Linear and Integer Programming (ADM II)

Martin Skutella

TU Berlin

WS 2007/08

An iteration of the simplex method

Let $B = (A_{B(1)} \dots A_{B(m)})$ be a basis matrix with a corresponding basic feasible solution x.

1. Let $\bar{c}^T := c^T - {c_B}^T B^{-1} A$. If $\bar{c} \geq 0$, then STOP; else choose j with $\bar{c}_j < 0$.
2. Let $u := B^{-1} A_j$. If $u \leq 0$, then STOP (optimal cost is $-\infty$).
3. Let $\theta^* := \min_{i: u_i > 0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$ for some $\ell \in \{1, \dots, m\}$.
4. Form new basis by replacing $A_{B(\ell)}$ with A_j; corresponding basic feasible solution y is given by

$$y_j := \theta^* \qquad \text{and} \qquad y_{B(i)} = x_{B(i)} - \theta^* u_i \quad \text{for } i \neq \ell.$$

Remark

We say that the nonbasic variable x_j *enters the basis* and the basic variable $x_{B(\ell)}$ *leaves the basis*.

Correctness of the simplex method

Theorem

Assume that the feasible set is nonempty and that every basic feasible solution is nondegenerate. Then, the simplex method terminates after a finite number of iterations. At termination, there are the following two possibilities:

1. *We have an optimal basis matrix B and an associated basic feasible solution x which is optimal.*
2. *We have found a vector d satisfying $Ad = 0$, $d \geq 0$, and $c^T d < 0$, and the optimal cost is $-\infty$.*

Prof sketch.

The simplex method makes progress in every iteration. Since there are only finitely many different basic feasible solutions, it stops after a finite number of iteration. □

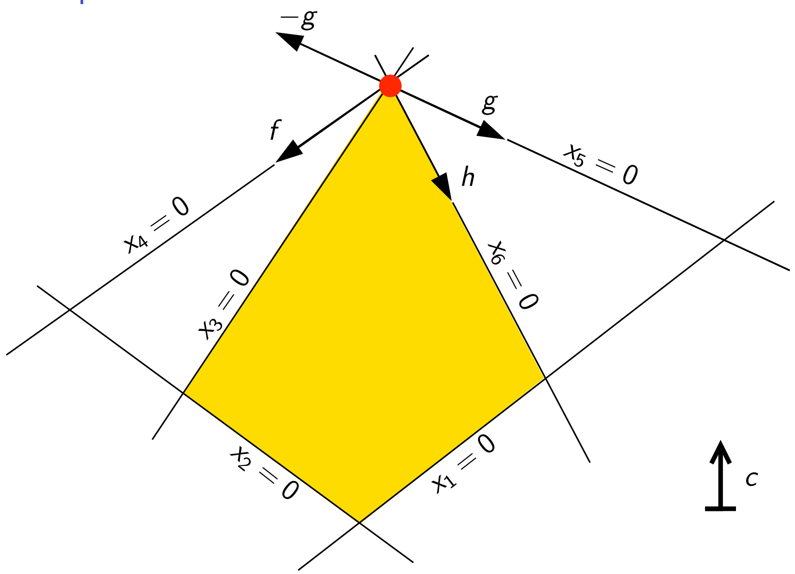
The simplex method for degenerate problems

- An iteration of the simplex method can also be applied if x is a degenerate basic feasible solution.
- In this case it might happen that $\theta^* := \min_{i: u_i > 0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell} = 0$ if some basic variable $x_{B(\ell)}$ is zero and $d_{B(\ell)} < 0$.
- Thus, $y = x + \theta^* d = x$ and the current basic feasible solution does not change.
- But replacing $A_{B(\ell)}$ with A_j still yields a new basis with associated basic feasible solution $y = x$.

Remark

Even if θ^* is positive, more than one of the original basic variables may become zero at the new point $x + \theta^* d$. Since only one of them leaves the basis, the new basic feasible solution y is degenerate.

