The Criss-Cross Method Can Take $\Omega(n^d)$ Pivots

Komei Fukuda*
Institute for Operations Research and Institute of Theoretical Computer Science
ETH Zentrum
Zurich, Switzerland
fukuda@ifor.math.ethz.ch

Bohdan Kaluzny
School of Computer Science
McGill University
Montreal, Canada
beezer@cs.mcgill.ca

ABSTRACT

Using deformed products of arrangements, we construct a family of linear programs with n inequalities in $\Re^d$ on which, in the worst-case, the least-index criss-cross method requires $\Omega(n^d)$ (for fixed d) pivots to reach optimality.

Categories and Subject Descriptors: F.2 [Analysis of Algorithms and Problem Complexity]: Miscellaneous; G.1.6 [Numerical Analysis]: Optimization–Linear Programming

General Terms: Thoery

Keywords: Linear Programming, Polyhedra, Deformed Products, Pivot Rules, Long Paths

1 Introduction

Despite an amplitude of research on convex polytopes, one of the most famous problems remains unsolved: Does there exist a *strongly polynomial* algorithm that solves linear programming? Most refinements (by pivot rules) for the *simplex method* have been shown to be exponential in the worst-case, and examples have been constructed (see [1] and [5]) where the maximal number of iterations that a simplex method can take (assymptotically) matches the the maximal number of vertices, $M(d,n)$, that a d-polytope with n facets can have. The lack of success with simplex methods, in this respect, lead researchers to study *criss-cross methods* which leave the boundary of the polytope and traverse the edges of the underlying oriented hyperplane arrangement using *admissible pivots*. In 1999, Fukuda and Terlaky [4] proved the existence of a short (linear in n and d) admissible pivot path between any two vertices of a polytope, motivating the research community with the prospect of finding a polynomial criss-cross method. But how long can an admissible pivot path be? This question was posed by D. Avis (private communication, 2002): What is the maximal number of vertices, $C(d,n)$, along a path taken by the *least-index criss-cross method* to solve a linear program in dimension d with n inequality constraints? Clearly

$$C(d,n) \leq \binom{n}{d} \leq n^d \quad \text{for } n \geq d \tag{1}$$

since the maximal number of vertices of an arrangement is $\binom{n}{d}$. In 1978, even before the birth of the least-index criss-cross method, Avis and Chvátal [2] unknowingly proved its exponential worst-case behaviour by exhibiting an example where the number of pivots taken by the simplex method with Bland's rule on a completely degenerate polytope is bounded from below by the d^{th} fibonacci number which is of the order $(1.618\cdots)^d$.

$$C(d,n) \geq \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^d - \left(\frac{1-\sqrt{5}}{2}\right)^d\right) \quad \text{for } n \geq 2d. \tag{2}$$

The result follows from the observation that the least-index criss-cross method and the simplex method with Bland's rule follow the same pivot path on a completely degenerate polytope. In 1990, Roos [8] constructed an example where the least-index criss-cross method follows the boundary of a nondegenerate polytope and requires an exponential number of pivots[1]. Until now, Roos' result provided the best known lower bound,

$$C(d,n) \geq 2^d \quad \text{for } n \geq 2d, \tag{3}$$

which, asymptotically, leaves a significant gap with the upper bound,

$$C(d,n) \text{ is } \Omega(2^d) \text{ and } O(n^d) \quad \text{for } n \geq 2d \text{ where } d \text{ is fixed.} \tag{4}$$

In fact, it remained unclear whether the criss-cross method could take a path of length longer than $M(d,n)$ which is $\Theta(n^{\lfloor \frac{d}{2} \rfloor})$ for fixed d by the Upper Bound Theorem (McMullen, [6]).

$$C(d,n) \overset{?}{\geq} M(d,n) \tag{5}$$

*Also affiliated with Institute of Mathematics / ROSO, EPFL, Lausanne, Switzerland.

[1] Both the Avis-Chvátal and Roos constructions are variants of the *Klee-Minty* examples [5].

In the present paper we show how to construct a family of examples which not only answer this question affirmatively, but also show that $C(d,n)$ is $\Omega(n^d)$ for fixed d, implying that criss-cross methods can visit nearly every vertex of the arrangement, and thus can perform even worse than simplex methods.

Theorem 1 (Main Theorem). *For fixed d, the function $C(d,n)$ grows like a polynomial of degree d in n:*

$$C(d,n) \text{ is } \Theta(n^d) \text{ for } n \geq 2d. \tag{6}$$

Our construction uses the powerful tool of a *deformed product of arrangements*, an extension of a *deformed product of polytopes* as defined recently by Amenta and Ziegler [1]. The reader familiar with polytopes, arrangements, and linear programming can breeze through section 2. Section 3 examines the behaviour of the least-index criss-cross method on the deformed product of polytopes, and by section 4 we are ready to construct a family of worst-case examples.

2 Preliminaries

2.1 Polyhedra

A *d-polyhedron* P is the intersection of n closed halfspaces in $\Re^d$, $P = \{x \in \Re^d : a_i^t x \leq \alpha_i \quad \text{for } 1 \leq i \leq n\}$ where $a_i \in \Re^d$, and $\alpha_i \in \Re$, or equivalently, the convex hull of a finite set of points in $\Re^d$, $P = \text{conv}\{p_1, ..., p_m\}$. The *faces* of P are all the subsets of the form $F = \{x \in P : a^t x = \alpha\}$ for some $a \in \Re^d$, and $\alpha \in \Re$, where $a^t x \leq \alpha$ is a *valid* inequality for P; meaning that $a^t x \leq \alpha$ is satisfied for all $x \in P$. The faces of P are themselves polyhedra and the faces of dimensions $0, 1, d-2, d-1$, and k are called *vertices*, *edges*, *ridges*, *facets,* and *k-faces* of P. A bounded polyhedron is called a *polytope* and a d-polytope is *simple* if every vertex lies on exactly d facets.

