. Basis $\mathcal{C}$ is said to be dual feasible if the reduced-cost vector is non-negative.

Applying the simplex method to the primal problem. We assume that a feasible starting basis is known, and use the phase-II simplex method. This is an iterative procedure that visits feasible bases. The method stops if the current reduced cost vector is non-negative. Otherwise a simplex step is performed: A column with a negative reduced cost is selected for entering the basis. The pivot element must be positive, and the selection of the outgoing column must guarantee the feasibility of the successive basis. We view the simplex method as a sequence of steps satisfying the above requirements. An implementation of the simplex method should of course uniquely determine the entering and the leaving column in each step. The technical jargon for the rule determining the entering column is pricing rule. In this paper we compare two pricing rules: the classic one that selects the column with the minimal (negative) reduced cost, and a new rule derived from the cutting-plane approach.

Applying the dual simplex method to the dual problem. We assume that a dual feasible starting basis is known, and use the phase-II dual simplex method. This is an iterative procedure that visits dual feasible bases. The method stops if the current basic solution vector is non-negative. Otherwise a dual simplex step is performed: A negative component is selected in the current basic solution, and the corresponding basic column leaves the basis. The pivot element must be negative, and the selection of the incoming column must guarantee the dual feasibility of the successive basis. We view the dual simplex method as a sequence of steps satisfying the above requirements. An implementation of the dual simplex method should of course uniquely determine the leaving and the entering column in each step.

2 A cutting-plane method approximating the feasible domain

Let $\boldsymbol{a}_j$ $(j = 1, \ldots, n)$ denote the columns of matrix A. Problem (1.D) can be written in the form

$$\min_{\boldsymbol{y} \in Y} \boldsymbol{b}^T\boldsymbol{y} \qquad \text{where} \quad Y := \left\{ \boldsymbol{y} \in \mathbb{R}^m \,\middle|\, \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{a}_j^T\boldsymbol{y} \geq c_j\ (j = 1, \ldots, n) \right\}. \tag{3}$$

Let us solve problem (3) by the cutting-plane method. This is an iterative method that builds a model of the feasible domain in the form

$$Y_{\mathcal{J}} := \left\{ \boldsymbol{y} \in \mathbb{R}^m \,\middle|\, \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{a}_j^T\boldsymbol{y} \geq c_j\ (j \in \mathcal{J}) \right\}, \tag{4}$$

where $\mathcal{J} \subseteq \{1, \ldots, n\}$ denotes the set of cuts found so far. The model problem

$$\min_{\boldsymbol{y} \in Y_{\mathcal{J}}} \boldsymbol{b}^T\boldsymbol{y} \tag{5}$$

is then solved and a new cut is added as needed. In this paper we assume that each occurring model problem has finite optimum. – This is achieved by a broad enough initial set of cuts as we assumed that the problem (1.D) has a finite optimum. Let us rehearse steps of the cutting-plane method as applied to the present problem:

Algorithm 1 *A cutting-plane method approximating the feasible polyhedron.*

```
1.0 Initialise.
    Let $\mathcal{J} \subset \{1, \dots, n\}$ be such that the model problem (5) has a finite optimum.

1.1 Solve the model problem (5).
    Let $\boldsymbol{y}^\star$ be an optimal solution.

1.2 Check.
    If $\boldsymbol{y}^\star \in Y$ then stop.

1.3 Update the model problem.
    Select deepest cut, i.e., $j^\star \in \arg\min_{1 \le j \le n} \{ \boldsymbol{a}_j^T \boldsymbol{y}^\star - c_j \}$.

    Let $\mathcal{J} := \mathcal{J} \cup \{j^\star\}$, and repeat from step 1.1.
```

The model problem can be formulated as (6.D), below; a relaxation of (1.D). This is a linear programming problem that makes a primal-dual pair with (6.P).

$$
\begin{array}{ll|ll}
 & A_{\mathcal{J}} \boldsymbol{x}_{\mathcal{J}} \le \boldsymbol{b} & & A_{\mathcal{J}}^T \boldsymbol{y} \ge \boldsymbol{c}_{\mathcal{J}} \\
(6.\mathrm{P}) & \boldsymbol{x}_{\mathcal{J}} \ge \boldsymbol{0} & (6.\mathrm{D}) & \boldsymbol{y} \ge \boldsymbol{0} \\
 & \max\ \boldsymbol{c}_{\mathcal{J}}^T \boldsymbol{x}_{\mathcal{J}}, & & \min\ \boldsymbol{b}^T \boldsymbol{y},
\end{array}
\tag{6}
$$

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 $j \in \mathcal{J}$. According to our assumption, both problems in (6) have a finite optimum.

Let us consider the series of model problems (6.D) occurring in the course of the cutting-plane process. Each successor problem in this series is obtained by adding a constraint to its predecessor. Having solved the predecessor problem by a simplex-type method, a dual feasible starting basis is easily constructed for the successor problem.

Remark 2 *Due to this warm-start potential, the dual simplex method is an effective and time-honoured means of solving the model problems in the cutting-plane process.*

Instead of solving the current model problem (6.D) by the dual simplex method, we can solve (6.P) by the simplex method. The following algorithm is a re-formulation of the previous one. Actually it is more specific in some points, like the handling of bases. We assume that a feasible basis is known for the primal problem (1.P).

Algorithm 3 *A re-formulation of algorithm 1.*

```
3.0 Initialise.
    Let $\mathcal{B} \subset \{1, \dots, n+m\}$ be a feasible starting basis for the primal problem (1.P).
    Let $\mathcal{J} \subset \{1, \dots, n\}$ consist of the non-slack columns in $\mathcal{B}$.

3.1 Solve the model problem (6.P) by the simplex method starting from basis $\mathcal{B}$.
    Let $\mathcal{B}^\star$ denote the optimal basis, and $\boldsymbol{y}^\star$ the corresponding shadow-price vector.

3.2 Check.
    If $\boldsymbol{a}_j^T \boldsymbol{y}^\star \ge c_j$ holds for $1 \le j \le n$, then stop.
```

3.3 Update the model problem.

Select $j^\star \in \arg\min_{1 \le j \le n} \left\{ \boldsymbol{a}_j^T \boldsymbol{y}^\star - c_j \right\}$.

Let $\mathcal{J} := \mathcal{J} \cup \{j^\star\}$, $\mathcal{B} := \mathcal{B}^\star$, and repeat from step *3.1.*

Observation 4 *Algorithm 3 can be interpreted as solving problem (1.P) by the simplex method.*

Indeed, $\mathcal{B}^\star$ is a feasible basis for problem (1.P). Step *3.2* performs the regular optimality check on the columns of the matrix A. Slack columns need not be checked because they are included in the model problem. Step *3.3* selects a column having minimal (negative) reduced cost. This is the classic simplex pricing rule. (Having moved to step *3.1* from step *3.3*, clearly $j^\star$ will be the first column entering the basis in course of solving the model problem.) The above classic simplex pricing rule is also called Dantzig pricing rule, after the seminal book [4].

Assume that in the simplex method of step *3.1,* the classic pricing rule is used: the column having minimal reduced cost enters the basis. This results a special pricing rule applied in the simplex method of algorithm 3:

Definition 5 Cutting-plane pricing rule. *At any stage of the method, let $\mathcal{J} \subset \{1, \dots, n\}$ denote the set of the indexes of those $\boldsymbol{x}$ variables that have already been basic. In the subsequent pricing pass, columns in $\mathcal{J} \cup \{n+1, \dots, n+m\}$ are preferred to columns in $\{1, \dots, n\} \setminus \mathcal{J}$, as candidates for entering basis.*

If there are columns in the set $\mathcal{J} \cup \{n+1, \dots, n+m\}$ with negative reduced costs, then we select the one having minimal reduced cost. If each column in $\mathcal{J} \cup \{n+1, \dots, n+m\}$ has a non-negative reduced cost, then we consider the set $\{1, \dots, n\} \setminus \mathcal{J}$. From the latter set we also select the column having minimal reduced cost.