Example



Pivot Selection

Question

How to choose j with $\overline{c}_j < 0$ and ℓ with $\frac{x_{B(\ell)}}{u_\ell} = \min_{i:u_i>0} \frac{x_{B(i)}}{u_i}$ if several possible choices exists in an iteration of the simplex algorithm?

The choice of j is critical for the overall behavior of the simplex method. Three popular choices are:

- **smallest subscript rule:** choose smallest j with $\overline{c}_j < 0$. (very simple; no need to compute entire vector $\overline{c}$; usually leads to many iterations)
- **steepest descent rule:** choose j such that $\overline{c}_j < 0$ is minimal. (relatively simple; commonly used for mid-size problems; does not necessarily yield the best neighboring solution)
- **best improvement rule:** choose j such that $\theta^* \overline{c}_j$ is minimal. (computationally expensive; used for large problems; usually leads to very few iterations)

3.3 Implementations of the simplex method

- The revised simplex method
- The full tableau implementation
- Comparison
- Practical performance ehancements

The revised simplex method

Observation

In order to execute one iteration of the simplex method efficiently, it suffices to know $B(1), \dots, B(m)$, the inverse B^{-1} of the basis matrix and the input data A, b, and c. It is then easy to compute:

$$x_B = B^{-1} b \qquad \overline{c}^T = c^T - c_B{}^T B^{-1} A$$

$$u = B^{-1} A_j \qquad \theta^* = \min_{i:u_i>0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$$

The new basis matrix is then

$$\overline{B} = \left(A_{B(1)} \dots A_{B(\ell-1)} A_j A_{B(\ell+1)} \dots A_{B(m)}\right)$$

Question

How to obtain $\overline{B}^{-1}$ efficiently?

How to obtain $\overline{B}^{-1}$ efficiently?

- Notice that $B^{-1}\overline{B} = (e_1 \dots e_{\ell-1}\, u\, e_{\ell+1} \dots e_m)$.
- By elementary linear algebra, $\overline{B}^{-1}$ can be obtained from B^{-1} as follows:
  Multiply the ℓth row of B^{-1} with $1/u_\ell$; then subtract u_i times the resulting ℓth row from the ith row, for $i \neq \ell$.
- These are exactly the elementary row operations needed to turn $B^{-1}\overline{B}$ into the identity matrix!
- Elementary row operations are the same as multiplying the matrix with corresponding elementary matrices from the left hand side.
- Equivalently:

Obtaining $\overline{B}^{-1}$ from B^{-1}

Apply elementary row operations to the matrix $(B^{-1}|u)$ to make the last column equal to the unit vector e_ℓ. The first m columns of the resulting matrix form the inverse $\overline{B}^{-1}$ of the new basis matrix $\overline{B}$.

An iteration of the "revised simplex method"

Given: $A_{B(1)}, \dots, A_{B(m)}$, an associated basic feasible solution x, and B^{-1}.

1. Let $p^T := {c_B}^T B^{-1}$ and compute the reduced costs $\overline{c}_j := c_j - p^T A_j$; if $\overline{c} \geq 0$, then STOP; else choose j with $\overline{c}_j < 0$.
2. Let $u := B^{-1}A_j$. If $u \leq 0$, then STOP (optimal cost is $-\infty$).
3. Let $\theta^* := \min_{i: u_i > 0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$ for some $\ell \in \{1, \dots, m\}$.
4. Form new basis by replacing $A_{B(\ell)}$ with A_j; corresponding basic feasible solution y is given by
$$y_j := \theta^* \qquad \text{and} \qquad y_{B(i)} = x_{B(i)} - \theta^* u_i \quad \text{for } i \neq \ell.$$
5. Apply elementary row operations to the matrix $(B^{-1}|u)$ to make the last column equal to the unit vector e_ℓ. The first m columns of the resulting matrix yield $\overline{B}^{-1}$.