Definition 1 (Combinatorially Equivalent Polytopes). *Two polytopes P and Q are combinatorially equivalent if there is a bijection between their vertices, vert$(P) = \{p_1, ..., p_m\}$ and vert$(Q) = \{q_1, ..., q_m\}$, such that for any subset $I \subseteq \{1, ..., m\}$, the convex hull conv$\{p_i : i \in I\}$ is a face of P if and only if conv$\{q_i : i \in I\}$ is a face of Q.*

Definition 2 (Normally Equivalent Polytopes). *Two polytopes P and Q are normally equivalent if they are combinatorially equivalent and each facet conv$\{p_i : i \in I\}$ of P is parallel to the corresponding facet conv$\{q_i : i \in I\}$ of Q.*

2.2 Hyperplane Arrangements

A *hyperplane* is a set $h = \{x \in \Re^d : a^t x = \alpha\}$ for some nonzero $a \in \Re^d$, and $\alpha \in \Re$. A finite set of hyperplanes H in $\Re^d$ induces a decomposition of $\Re^d$ into connected (relatively) open cells called an *arrangement* A_H. The $0, 1, 2, (d-1)$, and k-dimensional cells of A_H are termed vertices, edges, faces, facets, and k-cells. Two vertices of A_H are *adjacent* if they share $d-1$ hyperplanes, in other words they share an edge. Given a hyperplane arrangement $A_H \subseteq \Re^d$ and a linear functional $\varphi : \Re^d \to \Re$ such that $v_{\min}$ and $v_{\max}$ are the vertices of A_H that minimize resp. maximize φ with $0 \leq \varphi(v_{\min}) \leq \varphi(v_{\max}) \leq 1$, then we write $\varphi(vert(A_H)) \subseteq [0,1]$. We will assume that the hyperplanes are labeled as $H = \{h_1, h_2, \ldots, h_n\}$, and that hyperplanes are *oriented*: each has a positive side and negative side that are given by $\{x \in \Re^d : a^t x > \alpha\}$ and $\{x \in \Re^d : a^t x < \alpha\}$. An arrangement of oriented hyperplanes is an example of an *oriented matroid*, and shares all of its features (see [7]). In particular, each cell of the arrangement is represented as a *signed vector* in $\{+, 0, -\}^n$ indicating the position of the cell with respect to the hyperplanes of H. For a study on the combinatorial structure of arrangements, see [11] and [7].

Proposition 1. *A polytope $P \subseteq \Re^d$ induces an arrangement of oriented hyperplanes A_P.*

Definition 3 (Combinatorially Equivalent Arrangements). *Two arrangements are combinatorially equivalent if the two sets of sign vectors of cells of the arrangements are exactly the same (i.e. the underlying oriented matroids are exactly the same).*

Definition 4 (Normally Equivalent Arrangements). *Two hyperplane arrangements are said to be normally equivalent if they are combinatorially equivalent and the corresponding unit hyperplane normals coincide.*

2.3 Linear Programming

Linear Programming is the problem of maximizing a linear functional φ over a polyhedron $P \subseteq \Re^d$:

$$\max \varphi(x) : x \in P. \tag{7}$$

For the remainder of our discussion, we will assume that the linear program is *feasible* (P is non-empty), P is simple, and that φ is bounded by P.

Starting at a vertex defined by a set H of d intersecting hyperplanes, a *pivot* is the operation of exchanging a hyperplane $h_i \in H$ for a hyperplane $h_j \notin H$ that intersects the edge $H \backslash \{h_i\}$. This results in a second vertex defined by $H' := H \backslash \{h_i\} \cup \{h_j\}$ (henceforth abbreviated by $H - h_i + h_j$). *Pivot methods* attempt to solve a linear program, (P, φ), by moving along the edges of A_P from vertex to adjacent vertex, and *pivot rules* determine which adjacent vertex to travel to. The sequence of vertices visited is called the *pivot path*, and when a pivot rule guarantees convergence to optimality then we say it is *finite*. *Simplex (pivot) methods* try to solve a linear program by pivoting along the boundary edges of P from vertex $p_i \in P$ to vertex $p_j \in P$ such that $\varphi(p_i) < \varphi(p_j)$.

Definition 5 (Increasing Edges). *For a linear functional $\varphi : \Re^d \to \Re$, an edge $[v', v'']$ is φ-increasing if $\varphi(v'') > \varphi(v')$.*

Definition 6 (Increasing Rays). *Given a start point $s \in \Re^d$ and a vector $\overrightarrow{u} \in \Re^d$, a ray $r = (s, \overrightarrow{u})$ is the set of all points of the form $s + \lambda \overrightarrow{u}$ for all scalar $\lambda \geq 0$. For a linear functional $\varphi : \Re^d \to \Re$, a ray is φ-increasing if and only if $\varphi(\overrightarrow{u}) > 0$.*

Definition 7 (Primal Infeasible Vertex). *A vertex p_i of the arrangement induced by a linear program (P, φ) is primal infeasible if p_i is not a vertex of the polytope P. In other words, p_i violates at least one inequality of P.*

Definition 8 (Dual Infeasible Vertex). *A vertex p_i, defined by a set H of d intersecting hyperplanes of the arrangement induced by a linear program (P, φ), is dual infeasible if there is at least one ray starting at p_i that is φ-increasing and satisfies all $h \in H$ (every point on the ray lies on the nonnegative side of all hyperplanes $h \in H$).*

The *optimal* vertex which maximizes φ is a vertex that is both primal and dual feasible.

