CSC2411 - Linear Programming and Combinatorial Optimization*
Lecture 3: Geometric Aspects of Linear Programming and an Introduction to the Simplex Algorithm

Notes taken by Adam Tenenbaum

January 31, 2005

Summary: In this class, we discuss some geometrical interpretations of linear programs and a high-level description of the simplex algorithm. We also introduce methods for implementing the two phases of the simplex algorithm. These notes also include the tutorial presented on January 19.

1 Review

In our previous class, we took some important steps toward defining an efficient algorithm for solving linear programs (LP).

- We defined basic feasible solutions (BFS).
- We proved that BFS and extreme points are equivalent.
- We showed that if an optimum exists for a LP, then there is a BFS which is optimum.
  - This gave us an exponential algorithm in which we check all sets of m columns.
  - Candidate solutions arise when the columns are linearly independent and there is a nonnegative combination of the columns that equals b.
  - An easy corollary is that if there is a solution ($P \neq 0$), then there exists a solution which is a BFS.
- Finally, we proved the claim that if $\vec{x}$ is a BFS, then there is a $\vec{c}$ such that

$$\langle \vec{x}, \vec{c} \rangle < \langle \vec{y}, \vec{c} \rangle \quad \forall \vec{y} \in P, \quad \vec{y} \neq \vec{x}$$

* Lecture Notes for a course given by Avner Magen, Dept. of Computer Sciecne, University of Toronto.

– That is, it is possible to pick an objective function that makes the BFS the unique optimal solution.

We began the lecture with Caratheodory's Theorem.

Theorem 1.1 (Caratheodory's Theorem (1907)). *Let $\vec{b}, \vec{v_1}, \ldots, \vec{v_t} \in \mathbb{R}^n$ and*

$$\vec{b} = \sum_{i=1}^{t} x_i \vec{v_i}, \quad \textit{for} \;\; x_i \geq 0, \quad \sum_i x_i = 1.$$

Then there is a set I of size $|I| \leq n+1$ so that

$$\vec{b} = \sum_{i \in I} \alpha_i \vec{v_i} \quad \textit{for} \;\; \alpha_i \geq 0, \quad \sum_i \alpha_i = 1.$$

In other words, if $\vec{b}$ is a convex combination of the $\vec{v}_i$'s, then it can be represented by a convex combination of only $n+1$ $\vec{v}_i$'s.

Proof. To prove this theorem, we write finding the scalars α_i as a linear program:

$$\begin{pmatrix} | & & | \\ \vec{v}_1 & \cdots & \vec{v}_t \\ | & & | \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_t \end{pmatrix} = \vec{b}, \quad x_i \geq 0, \quad \sum_i x_i = 1.$$

We then include the constraint $\sum_i x_i = 1$ by inserting a row:

$$\begin{pmatrix} | & & | \\ \vec{v}_1 & \cdots & \vec{v}_t \\ | & & | \\ 1 & \cdots & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_t \end{pmatrix} = \begin{pmatrix} | \\ \vec{b} \\ | \\ 1 \end{pmatrix}, \quad x_i \geq 0.$$

A BFS for the system must exist since $P \neq 0$; as such, there is a solution $(x_1 \cdots x_t)^T \in P$ with at most $n+1$ non-zero coefficients. We take the positive components of that BFS to be the α_i's. □

Example 1.2 (Graphical example of Caratheodory's Theorem). In Figure 1, we demonstrate a graphical example of Caratheodory's Theorem. Here, the regular pentagon is the convex hull of its five vertices; however, an arbitrary point inside the pentagon can be represented using only three of the vertices. In the example shown, the point represented by a small circle is contained in the shaded triangle formed by only three of the vertices – a subset of the five points forming the convex hull.

