A FAMILY OF SIMPLEX VARIANTS SOLVING AN $m \times d$ LINEAR PROGRAM IN EXPECTED NUMBER OF PIVOT STEPS DEPENDING ON d ONLY

Ilan Adler[+]

Richard Karp[*]

Ron Shamir[+]

University of California

[*]Computer Science Division

[+]Department of Industrial Engineering and Operations Research

Berkeley, CA 94720

Table of Contents

A FAMILY OF SIMPLEX VARIANTS SOLVING AN $m \times d$ LINEAR PROGRAM IN EXPECTED NUMBER OF PIVOT STEPS DEPENDING ON d ONLY

Ilan Adler+
*Richard Karp**
Ron Shamir+

University of California
*Computer Science Division
+Department of Industrial Engineering and Operations Research
Berkeley, CA 94720

ABSTRACT

We present a family of variants of the Simplex method, which are based on a Constraint-By-Constraint procedure: the solution to a linear program is obtained by solving a sequence of subproblems with an increasing number of constraints. We discuss several probabilistic models for generating linear programs. In all of them the underlying distribution is assumed to be invariant under changing the signs of rows or columns in the problem data. A weak regularity condition is also assumed. Under these models, for linear programs with d variables and $m + d$ inequality constraints, the expected number of pivots required by these algorithms is bounded by a function of $\min(m, d)$ only. In particular this means that, for a fixed number of variables, the expected number of pivots is bounded by a *constant* when the number of constraints tends to infinity. Since Smale's original model [S1] satisfies our probabilistic assumptions, the same results apply to his model. We also present some results for models generating only feasible linear programs, and for Bland's pivoting rule. We conclude with a discussion of our probabilistic models, and show why they are inadequate for obtaining meaningful results unless d and m are of the same order of magnitude.

1. Introduction

The Simplex Method for Linear Programming, originated by Dantzig in 1947, is one of the most frequently used algorithms in industry and government. The ordinary measure of complexity of this method is the number of pivot steps it requires to solve a linear program, expressed as a function of the dimensions of the problem. Vast practical experience indicates that this function is linear, or at most polynomial [D], [KQ]. However, examples have been constructed for several variants of the Simplex method, showing that in the worst case the number of pivots may grow exponentially with the dimensions [KM], [J], [GaS], [Z], [Mu]. The Ellipsoid Algorithm [Kh] was demonstrated to solve linear programs in time which is polynomial in the length of the problem data in the worst case, but appears to be much slower than the Simplex method in practice.

Recently, several works have tried to explain the efficiency of the Simplex method by approaching the complexity issue probabilistically: Assuming some distribution of the problem data, this approach tries to show that the *average* number of pivots grows slowly with the problem's dimensions. To quote these results denote the number of variables in the problem by d and the number of inequalities by n, and assume $d \leq n$. We use c to denote a constant and $c(d)$ to denote a function of d only. Borgwardt [Bo1], [Bo2] showed that a parametric simplex variant requires an average of at most $c \cdot n \cdot d^2 \cdot (d+1)^2$ pivots for a probabilistic model which generates only feasible linear programs. Smale [S1], [S2] showed that the parametric Self Dual Simplex requires an average of at most $c(d)\,(\log\,(n-d))^{d(d+1)}$ pivots when the problem data is drawn from a spherically symmetric distribution. Adler [A] and Haimovich [H] demonstrated that some parametric Simplex variants require an average of at most d steps once a vertex of the feasible region is given, but their results do not have immediate consequences for the full (Phase I - II) Simplex method.

In this paper we define a family of Simplex variants which are based on a Constraint-By-Constraint (CBC) procedure: They obtain a solution to a linear program by solving a sequence of subproblems with an increasing number of constraints. We present three consecutively more general algorithms satisfying this property. We show that under probabilistic assumptions which are weaker than Smale's [S1], these algorithms require an average of no more than $c(d)$ pivots where $c(d)$ is between $d \cdot 1.5^d$ and 2^{5d}, depending on the algorithm and the probabilistic model. In particular, this implies that when d is fixed and m tends to infinity, the expected number of pivots required to solve the problem is bounded by a *constant*. All our probabilistic models require that the problem data satisfy a weak regularity condition with probability one. The strongest model requires that the problem data be generated by a distribution which is invariant under changing the sense of any subset of the inequalities defining the problem. Weaker models, which do not require invariance with respect to changing the signs of the non-negativity constraints (if such are included) are also investigated. Since Smale's original model [S1] satisfies these assumptions, this implies that these algorithms require an average of at most a constant number of pivots for Smale's model when one dimension of the problem is kept fixed and the other tends to infinity. We also show that Bland's pivoting rule, when combined with the 'Big M' method, is a special case of a Constraint-By-Constraint algorithm.

Finally we discuss the consequences of these results. We observe that, in all these models, there is a very high probability, when $m \gg d$, that a random problem will be infeasible. The Constraint-By-Constraint algorithms exploit this property by detecting infeasibility at an early iteration with high probability. Therefore the good behavior of these algorithms when $m \gg d$ results primarily from the probabilistic models and not from the nature of the Simplex method. Hence these models are inadequate for obtaining meaningful results unless d and m are of the same order of magnitude.

2. Preliminaries

For a matrix $A \varepsilon R^{m \times d}$, we denote by A_i or $A_{i\cdot}$ the i-th row of A, and by $A_{\cdot i}$ the i-th column of A. If S is a sequence of indices of rows (columns), we denote by A_S ($A_{\cdot S}$) the submatrix obtained by taking only the rows (columns) in S.

We shall deal with the *Linear Programming Problem* (LPP) in the form

$$
(P) \quad \begin{array}{ll} \min c^T x & \\ \text{s.t. } a_i^T x \geq b_i & i = 1,\dots,m. \\ x \geq 0 & \end{array}
$$

where $c, x, a_i \varepsilon R^d$, $b_i \varepsilon R$.

The constraints of the form $a_i^T x \geq b_i$ are called ***matrix constraints*** to be distinguished from the $x_i \geq 0$ *sign* (or nonnegativity) *constraints*. Define also:

$$A := \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}, \quad M := \begin{bmatrix} I \\ A \end{bmatrix}, \quad b := \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}, \quad v := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ b \end{pmatrix} \varepsilon R^n \quad \begin{array}{l} n := m + d \\ \Delta := \min(m, d) \end{array}$$

So equivalent presentations of (P) are

$$\begin{array}{c} \min c^T x \\ Ax \geq b \\ x \geq 0 \end{array} \qquad \text{and} \qquad \begin{array}{c} \min c^T x \\ Mx \geq v \end{array}$$

Occasionally we shall deal with LPP in the form

$$(\overline{P}) \qquad \begin{array}{c} \min c^T x \\ Ax \geq b \end{array}$$

where dimensions are as in (P), but we do not necessarily have nonnegativity constraints. Here we can identify $n = m$, $M = A$, $v = b$ and refer to both forms together as $\min c^T x$, $Mx \geq v$.

The *Parametric Objective Function LPP* is the problem

$$\begin{array}{c} \min (c + \lambda \overline{c})^T x \\ Mx \geq v \end{array} \qquad c, \overline{c} \, \varepsilon R^d, \ \lambda \varepsilon R$$

where the optimal solution for $\lambda = 0$ is given, and we wish to find the optimal solution for all values of the parameter λ. Here c is called the *objective* function and $\overline{c}$ the *co-objective*.

The Parametric problem can be solved by a well-known Phase II Simplex variant called the Parametric Objective Algorithm [GaS]. Under some nondegeneracy assumptions that will be described later, this algorithm has the following properties:

(1) It starts at a given vertex of the feasible set $F := \{x \mid Mx \geq v\}$ which is optimal with respect to $c^T x$ in F.

(2) When λ is increased, the optimal solution may change, generating a connected one-dimensional path, following vertices and edges of F. This path is called the *efficient path* generated by the algorithm.