3 Criss-Cross Methods

Criss-cross methods are pivot methods for solving a linear program (P, φ) whose pivot path can leave the boundary of P. The first criss-cross method was baptized by Zoints [12], and the first finite criss-cross method, the *least-index criss-cross method*, was discovered independently by Terlaky [9], and Wang [10].

3.1 The Least-Index Criss-Cross Method

As the name suggests, criss-cross methods have two types of pivots (with respect to an objective function φ): *admissible type I* pivots and *admissible type II* pivots.

Definition 9 (Admissible Type I Pivot). *For every primal infeasible vertex p defined by a set H of d intersecting hyperplanes, there exists an oriented hyperplane $h_j \notin H$ that is violated at p. A pivot from p to vertex p', defined by $H' := H - h_i + h_j$, is an admissible type I pivot if p' lies on the nonnegative side of h_i.*

If h_j is selected such that j is minimized, followed by selecting h_i to minimize i, then the pivot is a *least-index admissible type I* pivot.

Definition 10 (Admissible Type II Pivot). *For every dual infeasible vertex p defined by a set H of d intersecting hyperplanes, there exists a φ-increasing ray $(p, \overrightarrow{u})$ that lies on an edge $H \backslash \{h_i\}$ that is on the nonnegative side of $h_i \in H$. A pivot from p to vertex p', defined by $H' := H - h_i + h_j$, is an admissible type II pivot if there exists a point on $(p, \overrightarrow{u})$ that lies on the nonpositive side of h_j.*

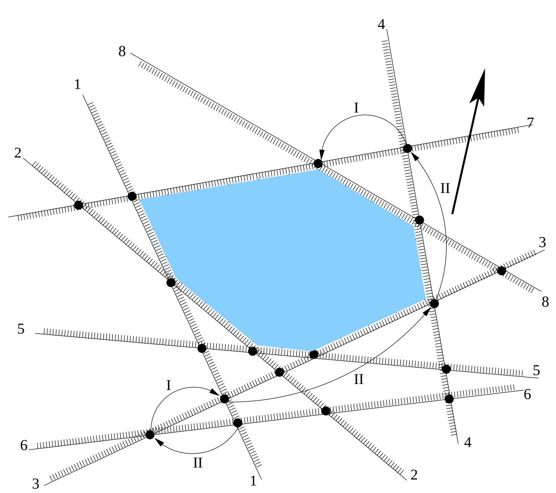

Figure 3.1

If h_i is selected such that i is minimized, followed by selecting h_j to minimize j, then the pivot is a *least-index admissible type II* pivot.

We will denote a pivot, exchanging h_i for h_j, by $pivot(i, j)$. Using these notions, we provide the geometric interpretation of the least-index criss-cross method. The reader can follow the algorithm on the example shown in Figure 3.1.

Algorithm 1 (The Least-Index Criss-Cross Method). *Given a linear program $(P, \varphi) \subseteq \Re^d$, a linear ordering of the inequalities of P, and a vertex p of A_P:*

Criss-Cross:

If p is optimal (both primal feasible and dual feasible) then ***Stop;***

If p is primal feasible then let $f := +\infty$. Otherwise let $f := j$ such that $pivot(i, j)$ is the least-index admissible type I pivot from p to p'. If no pivot exists then the linear program is primal inconsistent, ***Stop;***

If p is dual feasible then let $g := +\infty$. Otherwise let $g := i'$ such that $pivot(i', j')$ is the least-index admissible type II pivot from p to p''. If no pivot exists then the linear program is dual inconsistent, ***Stop;***

If $f < g$, then $\overline{p} := p'$. Otherwise $\overline{p} := p''$.

Pivot from p to $\overline{p}$, let $p := \overline{p}$ and go to ***Criss-Cross;***

The least-index criss-cross method is finite and solves a linear program (see [3] for simple proofs). From this point forward the criss-cross method will refer to the least-index criss-cross method.

3.2 Deformed Product of Arrangements

Recently Amenta and Ziegler [1] presented a construction of deformed products of polytopes that generalized all known constructions of linear programs with "many simplex pivots."

Definition 11 (Deformed Products of Polytopes). *Let $P \subseteq \Re^d$ be a convex polytope, and $\varphi : P \to \Re$ a linear functional with $\varphi(P) \subseteq [0, 1]$. Let $V, W \subseteq \Re^e$ be convex polytopes. Then the deformed product of (P, φ) and of (V, W) is*

$$(P, \varphi) \bowtie (V, W) := \left\{ \begin{pmatrix} x \\ v + \varphi(x)(w - v) \end{pmatrix} : \quad \begin{matrix} x \in P \\ v \in V, w \in W \end{matrix} \right\} \tag{8}$$

$$\subseteq \Re^{d+e}.$$

Definition 12 (Deformed Product Programs). *Define the deformed product program as*

$$\max \widehat{\alpha} \begin{pmatrix} x \\ u \end{pmatrix} = \alpha(u) : \quad \begin{pmatrix} x \\ u \end{pmatrix} \in Q = (P, \varphi) \bowtie (V, W). \tag{9}$$

The resulting linear program is the deformed product polytope Q with objective function $\max \alpha(u)$.

Our goal is to construct a family of deformed product programs on which the criss-cross method visits almost all vertices of the arrangement. We begin by analyzing the behaviour of the criss-cross method on the arrangement of hyperplanes of a deformed product program. Hence, we define the *deformed product of arrangements* to be the induced hyperplane arrangement of a deformed product of polytopes.