As mentioned in remark 2, the dual simplex method is traditionally used for the solution of the model problems in the cutting-plane process of algorithm 1. We now see that the whole cutting-plane process can be implemented as a single dual simplex method. This follows from observation 4, and the general equivalence between the simplex method applied to the primal problem on the one hand, and the dual simplex method applied to the dual problem on the other hand.

3 A cutting-plane method approximating the objective function

In this section we deal with special problems of the form (7). For motivation, let us consider the primal-dual pair of problems (1), and suppose we want to find a 'central' point in the feasible domain of problem (1.D). Constraints in (1.D) have the form $\boldsymbol{a}_j^T \boldsymbol{y} \ge c_j$ $(j = 1, \dots, n)$. Given a tolerance $\zeta > 0$, a vector $\boldsymbol{y}$ satisfies the jth constraint with this tolerance if $\boldsymbol{a}_j^T \boldsymbol{y} + \zeta \ge c_j$ holds. We want to find a point $\boldsymbol{y} \ge \boldsymbol{0}$ that satisfies each constraint with minimum tolerance. This is described by the following problem:

$$\begin{array}{l} A^T \boldsymbol{y} + \zeta \boldsymbol{1} \ge \boldsymbol{c} \\ \boldsymbol{y} \ge \boldsymbol{0}, \ \zeta \in \mathbb{R} \\ \min \ \zeta, \end{array} \tag{7}$$

where $\boldsymbol{1} = (1, \dots, 1)^T$ 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 (7) finds a maximum-radius ball that fits into the polyhedron $\left\{ \boldsymbol{y} \,\middle|\, A^T \boldsymbol{y} \ge \boldsymbol{c} \right\}$, such that the centre of the ball falls into the positive orthant.

Problem (7) is always feasible as ζ may take positive values. To ensure a bounded objective, we add the constraint $\boldsymbol{1}^T \boldsymbol{y} \le d$ where d is a (large) constant. For reasons of symmetry, we add this constraint in the form $-\boldsymbol{1}^T \boldsymbol{y} \ge -d$. We get (8.D) that makes a primal-dual pair with (8.P) :

$$\text{(8.P)} \quad \begin{array}{l} A\boldsymbol{x} - \xi \boldsymbol{1} \le \boldsymbol{0} \\ \boldsymbol{1}^T \boldsymbol{x} = 1 \\ \boldsymbol{x} \ge \boldsymbol{0}, \ \xi \ge 0 \\ \max \ \boldsymbol{c}^T \boldsymbol{x} - d\xi, \end{array} \qquad \Bigg| \qquad \text{(8.D)} \quad \begin{array}{l} A^T \boldsymbol{y} + \zeta \boldsymbol{1} \ge \boldsymbol{c} \\ -\boldsymbol{1}^T \boldsymbol{y} \ge -d \\ \boldsymbol{y} \ge \boldsymbol{0}, \ \zeta \in \mathbb{R} \\ \min \ \zeta. \end{array} \tag{8}$$

The above pair of problems are similar to those treated by Dantzig's self-dual parametric simplex method described in [4]. But the solution method we propose is different; in our case, e.g., d is constant.

Problem (8.D) can be formulated as minimisation of a polyhedral convex function over a simplex, formally

$$\min_{\boldsymbol{y}\in U} \varphi(\boldsymbol{y}) \quad \text{where} \quad \varphi(\boldsymbol{y}) := \max_{1\leq j\leq n} \left\{ c_j - \boldsymbol{a}_j^T \boldsymbol{y} \right\}, \quad U := \left\{ \boldsymbol{y} \in \mathrm{I\!R}^m \,\middle|\, \boldsymbol{y} \geq \boldsymbol{0},\, \boldsymbol{1}^T \boldsymbol{y} \leq d \right\}. \tag{9}$$

Clearly $(\boldsymbol{y}^\star, \zeta^\star)$ is an optimal solution of problem (8.D) if and only if $\boldsymbol{y}^\star$ is an optimal solution, and $\zeta^\star$ is the optimal objective value, of problem (9).

We solve problem (9) by a cutting-plane method that builds the model function

$$\varphi_{\mathcal{J}}(\boldsymbol{y}) := \max_{j\in\mathcal{J}} \left\{ c_j - \boldsymbol{a}_j^T \boldsymbol{y} \right\}, \tag{10}$$

where $\mathcal{J} \subseteq \{1, \ldots, n\}$ denotes the set of cuts found so far. Let us rehearse steps of the cutting-plane method as applied to the present problem:

Algorithm 6 *A cutting-plane method approximating the objective function.*

```
6.0 Initialise.
    Let $\mathcal{J} \subset \{1, \ldots, n\}$, $\mathcal{J} \neq \emptyset$.
6.1 Solve the model problem $\min_U \varphi_{\mathcal{J}}$.
    Let $\boldsymbol{y}^\star$ be an optimal solution.
6.2 Check.
    If $\varphi_{\mathcal{J}}(\boldsymbol{y}^\star) \geq \varphi(\boldsymbol{y}^\star)$ then stop.
6.3 Update the model function.
    Select $j^\star \in \{1, \ldots, n\}$ such that $l(\boldsymbol{y}) = c_{j^\star} - \boldsymbol{a}_{j^\star}^T \boldsymbol{y}$ be a support function to $\varphi(\boldsymbol{y})$ at $\boldsymbol{y}^\star$.

    Let $\mathcal{J} := \mathcal{J} \cup \{j^\star\}$, and repeat from step 6.1.
```

The model problem in step *6.1* has an optimal solution since the feasible domain is bounded and the objective function is finite due to $\mathcal{J} \neq \emptyset$. The criterion of step *6.2* clearly implies that $\boldsymbol{y}^\star$ is an optimal solution of problem (9) because we have

$$\min_U \varphi \;\geq\; \min_U \varphi_{\mathcal{J}} \;=\; \varphi_{\mathcal{J}}(\boldsymbol{y}^\star) \;\geq\; \varphi(\boldsymbol{y}^\star).$$

The model problem can be formulated as (11.D), below; a relaxation of (8.D). This is a linear programming problem that makes a primal-dual pair with (11.P).

$$\begin{array}{ll|ll}
\text{(11.P)} & \begin{array}{l} A_{\mathcal{J}} \boldsymbol{x}_{\mathcal{J}} - \xi \boldsymbol{1} \leq \boldsymbol{0} \\ \boldsymbol{1}^T \boldsymbol{x}_{\mathcal{J}} = 1 \\ \boldsymbol{x}_{\mathcal{J}} \geq \boldsymbol{0},\ \xi \geq 0 \\ \max\ \boldsymbol{c}_{\mathcal{J}}^T \boldsymbol{x}_{\mathcal{J}} - d\xi, \end{array} & \text{(11.D)} & \begin{array}{l} A_{\mathcal{J}}^T \boldsymbol{y} + \zeta \boldsymbol{1} \geq \boldsymbol{c}_{\mathcal{J}} \\ -\boldsymbol{1}^T \boldsymbol{y} \geq -d \\ \boldsymbol{y} \geq \boldsymbol{0},\ \zeta \in \mathrm{I\!R} \\ \min\ \zeta, \end{array}
\end{array} \tag{11}$$

where the notation is the same as in the analogous case (6).

Since (11.D) has an optimal solution, it follows from LP duality that (11.P) has an optimal solution. Clearly $(\boldsymbol{y}^\star, \zeta^\star)$ is an optimal solution of the dual problem (11.D) if and only if $\boldsymbol{y}^\star$ is an optimal solution of the model problem $\min_U \varphi_{\mathcal{J}}$, and $\zeta^\star$ is the optimal objective value.