(3) The path may terminate in a vertex of F, in which case that vertex is optimal for all λ greater than some $\bar{\lambda}$. It may also terminate in an unbounded ray of F, in which case the solution is unbounded (i.e. the objective function is unbounded from below over the feasible set) for λ greater than some $\hat{\lambda}$.

(4) The same phenomena happen when λ is decreased from zero. The connected union of the paths for $\lambda \geq 0$ and $\lambda \leq 0$ is called the *co-optimal path*.

Every inequality of the form $M_i x \geq v_i$ can be considered as a halfspace in R^d determined by the hyperplane $M_i x = v_i$ and a sign (or orientation) choice with respect to that hyperplane. The opposite sign choice would yield the inequality $M_i x \leq v_i$. Given k hyperplanes in R^d, $k \geq d$, every one of the 2^k sign choices determines a constraint set or an *instance*. A non-empty instance is called a *cell*. Under a nondegeneracy assumption (to be described later) every cell is d-dimensional. In that case we say that the hyperplanes form a *d-arrangement*.

When the parametric algorithm is used on each of the cells of a d-arrangement with the same objective and co-objective, a co-optimal path is generated in each cell. Assuming non-degeneracy, these paths have the following properties [A], [H]:

(1) Each vertex is optimal with respect to $c^T x$ in exactly one cell.

(2) Each vertex is on exactly $(d + 1)$ co-optimal paths in cells incident on it.

We shall denote by $\rho_\Gamma(A,b,c)$ the number of pivot steps required to solve LPP with that data by algorithm Γ. Assuming a specific probabilistic model over the data, we shall denote

$$\rho_\Gamma(m,d) := E[\rho_\Gamma(A,b,c)]$$

where the averaging is done over all $A \varepsilon R^{m \times d}$, $b \varepsilon R^m$, $c \varepsilon R^d$ according to that specific probability model.

3. The Parametric CBC Algorithm

We now describe an algorithm for solving linear programs. The algorithm solves a sequence of subproblems, each one containing one more constraint than its immediate predecessor. (We call algorithms satisfying this property *Constraint-By-Constraint* (CBC) algorithms). In each subproblem we start at a vertex supplied by its predecessor, and follow an efficient path until either feasibility with respect to the new constraint is obtained, or infeasibility is demonstrated. If the problem is feasible, the last efficient path provides the required solution. Since the Parametric Objective Algorithm is used in every subproblem we call this algorithm the *Parametric-CBC* (PCBC) algorithm.

In order to state the algorithm formally, define

$$X^{(k)} := \{x \in R^d \mid x \geq 0 \text{ and } a_i^T x \geq b_i \text{ for } i = 1,\ldots k\} \quad k = 0, 1,\ldots,m$$

$$X := X^{(m)}$$

$$e^T := (1,1,\ldots,1) \in R^d$$

Statement of the PCBC Algorithm:

Stage 0: (*For* $X^{(0)} = R_+^d$, 0 minimizes $e^T x$)

$\bar{x} \leftarrow 0$; Go to stage 1.

Stage k ($1 \leq k \leq m$): (Starts with $\bar{x}$ minimizing $e^T x$ in $X^{(k-1)}$)

If $a_k^T \bar{x} \geq b_k$ go to stage $k+1$ ($\bar{x}$ is also optimal in $X^{(k)}$).

Else use the Parametric Objective Algorithm to solve the parametric linear program

$$\min e^T x - \vartheta a_k^T x$$
$$x \in X^{(k-1)}$$

The algorithm starts at $\bar{x}$ and generates an efficient path of edges and vertices in $X^{(k-1)}$.

Case 1: Let $\hat{x}$ be the first point along the path which satisfies $a_k^T \hat{x} \geq b_k$. ($\hat{x}$ is a feasible vertex of $X^{(k)}$ and minimizes $e^T x$ in $X^{(k)}$). Go to stage $k + 1$ with $\bar{x} \leftarrow \hat{x}$.

Case 2: If the path terminates without reaching such $\hat{x}$–Stop, the problem is infeasible.

Stage $m + 1$: (Starts with $\bar{x}$ minimizing $e^T x$ in X).

Use the parametric objective algorithm to solve

$$\min e^T x + \vartheta c^T x$$

$$x \in X$$

The algorithm starts at $\bar{x}$ and the end vertex (or ray) of the efficient path provides the required solution.

4. Proof of Validity

In order to prove that the algorithm is valid we have to verify it recognizes infeasibility, unboundedness and optimality correctly. This will be established if we justify the claims in cases (1) and (2) in the description of the algorithm.

We only need to consider the case where the starting point $\bar{x}$ of stage l satisfies $a_l^T \bar{x} < b_l$. In that case the algorithm produces in step l a connected path following edges and vertices of $X^{(l-1)}$, stopping at the *first* point $\hat{x}$ on that efficient path satisfying $a_l^T x \geq b_l$. By the continuity of $a_l^T x$ along the path, at that point $\hat{x}$, $a_l^T \hat{x} = b_l$.

Denote the corresponding value of the parameter by $\hat{\vartheta}$. By efficiency of $\hat{x}$ with respect to $e - \hat{\vartheta} a_l$ we know that

$$(e - \hat{\vartheta} a_l)^T \hat{x} \leq (e - \hat{\vartheta} a_l)^T x \qquad \text{for all } x \in X^{(l-1)}$$

hence

$$e^T\hat{x} \le e^T x + \hat{\vartheta}(a_l^T\hat{x} - a_l^T x) \qquad \text{for all } x \,\varepsilon\, X^{(l-1)}$$

Every point $x \,\varepsilon\, X^{(l)}$ satisfies also $a_l^T x \ge b_l = a_l^T\hat{x}$. Hence

$$e^T\hat{x} \le e^T x \qquad \text{for all } x \,\varepsilon\, X^{(l)}$$

this justifies the statement in case (1).

To justify (2), note first that we cannot terminate in a *ray* in stage l without obtaining feasibility with respect to the $l-th$ constraint. This is true since if for $\vartheta \to \infty$ also $e^T x - \vartheta a_l^T x \to -\infty$ on some ray in $X^{(l-1)}$, the fact that $e^T x \ge 0$ on $X^{(l-1)}$ implies that $a_l^T x \to \infty$ on that ray in $X^{(l-1)}$, hence the $l-th$ constraint $a_l^T x \ge b_l$ must be satisfied at that point on that ray, so $X^{(l)} \ne \Phi$.

So we know that in case (2) we must terminate in an *optimal vertex* $\bar{x}$, satisfying

$$e^T\bar{x} - \vartheta a_l^T\bar{x} \le e^T x - \vartheta a_l^T x \qquad \forall x \,\varepsilon\, X^{(l-1)}, \forall \vartheta \ge \bar{\vartheta}.$$

and $a_l^T\bar{x} < b_l$.

Assume $X^{(l)} \ne \Phi$. Then there exists $\hat{x} \,\varepsilon\, X^{(l)}$, satisfying $a_l^T\hat{x} \ge b_l > a_l^T\bar{x}$. Hence for sufficiently large ϑ $e^T\hat{x} - \vartheta a_l^T\hat{x} < e^T\bar{x} - \vartheta a_l^T\bar{x}$ and $\hat{x} \,\varepsilon\, X^{(l-1)}$, a contradiction.

∎

Several comments can be made on the algorithm:

(1) The choice of $e \,\varepsilon\, R^d$ as the starting objective is quite arbitrary. In fact, any vector $u \,\varepsilon\, R^d$ for which $\min\{u^T x \mid x \,\varepsilon\, X^{(0)}\}$ is finite will do. So we can replace e by any non-negative vector.

(2) The algorithm is valid for every LPP, even if the data is degenerate. In that case we need only to introduce some anti-cycling device into the parametric algorithm we use in every stage (e.g. [D], [Bln]).

(3) The algorithm solves every LPP in any form, since by a proper transformation every LPP can be presented in an equivalent form (P).

5. The Probabilistic Models

Let us now define the basic ingredients which we use in our probabilistic models.