2 Geometric Aspects of Linear Programming

Example 2.1. Consider the following LP:

$$\begin{aligned} x_1 + x_2 + x_3 &= 1 \qquad (1) \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

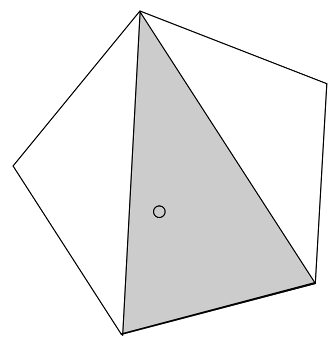

Figure 1: Convex hull containing a point inside a pentagon

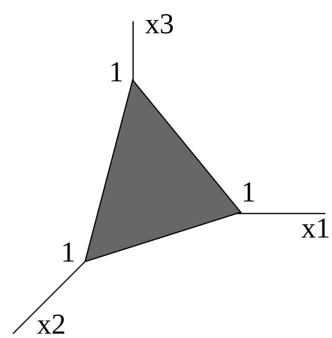

Figure 2: Feasible solutions to Equation (1)

The set of feasible solutions P to Equation (1) is represented by the shaded triangle shown in Figure 2. We can also consider reducing the dimension of the problem by removing the slack variable x_3:

$$\begin{aligned} x_1 + x_2 &\leq 1 \\ x_1, x_2 &\geq 0. \end{aligned} \tag{2}$$

We get (Figure 3) a somewhat similar triangle that is presented in two (rather than three) dimensions.

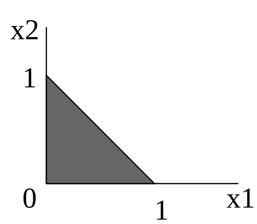

Figure 3: Feasible solutions to Equation (2)

Definition 2.2. A *hyperplane* H is the set

$$\{\vec{x} | \langle \vec{x}, \vec{a} \rangle = \alpha\}\ ,$$

where $\vec{a} \neq \vec{0} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$.

Definition 2.3. A *halfspace* H^+ is the set

$$\{\vec{x} | \langle \vec{x}, \vec{a} \rangle \geq \alpha\}\ ,$$

where $\vec{a} \neq \vec{0} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. Similarly, a halfspace H^- is the set

$$\{\vec{x} | \langle \vec{x}, \vec{a} \rangle \leq \alpha\}\ .$$

Note that the intersection of the two halfspaces H^+ and H^- is the hyperplane H ($H^+ \bigcap H^- = H$).

Observation 2.4. *If we take a LP in standard form, then our set of feasible solutions is*

$$P = \bigcap_{j=1}^{n} \{x_j \geq 0\} \bigcap \left(\bigcap_{i=1}^{m} \{\langle \vec{a_i}, \vec{x} \rangle = b_i\} \right)$$

Thus our set of feasible solutions is expressed as an intersection of halfspaces and hyperplanes. Since we can represent hyperplanes as an intersection of halfspaces, this can be rewritten as a finite intersection of halfspaces:

$$P = \bigcap_{j=1}^{n} \{x_j \geq 0\} \bigcap \left(\bigcap_{i=1}^{m} \{\langle \vec{a_i}, \vec{x} \rangle \geq b_i\} \right) \bigcap \left(\bigcap_{i=1}^{m} \{\langle \vec{a_i}, \vec{x} \rangle \leq b_i\} \right)$$

Since halfspaces are convex, the intersection of halfspaces is convex. Conversely, all convex spaces can be represented as an intersection of halfspaces (but not necessarily a finite intersection).

Definition 2.5. A *polyhedron* is the finite intersection of halfspaces. Furthermore, a polyhedron which is bounded (contained in a big enough cube) is a *polytope*.

Since the set of feasible solutions is always a finite intersection of halfspaces, it always forms polyhedra.

Definition 2.6. H is a *supporting hyperplane* for P if $H \bigcap P \neq \emptyset$ (P is not disjoint from H) and $P \subset H^-$ or $P \subset H^+$ (P is contained in either the upper or lower halfspace).

Definition 2.7. The intersection of a supporting hyperplane H with P is called a *face*.

Definition 2.8. The *dimension* of a set is the dimension of the smallest affine space containing it.

For example, the feasible sets in Figures 2 and 3 are of dimension 2.