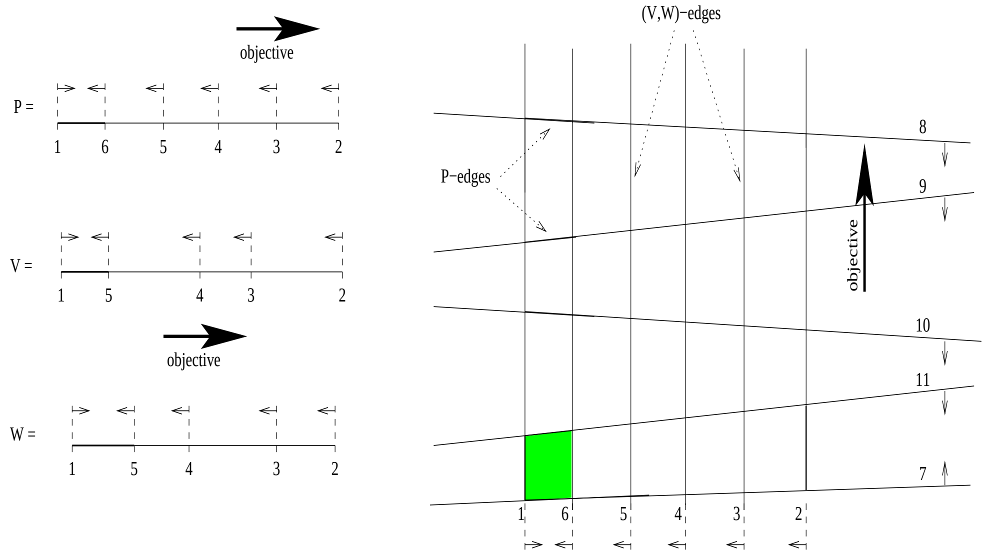

Figure 3.2

We extend some of the facts about deformed products that are proved in [1] to express the properties of the induced hyperplane arrangement of Q. We facilitate understanding by providing an example (see Figure 3.2) on which the reader is encouraged to verify the theorem's statements.

THEOREM 2. *Let $P \subseteq \Re^d$ be a d-polyhedron, A_P be the underlying hyperplane arrangement of P, $\varphi : \Re^d \to \Re$ a linear function such that $\varphi(vert(A_P)) \subseteq [0,1]$, and $V, W \subseteq \Re^e$ be normally equivalent e-polyhedra inducing normally equivalent hyperplane arrangements A_V and A_W, then:*

- *If A_P has m vertices and s hyperplanes, and if A_V and A_W have n vertices and t hyperplanes each, then $Q := (P, \varphi) \bowtie (V, W)$ is a $(d+e)$-polytope whose underlying arrangement A_Q has at least $m \cdot n$ vertices and exactly $s + t$ hyperplanes.*

- *Specifically, if $\{p_1, ..., p_m\}$ are vertices of A_P, $\{v_1, ..., v_n\}$ and $\{w_1, ..., w_n\}$ the vertices of A_V resp. A_W, then we can define $m \cdot n$ of the vertices of Q, denoted $\gamma(i, j)$, as:*

$$\gamma(i,j) = \begin{pmatrix} p_i \\ v_j + \varphi(p_i)(w_j - v_j) \end{pmatrix} : \quad \begin{array}{l} 1 \le i \le m \\ 1 \le j \le n. \end{array} \tag{10}$$

- *If A_P, A_V, and A_W are given by*

$$\begin{aligned} A_P &= \{a_k x \le \alpha_k \text{ for } 1 \le k \le s\}, \\ A_V &= \{b_l u \le \beta_l \text{ for } 1 \le l \le t\}, \text{ and} \\ A_W &= \{b_l u \le \beta'_l \text{ for } 1 \le l \le t\}, \end{aligned} \tag{11}$$

  then the arrangement of hyperplanes of the deformed product Q is given by

$$A_Q = \left\{ \begin{array}{c} a_k x \le \alpha_k \text{ for } 1 \le k \le s \\ (\beta_l - \beta'_l)\varphi(x) + b_l u \le \beta_l \text{ for } 1 \le l \le t \end{array} \right\}. \tag{12}$$

- *A cell C_Q^i of A_Q is the deformed product of some cell C_P^j with (C_V^k, C_W^k): $C_Q^i = (C_P^j, \varphi) \bowtie (C_V^k, C_W^k)$. C_Q^i is convex if $0 \le \varphi(y) \le 1$ for $y \in C_Q^i$.*

Let's examine the edges of a hyperplane arrangement underlying a deformed product program with the assumptions of Theorem 2. Specifically, we are interested in these edges of A_Q:

- *A_P-edges* of the form $[\gamma(i', j), \gamma(i'', j)]$ for $1 \le j \le n$ and for any $1 \le i', i'' \le m$ such that $p_{i'}$ and $p_{i''}$ are adjacent vertices of A_P and

- *$A_{(V,W)}$-edges* of the form $[\gamma(i, j'), \gamma(i, j'')]$ for $1 \le i \le m$ and for any $1 \le j', j'' \le n$ such that $v_{j'}$ and $v_{j''}$ are adjacent vertices of A_V (equivalently, $w_{j'}$ and $w_{j''}$ are adjacent vertices of A_W).