Like in the previous section, we will interpret the cutting-plane method as a simplex method. To this end, let us formulate the pair of problems (8) 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. Feasible bases for the primal problem are easily constructed; specifically, given a single x-column, one can select m out of the $m + 1$ artificial columns that make a feasible basis with the x-column.

Algorithm 7 *A simplex interpretation of algorithm 6.*

```
7.0 Initialise.
    Let $\mathcal{B} \subset \{1, \dots, n+1+m\}$ be a feasible basis for the primal problem (8.P).
    Let $\mathcal{J} \subset \{1, \dots, n\}$ consist of the $x$-columns belonging to $\mathcal{B}$.

7.1 Solve the model problem (11.P) by the simplex method starting form basis $\mathcal{B}$.
    Let $\mathcal{B}^\star$ denote the optimal basis, and $(\boldsymbol{y}^\star, \zeta^\star)$ the corresponding dual vector.

7.2 Optimality check.
    If $(\boldsymbol{y}^\star, \zeta^\star)^T(\boldsymbol{a}_j, 1) \geq c_j \ (j = 1, \dots, n)$ then stop.

7.3 Pricing.
    Let $j^\star \in \arg\min_{1 \leq j \leq n} \left\{ (\boldsymbol{y}^\star, \zeta^\star)^T(\boldsymbol{a}_j, 1) - c_j \right\}$.

    Let $\mathcal{J} := \mathcal{J} \cup \{j^\star\}$, $\mathcal{B} := \mathcal{B}^\star$, and repeat from step 7.1.
```

Algorithm 7 is clearly a specific version of algorithm 6. (In order to see the equivalence of the respective second and third steps, let us just substitute $\zeta^\star$ for $\varphi_{\mathcal{J}}(\boldsymbol{y}^\star)$ in algorithm *6.*)

Observation 8 *Algorithm 7 describes a simplex method applied to problem (8.P).*

Specifically, the pricing in step *7.3* selects a column having minimal (negative) reduced cost.

Assume that in the simplex method of step *7.1,* the classic pricing rule is used: the column having minimal reduced cost enters the basis. This is the cutting-plane pricing rule (definition 5) as applied to problem (8.P):

At any stage of the method, let $\mathcal{J} \subset \{1, \dots, n\}$ denote the set of the indexes of those $\boldsymbol{x}$ variables that have already been basic. In the subsequent pricing pass, columns in $\mathcal{J} \cup \{n+1, \dots, n+1+m\}$ are preferred to columns in $\{1, \dots, n\} \setminus \mathcal{J}$, as candidates for entering basis.

If there are columns in the set $\mathcal{J} \cup \{n+1, \dots, n+1+m\}$ with negative reduced costs, then we select the one having minimal reduced cost. If each column in $\mathcal{J} \cup \{n+1, \dots, n+1+m\}$ has a non-negative reduced cost, then we consider the set $\{1, \dots, n\} \setminus \mathcal{J}$. From the latter set we also select the column having minimal reduced cost.

Remark 9 *Assume now that in the simplex method of step 7.1, we use the above cutting-plane pricing rule. This results a further pricing rule in the simplex method of algorithm 7.*

This further pricing rule can of course be applied in step 7.1 ...

Summing up all these, we obtain a recursive structure with a respective precedence set at each level of the recursion.

Corollary 10 *The cutting-plane algorithm 6 can be implemented as a single dual simplex method applied to problem (8.D).*

Proof. According to observation 8, algorithm 6 can be implemented as a simplex method applied to problem (8.P). This in turn is equivalent to a dual simplex method applied to the dual problem (8.D).

In the appendix we give a proof to this equivalence. To keep discussion simple, though, we consider pairs of problems having only inequality constraints, in the form of (1). It remains to be seen that the result is applicable to the present pair of problems (8).

First we may assume that the system of inequalities $A^T\boldsymbol{y} \geq \boldsymbol{c}$, $-\boldsymbol{1}^T\boldsymbol{y} \geq -d$, $\boldsymbol{y} \geq \boldsymbol{0}$ has no feasible solution. This can be achieved by adding a large number γ to each component of the vector $\boldsymbol{c}$ in the pair of problems (8). The polyhedral problem (9) changes accordingly, the modified objective function being $\varphi(\boldsymbol{y}) + \gamma$. Thus the cutting-plane sequence of algorithm 6 remains the same. The simplex step sequences for (8.P) also remain the same: Let us consider an arbitrary basis of problem (8.P), and let $(\boldsymbol{y}, \zeta)$ denote the corresponding dual vector. If we add the same number γ to each component of $\boldsymbol{c}$, then the dual vector will change to $(\boldsymbol{y}, \zeta + \gamma)$, hence the reduced cost components remain unchanged.

Under the infeasibility assumption, the constraint $\zeta \geq 0$ is redundant in problem (12.D), below:

$$\text{(12.P)} \quad \begin{array}{l} A\boldsymbol{x} - \xi\mathbf{1} \leq \mathbf{0} \\ \mathbf{1}^T\boldsymbol{x} \leq 1 \\ \boldsymbol{x} \geq \mathbf{0},\ \xi \geq 0 \\ \max\ \boldsymbol{c}^T\boldsymbol{x} - d\xi, \end{array} \quad \Bigg| \quad \text{(12.D)} \quad \begin{array}{l} A^T\boldsymbol{y} + \zeta\mathbf{1} \geq \boldsymbol{c} \\ -\mathbf{1}^T\boldsymbol{y} \geq -d \\ \boldsymbol{y} \geq \mathbf{0},\ \zeta \geq 0 \\ \min\ \zeta. \end{array} \tag{12}$$

It means that there exists an optimal solution $(\boldsymbol{x}^\star, \xi^\star)$ of (12.P) such that $\mathbf{1}^T\boldsymbol{x}^\star = 1$ holds. Problems (8.P) and (12.P) differ only in the latter problem having an extra column (the slack column belonging to the constraint $\mathbf{1}^T\boldsymbol{x} \leq 1$). Hence any series of simplex steps that solves (8.P) can also be considered as a series of simplex steps that solves (12.P). (The slack vector of the constraint $\mathbf{1}^T\boldsymbol{x} \leq 1$ never enters basis in the course of such a series.)

The pair of problems (12) has only inequality constraints, hence the discussion in the appendix is applicable: any series of simplex steps that solves (12.P) can be interpreted as a series of dual simplex steps that solves (12.D). Since the slack vector of the constraint $\mathbf{1}^T\boldsymbol{x} \leq 1$ never enters basis in the course of our primal series, it follows that the variable ζ is always basic in the course of the corresponding dual series. Hence this dual series can be interpreted as a series of dual simplex steps that solves (8.D). $\square$

4 Computational study

We compared the classic Dantzig pricing rule and the cutting-plane pricing rule on general linear programming problems, and also on ball-fitting problems. (The latter type of special problems were discussed in section 3.)

4.1 Implementation issues

We implemented a basic but usable form of the simplex method: revised method with the product form of the inverse, and multiple pricing. (The implementation was part of the preparation of the Informatics Master's thesis of the third author of this paper.) The optimality tolerance of $1e-5$ was used in the test runs.

We can either use the classic Dantzig pricing, or the cutting-plane pricing. We implemented the latter in a form slightly different from that of definition 5: In our implementation, column indices of problem (8.P) are divided into three categories: The first category contains the index of the variable ξ and the indices of the slack variables. The second category contains the indices of those $\boldsymbol{x}$ variables that have already been basic. The third category contains the indexes of those $\boldsymbol{x}$ variables that have not yet been basic. We prefer first-category indexes to second-category ones, and second-category indexes to third-category ones. Given a category, indices falling into this category are ordered according to reduced costs.

We did not implement the recursive pricing rule of remark 9. For our ball-fitting test problems, we do not expect any significant improvement from the recursive pricing over the cutting-plane pricing; because using the latter, it took only 2 pivot steps at the average to recover optimality in step *7.1* of the solution.