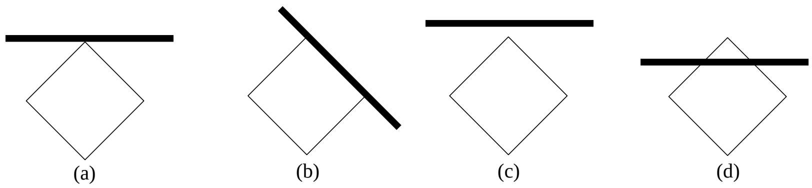

Figure 4: Graphical examples of hyperplanes

Example 2.9. Examples of faces and their dimensions.
A vertex of a polyhedron is a face of dimension 0.
An edge of a polyhedron is a face of dimension 1.
A facet of a polyhedron is a face of dimension $d-1$, where d is the dimension of the polyhedron.

Example 2.10 (Graphical examples of supporting hyperplanes). In Figure 4, we consider four different hyperplanes, and whether or not they are supporting hyperplanes. The two hyperplanes in (a) and (b) are supporting hyperplanes; those in (c) and (d) are not. Furthermore, the intersection of the hyperplane in (a) yields a zero-dimensional face (vertex), and that in (b) has a one-dimensional face (edge).

Lemma 2.11. *Let $P = \{x \geq 0 | Ax = b\}$. x is a vertex of P iff x is a BFS. Equivalently, x is a vertex of P iff x is an extreme point.*

Before we begin the proof, we note that since vertices are equivalent to BFS's and extreme point, it is enough to look at vertices when looking for the optimum solution.

Proof. Here, we prove that a BFS is a vertex. Recall that if $\vec{y}$ is a BFS, then there exists $\vec{c} \neq \vec{0}$ such that $\langle \vec{y}, \vec{c} \rangle < \langle \vec{z}, \vec{c} \rangle$ for $\vec{y} \neq \vec{z} \in P$. We define the hyperplane H as

$$H = \{\vec{x} | \langle \vec{x}, \vec{c} \rangle = \alpha\}, \quad \text{where} \quad \alpha = \langle \vec{y}, \vec{c} \rangle,$$

Since, for $\vec{z} \neq \vec{y}$ and $\vec{z} \in P$, $\langle \vec{z}, \vec{c} \rangle > \alpha$, it follows that $H \bigcap P = \{\vec{y}\}$. Furthermore, since for any $\vec{z} \in P$, $\langle \vec{z}, \vec{c} \rangle \geq \alpha$, we get $z \in H^+$, hence $P \subset H^+$. Therefore, H is a supporting hyperplane for P and its intersection with P (the point y) is a vertex. $\square$

Note that we have previously shown the equivalence between BFS's and extreme points; the proof of equivalence between vertices and extreme points will complete the proof of the lemma and is left as an exercise.

Definition 2.12. A BFS is associated with a set of indices I of size $|I| = m$, which correspond to m independent columns. Two distinct BFS's are called *neighbours* if

$$I' = I \setminus \{i_1\} \bigcup \{i_2\},$$

where I,I' are the corresponding basic indices, and i_1,i_2 are two distinct indices in $\{1, \ldots, n\}$.

In Lemma 2.11, we demonstrated a correspondence sending a BFS to a vertex. By using this definition of neighbouring BFS's (defined by two sets of valid indices which differ in one index), we see a connection to the geometrical view of neighbours. Here, swapping a single valid index is equivalent to moving from one vertex to another along an edge of a polytope. A proof for this claim can be found in Papadimitriou and Steiglitz.

3 Introduction to the Simplex Algorithm

The simplex algorithm has historically been the algorithm most commonly used to solve linear programs. While it suffers from worst case exponential complexity in time, and it is not believed that it can be made polynomial, it does perform very well on average. Next class, we will discuss the complexity of the simplex algorithm in detail.

We now provide a very high level description of the simplex algorithm. We want to take advantage of the structure of the LP, and to improve on our previous (brute force) algorithm by moving from BFS to BFS in a clever way, rather than visiting all $\begin{pmatrix} n \\ m \end{pmatrix}$ possible combinations (some of which may not be feasible).

This high level view of the simplex algorithm consists of three phases:

Algorithm 3.1 (High level view of Simplex Algorithm).

S: **S**tart at an arbitrary BFS.

M: **M**ove to a neighbouring BFS that improves (decreases) objective function.

T: **T**erminate when there is no such neighbour.

Observation 3.2 (Observations on the Simplex Algorithm). .