The full tableau implementation

Main idea

Instead of maintaining and updating the matrix B^{-1}, we maintain and update the $m \times (n+1)$-matrix
$$B^{-1}(b|A) = (B^{-1}b|B^{-1}A)$$
which is called the *simplex tableau*.

- The *zeroth column* $B^{-1}b$ contains x_B.
- For $i = 1, \dots, n$, the ith column of the tableau is $B^{-1}A_i$.
- The column $u = B^{-1}A_j$ corresponding to the variable x_j that is about to enter the basis is the *pivot column*.
- If the ℓth basic variable $x_{B(\ell)}$ exits the basis, the ℓth row of the tableau is the *pivot row*.
- The element $u_\ell > 0$ is the *pivot element*.

The full tableau implementation (cont.)

Notice that the simplex tableau $B^{-1}(b|A)$ represents the equality system $B^{-1}b = B^{-1}Ax$ which is equivalent to $Ax = b$.

Updating the simplex tableau

- At the end of an iteration, the simplex tableau $B^{-1}(b|A)$ has to be updated to $\overline{B}^{-1}(b|A)$.
- $\overline{B}^{-1}$ can be obtained from B^{-1} by elementary row operations (i.e. $\overline{B}^{-1} = Q\overline{B}^{-1}$ where Q is a product of elementary matrices).
- Thus, $\overline{B}^{-1}(b|A) = QB^{-1}(b|A)$ and the new tableau $\overline{B}^{-1}(b|A)$ can be obtained from the old one by applying the same elementary row operations.

The zeroth row of the simplex tableau

In order to keep track of the objective function value and the reduced costs, we consider the following augmented simplex tableau:

$-{c_B}^T B^{-1} b$	$c^T - {c_B}^T B^{-1} A$
$B^{-1} b$	$B^{-1} A$

or in more detail

$$\begin{array}{c|ccc} -{c_B}^T x_B & \overline{c}_1 & \cdots & \overline{c}_n \\ \hline x_{B(1)} & | & & | \\ \vdots & B^{-1}A_1 & \cdots & B^{-1}A_n \\ x_{B(m)} & | & & | \end{array}$$

Update after one iteration

The zeroth row is updated by adding a multiple of the pivot row to the zeroth row to set the reduced cost of the entering variable to zero.

An iteration of the full tableau implementation

Given: Simplex tableau associated with a feasible basis $A_{B(1)}, \dots, A_{B(m)}$.

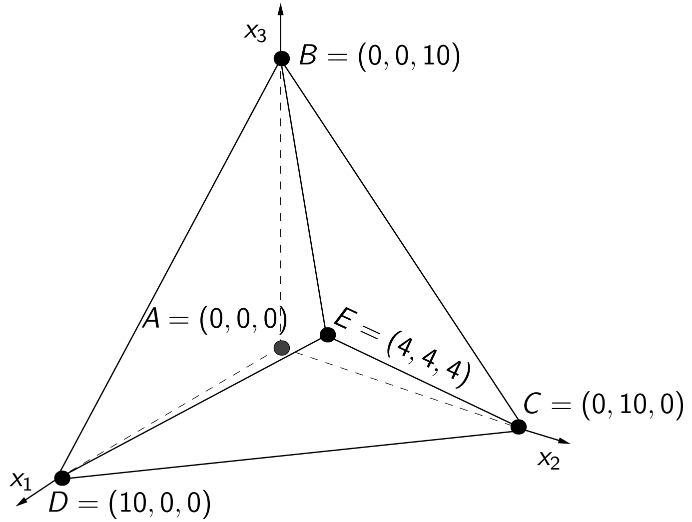
1. If $\overline{c} \geq 0$ (zeroth row), then STOP; else choose pivot column j with $\overline{c}_j < 0$.
2. If $u = B^{-1}A_j \leq 0$ (jth column), then STOP (optimal cost is $-\infty$).
3. Let $\theta^* := \min_{i:u_i>0} \frac{x_{B(i)}}{u_i} = \frac{x_{B(\ell)}}{u_\ell}$ for some $\ell \in \{1, \dots, m\}$ (see columns 0 and j).
4. Form new basis by replacing $A_{B(\ell)}$ with A_j.
5. Apply elementary row operations to the simplex tableau so that u_ℓ (pivot element) becomes one and all other entries of the pivot column become zero.