PROPOSITION 2. *Given a deformed product program (9), an A_P-edge $[\gamma(i', j), \gamma(i'', j)]$ is $\widehat{\alpha}$-increasing if and only if either $[p_{i'}, p_{i''}]$ is φ-increasing and $\alpha(w_j) > \alpha(v_j)$, or $[p', p'']$ is φ-decreasing and $\alpha(w_j) < \alpha(v_j)$.*

PROPOSITION 3. *A (V, W)-edge $[\gamma(i, j''), \gamma(i, j'')]$ is $\widehat{\alpha}$-increasing if and only if $[v_{j'}, v_{j''}]$ is α-increasing.*

The proofs of the preceeding statements follow naturally from the proofs given in [1] for deformed products of polytopes. We are now ready to analyze the behaviour of the criss-cross method on deformed product programs:

COROLLARY 1. *Let p_1 and p_l be the vertices of P that minimize respectively maximize φ with $0 \le \varphi(p_1) \le \varphi(p_l) \le 1$. Construct the deformed product program Q, as defined in (9). If we number the inequalities of Q such that the inequalities of P get smaller indices than the inequalities corresponding to (V, W), then the criss-cross method prefers to pivot along A_P-edges rather than $A_{(V,W)}$-edges.*

The result is that if the criss-cross method on (P, φ) for the objective function φ takes a path of length l from the p_1 to p_l, and for $-\varphi$ takes a path of length l' from p_l to p_1, then for $(Q, \widehat{\alpha})$ the criss-cross method will follow a path of length l from $\gamma(1, j)$ to $\gamma(l, j)$ if $\alpha(v_j) < \alpha(w_j)$, and a path of length l' from $\gamma(l, j)$ to $\gamma(1, j)$ if $\alpha(v_j) > \alpha(w_j)$.

4 The Construction

We construct the worst-case example by first building low dimensional examples where the criss-cross method takes many pivots. We then show how to take deformed products of these base cases to construct polyhedra in any dimension where the criss-cross method behaves badly.

Definition 13. *Let $C(d,n)$ be the maximal number of vertices along a path taken by the least-index criss-cross method for some linear objective function φ on a d-dimensional polyhedron with at most n facets.*

Definition 14. *Starting at the vertex of P that minimizes φ, let $H(d,n)$ be the maximal number of vertices visited along a path taken by the least-index criss-cross method for some linear objective function max φ on the arrangement of hyperplanes induced by a d-dimensional polyhedron with at most n facets.*

Clearly $C(d,n) \geq H(d,n)$.

Lemma 1. *For $n \geq 2$, $H(1,n) = n$*

Proof. Consider the following linear program defined by $max\ \varphi = x$ and the polytope given by the inequalities (indexed in order of appearance): $x \geq 0$ and $x \leq (n-i+1)\lambda$ for $2 \leq i \leq n$ and for some constant $\lambda > 0$. For example, for $n = 6$:

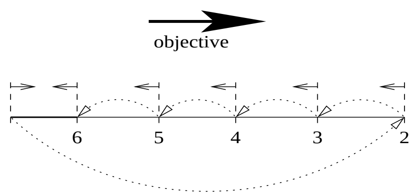

Figure 4a

Let v_i be the vertex defined by hyperplane h_i for $1 \leq i \leq n$. The criss-cross method takes a path of length n from vertex v_1 to vertex v_n when $\varphi = x$. The criss-cross method takes a path of length one from vertex v_n to vertex v_1 when $\varphi = -x$. □

Lemma 2. *There exists a pair of normally equivalent 1-polytopes, (V,W), defined by k inequalities each (hence k vertices), and a linear functional α, such that $\alpha(v_i) > \alpha(w_i)$ if i is even and $\alpha(v_i) < \alpha(w_i)$ if i is odd.*

Proof. Construct V as in Lemma 2. To build W, for each hyperplane h_i of V, construct h'_i of W by translating h_i in the positive x-direction by (some suitably small) $\epsilon > 0$ if i is odd and by $-\epsilon$ if i is even. The case when $k = 5$ in is illustrated in Figure 3.2. □

Note that $\alpha(v_k) < \alpha(w_k)$ when k is odd and $\alpha(v_k) > \alpha(w_k)$ if k is even. The following example illustrates the construction of a deformed product and the path that the criss-cross method takes on the underlying arrangement.

Example 1. *(See Figure 4b) Construct P (6 inequalities, variable x_1, $\lambda = 0.1$) and V (5 inequalities, variable x_2) as in Lemma 1, and W (5 inequalities, variable x_2) as in Lemma 2. Let $Q = (P,x) \bowtie (V,W)$ and order the inequalities of Q so that the inequalities coming from P are indexed smaller than those from (V,W). Consider the path that the criss-cross method takes on the deformed product program $(Q, \alpha = x_2)$.*