1. *The **S** phase is quite difficult! (See Question 3b in assignment 1, which asks us to show that Feasibility is "as hard" as Optimality in LP.)*
2. *We need to make sure that the termination condition is acceptable; i.e., that local optimality (in the sense of being better than neighbours) is enough to conclude global optimality.*
3. *While we can consider other forms (i.e. not the standard form for LP), we focus on the standard form as it makes it easier to run the simplex algorithm.*

3.1 The Start Phase

We have observed that finding a basic feasible solution from which the simplex algorithm can start is quite a hard problem. Here, we propose a method by which we can find such a BFS.

Consider our standard LP

$$\min_{\vec{x}} \langle \vec{x}, \vec{c} \rangle \quad s.t. \quad A\vec{x} = \vec{b}, \quad \vec{x} \geq 0.$$

Assume, without loss of generality, that all entries in $\vec{b}$ are non-negative since we can multiply a row in A and the corresponding b_i by -1 without changing the solution space.

We consider a different problem of solving

$$\min_{\vec{y}} \sum_i y_i \quad s.t. \quad A\vec{x} + \vec{y} = \vec{b}, \quad \vec{x}, \vec{y} \geq 0.$$

The constraints on this problem can be represented by concatenating the identity matrix I to the matrix A, i.e.

$$\begin{bmatrix} A & I \end{bmatrix} \begin{bmatrix} \vec{x} \\ \vec{y} \end{bmatrix} = \vec{b}.$$

We can then use the simplex algorithm to solve this problem; notice that in the new system, the S phase is easy: simply take $\vec{x} = 0, \vec{y} = \vec{b}$ as the BFS. If we can solve it with $\sum_i y_i = 0$, then we have found a solution to $A\vec{x} = \vec{b}$ and thus a BFS from which we can begin our original problem. Note that we do not have any difficulty finding a BFS from which to start the subproblem, as $\vec{x} = 0, \vec{y} = \vec{b}$ is a BFS for the subproblem with $\vec{y}$ as the basic variables (corresponding to the linearly independent columns in the appended identity matrix).

More precisely, at the end of this start phase, the simplex algorithm will give us an optimal solution $\begin{bmatrix} \vec{x}^\star & \vec{y}^\star \end{bmatrix}$ to the linear program. We will then have one of three possible cases:

1. $\vec{y}^\star \neq 0$. If this is the case, there are no BFS's for the original program; in other words, the original LP is infeasible.

2. $\vec{y}^\star = 0$ and $I = \{i | x_i^\star > 0\}$ is of size m. When this is the case, $\vec{x}^\star$ is a feasible solution to the original LP and I is a basis of linearly independent columns. Thus, $\vec{x}^\star$ is a BFS corresponding to I.

3. $\vec{y}^\star = 0$ and $|I| < m$. In this case, we can find additional columns in A to get $J \supset I$ of linearly independent columns. This is achieved by checking if column A_k is linearly independent of all columns $A_j, j \in I$ for each $k \notin J$; if it is, k is added to J. We call the resulting set our basis.

3.2 The Move Phase

Now that we have outlined a method for finding a BFS from which to start, we can describe the **M**ove phase of the simplex algorithm. We begin with some linear algebra and notation. We define the set

$$I = \{B(1), \cdots, B(m)\}$$

as the set of basic variables of our starting BFS, where $B(i)$ is the i^{th} index in the basis. We know that for the BFS $\vec{x}_0 = (x_{10}, x_{20}, \cdots, x_{n0})^T$,

$$\sum_{i=1}^{m} x_{i0} A_{B(i)} = \vec{b}. \tag{3}$$

Furthermore, each column of A which is not in the basis can be written as a linear combination of the columns which are in the basis:

$$A_j = \sum_{i=1}^{m} x_{ij} A_{B(i)}$$

Here, we define x_{ij} as some entries in a matrix which are used to form linear combinations of the basis columns.