The full tableau implementation: An example

A simple linear programming problem

$$\begin{array}{rrcrcrcr} \min & -10x_1 & - & 12x_2 & - & 12x_3 & & \\ \text{s.t.} & x_1 & + & 2x_2 & + & 2x_3 & \leq & 20 \\ & 2x_1 & + & x_2 & + & 2x_3 & \leq & 20 \\ & 2x_1 & + & 2x_2 & + & x_3 & \leq & 20 \\ & & & & & x_1, x_2, x_3 & \geq & 0 \end{array}$$

The feasible set visualized in $\mathbb{R}^3$



Introducing slack variables

$$\begin{array}{rrcrcrcl}
\min & -10x_1 & - & 12x_2 & - & 12x_3 & & \\
\text{s.t.} & x_1 & + & 2x_2 & + & 2x_3 & \leq & 20 \\
 & 2x_1 & + & x_2 & + & 2x_3 & \leq & 20 \\
 & 2x_1 & + & 2x_2 & + & x_3 & \leq & 20 \\
 & & & & & x_1, x_2, x_3 & \geq & 0
\end{array}$$

LP in standard form

$$\begin{array}{rrcrcrlll}
\min & -10x_1 & - & 12x_2 & - & 12x_3 & & & \\
\text{s.t.} & x_1 & + & 2x_2 & + & 2x_3 & + x_4 & & = 20 \\
 & 2x_1 & + & x_2 & + & 2x_3 & + x_5 & & = 20 \\
 & 2x_1 & + & 2x_2 & + & x_3 & + x_6 & & = 20 \\
 & & & & & & x_1, \ldots, x_6 & & \geq 0
\end{array}$$

Observation

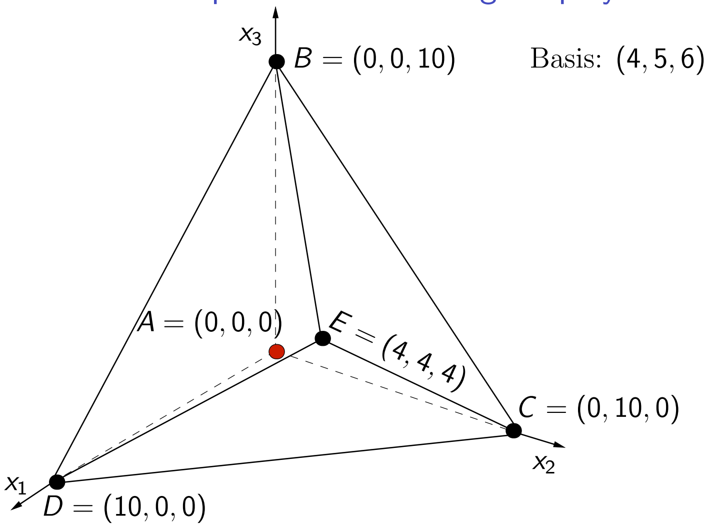
The right hand side of the system is non-negative. Therefore the point $(0, 0, 0, 20, 20, 20)$ is a basic feasible solution and we can start the simplex method with basis $B(1) = 4, B(2) = 5, B(3) = 6$.