4.2 Test problems

We considered 59 test problems from the NETLIB collection (the smaller ones). These are listed in table 1.

Given a NETLIB problem, we constructed a problem of the form (8.P) using only the matrix A and the objective vector $\boldsymbol{c}$ of the NETLIB problem. The constant d was always set so large that increasing d would not change the optimal basis of (8.P). Assumed each A-column has an Euclidean norm of 1, the dual problem (8.D) can be interpreted as fitting the largest ball into the polyhedron $\left\{\boldsymbol{y} \,\middle|\, A^T\boldsymbol{y} \geq \boldsymbol{c}\right\}$, such that the centre of the inscribed ball falls into the simplex $\left\{\boldsymbol{y} \,\middle|\, \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{1}^T\boldsymbol{y} \leq d\right\}$. Though we did not scale the columns, we dub the problem (8.P) the *max-ball variant* of the relevant NETLIB problem.

Given a NETLIB problem, we also constructed a problem of the form (13.P), below, using only the matrix A and the objective vector $\boldsymbol{c}$ of the NETLIB problem.

$$\begin{array}{rl|rl}
\text{(13.P)} & \begin{array}{l} A\boldsymbol{x} + \xi\boldsymbol{1} \geq \boldsymbol{0} \\ \boldsymbol{1}^T\boldsymbol{x} = 1 \\ \boldsymbol{x} \geq \boldsymbol{0},\ \xi \geq 0 \\ \min\ \boldsymbol{c}^T\boldsymbol{x} + d\xi, \end{array} & \text{(13.D)} & \begin{array}{l} A^T\boldsymbol{y} + \zeta\boldsymbol{1} \leq \boldsymbol{c} \\ \boldsymbol{1}^T\boldsymbol{y} \leq d \\ \boldsymbol{y} \geq \boldsymbol{0},\ \zeta \in \mathrm{I\!R} \\ \max\ \zeta. \end{array}
\end{array} \qquad (13)$$

The constant d was set so large that increasing d would not change the optimal basis of (13.P). Assumed each A-column $\boldsymbol{a}_j$ has an Euclidean norm of 1, the dual problem (13.D) can be interpreted as finding the smallest ball that intersects each of the half-spaces $\left\{\boldsymbol{y} \,\middle|\, \boldsymbol{a}_j^T\boldsymbol{y} \leq c_j\right\}$ $(j = 1, \dots, n)$, such that the centre of the ball falls into the simplex $\left\{\boldsymbol{y} \,\middle|\, \boldsymbol{y} \geq \boldsymbol{0},\ \boldsymbol{1}^T\boldsymbol{y} \leq d\right\}$. Though we did not scale the columns, we dub the problem (13.P) the *min-ball variant* of the relevant NETLIB problem.

The max-ball variants and the min-ball variants are special problems: each can be formulated as the minimisation of a convex objective over a simplex, as described in section 3.

We compared the Dantzig pricing rule and the cutting-plane pricing rule on the general NETLIB problems, and also on the special ball-variants.

4.3 Comparison of pricing rules on general LP problems

We solved the 59 NETLIB problems by the simplex method, comparing the Dantzig pricing rule and the cutting-plane pricing rule as described in section 2.

Our basic simplex implementation could solve $59 - 4 = 55$ problems when the Dantzig pricing was used, and $59 - 8 = 51$ problems when cutting-plane pricing was used. Out of the 59 test problems, 50 could be solved by both pricing rules. On average, solving a problem using the cutting-plane pricing took twice as long as it did when using the Dantzig pricing. The ratio between iteration counts was similar. (Our implementation is not a state-of-art solver, but the quality of the implementation is the same for the two pricing rules.)

Summing up, the cutting-plane pricing rule proved very much inferior to the Dantzig pricing rule on general problems.

4.4 Comparison of pricing rules on ball-variant problems

We solved the 118 ball-variant problems by the simplex method, comparing the Dantzig pricing rule and the cutting-plane pricing rule as described in section 3.

Our simplex implementation could solve $118 - 7 = 111$ problems when the Dantzig pricing was used, and $118 - 2 = 116$ problems when cutting-plane pricing was used. Out of the 118 test problems, 110 could be solved by both pricing rules. The total run times for these 110 problems were: 409,453 seconds for the classic pricing, and 373,209 seconds for the cutting-plane pricing. Iteration counts and run times are listed in tables 2 - 3.

We also constructed the performance profiles of our implementations. This measurement tool was introduced by Dolan and Moré (2002). As in the present paper we only have two methods to compare, we present the construction of performance profiles in a simplified form: Let $\mathcal{P}$ denote the set of problems, and let t_p and t'_p denote the times of solving problem $p \in \mathcal{P}$ with the classic simplex pricing and the cutting-plane pricing, respectively. We define performance ratios

$$r_p = \frac{t_p}{\min\{t_p, t'_p\}} \quad \text{and} \quad r'_p = \frac{t'_p}{\min\{t_p, t'_p\}} \qquad (p \in \mathcal{P}).$$

If a method failed to solve a problem, then we set the corresponding value to ∞. Let us consider r and r' as random variables with equally probable outcomes r_p and r'_p $(p \in \mathcal{P})$, respectively. The corresponding cumulative distribution functions are

$$\rho(\tau) = \frac{|\{\, p \in \mathcal{P} \mid r_p \leq \tau \,\}|}{|\mathcal{P}|} \quad \text{and} \quad \rho'(\tau) = \frac{|\{\, p \in \mathcal{P} \mid r'_p \leq \tau \,\}|}{|\mathcal{P}|} \qquad (\tau \in \mathbb{R}).$$

The performance profiles are the graphs of these cumulative distribution functions. A profile nearer the top left corner indicates a better algorithm.

The performance profiles of the classic (Dantzig) and the cutting-plane pricing rules are presented in figure 4. On the ball-variant problems, the cutting-plane pricing rule performs slightly better than the classic rule.

4.5 Comparison of the difficulty of general LP problems and their max-ball variant problems

In this test we compared the difficulty of solving a general linear programming problem, i.e., optimising a linear function over a polyhedron on the one hand; and of finding a central point in the same polyhedron on the other hand. – Namely, the problems compared are of the forms (1.D) and (8.D), but we constructed and solved the primal pairs.

Given a NETLIB problem, we constructed a problem of the form (1.P) using only the matrix A and the vectors $\boldsymbol{b}, \boldsymbol{c}$ of the NETLIB problem. We refer to the resulting problem as *canonical problem* in this comparison.

For each of our 59 NETLIB problems, we solved both the canonical problem and the max-ball variant problem using the cutting-plane pricing rule.

On average, the max-ball problems were solved in 85% less iterations than the canonical problems. (I.e., the ratio of iteration counts was about 1 : 6.) Considering running times, the max-ball problems were solved in half the time as the canonical problems. (The difference between the ratios of iteration counts and of solution times is due to longer iteration times in the ball-variant problems. This is the consequence of faster non-zero build up in our inverse representation. – We maintain a product-form inverse in our implementation. A more sophisticated inverse representation would of course mitigate nonzero build-up.)

To sum up our results, finding a central point in a polyhedron seems significantly easier than optimising a linear function over the same polyhedron. This observation is in accord with our former working experience: The simplex method is very flexible in handling feasibility issues, and this flexibility is exploited in finding a central point.

5 Conclusion and discussion

In this paper we showed that the simplex method can be interpreted as a convex cutting-plane method, assumed that a special pricing rule is used in the simplex implementation.

The cutting-plane pricing rule proved very much inferior to the classic Dantzig rule when compared on general LP problems. On the other hand, the cutting-plane rule proved slightly better than the Dantzig rule when tried on special LP problems that can be formulated as minimisation of a convex objective over a simplex.