Let a linear programming problem in form (P) be given by the data (A,b,c). We call the data *Vertex-Distinct* (VD) if all bases generated by the $d+m$ hyperplanes correspond to distinct vertices. This condition is satisfied if for every $d \times d$ submatrix M_S of $M = \begin{bmatrix} A \\ I \end{bmatrix}$ with rank d, rank $\begin{bmatrix} M_S & v_S \\ M_j & v_j \end{bmatrix} = d+1$ for all $j \notin S$. Note that this condition depends only on A and b.

For the same data with an additional co-objective $\bar{c} \in R^d$, call the data *Path-Unique* (PU) if the co-optimal paths generated by the parametric algorithm in each cell are uniquely defined. This condition is satisfied if for every $d \times d$ non-singular submatrix M_S, $(c+\lambda\bar{c})^T M_S^{-1}$ has at most one zero coordinate for any real λ. Usually we shall use $e \in R^d$ as the second objective, and we mention $\bar{c}$ only when its identity is not obvious. Note that this condition depends on $A, c, \bar{c}$ but not on b.

If the data is both VD and PU we call it *Weakly Regular* (WR). We also say that $(A,b,c,\bar{c})$ are in *weakly regular position*. If both $(A,b,c,\bar{c})$ and $(A^T,c,b,\bar{b})$ are WR [VD] we say that the data is *Twice-WR* (TWR) [twice-VD (TVD)].

A probabilistic model for the generation of the data which produces weakly regular (or VD, or PU) instances with probability one is called a *Weakly Regular* (or VD, or PU) *Model*.

A distribution of the data (A,b,c) will be called *Column Sign Invariant* (CSI) if it is invariant under changing the signs of every subset of the columns of A. If it is invariant under sign changes of columns of $\begin{bmatrix} c^T \\ A \end{bmatrix}$ we call it *Extended-CSI* (ECSI). If it is invariant under changing the signs of every subset of the rows of $[A, b]$ we call it *Row Sign Invariant* (RSI).

A distribution which is both RSI and CSI [ECSI] will be called *Sign Invariant* (SI) [Extended-SI (ESI)]. Note that Sign Invariance does not imply any condition on the objective c, but ESI does. Note also that the primal data is RSI if and only if the dual data is ECSI.

The advantage of the various sign invariant models to our work is that their probabilistic analysis can be done in essentially combinatorial techniques. Similar models were used by May and Smith [MS] for investigating random polytopes, and by Adler and Berenguer [AB1], [AB2] for investigating several issues in random linear programs.

A measure over sets of rays in R^n is called a *Spherically Symmetric Measure (SSM)* if for each set S its measure is $\mu(S \cap B^{n-1})$ where B^{n-1} is the unit sphere in R^n and μ is the normalized uniform measure on B^{n-1}. A measure over sets of vectors in R^n is called SSM if by identifying every vector with its corresponding ray (i.e. by ignoring the radial parts of the vectors) the resulting measure over rays in R^n is SSM.

A distribution of the data (A, b, c) is called *Spherically Symmetric* if A assumes a SSM in $R^{m \times d}$ and independently (b^T, c^T) assumes SSM in R^{m+d}.

The models used by Adler [A] and Haimovich [H] are Twice Weakly Regular and Extended Sign Invariant. (In fact, they require CSI in the matrix $\begin{bmatrix} \bar{c} & 0 \\ c & 0 \\ A & b \end{bmatrix}$ and assume that c and $\bar{c}$ are also randomized).

The model used by Smale [S1] is Twice Weakly Regular and Spherically Symmetric. Note that Spherical Symmetry implies Extended Sign Invariance, since a spherically symmetric measure is invariant under reflections of coordinates. So Smale's model is also ESI.

The model used by Borgwardt [Bo1],[Bo2] is Weakly Regular and requires that the rows of $\begin{bmatrix} c^T \\ \bar{c}^T \\ A \end{bmatrix}$ are i.i.d. and assume spherically symmetric measure in $R^d - \{0\}$. Hence this model is Extended-CSI but not RSI.

6. Analysis of the WR-RSI Model

Consider a (fixed) data A,b,c satisfying the Weak Regularity conditions. The 2^m LP instances obtained from that data by flipping signs of the matrix inequalities are equiprobable under the Row-Sign Invariance model. The same is true for the 2^k sub-instances obtained by using only the first k matrix inequalities together with the sign inequalities as we do in the PCBC algorithm. Note that in all sub-instances d more sign constraints are present, but their signs are kept fixed.

In stage $k + 1$ of the algorithm k matrix-constraints and d sign constraints are present, generating at most $\binom{k+d}{d}$ vertices. The algorithm follows efficient paths in all feasible instances (cells) generated. By Weak Regularity each vertex is on the co-optimal paths in exactly $d + 1$ cells [A], [H]. So an upper bound on the number of pivots performed in stage $k + 1$ is $\binom{k+d}{d}$ $(d + 1)$.

Every feasible subinstance in stage $k + 1$ may be completed in 2^{m-k} different ways to form an instance of the original problem, all of which are equi-probable. So the total number of pivots contributed in stage $k + 1$ to solving full instances is at most $\binom{k+d}{d}$ $(d + 1)\, 2^{m-k}$.

Summing over all stages we get that the total number of pivots performed in all the instances is bounded by

$$\sum_{k=1}^{m} \binom{k+d}{d} (d + 1)\, 2^{m-k}$$

Hence the average number of pivots per instance is bounded by

$$2^{-m} \sum_{k=1}^{m} \binom{k+d}{d} (d + 1)\, 2^{m-k} = (d + 1)\, 2^{d} \sum_{k=1}^{m} \binom{k+d}{d} 2^{-(k+d)} \leq (d + 1)\, 2^{d+1}$$

Where the last inequality follows from Lemma A in the Appendix. Since this result is independent of the data and requires only that it satisfies WR we can conclude:

Theorem 1: For a model satisfying the Row Sign Invariance and Weak Regularity with probability one, the PCBC algorithm requires an average of at most $(d + 1)\, 2^{d+1}$ pivots, independent of m.

Corollary 1: Under the above model when d is fixed and $m \to \infty$ the average number of pivots is bounded by a constant.

Corollary 2: For a TWR, ESI model

$$\rho_{PCBC}(m,d) \le (\Delta + 1)\, 2^{\Delta+1} \qquad \text{where } \Delta := \min(m,d)$$

Proof: Apply the Parametric CBC Algorithm either to the primal or to the dual, depending on whether $m \ge d$ or $d > m$. Since both problems satisfy the conditions of Theorem 1, the corollary follows.

∎

We shall improve this last result in the next section.

7. Analysis of the WR-SI model

In the previous section we did not use the fact that only vertices in the positive orthant may contribute pivot steps at any stage. We now use that fact in a model which is Column Sign Invariant.

Lemma 1: Let (A,b,c) be VD and assume that it is drawn from a Column-Sign-Invariant distribution. Then the probability that a vertex generated by (A,b) is in the positive orthant is 2^{-l}, where l is the number of tight matrix constraints at that vertex.

Proof: Let $\bar{x}$ be a vertex determined by the d equations

$$A_{\Delta_1} x = b_{\Delta_1}$$

$$x_{\Delta_2} = 0 \quad \text{with } |\Delta_1| + |\Delta_2| = d,\ |\Delta_1| = l,\ \bar{\Delta}_2 := \{1,\dots,d\} \sim \Delta_2$$

Denote $B := A_{\Delta_1}^{\bar{\Delta}_2}$, the $l \times l$ submatrix of A determined by the rows in Δ_1 and the columns not in Δ_2. $v := b_{\Delta_1}$, $z := x_{\bar{\Delta}_2}$. Then the above system is $Bz = v$ and we ask what is $Pr[z \geq 0]$ assuming B is taken from Column-Sign-Invariant distribution.

Changing signs of columns in B may be presented by multiplying it by an $l \times l$ *sign matrix* J satisfying

$$J_{ij} = \begin{cases} +1 \text{ or } -1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

BJ is the matrix obtained from B by flipping the signs of those columns k s.t. $J_{kk} = -1$. There are 2^l different sign matrices, and they all satisfy $J = J^{-1}$. Hence $Bz = v$ *iff* $B(JJ^{-1})z = v$, or $(BJ)(Jz) = v$.

