The (Dual) Simplex Method, Sensitivity, and Complexity

Yinyu Ye

Department of Management Science and Engineering

Stanford University

http://www.stanford.edu/~yyye

Linear Programming (LP)

$$(LP) \quad \text{minimize} \quad c^T x$$
$$\text{subject to} \quad Ax = b,\ x \geq 0.$$

$$(LD) \quad \text{maximize} \quad b^T y$$
$$\text{subject to} \quad A^T y + s = c,\ s \geq 0.$$

Invented by George Dantzig in 1947, the simplex method remains one of the two best solvers for LP.

Methodological Philosophy

The primal Simplex Algorithm maintains the primal feasibility and complementary slackness conditions while working toward dual feasibility. By contrast, the Dual Simplex Algorithm maintains dual feasibility and complementary slackness while working toward primal feasibility. In a sense, it is the primal Simplex Algorithm applied to the dual problem, but carried out in the format of the primal problem.

The (Dual) Simplex Method I: Test optimality

Given a dual basic feasible y_B satisfying

$$A_B^T y_B = c_B, \quad A_N^T y_B \leq c_N.$$

Let , the primal basic solution be

$$A_B x_B = b.$$

Then if $x_B \geq 0$, that is, $b \in \text{Cone}(A_B)$, y_B is optimal for the dual and x_B is optimal for the primal.

If not, we move to an adjacent basic feasible solution, that is, exactly one index of B is replaced. This solution represents a neighboring extreme point of the feasible region.

The (Dual) Simplex Method II: Find direction to go

Consider the transformed dual problem using $y' = A_B^T y$:

$$\begin{array}{lll} (LD') & \text{maximize} & x_B^T y' \\ & \text{subject to} & (A_B^{-1}A)^T y' \leq c. \end{array}$$

Note that $y' = c_B$ is a basic feasible solution to LD'.

Then, let

$$y'(\alpha) = c_B - \alpha e_i, \ \alpha \geq 0$$

where i is the ith variable in y whose coefficient is negative.

Choose α big enough such that $y'(\alpha)$ hit the boundary of the feasible region. Then we have a new basic feasible $y'(\alpha^*)$ for LD' or $y = (A_B^{-1})^T y'(\alpha^*)$ for LD. The new basic feasible solution is adjacent to the old one.

The (Dual) Simplex Method III: Pivot procedure

The algorithm works with pivot steps like those of the Primal Simplex Algorithm but uses different criteria for pivot selection and termination.

Once the problem is put into the canonical form, the method checks whether it is time to terminate. This will be the case if either of the following conditions is met by the current system:

1. $\bar{b} \geq 0$;
2. $b_o < 0$ and $\bar{A}_{oj} \geq 0$ for all $j \in \{1, \ldots, n\}$.

In the first case, the current basic solution is primal feasible. In the second case, the infeasibility of the primal is revealed.

The (Dual) Simplex Method IV: Outgoing and Entering Variables

Let $x_o,\ o \in B$ be the outgoing variable.

If neither of these conditions obtains, we have $\bar{b}_o < 0$ and $\bar{A}_{oj} < 0$ for some j. One of these negative numbers a_{rj} will be chosen as the pivot element.

Let $x_o,\ o \in B$ be the outgoing variable.

Under the rules of the algorithm, it is necessary to maintain the dual feasibility (nonnegativity of the coefficients in the objective function row). This is accomplished by chosing the pivot column e according to the minimum ratio rule

$$e \in \text{ argmin}_{j \in N} \{ \frac{r_j}{-\bar{A}_{oj}} : \bar{A}_{oj} < 0 \}.$$

Then, x_e is the entering variable.

Uniqueness

Homework 6: At an optimal basic solution 1) if the reduced cost of every nonbasic variable is positive, then the optimal primal solution is unique; 2) if every primal basic variable is positive, then the optimal dual solution is unique.

Resolving Cycling in the Simplex Algorithm

In a system of rank m, a (basic) solution that uses fewer than m columns to represent the right-hand side vector is said to be degenerate. Otherwise, it is called nondegenerate.

A basic feasible solution will be nondegenerate if and only if its m basic variables are positive.

Why is degeneracy a problem?

The Simplex Algorithm can cycle when a degenerate basic feasible solution crops up in the course of executing the algorithm.