A geometrical viewpoint of adding column A_j to the basis is that the basis increases from m to $m+1$ components. The previous basis (which belonged to a BFS) corresponded to a vertex; linear combinations of the previous basis with a scaled version of the new column θA_j (where $0 \leq \theta \leq \theta_0$) correspond to moving along an edge. The "upper bound" of θ_0 will be explained shortly. We can write the scaled column as

$$\theta A_j = \sum_{i=1}^{m} \theta x_{ij} A_{B(i)}.$$

By adding and subtracting θA_j to Equation (3), we get

$$\sum_{i=1}^{m} (x_{i0} - \theta x_{ij}) A_{B(i)} + \theta A_j = \vec{b}.$$

In other words, we get the values of all the basic variables in a solution, provided that j's variable is θ and all the rest are still zero.

As θ increases, the "new" x_{i0} $(x_{i0} - \theta x_{ij})$ changes. If $\theta \geq 0$ and $x_{i0} - \theta x_{ij} \geq 0$, we still have a feasible solution (i.e. all x_j for j not in I are still 0).

If, further, there exists an i for which $x_{i0} - \theta x_{ij} = 0$, then we reach a solution with at most m nonzero coordinates which we will see is a new BFS, with j in the new basis.

Figure 5 illustrates the operation of moving from one BFS to another. Here, the move is made from $\vec{x}$ to the new vertex along the direction $\vec{d} = \begin{pmatrix} -x_{1j} \\ 1 \\ -x_{2j} \\ -x_{3j} \end{pmatrix}$.

Example 3.3 (Three examples, $m = 3$ and $j = 4$). Here, we consider three examples of finding θ_0, the maximum θ for which a scaled version of a column can be subtracted.

Example A

$$\begin{array}{ccrrr} x_{i0} & = & 4 & 1 & 6 \\ x_{i4} & = & 8 & -2 & -9 \end{array}$$

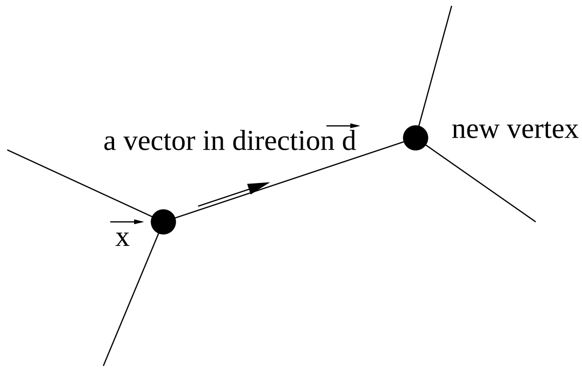

Figure 5: Illustrating a **M**ove step in the simplex algorithm

Here, $\theta_0 = \frac{1}{2}$; subtracting a scalar multiple of the row x_{i4} with θ any greater would cause the first element to be negative.

Example B

$$\begin{array}{lcrrr} x_{i0} & = & 0 & 2 & 5 \\ x_{i4} & = & 1 & -1 & -6 \end{array}$$

Here, $\theta_0 = 0$. No move can be made since any positive θ would cause the first element to be negative. This is the case whenever there is a zero entry in x_{i0} and a positive corresponding entry in x_{ij}. A BFS with less than m nonzero elements such as athe one shown here is called a "degenerated basis".

Example C

$$\begin{array}{lcrrr} x_{i0} & = & 0 & 1 & 6 \\ x_{i4} & = & 0 & -1 & -2 \end{array}$$

Here, $\theta_0 = \infty$. An arbitrarily large scalar θ can be used without causing any elements to be negative. This case occurs when for all i, $x_{ij} \leq 0$, and so increasing θ doesn't get us "any closer" to zero in any of the coordinates.

We formally define θ_0 as:

$$\theta_0 \stackrel{def}{=} \min_{\{i|x_{ij}>0\}} \frac{x_{i0}}{x_{ij}} = \frac{x_{l0}}{x_{lj}}.$$

The minimization is only performed over positions i where $x_{ij} > 0$ since we have seen that they are the ones that can cause problems. The three examples shown above are typical of the three possible solutions to θ_0:

A: $\theta_0 > 0$. This is the "usual" case. We can increase θ, but up to a certain value. We move to a new BFS.

B: $\theta_0 = 0$. This occurs when there is a zero in x_{i0} and a positive value in x_{ij}. We cannot increase θ, but we can still make j a new basic variable and throw away the i^{th} current basic variable.

C: $\theta_0 = \infty$. This occurs when there are no issues with nonnegativity – all $x_{ij} \leq 0$.