A variety of simplex pricing rules have been developed in the past decades; see Terlaky and Zhang [32] for a survey. In the successful simplex-based solvers, several pricing rules are implemented, and the solver employs the one that best fits the situation. Such a general pricing scheme was proposed by Maros [21]. We do not expect the cutting-plane pricing rule to be competitive with a sophisticated pricing scheme. On the other hand, we think that including the cutting-plane rule may be advantageous to a pricing scheme.

5.1 On the utility of finding the largest inscribed ball

When applying the plain cutting-plane approach to a constrained convex program, the optimal solution of the current model problem is selected as the subsequent test point. E.g., in algorithm 1 (section 2), the subsequent test point is $\boldsymbol{y}^\star$.

A useful idea is finding a 'central' point in the current polyhedral model of the feasible domain, and selecting this as the subsequent test point. Different centres were proposed and tried, the most widely used being the analytic centre. The Analytic Centre Cutting-Plane Method was introduced by Sonnevend [29], and further developed by Goffin, Haurie, and Vial [10]. This method proved effective for many problems. A recent development in this direction is described in [23]. Here an interior-point constraint-generation algorithm is presented for semi-infinite linear programming, with a successful application to a healthcare problem.

Considering the ease of finding the largest inscribed ball (section 4.5), we think that the centre of this ball can also be used as the subsequent test point when solving a constrained convex problem by the cutting-plane method. Verification of this conjecture requires further experimentation.

In case of linear problems, algorithm 1 (section 2) can be easily modified according to the above observation: Let the subsequent test point be the centre of the largest ball inscribed into the current model of the feasible domain. However, we expect this method to be effective only if the original problem has an excessively large number of constraints, and especially in case of semi-infinite programming problems.

5.2 Comments on the primal-dual equivalence

The following question is raised in [11]: given the equivalence of the simplex method and the dual simplex method, how to explain their very different numerical behaviour?

The difference in basis sizes gives a partial explanation: if $m >> n$ then the dual method handles the inverses of smaller matrices. However, this observation holds true only if traditional basis inversion routines are used: Maros [22] developed a special *row basis* technique that enables simplex implementations using a working basis of size $\min(n, m)$. (The row basis technique is related to the equivalence of the simplex and the dual simplex methods.)

Using the basis correspondence described in the appendix, special inversion modules can be developed that handle even smaller working bases. We can store and keep updating the inverse of the sub-matrix S in figure 3. A product-form inverse of the variable-size S matrix is straightforward to implement. Such an inversion module was used in [8], without reference to the primal-dual equivalence. Application of this inversion routine resulted a substantial decrease in running time. (Though it should be mentioned that in the special problems handled in [8], even the sub-matrices S had a special structure that could be exploited.)

Individual upper bounds of the variables are an important factor of the different behaviour of the simplex and the dual simplex method. The primal-dual equivalence does not extend to individual upper bounds, hence a method making use of such bounds does not have an equivalent counterpart. Such variants of the dual simplex method were recently proposed by Maros [19], [20], together with computational studies that demonstrate the effectiveness of the new variants. Another advantage of the dual simplex method is observed in [22]: Devex-type pricing requires little additional cost.

References

[1] Borgwardt, K.H. (1980). *The simplex method: a probabilistic analysis.* Number 1 in Algorithms and Combinatorics. Springer-Verlag, New York.

[2] Cheney, E.W. and A.A. Goldstein (1959). Newton's method for convex programming and Tchebycheff approximation. *Numeriche Mathematik* **1**, 253-268.

[3] Dantzig, G.B. (1951). Maximization of a linear function of variables subject to linear inequalities. In: *Activity Analysis of Production and Allocation,* (T.C. Koopmans, editor). Wiley, New York, 339-347.

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

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

[6] Dolan, E.D. and J.J. Moré (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming* **91**, 201-213.

[7] Fábián, C.I., G. Mitra, and D. Roman. Processing Second-Order Stochastic Dominance models using cutting-plane representations. *Mathematical Programming, Ser. A* DOI 10.1007/s10107-009-0326-1.

[8] Fábián, C.I., A. Prékopa, and O. Ruf-Fiedler (2002). On a dual method for a specially structured linear programming problem. *Optimization Methods and Software* **17**, 445-492.

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

[10] Goffin, J.-L., A. Haurie, and J.-P. Vial (1992). Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science* **38**, 284-302.

[11] Kall, P. and J. Mayer (2005). *Stochastic Linear Programming. Models, Theory, and Computation.* International Series in Operations Research & Management Science. Springer.

[12] Kelley, J.E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics* **8**, 703-712.

[13] Klee, V. and G.J. Minty (1972). How good is the simplex algorithm? In: *Inequalities III.* (O. Shisha, editor). Academic Press, 159-175.

[14] Klein Haneveld, W.K. and M.H. van der Vlerk (2006). Integrated chance constraints: reduced forms and an algorithm. *Computational Management Science* **3**, 245-269.

[15] *Linear Inequalities and Related Systems.* (H.W. Kuhn and A.W. Tucker, editors.) Princeton University Press, 1956.

[16] Künzi-Bay, A. and J. Mayer (2006). Computational aspects of minimizing conditional value-at-risk. *Computational Management Science* **3**, 3-27.

[17] Lemke, C.E. (1954). The dual method of solving the linear programming problem. *Naval Research Logistics Quarterly* **1**, 36-47.

[18] Luedtke, J. (2008). New formulations for optimization under stochastic dominance constraints. *SIAM Journal on Optimization* **19**, 1433-1450.

[19] Maros, I. (2003). A generalized dual phase-2 simplex algorithm. *European Journal of Operational Research* **149**, 1-16.

[20] Maros, I. (2003). A piecewise linear dual phase-1 algorithm for the simplex method. *Computational Optimization and Applications* **26**, 63-81.

[21] Maros, I. (2003). A general pricing scheme for the simplex method. *Annals of Operations Research* **124**, 193-203.

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

[23] Oskoorouchi, M.R., H.R. Ghaffari, T. Terlaky, and D.M. Aleman. An interior point constraint generation algorithm for semi-infinite optimization with healthcare application. *Mathematics of Operations Research*, to appear in 2011.

[24] Padberg, M. (1995). *Linear optimization and extensions.* Springer.

[25] Prékopa, A. (1968). *Linear Programming* (in Hungarian). Bolyai János Mathematical Society, Budapest.

[26] Prékopa, A. (1992). A very short introduction to linear programming. *RUTCOR Lecture Notes* 2-92.

[27] Prékopa, A. (2011). Private communication.

[28] Rudolf, G. and A. Ruszczyński (2008). Optimization problems with second order stochastic dominance constraints: duality, compact formulations, and cut generation methods. *SIAM Journal on Optimization* **19**, 1326-1343.

[29] Sonnevend, G. (1988). New algorithms in convex programming based on a notion of 'centre' (for systems of analytic inequalities) and on rational expectations. In: *Trends in Mathematical Optimization: Proceedings of the 4th French-German Conference on Optimization* (K.H. Hoffmann, J.-B. Hirriart-Urruty, C. Lemarchal, J. Zowe, editors). Volume **84**, pp. 311-327 of the *International Series of Numerical Mathematics.* Birkhäuser, Basel, Switzerland.

[30] Spielman, D.A. and S.-H. Teng (2004). Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. *Journal of the ACM* **51**, 385-463.

[31] Terlaky, T. (2010). Private communication.

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

[33] Vanderbei, R.J. (2001, 2008). *Linear Programming. Foundations and Extensions.* International Series in Operations Research & Management Science. Springer.

Appendix: a brief proof to the primal-dual equivalence

We work on the pair of problems (2). The primal problem and a basis $\mathcal{B}$ are illustrated in figure 1, the basis matrix being denoted by B. The dual problem and a basis $\mathcal{C}$ are illustrated in figure 2, the basis matrix being denoted by C.