As all sign inversions of columns BJ are equi-probable, so are all Jz obtained from them. Since (B,v) is VD $\bar{z} := B^{-1}v$ satisfies $\bar{z}_i \neq 0 \ \forall_i$. Hence $Pr[J\bar{z} \in S] = \frac{1}{2^l}$ for every orthant S of R^l.

∎

Corollary 3: For VD data (A,b,c) satisfying CSI, the expected number of vertices in R_+^d is $\sum_{l=0}^{d} \binom{m}{l} \binom{d}{d-l} 2^{-l}$

Consider now a model which assumes Weak Regularity, Row Sign Invariance as well as Column Sign Invariance. The reasoning of the previous section obviously holds and by the above corollary we can replace $\binom{k+d}{d}$ by $\sum_{l=0}^{d} \binom{k}{l} \binom{d}{d-l} 2^{-l}$ as the expected number of vertices which contribute pivots in stage $k+1$.

Hence the average number of pivots is bounded by

$$(d+1) \sum_{k=1}^{m} \left[\sum_{l=0}^{d} \binom{k}{l} \binom{d}{d-l} 2^{-l} \right] 2^{-k}$$

And using Lemma B (see Appendix) we get that this sum for $m \to \infty$ is equal to $2(d+1) \cdot 1.5^d$. So we conclude:

Theorem 2: For a model satisfying Weak Regularity and Sign Invariance, the average number of pivots required by the PCBC Algorithm is bounded by $2(d+1)\,1.5^d$, independent of m.

Corollary 4: For a model which is TWR and ESI

$$\rho_{PCBC}(m,d) \le 2(\Delta+1)\,1.5^{\Delta} \quad \text{where } \Delta = \min(m,d).$$

8. Analysis of a Feasible Model

Let us now turn to a model which generates only feasible linear programs. This model generates problems of one of the forms used by Borgwardt [Bo3], namely

$$(P') \qquad \begin{array}{c} \max c^T x \\ Ax \le e \\ x \ge 0 \end{array}$$

where $A \varepsilon R^{m \times d}$, $e^T := (1,1,\ldots,1) \varepsilon R^m$.

Instead of solving (P') which is always feasible since $x = 0$ is feasible, we shall solve its dual

$$(D') \qquad \begin{array}{c} \min e^T y \\ A^T y \ge c \\ y \ge 0 \end{array}$$

This program has a bounded solution for all A, c for which it is feasible, since the objective function value is bounded below by 0.

We use the PCBC Algorithm to solve (D'). The only difference is that stage $d+1$ is unnecessary since if we reach that stage we already have a vertex of X optimal with respect to $e^T x$.

Since our choice of objective function e in the first m stages of the PCBC Algorithm was made especially in order to guarantee boundedness of the solution, nothing is changed in our analysis from the previous sections and we conclude:

Theorem 3: For a model of feasible LPs of the form (P') which is Dual Weakly Regular and Extended Column-Sign Invariant, the PCBC Algorithm requires an average of at most $(m + 1)\ 2^{m+1}$ pivots, independent of d.

Proof: Apply Theorem 1 to the dual of (P') which has data (A^T, c, e) with $A^T \varepsilon R^{d \times m}$.

∎

Note that we cannot replace here m by $\min(m,d)$ as we did in Corollary 2, since we do not have here the choice between solving (P') and (D'). However we can improve the exponent if we assume that the data is also RSI.

Theorem 4: For a feasible model which is dual WR, ECSI and $[A]$ is also RSI

$$\rho_{PCBC}(m,d) \leq 2(m + 1)\ 1.5^m$$

independent of d.

Proof: Apply Theorem 2 to the dual of (P').

∎

9. The CBC Algorithm

We now define a new algorithm to solve the Linear Programming Problem. This algorithm uses the Constraint-by-Constraint idea, but it allows one to use *any primal* algorithm at every stage. It turns out that this very general algorithm (which we call the CBC Algorithm) is also analyzable in our probabilistic models.

Statement of the CBC Algorithm:

Stage 0: ($\bar{x} = 0$ is a feasible vertex of $X^{(0)}$)
Set $\bar{x} \leftarrow 0$; Go to stage 1.

Stage k ($1 \le k \le m$): ($\bar{x}$ is a feasible vertex of $X^{(k-1)}$)
Use any primal algorithm to solve $\max\{a_k^T x \mid x \varepsilon X^{(k-1)}\}$ starting at $\bar{x}$.
If the optimum does not satisfy $a_k^T x \ge b_k$ - Stop. $X^{(k)} = \Phi$ so $X = \Phi$.
Otherwise choose any vertex $\hat{x}$ of $X^{(k)}$ obtained along the path. Go to step $k + 1$ with $\bar{x} \leftarrow \hat{x}$.

Stage $m + 1$: ($\bar{x}$ is a feasible vertex of X)
Use any primal algorithm to solve $\min\{c^T x \mid x \varepsilon X\}$ starting at $\bar{x}$.
The optimal vertex (or ray) generated provides the required solution.

The proof that the algorithm is valid is obtained from the following invariant assertions:

(i) If $\max\{a_k^T x \mid x \varepsilon X^{(k-1)}\} < b_k$ then $X^{(k)} = \Phi$.

(ii) If $\max\{a_k^T x \mid x \varepsilon X^{(k-1)}\} \ge b_k$ then $X^{(k)} \ne \Phi$ and every primal algorithm starting at $\bar{x}$ reaches a feasible vertex of $X^{(k)}$.

Proof: (i) is immediate. To prove (ii) note that either in the beginning of stage $k + 1$ $a_k^T \bar{x} \ge b_k$ and then $\bar{x}$ is a vertex in $X^{(k)}$, or at some point along the path $a_k^T \hat{x}$ increases to a value of b_k, and hence $\hat{x}$ is a vertex in $X^{(k)}$.

∎

10. Analysis of the CBC Algorithm

The only crucial property of the CBC Algorithm in order to carry out the analysis is that in every subproblem we follow only *feasible* vertices of the corresponding set. Assuming non-degeneracy each vertex of the feasible set may be used in at most one pivot. So, the average number of pivots in each stage is bounded by the average number of feasible vertices per instance in that stage.

In stage $k+1$ the $k+d$ hyperplanes generate at most $\binom{k+d}{d}$ vertices. Under the Vertex Distinctness assumption each vertex is incident on exactly 2^d instances. Hence at most $\binom{k+d}{d}2^d$ pivots are performed in that stage. Under the RSI assumption all 2^k instances generated at stage $k+1$ are equi-probable. So the average number of pivots in stage $k+1$ is at most $\binom{k+d}{d}2^d\,2^{-k}$, and the average number of pivots summed over all stages is bounded by

$$2^d \sum_{k=0}^{m} \binom{k+d}{d} 2^{-k} = 2^{2d} \sum_{j=d}^{m+d} \binom{j}{d} 2^j \le 2^{2d+1}$$

where in the last inequality we used Lemma A from the Appendix. So we conclude:

Theorem 5: Under a model which satisfies Vertex Distinctness and Row-Sign Invariance, every variant of the CBC Algorithm requires on the average at most 2^{2d+1} pivots, independent of m.

If we add the Column-Sign Invariance assumption we can improve the bound using Lemma 1 and the following

Lemma 2: Let $\bar{x}$ be a vertex generated by the d hyperplanes

$$\begin{aligned} a_i x &= b_i \qquad & i &= 1,\ldots,l \\ x_i &= 0 \qquad & i &= l+1,\ldots,d \end{aligned}$$

Then out of the 2^d cells generated around $\bar{x}$ by those hyperplanes, exactly 2^l are in R_+^d.

Proof: The 2^d cells are generated by replacing the equality signs by inequalities in all possible ways. However, in R_+^d the last $d-l$ signs are restricted to $x_i \ge 0$. Under that restriction all 2^l cells generated by the other inequalities are in R_+^d.