Cycling Example

$$
\begin{array}{lrcrcrcrcrcl}
\min & -2x_1 & - & 3x_2 & + & x_3 & + & 12x_4 & & & & \\
\text{s.t.} & -2x_1 & - & 9x_2 & + & x_3 & + & 9x_4 & +x_5 & & & = 0 \\
& \frac{1}{3}x_1 & + & x_2 & - & \frac{1}{3}x_3 & - & 2x_4 & & +x_6 & & = 0 \\
& x_1, & & x_2, & & x_3, & & x_4, & x_5, & x_6 & & \geq 0
\end{array}
$$

Initially, the basic variables are $\{x_5, x_6\}$ and it is in the canonical form. The pivot sequence shown in the table below leads back to the original system after 6 pivots.

Pivot number	1	2	3	4	5	6
Basic var. out	x_6	x_5	x_2	x_1	x_4	x_3
Basic var. in	x_2	x_1	x_4	x_3	x_6	x_5

Methods for Resolving Cycling

There are several methods for resolving degeneracy in linear programming.

Among these are:

1. Perturbation of the right-hand side.
2. Lexicographic ordering.
3. Application of Bland's pivot selection rule.

Bland's Rule

It is a double least-index rule consisting of the following two parts:

(i) Among all candidates for the entering column (i.e., those with $r_j < 0$), choose the one with the smallest index, say s.

(ii) Among all rows i for which the minimum ratio test results in a tie, choose the row r for which the corresponding basic variable has the smallest index, j_r.

Theorem 1 *Under Bland's pivot selection rule, the Simplex Algorithm cannot cycle.*

Parametric Linear Programming Problem

The objective coefficient vector becomes $\mathbf{c} + \lambda\mathbf{g}$ or the right-hand-side vector of the form $\mathbf{b} + \lambda\mathbf{d}$ where the parameter λ belongs to an interval.

Denote this problem by LP(λ):

$$\begin{array}{lll} \text{LP}(\lambda) & \text{minimize} & (\mathbf{c} + \lambda\mathbf{g})^{\mathbf{T}}\mathbf{x} \\ & \text{subject to} & A\mathbf{x} = \mathbf{b} + \lambda\mathbf{d}, \\ & & \mathbf{x} \geq \mathbf{0}. \end{array}$$

Geometrical Observations

1. We know that for the function $\mathbf{c}^T\mathbf{x}$, the vector $\mathbf{c}$ denotes the direction of steepest ascent, so that $-c$ denotes the direction of steepest descent. Thus, parameterizing the cost function according to the rule $\mathbf{c} + \lambda\mathbf{g}$ changes the gradient, the normal direction of the objective hyperplane.

2. If $\mathbf{b}$ is replaced by $\mathbf{b} + \lambda\mathbf{d}$, as λ varying the point $\mathbf{b} + \lambda\mathbf{d}$ moves away from $\mathbf{b}$ in the direction $\mathbf{d}$ (depending on the sign of λ). This raises the question of whether or not the point $\mathbf{b} + \lambda\mathbf{d}$ lies in the cone generated by A_B or even A?

Getting Started

Let us consider λ around 0.

A key question in these parametric problems is: how much can the parameter λ be changed before the current optimal basic solution of LP(0) is lost?

Theorem 2 *The optmal basis of LP(0) remains optimal for LP(λ) if and only if*

$$A_B^{-1}(\mathbf{b} + \lambda\mathbf{d}) \geq \mathbf{0} \quad \textit{and} \quad (\mathbf{c} + \lambda\mathbf{g}) - \mathbf{A^T}(\mathbf{A_B^T})^{-1}(\mathbf{c} + \lambda\mathbf{g})_\mathbf{B} \geq \mathbf{0}.$$

This will establish an interval on λ in which the optmal basis of LP(0) remains optimal.

Sensitivity Analysis: Right-Hand-Side

The problem before us is to find (for each $i = 1, \ldots, m$) the range of values of the scalar λ for which the basis A_B remains optimal for the new RHS $\mathbf{b} + \lambda \mathbf{e}_i$, where $\mathbf{e}_i$ is the vector of all zero except 1 in the ith position.