A one-to-one mapping between primal and dual bases

Let $\mu : \{1, \ldots, n+m\} \to \{1, \ldots, m+n\}$ be defined as

$$\mu(\ell) = \begin{cases} m+\ell & \text{if } \ell \leq n, \\ \\ \ell - n & \text{if } n < \ell. \end{cases}$$

μ maps columns of the primal problem to rows of the dual problem. Specifically, columns of matrix A are mapped to surplus rows, and slack columns are mapped to rows of matrix A.

Given a subset $\mathcal{B}$ of the columns of the primal problem, we can construct a subset $\mathcal{C}$ of the rows of the dual problem according to the rule:

$$\text{Let } \mu(\ell) \in \mathcal{C} \text{ if and only if } \ell \notin \mathcal{B}. \tag{14}$$

In words:

- a row of matrix A is selected into $\mathcal{C}$ if and only if the corresponding slack vector is not in $\mathcal{B}$,
- a surplus row is selected into $\mathcal{C}$ if and only if the corresponding column of A is not in $\mathcal{B}$.

Proposition 11 *The above rule results a one-to-one mapping between primal and dual bases.*

Proof. The rule obviously results a one-to-one mapping between the collection of subsets $\mathcal{B} \subset \{1, \ldots, n+m\}$, $|\mathcal{B}| = m$; and the collection of subsets $\mathcal{C} \subset \{1, \ldots, m+n\}$, $|\mathcal{C}| = n$.

It remains to be seen that $\mathcal{C}$ is a basis of the dual problem if and only if $\mathcal{B}$ is a basis of the primal problem. We illustrate this in figure 3 : matrix B consists of columns belonging to set $\mathcal{B}$, and matrix C consists of rows belonging to set $\mathcal{C}$. Let us consider the shaded square matrix S that is the 'intersection' of B and C. Clearly B is invertible if and only if S is invertible. Similarly, C is invertible if and only if S is invertible. □

Observation 12 *Assume that $\mathcal{B}$ and $\mathcal{C}$ are corresponding primal and dual bases (i.e., assigned to each other by the mapping of proposition 11).*

Let $(\boldsymbol{x}, \boldsymbol{u})$ denote the basic solution belonging to $\mathcal{B}$. Then $\boldsymbol{x}$ is the shadow-price vector belonging to $\mathcal{C}$, and $(\boldsymbol{u}, \boldsymbol{x})$ is the reduced-cost vector belonging to $\mathcal{C}$.

Let $(\boldsymbol{y}, \boldsymbol{v})$ denote the basic solution belonging to $\mathcal{C}$. Then $\boldsymbol{y}$ is the shadow-price vector belonging to $\mathcal{B}$, and $(\boldsymbol{v}, \boldsymbol{y})$ is the reduced-cost vector belonging to $\mathcal{B}$.

Corollary 13 *Assume that $\mathcal{B}$ and $\mathcal{C}$ are corresponding primal and dual bases.*

$\mathcal{C}$ is a dual feasible basis of the dual problem if and only if $\mathcal{B}$ is a feasible basis of the primal problem.

Extended-sense simplex and dual simplex steps

Characteristics of the simplex and the dual simplex steps were summed up in section 1.2. In this section we relax the sign restrictions on the pivot elements. Moreover, we show that in the absence of degeneracy, these restrictions are redundant.

Extended-sense simplex steps for the primal problem. Let $\mathcal{B}$ and $\mathcal{B}'$ be feasible bases of the primal problem. Suppose $\mathcal{B}$ and $\mathcal{B}'$ are neighbouring bases, i.e., considering them as sets, their set-difference $\mathcal{B}' \setminus \mathcal{B}$ has a single element that we denote by ℓ. (In words, ℓ is the index that enters basis in course of step $\mathcal{B} \to \mathcal{B}'$.) Then $\mathcal{B} \to \mathcal{B}'$ is an extended-sense simplex step if and only if there is a negative component at position ℓ of the reduced-cost vector belonging to $\mathcal{B}$. The difference between a regular simplex step and an extended-sense simplex step is that negative pivot elements are not excluded in the latter.

We say that $\mathcal{B} \to \mathcal{B}'$ is a degenerate step if the same basic solution belongs to $\mathcal{B}$ and $\mathcal{B}'$. If an extended-sense simplex step $\mathcal{B} \to \mathcal{B}'$ is not degenerate, then it is a regular simplex step, because pivoting on a negative element would obviously destroy feasibility.

Extended-sense dual simplex steps for the dual problem. Let $\mathcal{C}$ and $\mathcal{C}'$ be dual feasible bases of the dual problem. Suppose $\mathcal{C}$ and $\mathcal{C}'$ are neighbouring bases, i.e., their set-difference $\mathcal{C} \setminus \mathcal{C}'$ has a single element that we denote by k. (In words, k is the index that leaves basis in course of step $\mathcal{C} \to \mathcal{C}'$.) Then $\mathcal{C} \to \mathcal{C}'$ is an extended-sense dual simplex step if and only if there is a negative component at position k of the basic solution belonging to $\mathcal{C}$. The difference between a regular dual simplex step and an extended-sense dual simplex step is that positive pivot elements are not excluded in the latter.

We say that $\mathcal{C} \to \mathcal{C}'$ is a dual degenerate step if the same reduced-cost vector belongs to $\mathcal{C}$ and $\mathcal{C}'$. If an extended-sense dual simplex step $\mathcal{C} \to \mathcal{C}'$ is not dual degenerate, then it is a regular dual simplex step, because pivoting on a positive element would obviously destroy dual feasibility.

Equivalence of extended-sense simplex and extended-sense dual simplex steps. Let $(\mathcal{B}, \mathcal{C})$ and $(\mathcal{B}', \mathcal{C}')$ be pairs of corresponding primal and dual bases. The mapping of proposition 11 obviously keeps neighbours, hence the primal bases $\mathcal{B}$ and $\mathcal{B}'$ are neighbouring if and only if the dual bases $\mathcal{C}$ and $\mathcal{C}'$ are neighbouring. In the following discussion we assume that they are neighbouring.

Corollary 13 takes care of feasibility issues: $\mathcal{C}$ is a dual feasible basis if and only if $\mathcal{B}$ is a feasible basis, and $\mathcal{C}'$ is a dual feasible basis if and only if $\mathcal{B}'$ is a feasible basis. In what follows we assume that all these feasibility requirements are satisfied.

Proposition 14 *$\mathcal{C} \to \mathcal{C}'$ is an extended-sense dual simplex step if and only if $\mathcal{B} \to \mathcal{B}'$ is an extended-sense simplex step.*

Proof. Let ℓ denote the index that enters basis in course of step $\mathcal{B} \to \mathcal{B}'$, and let k denote the index that leaves basis in course of step $\mathcal{C} \to \mathcal{C}'$. According to rule (14), we have $k = \mu(\ell)$.

Let $(\boldsymbol{y}, \boldsymbol{v}) \in \mathbb{R}^{m+n}$ denote the basic solution belonging to $\mathcal{C}$. According to the second part of observation 12, the reduced-cost vector belonging to $\mathcal{B}$ is $(\boldsymbol{v}, \boldsymbol{y})$. From the properties of mapping μ, we obviously have $(\boldsymbol{y}, \boldsymbol{v})_{\mu(\ell)} = (\boldsymbol{v}, \boldsymbol{y})_\ell$.

From the above definitions, $\mathcal{B} \to \mathcal{B}'$ is an extended-sense simplex step if and only if $(\boldsymbol{v}, \boldsymbol{y})_\ell < 0$, and $\mathcal{C} \to \mathcal{C}'$ is an extended-sense dual simplex step if and only if $(\boldsymbol{y}, \boldsymbol{v})_k < 0$. □

Equivalence of the simplex and the dual simplex method

We are going to show that the simplex method applied to the primal problem on the one hand, and the dual simplex method applied to the dual problem on the other hand, are equivalent methods. If no degeneracy is present, then extended-sense simplex steps are regular simplex steps, and extended-sense dual simplex steps are regular dual simplex steps; hence the statement is a straight consequence of proposition 14.