Assuming CSI, from Lemma 1 and 2 we get that the total number of pivots performed in stage $k + 1$ in the positive orthant is bounded by

$$\sum_{l=0}^{d} \binom{k}{l} \binom{d}{l} 2^{-l} 2^{l} = \sum_{l=0}^{d} \binom{k}{l} \binom{d}{l} = \binom{k+d}{d}$$

Hence the expected number of pivots is bounded by

$$\sum_{k=0}^{m} \binom{k+d}{d} 2^{-k} = 2^{d} \sum_{j=d}^{m+d} \binom{j}{d} 2^{-j} \leq 2^{d+1}$$

where the last inequality uses Lemma A. So we get the following result:

Theorem 6:

(1) For a VD, SI model $\rho_{csc}(m,d) \leq 2^{d+1}$

(2) For a Twice-VD, ESI model $\rho_{csc}(m,d) \leq 2^{\Delta+1}$.

For the feasible model (P') we can apply all our arguments to the dual (D') as in section 8. Since we did not make any assumptions on the objective function distribution in order to obtain Theorems 5 and 6(1), these results hold when the objective function in (D') is kept positive. So we get:

Theorem 7: For a feasible model of the form (P') if it is Dual-VD and ECSI then

$$\rho_{csc}(m,d) \leq 2^{2m+1}$$

Theorem 8: For a feasible, Dual-VD, ESI model

$$\rho_{csc}(m,d) \leq 2^{m+1}$$

11. Linear Programs Without Non-negativity Constraints

We now want to consider linear programs of the form

$$(\bar{P}) \qquad \begin{array}{l} \min c^T x \\ Ax \geq b \end{array} \qquad A \varepsilon R^{n \times d}, n \geq d$$

It is well known that such a problem, which does not include sign constraints, can be presented in equivalent form $(\hat{P})$ with sign constraints. We shall show that the Row Sign Invariance assumption on $(\bar{P})$ is equivalent to ESI on $(\hat{P})$. This will enable us to use the results of previous sections.

Define first $A =: \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ $b =: \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ where $A_1 \varepsilon R^{d \times d}$, $b_1 \varepsilon R^d$ $m := n - d$
$\bar{c} := c^T A_1^{-1}$ $\bar{A} := A_2 A_1^{-1}$ $\bar{b} := b_2 - A_2 A_1^{-1} b_1$

Then the equivalent LP is:

$$(\hat{P}) \qquad \begin{array}{l} \min \bar{c}^T x \\ \bar{A} x \geq \bar{b} \\ x \geq 0 \end{array} \qquad \bar{A} \varepsilon R^{m \times d}$$

Lemma 3: A model of $(\bar{P})$ is RSI if and only if the corresponding model on $(\hat{P})$ is ESI.

Proof: Let us use the notation of sign matrices, introduced in section 7. Let $J_1 \varepsilon R^{d \times d}$, $J_2 \varepsilon R^{m \times m}$ be sign matrices. Let $\bar{P}(J_1, J_2)$ be the problem $\bar{P}$ with the corresponding sign assignment to the rows:

$$\bar{P}(J_1, J_2) \qquad \begin{array}{l} \min c^T x \\ J_1 A_1 x \geq J_1 b_1 \\ J_2 A_2 x \geq J_2 b_2 \end{array}$$

The first set of inequalities is equivalent to

$$J_1 A_1 x - u = J_1 b_1 \qquad \text{for some } u \varepsilon R^d, \ u \geq 0.$$

or, since $J_1 = J_1^{-1}$

$$x = A_1^{-1} b_1 + A_1^{-1} J_1 u, \qquad u \geq 0$$

Introduce x into the second set of inequalities to get:

$$J_2 A_2 [A_1^{-1} b_1 + A_1^{-1} J_1 u] \geq J_2 b_2$$

or

$$J_2 \bar{A} J_1 u \geq J_2 \bar{b}$$

And using the expression for x in the objective function,

$$c^T x = c^T [A_1^{-1} b_1 + A_1^{-1} J_1 u] = \bar{c}^T J_1 u + \text{constant}.$$

So $\overline{P}(J_1, J_2)$ is equivalent to

$$\hat{P}(J_1, J_2) \qquad \begin{array}{l} \min \bar{c}^T J_1 u \\ J_2 \bar{A} J_1 u \geq J_2 \bar{b} \\ u \geq 0 \end{array}$$

Under the RSI assumption for $(\overline{P})$, all the 2^n instances $\overline{P}(J_1, J_2)$ are equi-probable. So by the equivalence just established all the instances $\hat{P}(J_1, J_2)$ are also equi-probable. So $\hat{P}$ is ESI. The converse follows in the same way.

∎

By the above discussion we see that the problems without sign constraints can be dealt with in the same manner as those discussed earlier. In particular Lemma 3 and Corollary 4 imply:

Theorem 9: For a model of LPPs of the form $(\overline{P})$, assuming TWR, RSI

$$\rho_{PCBC}(n,d) \leq 2(\delta + 1)\, 1.5^{\delta} \qquad \text{where } \delta := \min(n - d, d)$$

And Lemma 3 and Theorem 6(b) imply:

Theorem 10: For a model of LPPs of the form $(\bar{P})$ assuming TVD, RSI

$$\rho_{CBC}(n,d) \le 2^{\delta+1} \qquad \text{where } \delta := \min(n-d,d)$$

Since $(\hat{P})$ is ESI if and only if the matrix $\begin{bmatrix} \bar{A} & \bar{b} \\ I & 0 \end{bmatrix}$ is RSI, this section just clarifies that the sign constraints do not play any special role in either the model of the algorithm.

12. A Generalized CBC Algorithm

We now present and analyze a generalized version of the CBC algorithm. In the PCBC Algorithm we allowed only efficient basic solutions to be used in every stage. In the CBC Algorithm we allowed all primal feasible bases to be used. In the Generalized CBC (GCBC) Algorithm presented below we relax that requirement and allow the use of *both feasible and infeasible* bases in each stage. The only requirement we make is that the algorithm should proceed in a constraint-by-constraint manner, hence reaching the $k-th$ subproblem only if the $(k-1)^{st}$ is feasible.

We shall show that under our probabilistic models the GCBC Algorithm still maintains an average of constant time when d is fixed for any m. We also show that the "Big M" method for Phase I together with Bland's rule are a special case of the GCBC.

Statement of the GCBC Algorithm:

Stage 0: Let $\bar{s}$ be the unique basis of $X^{(0)}$.
$\bar{s}$ is also a feasible basis. Go to Stage 1.

Stage k $(1 \le k \le m)$:
Scan bases (both feasible and infeasible) of $X^{(k)}$ until either

(1) a feasible basis $\hat{s}$ of $X^{(k)}$ is found. Go to Stage $k+1$, or

(2) Infeasibility of $X^{(k)}$ is demonstrated. Stop, $X = \Phi$.

Stage $m + 1$:

Scan bases of X to find an optimal or unbounded solution.

Analysis for the ESI Model:

Stage k is reached only if $X^{(k-1)}$ is feasible. The probability of $X^{(k-1)}$ being feasible is [AB2]: $2^{-(k+d-1)} \sum_{i=0}^{d} \binom{k+d-1}{i}$. If that stage is reached, at most all vertices in that d-arrangement can be pivoted on, and their number is $\binom{k+d}{d}$. Hence an upper bound on the expected number of pivots is:

$$\sum_{k=1}^{m+1} \binom{k+d}{d} 2^{-(k+d-1)} \sum_{i=0}^{d} \binom{k+d-1}{i}$$

Partition the sum on k to $k = 1,\ldots,d$ and $k = d+1,\ldots,m$. The first sum gives:

$$\sum_{k=1}^{d} \binom{k+d}{d} 2^{-(k+d-1)} \sum_{i=0}^{d} \binom{k+d-1}{i} \le 2 \sum_{k=1}^{d} \binom{k+d}{d} 2^{-(k+d)} \binom{2d}{d} (d+1) = O(\sqrt{d} \cdot 2^{2d})$$

The second sum gives:

$$\sum_{k=d+1}^{m+1} \binom{k+d}{d} 2^{-(k+d-1)} \sum_{i=0}^{d} \binom{k+d-1}{i} \le 2(d+1) \sum_{k=d+1}^{m+1} \binom{k+d}{d} 2^{-(k+d)} \binom{k+d-1}{d}$$

$$= 2(d+1) \sum_{k=d+1}^{m+1} \binom{k+d}{d}^2 2^{-(k+d)} \frac{k}{k+d} \le$$

$$\le 2(d+1) \sum_{l=2d+1}^{m+1} \binom{l}{d}^2 2^{-l} = O(2^{4d})$$

where the last equality follows from Lemma C in the Appendix.