Setting up the simplex tableau

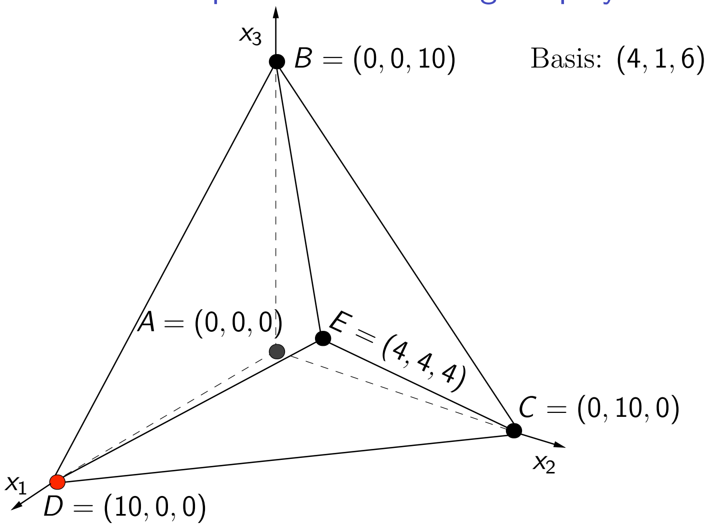
		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	0	−10	−12	−12	0	0	0	
$x_4 =$	20	1	2	2	1	0	0	20
$x_5 =$	20	2	1	2	0	1	0	10
$x_6 =$	20	2	2	1	0	0	1	10

- Determine pivot column (e.g. take smallest subscript rule).
- $\bar{c}_1 < 0 \quad \Rightarrow \quad x_1$ enters the basis.
- Find pivot row with $u_i > 0$ and $\frac{x_{B(i)}}{u_i}$ minimum.
- Rows 2 and 3 both attain minimum.
- Choose $i = 2$ with $B(i) = 5$. $\Rightarrow$ x_5 leaves the basis.
- Perform basis change: Eliminate other entries in the pivot column.
- Obtain new basic feasible solution $(10, 0, 0, 10, 0, 0)$, cost value -100.

Geometric interpretation in the original polyhedron



Geometric interpretation in the original polyhedron

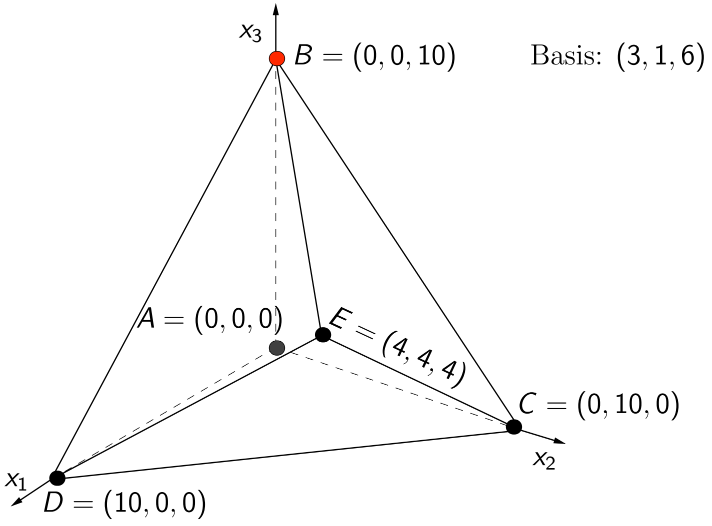


The next iterations

		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$
	100	0	-7	-2	0	5	0	
$x_4 =$	10	0	1.5	1	1	-0.5	0	10
$x_1 =$	10	1	0.5	1	0	0.5	0	10
$x_6 =$	0	0	1	-1	0	-1	1	–

- $\bar{c}_2, \bar{c}_3 < 0 \;\Rightarrow\;$ Two possible choices for pivot column.
- Choose x_3 for entering the new basis.
- $u_3 < 0 \;\Rightarrow\;$ Third row will not be a choice for pivot row.
- Choose x_4 to leave basis.
- New basic feasible solution: $(0, 0, 10, 0, 0, 10)$, correspondig to point B in the original polyhedron.