A_B remains optimal if

$$A_B^{-1}(\mathbf{b} + \lambda \mathbf{e}_i) = \bar{\mathbf{b}} + \lambda(A_B^{-1}\mathbf{e}_i) \geq \mathbf{0}.$$

Then the new optimal objective value is changed from the old one by $\lambda \cdot y_i^*$ where $\mathbf{y}^*$ is the optimal dual solution of LP(0):

$$\mathbf{c}_B^T A_B^{-1}(\mathbf{b}+\lambda \mathbf{e}_i) = (\mathbf{y}^*)^T(\mathbf{b}+\lambda \mathbf{e}_i) = (\mathbf{y}^*)^T\mathbf{b}+\lambda \cdot (\mathbf{y}^*)^T\mathbf{e}_i = (\mathbf{y}^*)^T\mathbf{b}+\lambda \cdot y_i^*.$$

Sensitivity Analysis: Cost Coefficient

The problem before us is to find (for each $j = 1, \ldots, n$) the range of values of the scalar λ for which the basis A_B remains optimal for the new RHS $\mathbf{c} + \lambda \mathbf{e}_j$, where $\mathbf{e}_j$ is the vector of all zero except 1 in the jth position.

A_B remains optimal if

$$(\mathbf{c}+\lambda\mathbf{e}_j)_N - A_N^T(A_B^T)^{-1}(\mathbf{c}+\lambda\mathbf{e}_{i_j})_B = \begin{cases} \mathbf{r}_N - \lambda(\mathbf{e}_{i_j}^T \bar{A})_N^T \geq \mathbf{0} & \text{if } j \in B \\ \mathbf{r}_N + \lambda \mathbf{e}_j \geq \mathbf{0} & \text{otherwise} \end{cases}.$$

Then the new optimal objective value is changed from the old one by $\lambda \cdot x_j^*$ where $\mathbf{x}^*$ is the optimal primal solution of LP(0):

$$(\mathbf{c} + \lambda \mathbf{e}_j)_B^T A_B^{-1} \mathbf{b} = (\mathbf{c} + \lambda \mathbf{e}_j)_B^T \mathbf{x}_B^*$$

$$= (\mathbf{c} + \lambda \mathbf{e}_j)^T \mathbf{x}^* = \mathbf{c}_B^T \mathbf{x}_B^* + \lambda \mathbf{e}_j^T \mathbf{x}^* = \mathbf{c}_B^T \mathbf{x}_B^* + \lambda \cdot x_j^*.$$