Claim 3.4. *If we define x'_{i0} as follows,*

$$x'_{i0} = \begin{cases} x_{i0} - \theta_0 x_{ij} & i \neq l \\ \theta_0 & i = l \end{cases},$$

and we define $B'(l) = j$ and $B'(i) = B(i)$ for $i \neq l$, then x'_{i0} is a new BFS with B' as the new basis.

Proof. We start our proof by showing that x'_{i0} is feasible; this is true since both $x_{i0} - \theta_0 x_{ij}$ and θ_0 are non-negative. Next we show that B' is still a basis.

In order to prove that B' is still a basis, we will show columns $A_{B'(1)}, A_{B'(2)}, \cdots, A_{B'(m)}$ are linear independent. Let j be the column entering the basis. Then A_j can be written as a linear combination of the columns which are in the original basis B:

$$A_j = \sum_{i=1}^{m} x_{ij} A_{B(i)}$$

By the choice of l, we know $x_{lj} > 0$. So we can get the following equality:

$$A_{B(l)} = \frac{A_j - \sum_{i=1, i\neq l}^{m} x_{ij} A_{B(i)}}{x_{lj}}$$

This shows $A_{B(l)}$ is expressible as a linear combination of entering column A_j and the other columns except itself in the basis B. Then we know subspace spanned by columns $A_{B(1)}, \cdots, A_{B(l)}, \cdots, A_{B(m)}$ is the same as the subspace spanned by columns $A_{B'(1)}, \cdots, A_{B'(m)}, A_j$:

$$span(A_{B(1)}, \cdots, A_{B(l)}, \cdots, A_{B(m)}) = span(A_{B'(1)}, \cdots, A_{B'(m)}, A_j) = \mathbb{R}^m$$

So B' is still a basis. □

4 Tutorial: January 19, 2005

4.1 Outlining the Linear Program

We began tutorial by considering the following LP which minimizes the function $f(x) = 2x_2 + x_4 + 5x_7$ subject to the constraints:

$$\begin{array}{ccccccccc} x_1 & +x_2 & +x_3 & +x_4 & & & & = & 4 \\ x_1 & & & & +x_5 & & & = & 2 \\ & & x_3 & & & +x_6 & & = & 3 \\ & 3x_2 & +x_3 & & & & +x_7 & = & 6 \end{array}$$

and $x_i \geq 0$ for all $1 \leq i \leq 7$.

Our first goal will be to transform this LP from standard form to canonical form. Note that this is a problem with $x \in \mathbb{R}^n$, and with m equality constraints. This set of equality constraints can be represented in matrix form as $A\vec{x} = \vec{b}$. Also note that in this particular problem, the matrix A is in the form $A = [H|I]$; that is, the $m \times m$ matrix at the right of A is the identity matrix. If one is faced with a problem in which this is not the case, it is possible to transform the $m \times m$ matrices in A to the identity matrix through elementary row operations.

By noting that for the variables corresponding to the identity part of A,

$$x_i = b_{i-(n-m)} - \sum_{j=1}^{n-m} A_{(i-(n-m))j} x_j, \quad i = n-m+1, \ldots, n,$$

where A_{ij} is the (i, j) entry in the matrix A, and also noting that $x_i \geq 0$, these m equalities can be written as inequalities in x_i, $i = 1, \ldots, n-m$:

$$\sum_{j=1}^{n-m} A_{(i-(n-m))j} x_j \leq b_{i-(n-m)}.$$

The remaining variables x_i, $i = 1, \ldots, n-m$ satisfy the inequalities $x_i \geq 0$. Thus, we have expressed the problem in canonical form:

$$\begin{array}{c|cccc}
A_4 & x_1 & +x_2 & +x_3 & \leq 4 \\
A_5 & x_1 & & & \leq 2 \\
A_6 & & & x_3 & \leq 3 \\
A_7 & & 3x_2 & +x_3 & \leq 6 \\
A_1 & x_1 & & & \geq 0 \\
A_2 & & x_2 & & \geq 0 \\
A_3 & & & x_3 & \geq 0
\end{array}$$

Note that the problem in canonical form consists of n inequalities, which we label A_i. We also note that we have reduced the domain of the problem from $\mathbb{R}^n$ to $\mathbb{R}^{n-m}$ by removing the slack variables. From this form, it is much easier to view the problem geometrically.