The feasible set visualized in $\mathbb{R}^3$



The next iterations

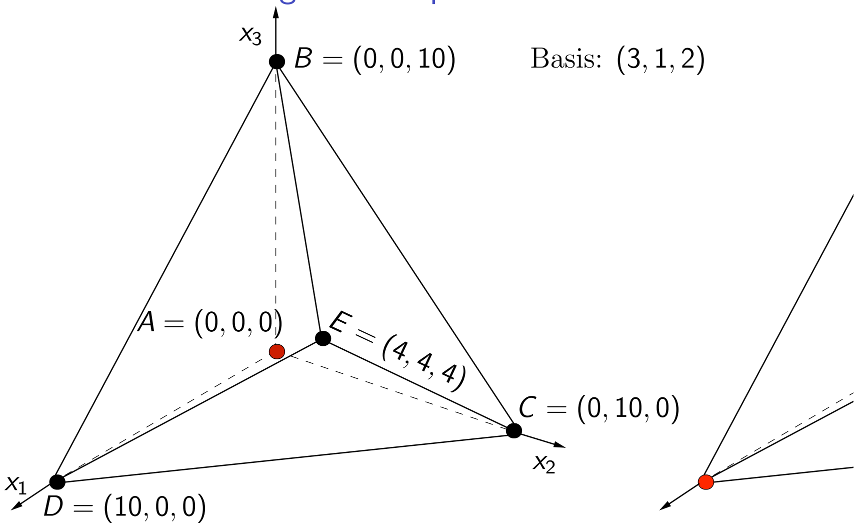
		x_1	x_2	x_3	x_4	x_5	x_6	$\frac{x_{B(i)}}{u_i}$	
	120	0	-4	0	2	4	0		
$x_3 =$	10	0	1.5	1	1	-0.5	0	$\frac{20}{3}$	
$x_1 =$	0	1	-1	0	-1	1	0	–	
$x_6 =$	10	0	2.5	0	1	-1.5	1	4	$< \frac{20}{3}$

x_2 enters the basis, x_6 leaves it. We get

		x_1	x_2	x_3	x_4	x_5	x_6
	136	0	0	0	3.6	1.6	1.6
$x_3 =$	4	0	0	1	0.4	0.4	-0.6
$x_1 =$	4	1	0	0	-0.6	0.4	0.4
$x_2 =$	4	0	1	0	0.4	-0.6	0.4

and the reduced costs are non-negative.
Thus $(4, 4, 4, 0, 0, 0)$ is an optimal solution with cost value -136, corresponding to point $E = (4, 4, 4)$ in the original polyhedron.

The iterations from geometric point of view



Cycling

If a linear programming problem is degenerate, the simplex method might end up in an infinite loop (*cycling*).

An example

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	$-3/4$	20	$-1/2$	6	0	0	0
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Pivoting rules

1. **Column selection:** We select a nonbasic variable with the most negative reduced cost $\bar{c}_j$ to be the one that enters the basis, i.e. steepest descent rule.
2. **Row selection:** Out of all basic variables that are eligible to exit the basis, we select the one with the smallest subscript.

Iteration 1

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	$-3/4$	20	$-1/2$	6	0	0	0	
$x_5 =$	0	$1/4$	-8	-1	9	1	0	0	0
$x_6 =$	0	$1/2$	-12	$-1/2$	3	0	1	0	0
$x_7 =$	1	0	0	1	0	0	0	1	–

Basis change: x_1 enters the basis x_5 leaves.

Bases visited

$(5, 6, 7)$

Iteration 2

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	0	-4	$-7/2$	33	3	0	0	
$x_1 =$	0	1	-32	-4	36	4	0	0	–
$x_6 =$	0	0	4	$3/2$	-15	-2	1	0	0
$x_7 =$	1	0	0	1	0	0	0	1	–

Basis change: x_2 enters the basis x_6 leaves.

Bases visited

$(5, 6, 7) \rightarrow (1, 6, 7)$

Iteration 3

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	0	0	$-4/2$	18	1	1	0	
$x_1 =$	0	1	0	8	156	-12	4	0	0
$x_2 =$	0	0	1	$3/8$	$-15/4$	$-1/2$	$1/4$	0	0
$x_7 =$	1	0	0	1	0	0	0	1	1

Basis change: x_3 enters the basis x_1 leaves.