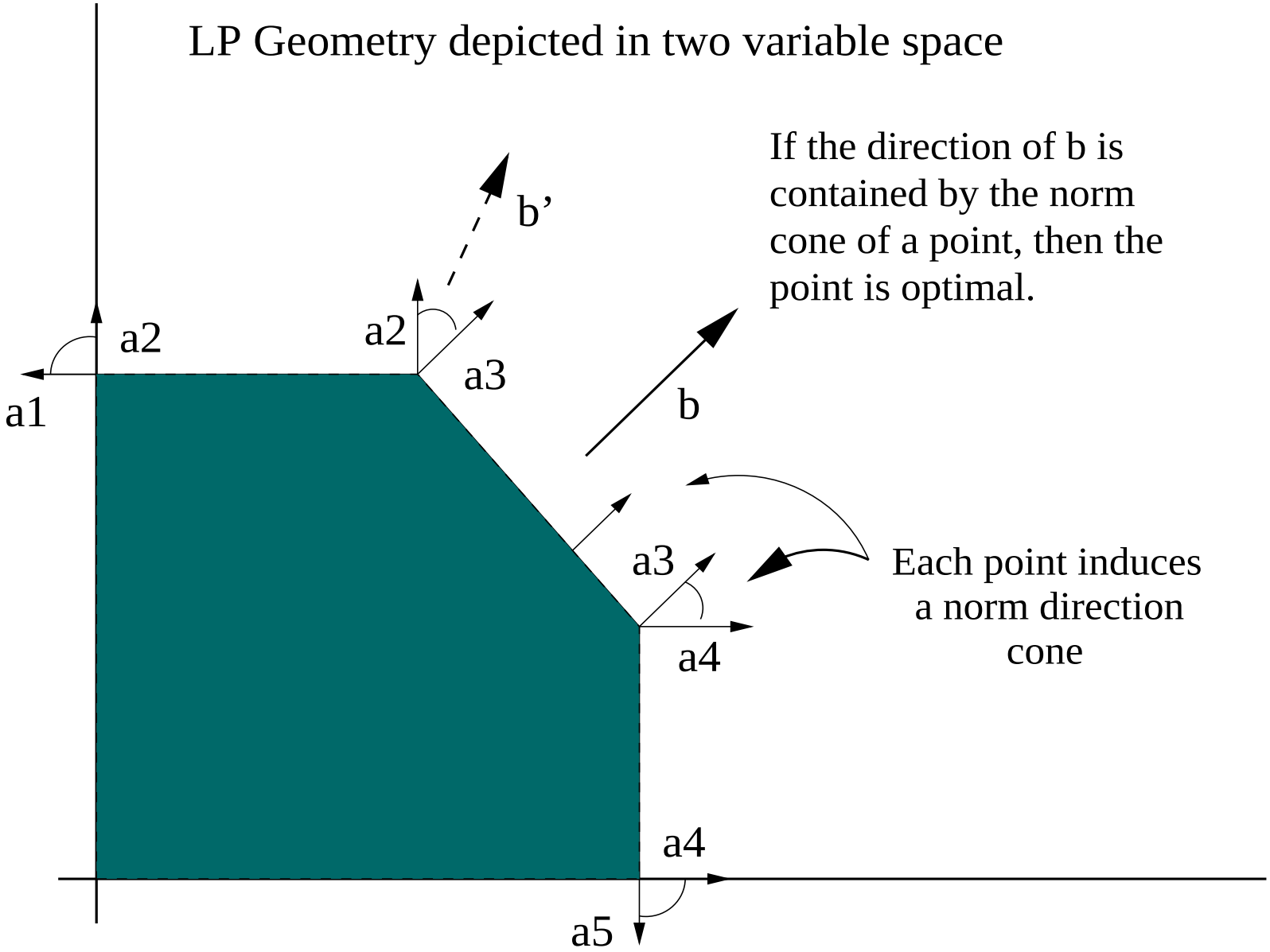
Geometry of linear programming

Consider

$$\begin{array}{lrrrr} \text{maximize} & x_1 & +2x_2 & & \\ \text{subject to} & x_1 & & \leq 1 \\ & & x_2 & \leq 1 \\ & x_1 & +x_2 & \leq 1.5 \\ & x_1, & x_2 & \geq 0. \end{array}$$

Final Tableau:

B	0	0	0	1	1	$2\frac{1}{2}$
3	0	0	1	1	-1	$\frac{1}{2}$
2	0	1	0	1	0	1
1	1	0	0	-1	1	$\frac{1}{2}$


LP Geometry depicted in two variable space
If the direction of b is contained by the norm cone of a point, then the point is optimal.
b'
a2
a2
a3
a1
b
a3
Each point induces a norm direction cone
a4
a4
a5

Wyndor Example

$$
\begin{array}{lrrrrrcr}
\max & 30x_1 & +50x_2 & & & & & \\
\text{subject to} & x_1 & & +x_3 & & & = & 4 \\
& & 2x_2 & & +x_4 & & = & 12 \\
& 3x_1 & +2x_2 & & & +x_5 & = & 18
\end{array}
$$

	x_1	x_2	x_3	x_4	x_5	1
B	-30	-50	0	0	0	0
3	1	0	1	0	0	4
4	0	2	0	1	0	12
5	3	2	0	0	1	18

	x_1	x_2	x_3	x_4	x_5	1
B	0	0	0	15	10	360
3	0	0	1	1/3	-1/3	2
2	0	1	0	1/2	0	6
1	1	0	0	-1/3	1/3	2

How Good is the Simplex Method

Very good on average, but the worse case ...?

When the simplex method is used to solve a linear program the number of iterations to solve the problem starting from a basic feasible solution is typically a small multiple of m, e.g., between $2m$ and $3m$.

At one time researchers believed—and attempted to prove—that the simplex algorithm (or some variant thereof) always requires a number of iterations that is bounded by a polynomial expression in the problem size.