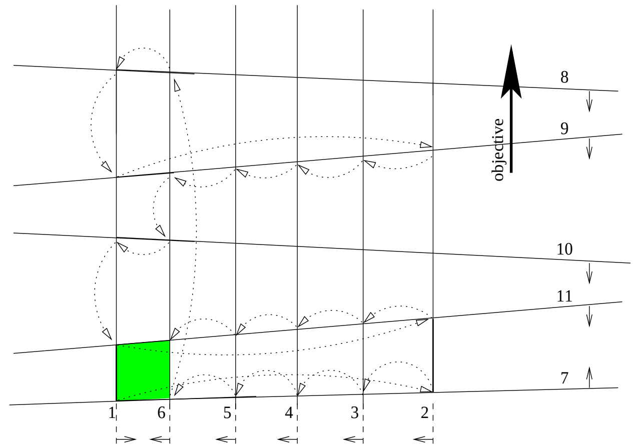

Figure 4b

Theorem 3. *For $k \geq 2$ and $n > d \geq 0$,*

$$H(d+1, n+k) \geq \left\lceil \frac{k}{2} \right\rceil \cdot H(d,n) \tag{13}$$

Proof. Take a polytope $P \subseteq \Re^d$ with n inequalities for which the least-index criss-cross method for a linear functional $\varphi(x)$ (rescaled such that $\varphi(vert(A_P)) \subseteq [0,1]$) follows a criss-cross method path of length $l = H(d,n)$ starting at vertex p_1 and ending at vertex p_l. Now construct the deformed product program, as in (9), where V, W and α are defined according to Lemma 2. By Corollary 1 we get that the criss-cross method applied to (Q,α) first follows a P-path with l vertices from $\gamma(1,1)$ to $\gamma(l,1)$, then after one (V,W)-pivot it follows a P-path of length one from $\gamma(l,2)$ to $\gamma(1,2)$, then after one (V,W)-pivot it follows a P-path with l vertices from $\gamma(1,3)$ to $\gamma(l,3)$, then after one (V,W)-pivot it follows a P-path of length one from $\gamma(l,4)$ to $\gamma(1,4)$, etc... The complete path will visit $\lceil \frac{k}{2} \rceil l + \lfloor \frac{k}{2} \rfloor$ vertices arriving at $\gamma(1,k)$ or $\gamma(l,k)$, depending on whether k is even or odd. □

Remark 1. *We could use this result to construct examples, by induction, where $C(d,n)$ is $\Omega(n^d)$ asymptotically for fixed d. However we choose to postpone this analysis since iterative deformed products with the 1-dimensional construction would contain a large number of redundant constraints, in fact $n - 2d$ of them.*

Lemma 3. *For $n \geq 3$, $H(2,n)$ is $\Omega(n^2)$.*

Proof. Consider the following construction: let the i^{th} inequality of P be defined as

$$-(i-1)x_1 - (n-i)x_2 \leq -2(i-1)(n-i). \tag{14}$$

This construction ensures that the x_1 intercept of the i^{th} inequality is greater than the x_1 intercept of the $(i-1)^{th}$ while the x_2 intecept of the i^{th} inequality is less than that of the $(i-1)^{th}$ (see Figure 4c for an example with $n = 7$). The least-index criss-cross method on the linear program *min* $\alpha = x_2$, subject to $\binom{x_1}{x_2} \in P$, starting at the vertex defined by the intersection of hyperplanes 1 and n (which we denote $(1,n)$), will take $n-1$ *type I* pivots $(1,n) \to (2,n) \to \cdots \to (n-1,n)$, and then from $(n-1,n)$ take one *type II* pivot to $(1,n-1)$, and then take $n-2$ *type I* pivots to $(n-2,n-1)$, and then one *type II* pivot to $(1,n-2)$, and then take $n-3$

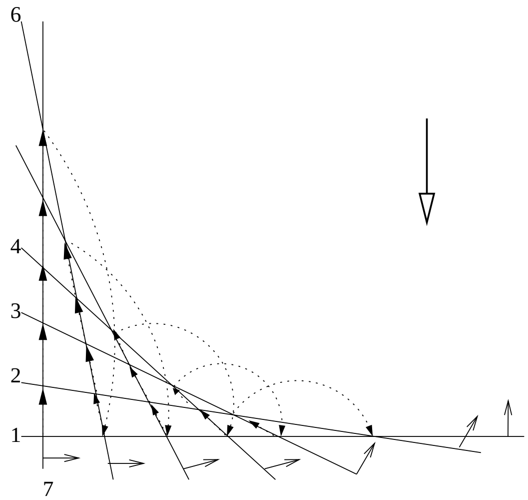

Figure 4c

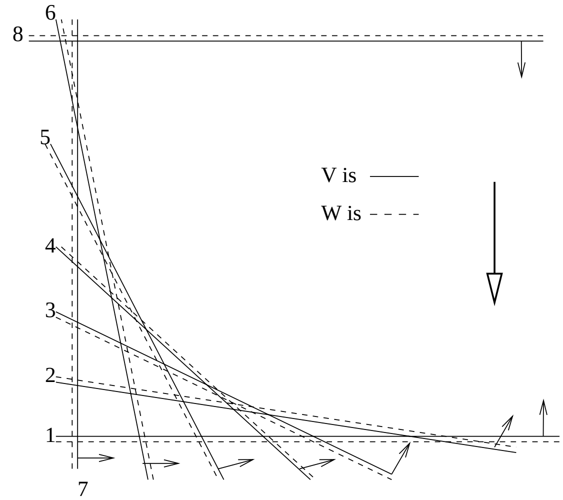

Figure 4d

type I pivots to $(1, n-3)$, etc... and ending with one *type II* pivot from $(2,3)$ to $(1,2)$, visiting a total of $\frac{n(n-1)}{2} = \binom{n}{2}$ vertices. □

REMARK 2. *There are $n-2$ type II pivots, and $\frac{(n-1)(n-2)}{2}$ type I pivots. For every type I pivot from intersection (i,j) to (g,j), if i is odd then g is even, and if i is even then g is odd. Every type II pivot has the form (i,j) to $(1,i)$ where $j = i+1$.*

LEMMA 4. *There exist normally equivalent 2-dimensional polyhedra V and W with k facets ($k \geq 4$), and objective function min α for which the criss-cross method takes $\Theta(k^2)$ pivots such that corresponding vertices v of V and w of W, defined by the intersection of hyperplanes h_i and h_j for $i < j$, have the following property: $\alpha(v) > \alpha(w)$ when i is odd, and $\alpha(v) < \alpha(w)$ when i is even.*