Let $\mathcal{B} \to \mathcal{B}'$ be an extended-sense simplex step, and let $\mathcal{C} \to \mathcal{C}'$ be the corresponding extended-sense dual simplex step. By the first part of observation 12, $\mathcal{C} \to \mathcal{C}'$ is dual degenerate if and only if $\mathcal{B} \to \mathcal{B}'$ is degenerate.

Assume now that $\mathcal{B} \to \mathcal{B}'$ is a regular simplex step. Obviously there exists a perturbation of the right-hand-side vector that makes $\mathcal{B} \to \mathcal{B}'$ non-degenerate, while retaining it a simplex step. This shows that $\mathcal{C} \to \mathcal{C}'$ is a regular dual simplex step. (If $\mathcal{C} \to \mathcal{C}'$ is known to be a regular dual simplex step, then a similar procedure can be applied to show that $\mathcal{B} \to \mathcal{B}'$ is a regular simplex step.)

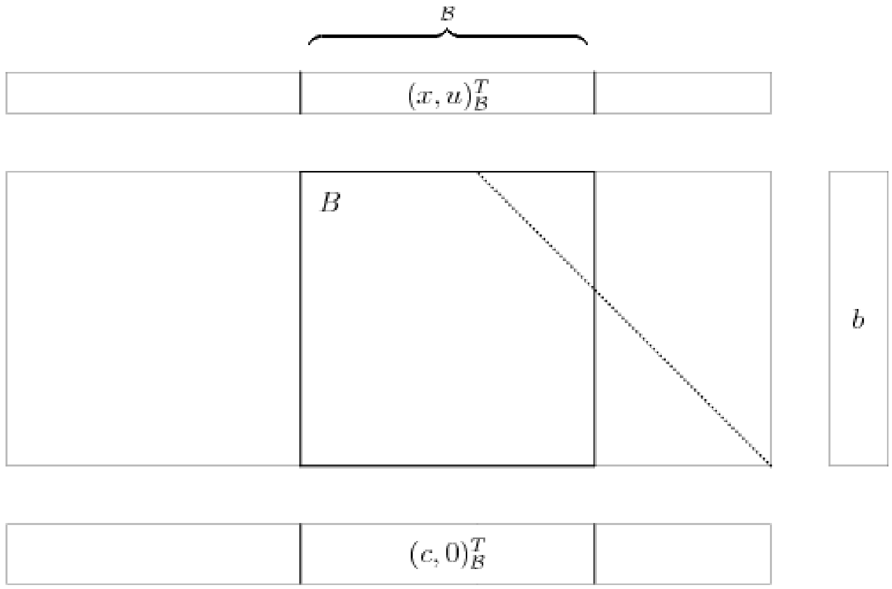

Figure 1: Basis $\mathcal{B}$ in primal problem (2.P)

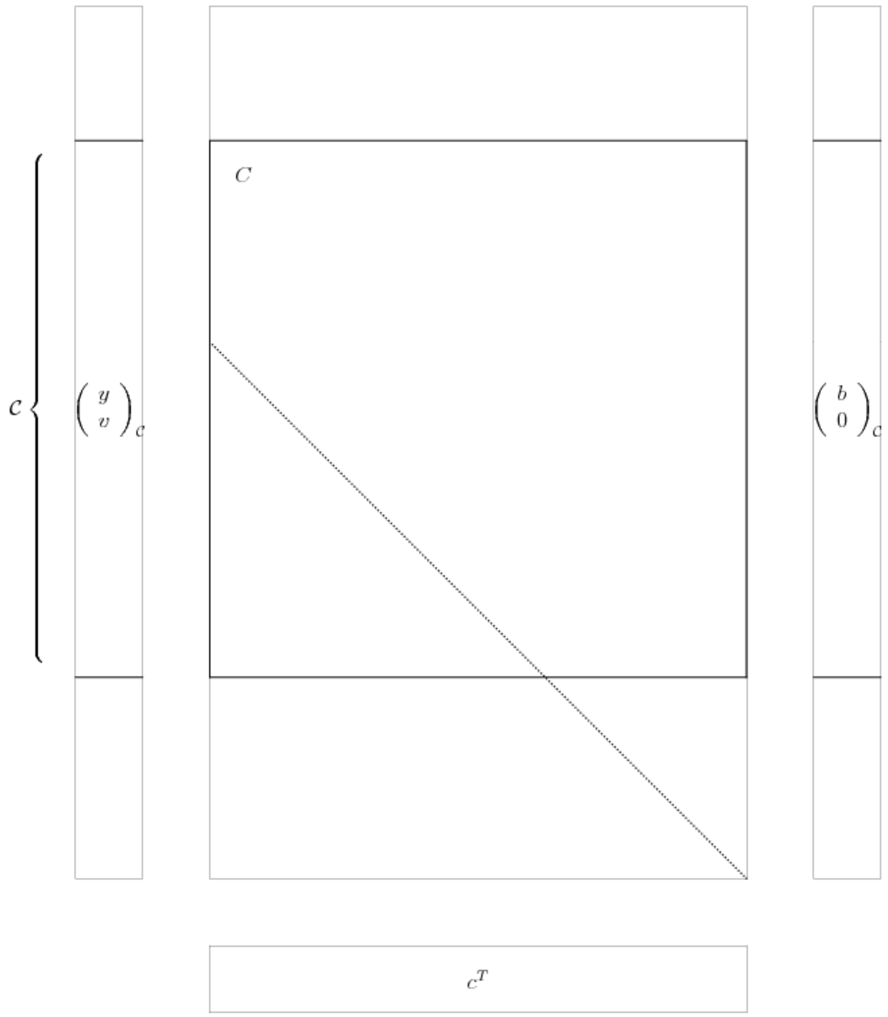
$\mathcal{C}$
$\begin{pmatrix} y \\ v \end{pmatrix}_{\mathcal{C}}$
C
$\begin{pmatrix} b \\ 0 \end{pmatrix}_{\mathcal{C}}$
c^T

Figure 2: Basis $\mathcal{C}$ in dual problem (2.D)