4.2 Geometrical Interpretation

As we have seen in class, each inequality represents a halfspace. Furthermore, the intersection of these n halfspaces forms the space of feasible solutions P. Finally, since halfspaces are convex, their intersection is convex, and as such forms a convex polytope in $\mathbb{R}^{n-m}$.

In Figure 6, we see that there are $n = 7$ faces labelled as $A_1, \ldots, A_7$. This labelling is intentional, as each face corresponds to the hyperplane which is defined when any of the corresponding inequalities A_i is satisfied with equality.

Recall that a BFS is an $\vec{x}$ vector in $\mathbb{R}^n$ with at least $n-m$ zeros (in standard form). Since a zero in position i standard form maps to the equality of A_i in canonical form, a BFS corresponds to the intersection of $n-m$ hyperplanes (corresponding to the $n-m$ zeros in $\vec{x}$). Since the polytope we have constructed from the canonical form exists in $\mathbb{R}^{n-m}$, we know that to the intersection of $n-m$ hyperplanes will define a point; furthermore, since these hyperplanes correspond to faces of the polytope, the intersection of $n-m$ hyperplanes defines a vertex of the polytope.

Consider the BFS with the basis $B = \{A_1, A_2, A_3, A_6\}$. Recall that this notation means that A_4,A_5, and A_7 do not belong to the basis, or equivalently that $x_4 = x_5 = x_7 = 0$ in standard form. We can see from Figure 6 that the intersection of the planes A_4,A_5, and A_7 is the point $(2, 2, 0)$. By substituting the values $x_4 = x_5 = x_7 = 0$ into standard form, we find the BFS $\vec{x} = (2, 2, 0, 0, 0, 3, 0)$, which has $(x_1, x_2, x_3) = (2, 2, 0)$.

Note that while the intersection of any $n-m$ hyperplanes will map to a BFS, it is possible that there are multiple mappings like this. For example, consider the BFS with basis $B' = \{A_1, A_2, A_4, A_6\} \neq B$. Here, $x_3 = x_5 = x_7 = 0$, which gives us the same solution as the previous BFS (the point $(2, 2, 0)$). This occurs because the previous BFS had more than $n-m = 3$ zeroes. In fact, since it had 4 zeroes, the point $(2, 2, 0)$ corresponds to the intersection of the 4 hyperplanes A_3, A_4, A_5, A_7. We refer to such BFS's as *degenerate*.

4.3 Moving to another BFS

Now, consider starting from the BFS with basis $B = \{A_1, A_3, A_6, A_7\}$. This corresponds to the point $(2, 0, 2)$ in $\mathbb{R}^3$. As we have seen in class, moving from one BFS to another is accomplished by relaxing one of the equalities A_2, A_4, A_5 to an inequality. Here, we consider removing the equality A_2 on x_2. Then, we will add a scaled version of the variable x_2, namely θx_2. Geometrically, this action corresponds to moving along the line towards the point $(2, 2, 0)$. This occurs since we have removed the equality A_2, and the intersection of the hyperplanes corresponding to the remaining inequalities A_4, A_5 is the line connecting the two points $(2, 0, 2)$ and $(2, 2, 0)$. As we have seen in our description of the simplex algorithm, we will continue moving along this line until one of the inequalities reaches an equality. In this case, this will occur at the point $(2, 2, 0)$ where $x_3 = 0$. At that point, the intersection of hyperplanes corresponding to equalities A_3, A_4, A_5 forms the vertex $(2, 2, 0)$ which is a BFS.

4.4 Concluding Remarks

To conclude the tutorial, we briefly discussed some implementation issues associated with the simplex algorithm.

- How do we decide which edge to move along?
  - Consider decrease in objective function in free variable (i.e. the one that we change when moving along the edge)?
  - Pick the steepest edge?
  - Pick the edge whose terminal vertex is farthest away (i.e. max distance)?

- We also noted that with degenerate BFS's, we must be careful not to get trapped in a loop. If we only consider the slope of the edge (i.e. its gradient) connecting degenerate vertices, we may think that moving will improve the objective function. However, since the two points are "separated by distance zero", clearly no improvement can be made.