So we conclude:

Theorem 11: For TWR, ESI model

$$\rho_{GCBC}(m, d) = O(2^{4\Delta})$$

Analysis of a RSI Model:

Recall that in this model we have d fixed nonnegativity constraints, and only the m matrix inequalities change their directions. So there are 2^k subinstances in the subproblem containing k matrix inequalities and d nonnegativity constraints. Observe that the nonnegativity constraints can only eliminate some of the cells generated by the matrix constraints but they cannot add new feasible cells. So the total number of cells in stage k is bounded by the total number of cells generated by the matrix inequalities. This number is 2^k if $k \le d$ and $\sum_{i=0}^{d} \binom{k}{i}$ if $k \ge d$ [Bu]. So we conclude:

$$P_k := Pr\,[\text{an instance in stage } k \text{ is feasible}] \le \begin{cases} 1 & \text{if } k \le d \\ 2^{-k} \sum_{i=0}^{d} \binom{k}{i} & \text{if } k \ge d \end{cases}$$

In stage $k + 1$ there are $\binom{k+d+1}{d}$ vertices, and the GCBC algorithm may 'visit' them only if $X^{(k)}$ is feasible. Hence

$$\rho_{GCBC}(m,d) \le \sum_{k=0}^{m+1} P_k \binom{k+d+1}{d} \le \sum_{k=0}^{d} \binom{k+d+1}{d} + \sum_{k=d+1}^{m+1} \binom{k+d+1}{d} 2^{-k} \sum_{i=0}^{d} \binom{k}{i} =$$

$$= \binom{2d+2}{d+1} - 1 + \sum_{k=d+1}^{2d} \binom{k+d+1}{d} 2^{-k} \sum_{i=0}^{d} \binom{k}{i} + \sum_{k=2d+1}^{m+1} \binom{k+d+1}{d} 2^{-k} \sum_{i=0}^{d} \binom{k}{i} \le$$

$$\le \binom{2d+2}{d+1} + d \cdot \binom{3d+1}{d} 2^{-(d+1)} \cdot (d+1) \binom{2d}{d} + \sum_{k=2d+1}^{m+1} \binom{k+d+1}{d} 2^{-k} (d+1) \binom{k}{d} =$$

$$= O(2^{2d}) + O(d^2 \cdot 2^{3d}\, 2^{-d} \cdot 2^{2d} \cdot d^{-1}) + \sum_{k=2d+1}^{m+1} \binom{k}{d}^2 2^{d+1}\, 2^{-k} (d+1)$$

$$= O(d \cdot 2^{4d}) + (d+1)\, 2^{d+1} \sum_{k=2d+1}^{m+1} \binom{k}{d}^2 2^{-k} = O(d \cdot 2^{4d} + d \cdot 2^{d}\, 2^{4d}\, d^{-1}) = O(2^{5d})$$

So we conclude:

Theorem 12: In the WR, RSI model $\rho_{GCBC}(m,d) = O(2^{5d})$.

Relation to Bland's Algorithm:

Let us now investigate the relation of the GCBC algorithm to Bland's rule. Bland's algorithm [Bln] maintains primal feasibility and in every iteration the next variable chosen to enter the basis is the one with the least index which has a negative reduced cost coefficient. In other words, if $\bar{c}$ is the current reduced cost vector, then $\min\{i \mid \bar{c}_i < 0\}$ is the index of the entering variable. So variable $t+1$ enters the basis only if the subproblem

$$P^{(t)}: \quad \begin{array}{ll} \min \sum_{i=1}^{t} c_i x_i & \\ s.t. \ \sum_{j=1}^{t} A_{ij} x_j \geq b_i & i = 1,\ldots,m \\ x \geq 0 & \end{array}$$

has an optimal solution.

The corresponding dual subproblem is

$$D^{(t)}: \quad \begin{array}{ll} \max b^T y & \\ A_k^T \cdot y \leq c_k & k = 1,\ldots,t \\ y \geq 0 & \end{array}$$

So in terms of the dual problem Bland's algorithm maintains *dual* feasibility and proceeds in a constraint-by-constraint manner, reaching subproblem $D^{(t+1)}$ only if $D^{(t)}$ is optimal.

In the above discussion we did not specify how the Bland's algorithm is initialized. In fact, Bland does not specify that himself, since he describes a pivoting rule which may be implemented in any algorithm maintaining primal feasibility. One possible way to obtain initial feasibility is by introducing an artificial variable:

$$P(M) \quad \begin{array}{ll} \min M x_0 + c^T x & \\ x_0 e + A x \geq b & \\ x_0, x \geq 0 & x_0, M \in R, \ e := (1,1,\ldots,1) \in R^m \end{array}$$

A starting feasible solution for solving $P(M)$ is $(x_0, x) = (b_k, 0)$ where $b_k = \max_i b_i$. (We assume that $b \nleq 0$, since otherwise $x = 0$ is feasible and no artificial variable is needed). If M is sufficiently large, a finite solution to $P(M)$ with $x_0 > 0$ implies that P is infeasible, and an unbounded solution with $x_0 > 0$ implies P is unbounded. This method is known as the "Big M" method [D], [Ch].

Assume now that the "Big M" method is used for initialization, and the Bland rule is used for pivoting. The dual subproblem is stage k is:

$$D^{(t)}_{(M)} \quad \begin{array}{l} \max b^T y \\ e^T y \leq M \\ A_k^T \cdot y \leq c_k \qquad k = 1,\ldots,t \\ y \geq 0 \end{array}$$

If we assume that the dual data is sampled from a RSI distribution, then the situation is very similar to that analyzed in the previous subsection. The only difference is that the extra constraint $e^T y \leq M$ is present in all subproblems. Since all instances satisfy $y \geq 0$ in this model, then the *same* constraint $e^T y \leq M$ can be used to bound all instances. Also, by choosing M sufficiently large, no feasible cells in $D^{(k)}$ are eliminated in $D^{(k)}_{(M)}$. So we can carry the same analysis replacing $\binom{k+d+1}{d}$ by $\binom{k+d+2}{d}$, and the same upper bound is obtained. Summarizing the above we get:

Theorem 13: The Simplex variant obtained by "Big M" initialization and Bland's pivoting rule has the following properties:

(1) It performs a sequence of pivots corresponding to a special case of the GCBC algorithm (with an extra bounding constraint) in the dual problem.

(2) If the data is dual-WR and ECSI, then it requires on the average no more than $O(2^{5m})$ pivots, independent of d.

Note that a better bound may be obtained for the Big M-Bland algorithm by taking into account the fact that only dual feasible bases can be pivoted on.

13. Summary and Discussion

We have presented a family of Simplex variants which proceed in a constraint-by-constraint manner, and have described three successively more general algorithms within this family. The main features of these algorithms are summarized in the following table:

Algorithm | Property	Primal Feasibility	Cooptimality	Free Choice within the Algorithm
PCBC	preserved	preserved	Starting objective
CBC	preserved	not preserved	Any primal algorithm can be used in every stage.
GCBC	not preserved	not preserved	Any vertex following algorithm can be used in every stage.

Table 1: Properties of the algorithms within every stage.