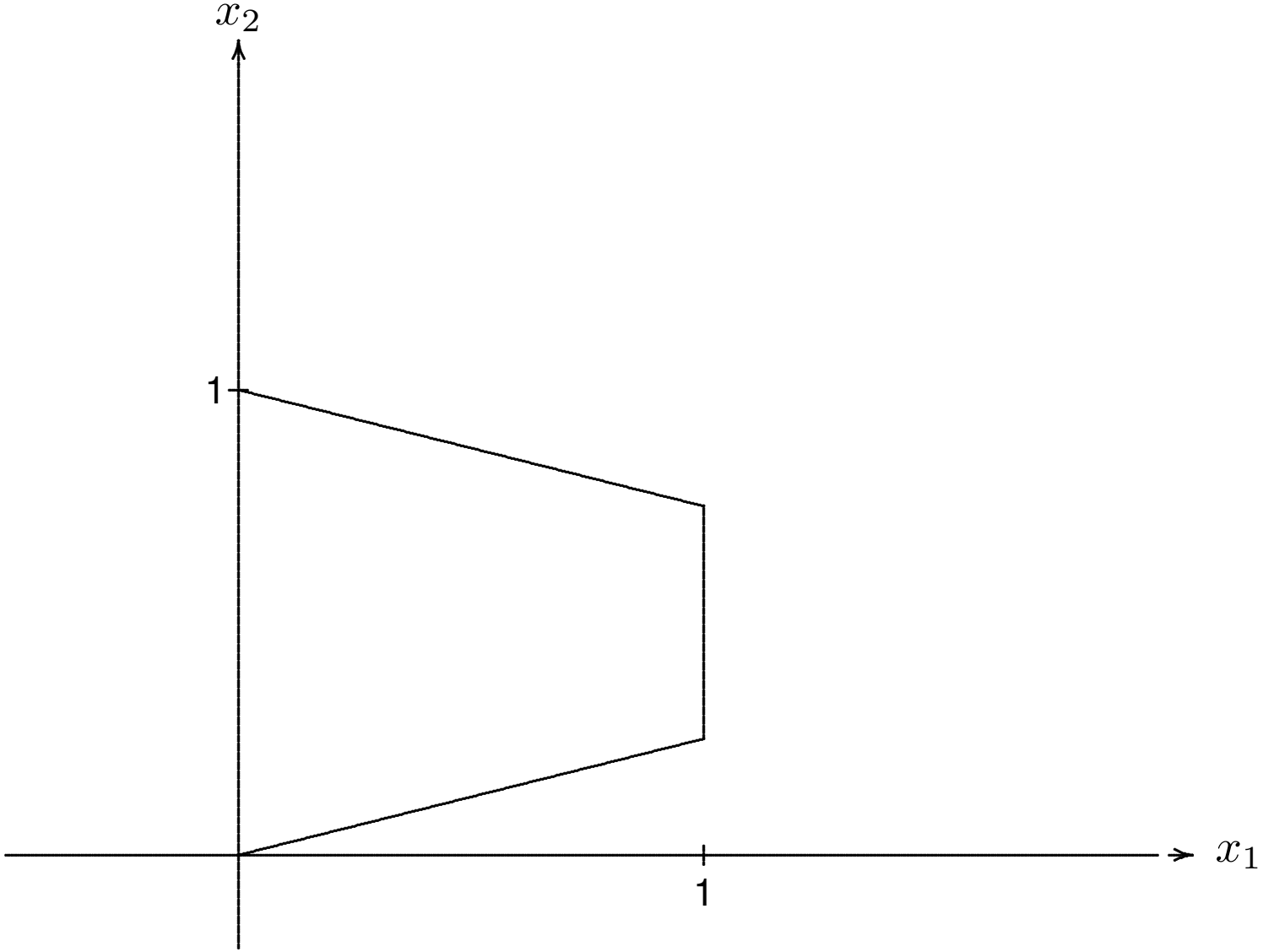
Klee and Minty Example

Consider

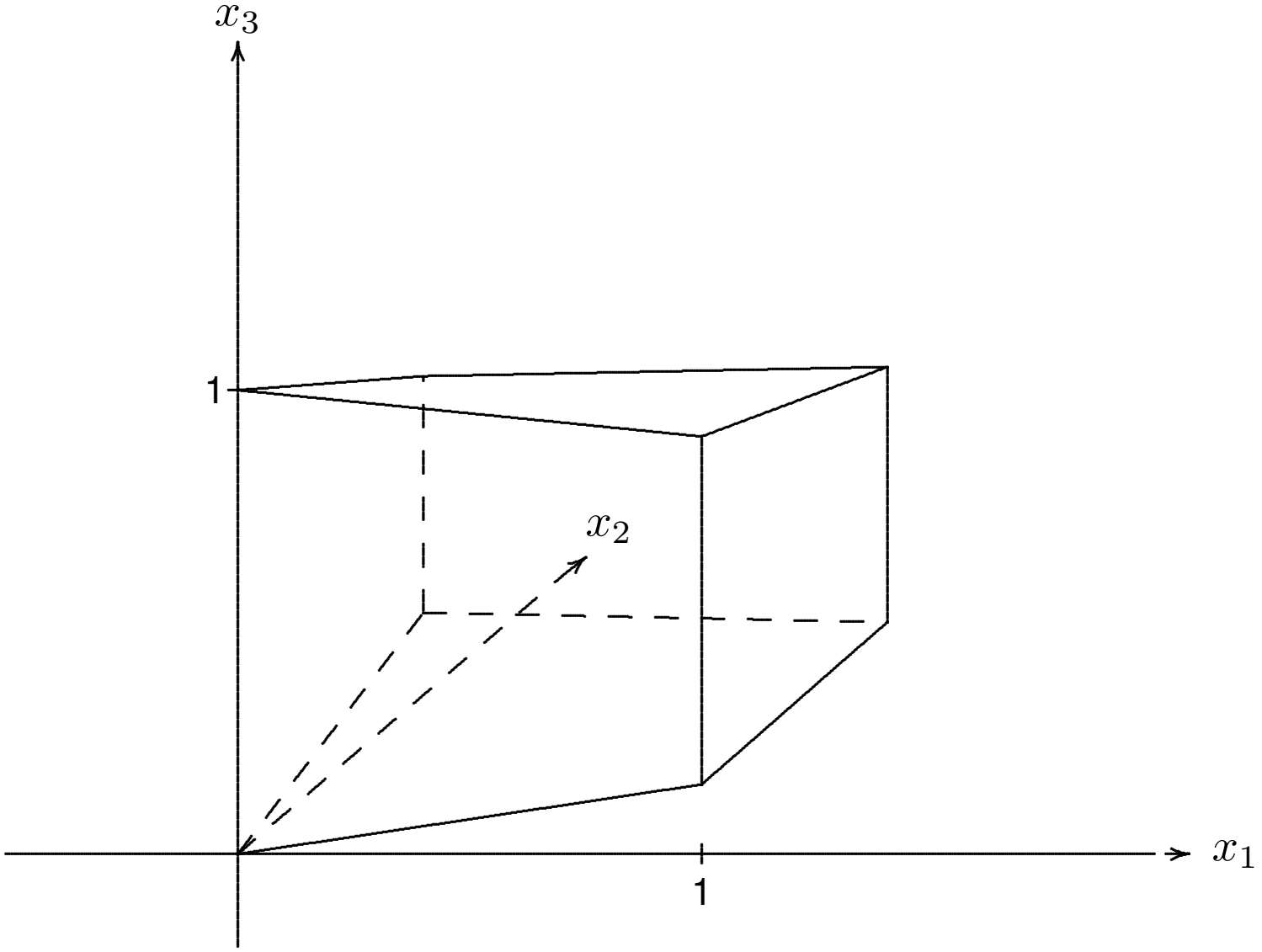
$$\begin{array}{ll} \max & x_n \\ \text{subject to} & x_1 \geq 0 \\ & x_1 \leq 1 \\ & x_j \geq \epsilon\, x_{j-1} \qquad j = 2, \ldots, n \\ & x_j \leq 1 - \epsilon\, x_{j-1} \quad j = 2, \ldots, n \end{array}$$

where $0 < \epsilon < 1/2$. This presentation of the problem emphasizes the idea (see the figures below) that the feasible region of the problem is a perturbation of the n-cube.

In the case of $n = 2$ and $\epsilon = 1/4$, the feasible region of the linear program above looks like



For the case where $n = 3$, the feasible region of the problem above looks like



The formulation above does not immediately reveal the standard form representation of the problem. Instead, we consider a different one, namely

$$\begin{aligned} \max \quad & \sum_{j=1}^{n} 10^{n-j} x_j \\ \text{subject to} \quad & 2\sum_{j=1}^{i-1} 10^{i-j} x_j + x_i \leq 100^{i-1} \quad i = 1, \ldots, n \\ & x_j \geq 0 \qquad\qquad\qquad\quad j = 1, \ldots, n \end{aligned}$$

The problem above[a] also be used is easily cast as a linear program in standard form. Unfortunately, it is less apparent how to exhibit the relationship between its feasible region and a perturbation of the unit cube.