- Finally, we used our geometrical interpretation to reassert that local optima are global optima. This holds since the polytope is convex.

References

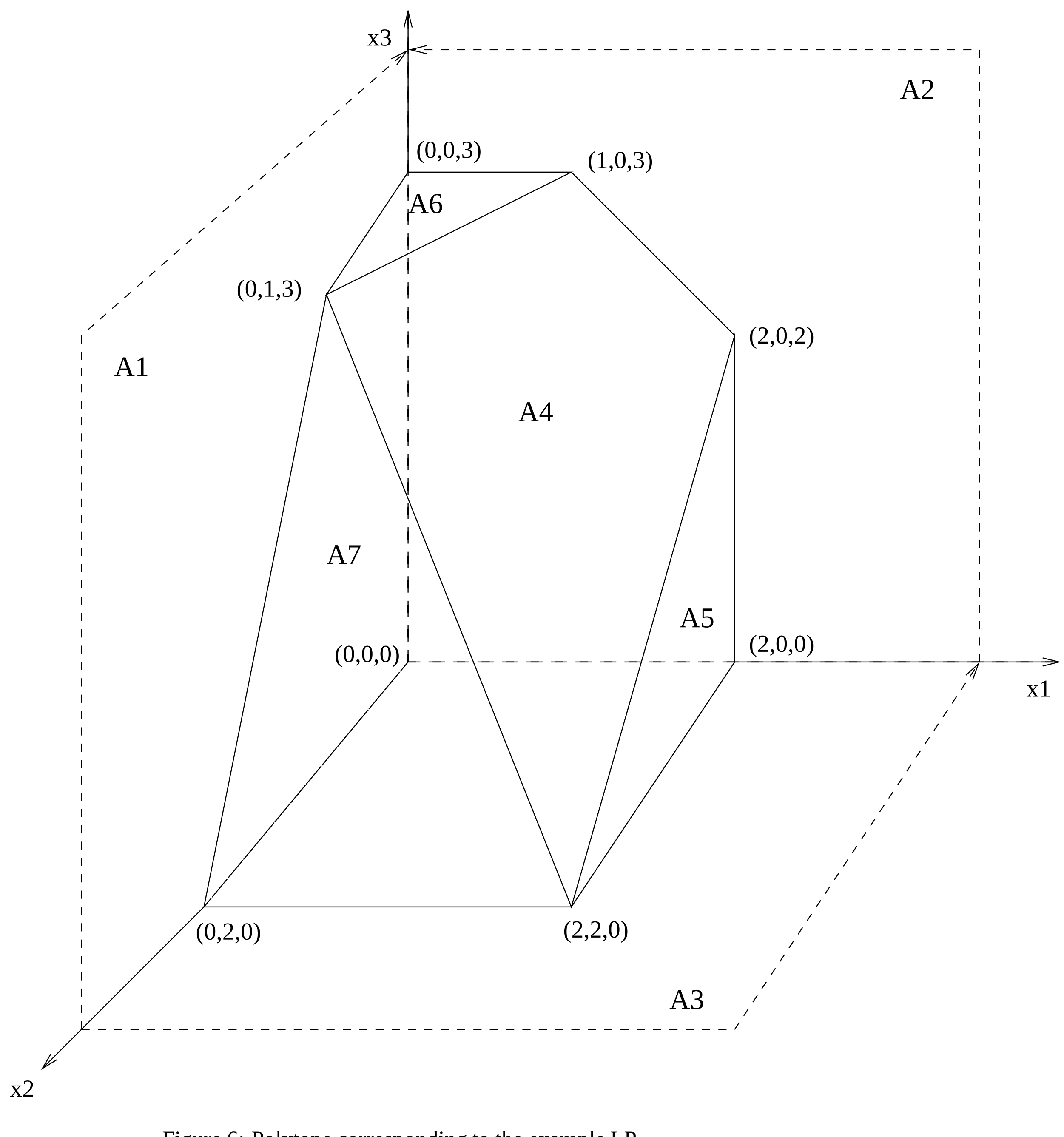

Figure 6: Polytope corresponding to the example LP