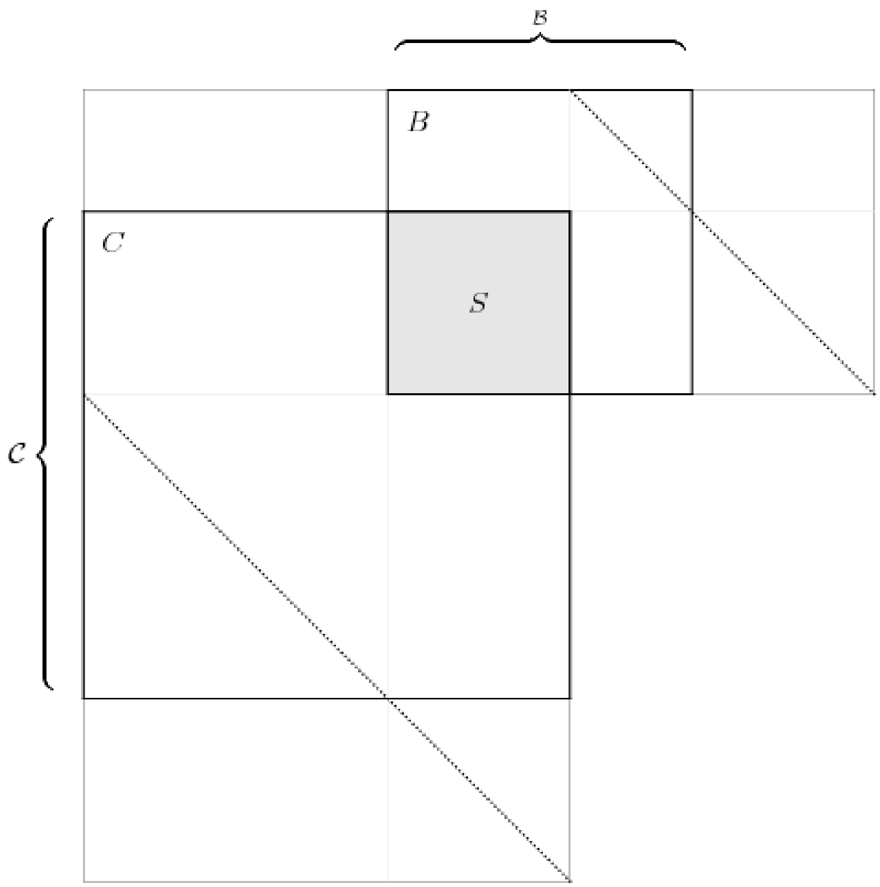

Figure 3: Corresponding bases $\mathcal{B}$ and $\mathcal{C}$

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: NETLIB test problems

problem name	classic pricing rule		cutting-plane pricing rule	
	iterations	run time (millisec)	iterations	run time (millisec)
ADLITTLE	24	16	41	15
AFIRO	22	15	23	15
AGG	213	3625	219	2078
AGG2	-	-	717	20843
AGG3	-	-	1030	34766
BANDM	426	7797	-	-
BEACONFD	128	140	13	15
BLEND	73	47	102	78
BNL1	36	406	30	359
BOEING1	67	329	70	344
BOEING2	20	15	22	31
BORE3D	2	1	5	15
BRANDY	1	1	1	1
CAPRI	36	109	10	16
DEGEN2	725	33828	1634	34953
E226	202	875	467	2516
ETAMACRO	4	15	5	31
FFFFF800	167	4672	55	578
FINNIS	2	16	3	31
FIT1D	68	62	29	32
FIT2D	54	375	34	234
FORPLAN	88	141	64	93
GFRD-PNC	18	219	12	156
GROW15	11	47	2	16
GROW22	13	110	1	15
GROW7	8	15	2	1
ISRAEL	-	-	949	3829
KB2	6	15	9	1
LOTFI	41	63	36	31
PEROLD	-	-	87	2297
PILOT4	11	62	10	47
RECIPELP	8377	13172	57	31
SC105	143	187	227	281
SC205	312	2438	387	2922
SC50A	71	46	62	31
SC50B	64	31	64	46
SCAGR25	23	156	23	141
SCAGR7	3	1	6	1
SCFXM1	5	15	2	16
SCFXM2	7	93	7	93
SCORPION	13	62	1	1
SCRS8	48	297	46	281
SCTAP1	196	1453	150	1172
SCSD1	88	125	59	78
SCSD6	9	31	21	63
SEBA	3	31	3	31
SHARE1B	262	375	235	234
SHARE2B	109	110	92	78
SHELL	594	38281	572	36844
SHIP04L	58	578	28	250
SHIP04S	32	218	65	578
STAIR	126	1485	93	453
STANDATA	59	438	8	31
STANDGUB	24	188	8	31
STANDMPS	211	5766	8	47
STOCFOR1	74	93	59	62
TUFF	136	1344	120	985
VTP-BASE	-	-	2	1
WOOD1P	187	1782	1079	13406

Table 2: Max-ball variants test results

problem name	classic pricing rule		cutting-plane pricing rule	
	iterations	run time (millisec)	iterations	run time (millisec)
ADLITTLE	6	1	4	16
AFIRO	8	1	8	1
AGG	9	31	12	47
AGG2	9	47	13	63
AGG3	9	47	13	63
BANDM	19	62	14	47
BEACONFD	1	15	1	1
BLEND	21	15	20	16
BNL1	9	94	15	203
BOEING1	41	172	22	47
BOEING2	43	47	36	31
BORE3D	316	859	113	250
BRANDY	60	140	144	422
CAPRI	21	31	79	172
DEGEN2	10	31	5	16
E226	4	15	4	16
ETAMACRO	28	141	23	125
FFFFF800	948	87093	1	16
FINNIS	16	63	19	79
FIT1D	3	16	3	15
FIT2D	3	32	5	46
FORPLAN	64	109	57	94
GFRD-PNC	5	47	5	78
GROW15	809	16234	817	14734
GROW22	2346	85344	2274	75593
GROW7	244	562	1179	6829
ISRAEL	-	-	3	1
KB2	81	46	104	47
LOTFI	9	16	1	1
PEROLD	177	7188	427	22953
PILOT4	-	-	-	-
RECIPELP	4	1	27	15
SC105	24	16	20	16
SC205	14	16	17	31
SC50A	19	1	20	16
SC50B	9	15	18	1
SCAGR25	2099	53344	2178	64641
SCAGR7	88	47	312	422
SCFXM1	3	1	3	16
SCFXM2	6	63	6	78
SCORPION	3	15	7	31
SCRS8	3	31	2	31
SCTAP1	4	15	7	16
SCSD1	23	31	41	47
SCSD6	9	31	31	94
SEBA	516	18234	513	19265
SHARE1B	24	31	24	15
SHARE2B	63	62	60	47
SHELL	546	21125	725	57375
SHIP04L	47	500	50	516
SHIP04S	50	437	53	422
STAIR	47	204	26	125
STANDATA	2	16	93	672
STANDGUB	29	219	8	47
STANDMPS	2	31	93	1484
STOCFOR1	52	31	9	1
TUFF	2	16	121	1109
VTP-BASE	43	78	62	125
WOOD1P	241	2328	243	2344

Table 3: Min-ball variants test results

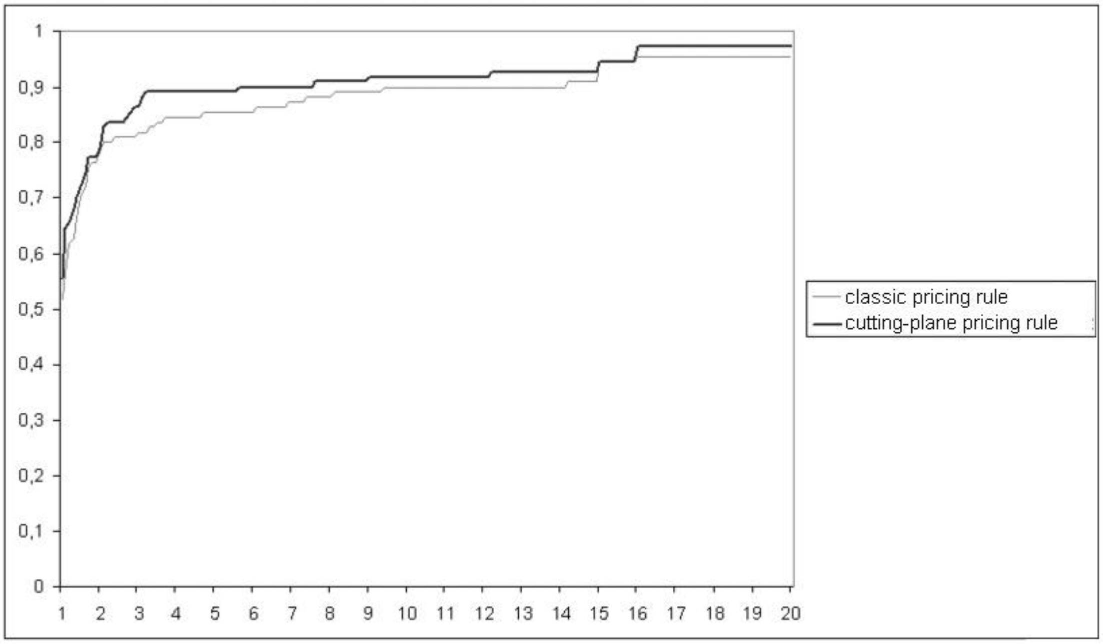

Figure 4: Performance profiles of the classic and the cutting-plane pricing rules on ball-variant problems