[a] It should be noted that there is no need to express this problem in terms of powers of 10. Using any constant $C > 1$ would yield the same effect (an exponential number of pivot steps).

Example

$$
\begin{array}{lrcrcrcr}
\max & 100x_1 & + & 10x_2 & + & x_3 & & \\
\text{subject to} & x_1 & & & & & \leq & 1 \\
& 20x_1 & + & x_2 & & & \leq & 100 \\
& 200x_1 & + & 20x_2 & + & x_3 & \leq & 10,000
\end{array}
$$

In this case, we have three constraints and three variables (along with their nonnegativity constraints). After adding slack variables, we get a problem in standard form. The system has $m = 3$ equations and $n = 6$ nonnegative variables. In tableau form, the problem is

T^0

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	100	10	1	0	0	0	0
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	200	20	1	0	0	1	10,000
				•	•	•	

The bullets below the tableau indicate the columns that are basic.

T^1

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	10	1	–100	0	0	–100
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	20	1	–200	0	1	9,800
	●				●	●	

T^2

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	0	1	100	–10	0	–900
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	0	1	200	–20	1	8,200
	●	●				●	

T^3

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	–100	0	1	0	–10	0	–1,000
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	–200	0	1	0	–20	1	8,000
		•		•		•	

T^4

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	100	0	0	0	10	–1	–9,000
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	–200	0	1	0	–20	1	8,000

● ● ●

T^5

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	0	0	–100	10	–1	–9,100
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	0	1	200	–20	1	8,200

● ● ●

T^6

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	0	–10	0	100	0	–1	–9,900
0	1	0	0	1	0	0	1
0	0	1	0	–20	1	0	80
0	0	20	1	–200	0	1	9,800
	●		●		●		

T^7

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	1
1	–100	–10	0	0	0	–1	–10,000
0	1	0	0	1	0	0	1
0	20	1	0	0	1	0	100
0	200	20	1	0	0	1	10,000
			•	•	•		

$$(x_1, x_2, x_3, x_4, x_5, x_6) = (0, 0, 10^4, 1, 10^2, 0)$$

is optimal and that the objective function value is $10,000.$

Along the way, we made $2^3 - 1 = 7$ pivot steps. The objective function made a strict increase with each change of basis.

Remark. The instance of the linear program (1) in which $n = 3$ leads to $2^3 - 1$ pivot steps when the greedy rule is used to select the pivot column. The general problem of the class (1) takes $2^n - 1$ pivot steps. To get an idea of how bad this can be, consider the case where $n = 50$. Now $2^{50} - 1 \approx 10^{15}$. In a year with 365 days, there are approximately 3×10^7 seconds. If a computer were running continuously and performing T iterations of the Simplex Algorithm per second, it would take approximately

$$\frac{10^{15}}{3T \times 10^8} = \frac{1}{3T} \times 10^8 \quad \text{years}$$

to solve the problem using the Simplex Algorithm with the greedy pivot selection rule.

An interesting connection

Consider the eight vectors $v^k = (v_1^k, v_2^k, v_3^k)$ where $k = 0, 1, \ldots, 7$ and

$$v_j^k = \begin{cases} 1 & \text{if } x_j \text{ is basic in tableau } k \\ 0 & \text{otherwise} \end{cases}$$

Looking at the eight tableaus $\mathrm{T}^0, \mathrm{T}^1, \ldots, \mathrm{T}^7$, we see that