PROOF. Let V be a 2-polyhedron with $k-1$ inequalities as defined in (14) and define the k^{th} inequality as

$$0x_1 + x_2 \leq 2(k-2).$$

Set $\alpha = x_2$. The k^{th} inequality ensures that V bounds both α and $-\alpha$, and that the x_2 intercept of the k^{th} inequality is greater than that of the $(k-2)^{th}$. Build W as follows: for $1 \leq i \leq k-1$ take the i^{th} inequality of V, $b_i x \leq \beta$, and define the i^{th} inequality of W to be $b_i x \leq \beta'$ where $\beta' = \beta - \epsilon$ if i is odd and $\beta' = \beta + \epsilon$ if i is even (see Figure 4d). ϵ is chosen to be positive and suitably small. Let the k^{th} inequality of W be $b_k x \leq \beta'$ where $\beta' = \beta + \epsilon$. Now lets examine corresponding vertices v of V and w of W defined by the intersection of hyperplanes h_i and h_j for $i < j$. Using basic trigonometry we can prove that if i is odd then $\alpha(v) > \alpha(w)$, and otherwise if i is even then $\alpha(v) < \alpha(w)$ (see Appendix). □

REMARK 3. *Starting at $(n-2, n)$ the criss-cross method on V (or W) will take one type II pivot to $(1,n)$ and then follow the path described in Lemma 3. There are $k-2$ type II pivots, and $\frac{(k-2)(k-3)}{2}$ type I pivots (see Remark 2 setting $n = k-1$ and adding one additional type II pivot from $(k-2,k)$ to $(1,k)$).*

DEFINITION 15 (SWITCH PIVOT). *Given two normally equivalent polyhedra V and W, and a linear objective function α we define a switch pivot to be a pivot from v_i to v_j (w_i to w_j) such that if $\alpha(v_i) > \alpha(w_i)$ then $\alpha(v_j) < \alpha(w_j)$, otherwise if $\alpha(v_i) < \alpha(w_i)$ then $\alpha(v_j) > \alpha(w_j)$.*

LEMMA 5. *Let (V, W) be as defined in Lemma 3. Starting at the intersection of h_{k-1} and h_k the least-index criss cross method takes a $\Theta(k^2)$ path to the intersection of h_1 and h_2 on which there are $\Theta(k^2)$ switch pivots.*

PROOF. Every *type I* pivot is a switch pivot, and every second *type II* pivot is a switch pivot,

$$\#\text{ of switch pivots} \geq \frac{k^2}{K} = \Theta(k^2) \tag{15}$$

for some constant $K > 1$ and all $k \geq 3$. □

THEOREM 4. *For $k \geq 3$, $n > d \geq 0$, and some constant $K > 1$,*

$$H(d+2, n+k) \geq \frac{k^2}{2K} \cdot H(d,n). \tag{16}$$

PROOF. Take a polytope $P \subseteq \Re^d$ with n inequalities for which the least-index criss-cross method for a linear functional $\varphi(x)$ (rescaled such that $\varphi(vert(A_P)) \subseteq [0,1]$) follows a criss-cross method path of length $l = H(d,n)$ starting at vertex p_1 and ending at vertex p_l. Let l' be the length of the criss-cross method path from p_l to p_1 for $-\varphi$. Now construct the deformed product program, as in (9), where V, W and α are defined according to Lemma 4. Let v_{opt} be the optimal vertex of (V,W). By Corollary 1, we get that the criss-cross method applied to (Q,α) first follows a P-path with l vertices from $\gamma(1,1)$ to $\gamma(l,1)$, and then after a (V,W)-*switch pivot* it follows a P-path of length l', and then after a (V,W)-*switch pivot* it follows a P-path of length l, and then after a (V,W)-*switch pivot* it follows a P-path of length l', etc... The complete path will visit at least $\frac{1}{2}\frac{k^2}{K}l + \frac{1}{2}\frac{k^2}{K}l'$ vertices ending at (l, opt). □

COROLLARY 2. *For $n \geq 2d \geq 2$, $C(d,n) = \Omega((\frac{n}{d})^d)$. More specifically, for some constant $K > 1$:*

$$C(d,n) \geq \left\lfloor \frac{2n}{d\sqrt{2K}} \right\rfloor^d \quad \textit{if } d \textit{ is even}, \tag{17}$$

and

$$C(d,n) \geq \left\lfloor \frac{2n}{(d+1)\sqrt{2K}} \right\rfloor^d \qquad \textit{if } d \textit{ is odd.} \tag{18}$$

PROOF. *(By induction)* Lets begin with the even case, when $d = 2m$ for all $m \geq 0$, let $n = km$:

$$\begin{aligned} H(2m, km) &\geq \frac{k^2}{2K} H(2(m-1), k(m-1)) \\ &\vdots \quad m \text{ times} \\ &\geq \left(\frac{k^2}{2K}\right)^{m-1} H(2,k) \\ &= \frac{k^{2m}}{(2K)^m} \qquad \text{by Lemma 3.} \end{aligned}$$

Substituting for $m = \frac{d}{2}$ and $k = \lfloor \frac{2n}{d} \rfloor$, we get

$$H(d,n) \geq \left\lfloor \frac{2n}{d\sqrt{2K}} \right\rfloor^d. \tag{19}$$

For the odd case, when $d = 2m+1$ for all $m \geq 0$, let $n = k(m+1)$:

$$\begin{aligned} H(2m+1, k(m+1)) &\geq \frac{k^2}{2K} H(2(m-1)+1, km) \\ &\vdots \quad m \text{ times} \\ &\geq \left(\frac{k^2}{2K}\right)^{m} H(1,k) \\ &= \frac{k^{2m+1}}{(2K)^m} \qquad \text{by Lemma 1.} \end{aligned}$$