Bases visited

$(5, 6, 7) \rightarrow (1, 6, 7) \rightarrow (1, 2, 7)$

Iteration 4

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	1/4	0	0	−3	−2	3	0	
$x_3 =$	0	1/8	0	1	−21/2	−3/2	1	0	–
$x_2 =$	0	−3/64	1	0	3/16	1/16	−1/8	0	0
$x_7 =$	1	−1/8	0	0	21/2	3/2	−1	1	2/21

Basis change: x_4 enters the basis x_2 leaves.

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7) \rightarrow (3,2,7)$

Iteration 5

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	−1/2	16	0	0	−1	1	0	
$x_3 =$	0	−5/2	56	1	0	2	−6	0	0
$x_4 =$	0	−1/4	16/3	0	1	1/3	−2/3	0	0
$x_7 =$	1	5/2	−56	0	0	−2	6	1	–

Basis change: x_5 enters the basis x_3 leaves.

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7) \rightarrow (3,2,7) \rightarrow (3,4,7)$

Observation

After 4 pivoting iterations our basic feasible solution still has not changed.

Iteration 6

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	$\frac{x_{B(i)}}{u_i}$
	3	−7/4	44	1/2	0	0	−2	0	
$x_1 =$	0	−5/4	28	1/2	0	1	−3	0	–
$x_2 =$	0	1/6	−4	−1/6	1	0	1/3	0	0
$x_7 =$	1	0	0	1	0	0	0	1	–

Basis change: x_6 enters the basis x_4 leaves.

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7) \rightarrow (3,2,7) \rightarrow (3,4,7)$
$\rightarrow (5,4,7)$

Back at the beginning

		x_1	x_2	x_3	x_4	x_5	x_6	x_7
	3	−3/4	20	−1/2	6	0	0	0
$x_5 =$	0	1/4	−8	−1	9	1	0	0
$x_6 =$	0	1/2	−12	−1/2	3	0	1	0
$x_7 =$	1	0	0	1	0	0	0	1

Bases visited

$(5,6,7) \rightarrow (1,6,7) \rightarrow (1,2,7) \rightarrow (3,2,7) \rightarrow (3,4,7)$
$\rightarrow (5,4,7) \rightarrow (5,6,7)$

This is the same basis that we started with.

Conclusion

Continuing with the pivoting rules we agreed on at the beginning, the simplex method will never terminate in this example.

Comparison of full tableau and revised simplex methods

The following table gives the computational cost of one iteration of the simplex method for the two variants introduced above.

	full tableau	revised simplex
memory	$O(mn)$	$O(m^2)$
worst-case time	$O(mn)$	$O(mn)$
best-case time	$O(mn)$	$O(m^2)$

Conclusion

- For implementation purposes, the revised simplex method is clearly preferable due to its smaller memory requirement and smaller average running time.
- The full tableau method is convenient for solving small LP instances by hand since all necessary information is readily available.

Practical performance enhancements

Numerical stability

The most critical issue when implementing the (revised) simplex method is numerical stability. In order to deal with this, a number of additional ideas from numerical linear algebra are needed.

- Every update of B^{-1} introduces roundoff or truncation errors which accumulate and might eventually lead to highly inaccurate results. Solution: Compute the matrix B^{-1} from scratch once in a while.
- Instead of computing B^{-1} explicitly, it can be stored as a product of matrices $Q_k \cdot Q_{k-1} \cdot \ldots \cdot Q_1$ where each matrix Q_i can be specified in terms of m coefficients. Then $\overline{B}^{-1} = Q_{k+1} \cdot B^{-1} = Q_{k+1} \cdot \ldots \cdot Q_1$. This might also save space.
- Instead of computing B^{-1} explicitly, compute and store an LR-decomposition.