The probabilistic models we used required certain weak regularity conditions and sign invariance properties. The sign invariance requirements for the models are summarized in the following table:

The Model	Invariance with respect to all sign changes in
CSI	columns of A
ECSI	columns of $\begin{bmatrix} c^T \\ A \end{bmatrix}$
RSI	rows of $[A, b]$
SI	columns of A, rows of $[A, b]$
ESI	columns of $\begin{bmatrix} c^T \\ A \end{bmatrix}$, rows of $[A, b]$

Table 2: Sign invariance properties of the probabilistic models (Invariance is stated for fixed data A, b, c inducing the LP $\min c^T x,\ Ax \geq b,\ x \geq 0$*).*

The upper bounds obtained for the expected number of pivots performed by each algorithm under each model are summarized in the following table:

Model | Algorithm	PCBC	CBC	GCBC
RSI	$(d+1)\,2^{d+1}$	2^{2d+1}	$O(2^{5d})$
SI	$2(d+1)\,1.5^{d}$	2^{d+1}	$O(2^{5d})$
ESI	$2(\Delta+1)\,1.5^{\Delta}$	$2^{\Delta+1}$	$O(2^{4d})$
feasible, ESCI	$(m+1)\,2^{m+1}$	2^{2m+1}	--
feasible, dual-SI	$2(m+1)\,1.5^{m}$	2^{m+1}	--

Table 3: Upper bounds on $\rho(m,d)$ under several probabilistic models.

Leaving aside the feasible models for a moment, we see that all three algorithms require a number of pivots bounded by functions of d only. These functions vary with the algorithm and the model, but they are all exponential in d.

These results seem "strong" when $m \gg d$, and specifically when d is fixed and m tends to infinity, since in that case $\rho(m,d)$ is bounded by a constant. However, we believe that all these models have a basic problematic characteristic, which underscores these results: When $m \gg d$ ($d \gg m$) all but a vanishing fraction of the problems generated by the models will be infeasible (unbounded) [AB2]. So in those situations we are essentially counting the number of pivots performed until infeasibility or unboundedness of the problem is demonstrated. It seems that detecting infeasibility or unboundedness is *an easier problem* than solving a comparable linear program which has an optimal solution. The reasoning is that while there are many bases which demonstrate infeasibility (or unboundedness) in an infeasible (unbounded) problem, there may be a unique optimal basis in an optimal problem, and it may take longer to find it. Hence the relevance of these results to the observed good performance of the Simplex method is questionable. It is still possible, of course, that for the same models and the same algorithms better bounds will be obtained (perhaps polynomial rather than exponential in d) rendering the results more meaningful.*

*See the final note to this section.

Another interpretation of the results, closely related to the above discussion, is the following: The seemingly good behavior of the algorithms is mainly due to the probabilistic models we used, and *not due to the Simplex method.* The CBC and the GCBC algorithms are very general procedures, and can be viewed as enumeration algorithms. The GCBC is more of a "*reductio ad absurdum*" than a practical algorithm, since it satisfies *only* the Constraint-by-Constraint idea. Namely, it may scan *all* bases of $P^{(k)}$, but it will do that only if $P^{(k-1)}$ is feasible. After about d^2 constraints the probability of an instance being feasible is minute, and it decreases with k more rapidly than the number of bases grows with k [AB2]. So the contribution to the expected number of pivots by additional constraints is negligible, and even an enumeration algorithm yields an expected number of pivots bounded by a function of d only. Again this does not exclude the possibility of improved analysis for the PCBC algorithm, reflecting on the nature of the Simplex method rather than on the model.*

We should mention here that the same drawbacks are present in Smale's results. Our models generalize Smale's original probabilistic model [S1], but do not apply to his second, more general model [S2], which assumes symmetry with respect to coordinate permutations, rather than symmetry with respect to rotations. However, Smale's results for both models have been obtained for fixed d and $m \to \infty$, and it was later demonstrated by Blair [Blr] that in that case most of the "good behavior" of the algorithm is due to the small chance of a column to be in any basis generated by the algorithm. So again this is still a reflection more on the model than on the Simplex variant used.

Let us now turn to the feasible models: The results we obtained for the feasible models depend only on m, the number of matrix constraints. When m is fixed and the number of variables increases in these models, almost all instances will become unbounded. This raises the same difficulty with interpreting our results as before. Borgwardt [Bo2], however, gets a result of order $d^4 \cdot m$ for another feasible model. By fixing d and increasing m the probability of the problem being both feasible and bounded in his model tends to one. Hence Borgwardt does get a linear bound for this case which we consider more difficult.

*See the final note to this section.

The fact that Bland's rule can be viewed as a dual Constraint-By-Constraint procedure is also interesting. We suspect that some other Simplex variants may also have this property "in disguise", and recognizing this property may facilitate their probabilistic analysis. Borgwardt's full (Phase I-II) algorithm is also a variable-by-variable algorithm, presentable as a CBC-like algorithm after dualizing. However, his algorithm was presented only in the context of linear programs without nonnegativity constraints for which the zero vector is feasible, and it cannot be used to solve general linear programs which do not assume these properties.

A Final Note

There have been several interesting developments since the completion of this work:

1. Megiddo [Me1] showed that the Self-Dual Simplex variant requires on the average at most $\bar{c}(d)$ pivots, under the same probabilistic assumptions made by Smale in [S1]. So his result is of the same type as ours. The constant $\bar{c}(d)$ he gets is, however, superexponential in d.

2. We managed to show [AKS] that the PCBC Algorithm, when used with a special starting objective (rather than with e, as presented here), maintains an average number of pivots which is at most *quadratic* in $\min(m,d)$, under the ESI model with slightly stronger regularity conditions.

3. At the same time, under the same probabilistic assumptions, Todd [T] and Adler and Megiddo [AM] obtained an $O(min(m,d)^2)$ bound on the expected number of pivots for the Self-Dual variant of the Simplex algorithm with a special kind of initialization.

4. After the completion of these three investigations, Megiddo [Me2] observed that, although the PCBC algorithm and the Self-Dual algorithm are in general quite different, the special initialization schemes used in the analyses result in the execution of exactly the same sequence of pivots in both algorithms. Thus all three investigations are concerned with the same Simplex variant.

REFERENCES

[A] Adler, I., "The Expected Number of Pivots Needed to Solve Parametric Linear Programs and the Efficiency of the Self-Dual Simplex Method", Working Paper, Dep. of *IEOR*, University of California, Berkeley (June 1983).

[AB1] Adler, I. and Berenguer, S. E., "Random Linear Programs", Operations Research Center report no. 81-4, University of California, Berkeley (1981).

[AB2] Adler, I. and Berenguer, S. E., "Generating Random Linear Programs", Manuscript, Department of Industrial Engineering and Operations Research, University of California, Berkeley (1983). Submitted to *Mathematical Programming*.

[AKS] Adler, I., Karp, R. M., Shamir, R., "A Simplex Variant Solving an $m \times d$ Linear Program in $O(min(m^2,d^2))$ Expected Number of Pivots Steps", Report UCB CSD 83/158, Computer Science Division, University of California, Berkeley (December 1983).

[AM] Adler, I. and Megiddo, N., "A Simplex-type Algorithm Solves Linear Programs of Order $m \times n$ in Only $O((min(m,n))^2)$ Steps on the Average", preliminary report (November 1983).

[Bln] Bland, R. G.,"New Finite Pivoting Rules For the Simplex Method" *Math. of OR* vol. 2,no. 2, pp.103-107 (1977).

[Blr] Blair, C., "Random Linear Programs with Many Variables and Few Constraints". Faculty Working Paper No.946, College of Commerce and Business Administration, University of Illinois at Urbana-Champaign (April 1983).

[Bo1] Borgwardt, K. H., "Some Distribution - Independent Results About the Asymptotic Order of the Average Number of Pivot Steps of the Simplex Method", *Math. of OR*. vol. 7, no. 3, pp. 441-462 (August 1982).

[Bo2] Borgwardt, K. H., "The Average Number of Pivot Steps Required by the Simplex-Method is Polynomial", *Zeitschrift für OR*, vol. 26, pp. 157-177 (1982).