Substituting for $m = \frac{d-1}{2}$ and $k = \left\lfloor \frac{2n}{d+1} \right\rfloor$, we get

$$H(d,n) \geq \left\lfloor \frac{2n}{(d+1)\sqrt{2K}} \right\rfloor^d. \tag{20}$$

The condition $n \geq 2d$ guarantees $k \geq 4$. $\square$

REMARK 4. *The construction has no redundant constraints when d is even, and $\left\lfloor \frac{2n}{d+1} \right\rfloor - 2$ redundant constraints when d is odd.*

COROLLARY 3 (MAIN THEOREM). *For fixed dimension d, the function $C(d,n)$ grows like a polynomial of degree d in n:*

$$C(d,n) \textit{ is } \Theta(n^d) \textit{ for } n \geq 2d \textit{ where } d \textit{ is fixed.} \tag{21}$$

5 Conclusion

Using a construction of deformed product programs, we proved that the worst-case path length that the least-index criss-cross method for solving a linear program can take is $\Omega(n^d)$ for a d-polyhedron defined by n halfspaces (when d is fixed). This result provides a tighter lower bound that assymptotically achieves the upperbound, and also shows that the least-index criss-cross method is worse than simplex methods in the worst case. Despite this negative result, criss-cross methods remain perhaps the best hope of finding a strongly polynomial algorithm for linear programming (see [4]).

References

[1] N. Amenta and G. Ziegler. Deformed products and maximal shadows of polytopes. *Contemporary Mathematics*, 223:57–90, 1999.

[2] D. Avis and V. Chvátal, Notes on Bland's rule. *Mathematical Programming Study*, 8:24–34, 1978.

[3] K. Fukuda and T. Matsui. On the finiteness of the criss–cross method. *European Journal of Operations Research*, 52:119–124, 1991.

[4] K. Fukuda and T. Terlaky. On the existence of a short admissible pivot sequence for feasibility and linear optimization problems. *Pure Mathematics and Applications, Mathematics of Optimization*, 10(4):431–447, 2000.

[5] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities–III*, pages 159–175. Academic Press, New York, 1972.

[6] P. McMullen. The maximum number of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.

[7] J. Richter-Gebert and G. Ziegler. Oriented matroids. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 111–132. CRC Press, New York, 1997.

[8] C. Roos. An exponential example for Terlaky's pivoting rule for the criss-cross simplex method. *Mathematical Programming*, 46:78–94, 1990.

[9] T. Terlaky. A convergent criss-cross method. *Math. Oper. und Stat. ser. Optimization*, 16(5):683–690, 1985.

[10] Z. Wang. A conformal elimination free algorithm for oriented matroid programming. *Chinese Annals of Mathematics*, 8(B1), 1987.

[11] T. Zaslavsky. Facing up to arrangements: face-count formulas for partitions of space by hyperplanes. *Mem. Amer. Math. Soc.*, 1(No 154 MR 50), 1975.

[12] S. Zionts. The criss-cross method for solving linear programming problems. *Management Science*, 15(7):426–445, 1969.

APPENDIX

Here we complete the proof of Lemma 4, namely if i is odd then $\alpha(v) > \alpha(w)$, and otherwise if i is even then $\alpha(v) < \alpha(w)$:

Case 1: i is odd and j odd. This case is illustrated in Figure 5a. n_i and n_j represent the normals of h_i and h_j respectively, or if you wish the direction of translation by ϵ: $|n_i| = |n_j| = \epsilon$. Let $\delta = |d|, \theta_1 = angle(A)$, and $\theta_2 = angle(B)$. By construction, $0° \leq \theta_1 < \theta_2 \leq 90°$, and $\delta = \epsilon \sin\theta_1$ where $\sin\theta_1 \geq 0°$. Now $\alpha(w) < \alpha(v-\delta) \leq \alpha(v)$ which implies $\alpha(v) > \alpha(w)$.

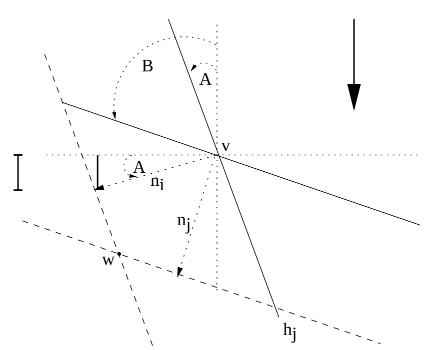

Figure 5a

Case 2: i is even and j even. This case is symmetric to *case 1*, hence $\alpha(v) < \alpha(w)$.

Case 3: i is odd and j even. This case is illustrated in Figure 5b. n_i and n_j represent the normals of h_i and h_j respectively, the direction of translation by ϵ: $|n_i| = |n_j| = \epsilon$. Let $\delta = |d|, \theta_1 = angle(A)$, and $\theta_2 = angle(B)$. By construction, $0^\circ \leq \theta_1 < \theta_2 \leq 90^\circ$, and $\delta = \epsilon \sin(90^\circ - \theta_2)$ where $\sin(90^\circ - \theta_2) > 0$. Now $\alpha(w) \leq \alpha(v - \delta) < \alpha(v)$ which implies $\alpha(v) > \alpha(w)$.

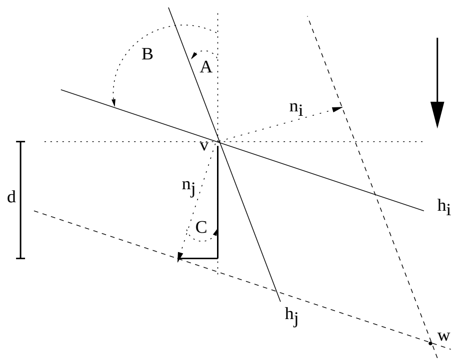

Figure 5b

Case 4: i is even and j odd. This case is symmetric to *case 3*, hence $\alpha(v) < \alpha(w)$.