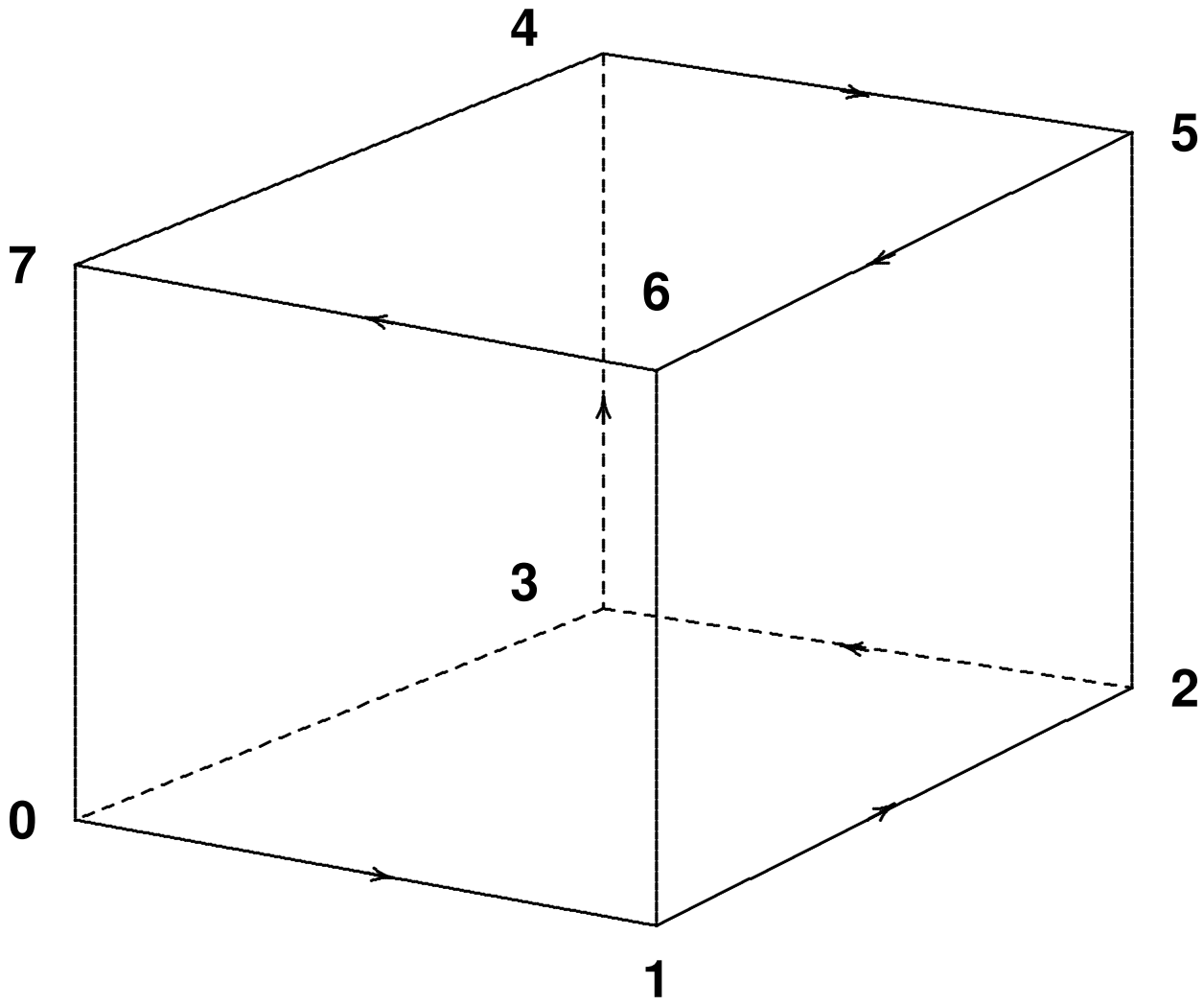
$$v^0 = (0,0,0) \qquad v^4 = (0,1,1)$$

$$v^1 = (1,0,0) \qquad v^5 = (1,1,1)$$

$$v^2 = (1,1,0) \qquad v^6 = (1,0,1)$$

$$v^3 = (0,1,0) \qquad v^7 = (0,0,1)$$

Now suppose we regard these vectors as the coordinates of the vertices of the 3-cube $[\,0,1]$.


4
5
7
6
3
2
0
1

The figure above illustrates the fact that the sequence of vectors v^k corresponds to a path on the edges of the 3-cube. The path visits each vertex of the cube once and only once. Such a path is said to be Hamiltonian.

There is an amusing recreational literature that connects Hamiltonian path with certain puzzles. See Martin Gardner, "Mathematical games, the curious properties of the Gray code and how it can be used to solve puzzles," *Scientific American* 227 (August 1972) pp. 106-109. See also, S.N. Afriat, *The Ring of Linked Rings*, London: Duckworth, 1982.