[Bo3] Borgwardt, K. H., "Zum Rechenaufwand von Simplexverfahren", *Methods of OR* 31, pp 83-97 (1978).

[Bu] Buck, R. C., "Partition of Space", *American Mathematical Monthly* vol. 50, pp 541-544 (1943).

[Ch] Charnes, A., "Optimality and Degeneracy in Linear Programming", *Econometrica* vol. 20,no.2 pp 160-170 (1952)

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

[GaS] Gass, S. I. and Saaty, E. L., "The Computational Algorithm for the Parametric Objective Function", *Naval Res. Logistic Quart.*, vol. 2, no. 1, pp. 39-45 (June 1955).

[GoS] Goldfarb, D. and Sit, W.Y., "Worst Case Behavior of the Steepest Edge Simplex Method", *Discrete Applied Math.*, vol. 1, pp 277-285 (1979).

[H] Haimovich, M., "The Simplex Algorithm is Very Good! - On the Expected Number of Pivot Steps and Related Properties of Random Linear Programs" Draft, 415 Uris Hall, Columbia University, New York (April 1983).

[J] Jeroslow, R. G., "The Simplex Algorithm with the Pivot Rule of Maximizing Improvement Criterions", *Discrete Mathematics* 4, pp. 367-378 (1973).

[Kh] Khachian, L. G., "A Polynomial Algorithm for Linear Programming", *Doklady Akad. Nauka USSR*, 244, No. 5, pp. 1093-96 (1979).

[KM] Klee, V. and Minty, G. J., "How Good is the Simplex Algorithm?", in Shish, O., ed., *Proceeding of the 3rd Symposium on Inequalities*, AP, New York, pp. 159-175 (1972).

[KQ] Kuhn, H. W. and Quandt, R. E., "An Experimental Study of the Simplex Method", in Metropolis, et al, eds. *Proceeding of the 15th Symposium in Applied Mathematics*, AMS, Providence, R. I., pp. 107-124 (1963).

[MS] May, J. H. and Smith, R. L., "Random Polytopes: Their Definition, Generalization and Aggregate Properties", *Mathematical Programming*, 24, pp. 39-54 (1982).

[Me1] Megiddo, N., "Improved Asymptotic Analysis of the Average Number of Steps Performed by the Self Dual Simplex Algorithm", Preliminary report, Department of Computer Science, Stanford University and Xerox PARC (September 1983).

[Me2] Megiddo, N., "A Note on the Generality of the Self-Dual Algorithm With Various Starting Points", Department of Computer Science, Stanford University and Xerox PARC (December

1983).

[Mu] Murty, G. K., "Computational Complexity of Parametric Linear Programming", *Mathematical Programming* vol. 19, pp. 213-219 (1980).

[S1] Smale, S., "On the Average Speed of the Simplex Method of Linear Programming", *Mathematical Programming* 27 (1983) to appear.

[S2] Smale, S. "The Problem of the Average Speed of the Simplex Method", *Proc. of the 11th International Symp. on Mathematical Programming*, Universitat Bonn, pp. 530-539 (August 1982).

[T] Todd, M. J., "Polynomial Expected Behaviour of a Pivoting Algorithm for Linear Complementary and Linear Programming Probles", Technical Report No. 595, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, New York (November 1983)

[Z] Zadeh, N., "A Bad Network Example for the Simplex Method and Other Minimum Cost Flow Algorithms", *Mathematical Programming* vol. 5, pp. 255-266 (1973).

APPENDIX

Lemma A: For every integer i $\sum_{k=i}^{\infty} \frac{1}{2^k} \binom{k}{i} = 2.$

Proof: By induction on i:

$$\text{For } i = 0 \quad \sum_{k=0}^{\infty} \frac{1}{2^k} \binom{k}{0} = \sum_{k=0}^{\infty} \frac{1}{2^k} = 2.$$

$$\text{For } i = 1 \quad \sum_{k=1}^{\infty} \frac{1}{2^k} \binom{k}{1} = \sum_{k=1}^{\infty} \frac{k}{2^k} = 2.$$

Assume that for i $F(i) := \sum_{k=i}^{\infty} \frac{1}{2^k} \binom{k}{i} = 2.$

Then

$$F(i+1) = \sum_{k=i+1}^{\infty} \frac{1}{2^k} \binom{k}{i+1} = \sum_{k=i+1}^{\infty} \frac{1}{2^k} \left[\binom{k-1}{i+1} + \binom{k-1}{i} \right]$$

$$= \frac{1}{2} \sum_{k-1=i}^{\infty} \frac{1}{2^{k-1}} \binom{k-1}{i+1} + \frac{1}{2} \sum_{k-1=i}^{\infty} \frac{1}{2^{k-1}} \binom{k-1}{i}$$

$$= \frac{1}{2} F(i+1) + \frac{1}{2} \cdot 2 \qquad \text{by the induction hypothesis}$$

So

$$\frac{1}{2} F(i+1) = 1 \rightarrow F(i+1) = 2.$$

■

Lemma B: $\sum_{k=1}^{m} \sum_{j=0}^{d} \binom{d}{j} \binom{k}{j} 2^{-j} 2^{-k} \leq 2 \cdot 1.5^d$

with equality for $m \rightarrow \infty$.

Proof: $\sum_{k=1}^{m} \sum_{j=0}^{d} \binom{d}{j} \binom{k}{j} 2^{-j} 2^{-k} =$

$$= \sum_{j=0}^{d} 2^{-j} \binom{d}{j} \sum_{k=1}^{m} 2^{-k} \binom{k}{j} \le$$

$$\le 2 \sum_{j=0}^{d} 2^{-j} \binom{d}{j} = \qquad \text{by Lemma A}$$

$$= 2 \sum_{j=0}^{d} \binom{d}{j} \left(\frac{1}{2}\right)^{j} \cdot (1)^{d-j} =$$

$$= 2\left(1 + \frac{1}{2}\right)^{d} = 2 \cdot 1.5^{d} \qquad \text{by the Binomial Theorem}$$

And the only inequality we used is tight for $m \to \infty$.

Lemma C: $\sum_{l=2m}^{\infty} \binom{l}{m}^2 2^{-l} = 0\left(\frac{1}{m} \cdot 2^{4m}\right)$

Proof: $\sum_{l=2m}^{\infty} \binom{l}{m}^2 2^{-l} = \sum_{j=0}^{\infty} \binom{2m+j}{m}^2 2^{-j} 2^{-2m} =$

$$= 2^{-2m} \sum_{j=0}^{\infty} \left[\binom{2m}{m} \frac{2m+1}{m+1} \cdot \frac{2m+2}{m+2} \cdots \frac{2m+j}{m+j}\right]^2 2^{-j}$$

Note that $\frac{2m+j}{m+j} \le 2$ for all j, and for $j \ge 2m$ $\frac{2m+j}{m+j} \le \frac{4}{3}$. So

$$\le 2^{-2m} \binom{2m}{m}^2 \left\{\sum_{j=0}^{2m-1} (2^j)^2 2^{-j} + \sum_{j=2m}^{\infty} \left[2^{2m} \left(\frac{4}{3}\right)^{j-2m}\right]^2 2^{-j}\right\}$$

$$= 2^{-2m} \binom{2m}{m}^2 \left\{2^{2m} + 2^{4m} \sum_{j=2m}^{\infty} \left(\frac{16}{9}\right)^{j-2m} \cdot 2^{-j}\right\} =$$

$$= 2^{-2m} \binom{2m}{m}^2 \left\{2^{2m} + 2^{4m} 2^{-2m} \sum_{k=0}^{\infty} \left(\frac{16}{18}\right)^{k}\right\} =$$

$$= 2^{-2m} \binom{2m}{m}^2 \left\{2^{2m} + 2^{2m} \frac{1}{1 - 8/9}\right\} = 10 \binom{2m}{m}^2$$

Since $\binom{2m}{m} = O\left(\frac{2^{2m}}{\sqrt{m}}\right)$, the results follows.