MAT 3007 – Optimization

Simplex Method and Simplex Tableau

Lecture 06 *June 18th*

Andre Milzarek iDDA / CUHK-SZ

Repetition

Recap: An Iteration of the Simplex Method

Initialization: We start from a BFS x (with corresponding basis B).

1. We first compute the reduced costs $\bar{c}$:

$$\bar{c}_j = c_j - c_B^\top A_B^{-1} A_j.$$

   - If none of the reduced costs is negative, then x is optimal.
   - Otherwise choose some j such that $\bar{c}_j < 0$.

2. Compute the jth basic direction $d = [-A_B^{-1}A_j; 0; \ldots; 1; \ldots; 0]$.
   - If $d \geq 0$, then the problem is unbounded, i.e., the optimal value is $-\infty$.
   - Otherwise, compute $\theta^* = \min_{i \in B, d_i < 0}\{-\frac{x_i}{d_i}\}$.

3. Set $y = x + \theta^* d$. Then, the point y is the new BFS with index j replacing $B(\ell)$ in the basis, where $B(\ell)$ is the index attaining the minimum in θ^*.

4. Repeat these procedures.

Simplex Iteration: Discussion

Theorem: Properties of y

Let x be a BFS with basic indices B and let $y = x + \theta^* d$ be generated by the simplex iteration. Then, y is a basic feasible solution associated with the basic indices

$$\{B(1), ..., B(\ell - 1), j, B(\ell + 1), ..., B(m)\}.$$

Observation:

- Typically, y achieves a lower objective value or the method stops with an optimal solution (if the reduced costs are nonnegative).
- If there exists i with $x_{B(i)} = 0$, then the objective functions might stay the same ($\rightsquigarrow \theta^* = 0$).
- We continue to talk about these degenerate cases today.

Discussion and Logistics

Remaining Open Questions:

- Given several j with $\bar{c}_j < 0$, which jth basic direction should we choose?
- Suppose there are multiple ℓ with $\theta^* = -x_{B(\ell)}/d_{B(\ell)}$, how should we update the basic indices?
- How can we find an initial BFS?
- Practical implementation of the simplex method?

Logistics:

- Homework 1 is due on Friday, June 19th, 11am. Assignment 2 will be available on Thursday, June 18th.

Degeneracy

Degeneracy

We call a basic feasible solution x (non)degenerate if (none) some of the basic variables are 0.

- We have seen that the objective value will strictly decrease after one simplex method iteration if the initial BFS is nondegenerate.
- In the degenerate case, the objective can stay the same. ($\leadsto$ We then have $\theta^* = 0$).
- Geometrically: More than n planes intersect in one point in $\mathbb{R}^n$.

$\leadsto$ We can still change the basic index from i (i leaving the basis) to j (j entering the basis) and proceed to the next iteration.

$\leadsto$ Prevent cycles!

If not dealt properly, cycling can happen. Consider the LP:

$$A = \begin{pmatrix} -2 & -9 & 1 & 9 & 1 & 0 \\ 1/3 & 1 & -1/3 & -2 & 0 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$
$$c^\top = (-2, -3, 1, 12, 0, 0).$$

If we set $B = \{5, 6\}$ initially, then the sequence shown below leads to a cycle (objective value doesn't change, and there is always an index with negative reduced cost):

Step #	1	2	3	4	5	6
Exiting	x_6	x_5	x_2	x_1	x_4	x_3
Entering	x_2	x_1	x_4	x_3	x_6	x_5
Basis	$\{5,2\}$	$\{1,2\}$	$\{1,4\}$	$\{3,4\}$	$\{3,6\}$	$\{5,6\}$

We will show that cycles can be avoided by designing how to choose incoming/outgoing basis when there are multiple choices.

Pivoting Rules: Choosing the Entering Basis

In the description of the algorithm, we said that at each feasible point, we can choose any j with negative reduced cost to enter the basis in the next iteration.

Sometimes, there are more than one j with $\bar{c}_j < 0$. In this case, we need to introduce some rules to choose the entering basis.

Here are several possible rules:

1. Smallest Index Rule: Choose the smallest index j with $\bar{c}_j < 0$.
2. Most Negative Rule: Choose the smallest $\bar{c}_j$.
3. Most Decrement Rule: Choose j with the smallest $\theta^* \bar{c}_j$.

Recall that

$$\theta^* = \min_{\{i \in B \mid d_i < 0\}} -\frac{x_i}{d_i}.$$

We choose one index that attains this minimum to leave the basis.

It is possible that there are two or more indices that attain the minimum (tie). Then we also need a rule to decide the new basis.

- The most commonly used rule is the smallest index rule.

When this tie happens, the next BFS will be degenerate. (Why?)

Theorem: Bland's Rule

If we use the smallest index rule for choosing both the entering basis and the exiting basis, then no cycle will occur in the simplex algorithm.

- Using the Bland's rule when applying the simplex method, we can guarantee to stop within a finite number of iterations at an optimal solution.

Finding an Initial BFS

So far, we assumed that we start with a certain BFS:

- This can be done easily if the standard form is derived by adding slacks to each constraint. (Why?)

However, in general, it is not necessarily easy to get an initial BFS from the standard form. For example:

$$\begin{array}{rrrrrcr}
\text{minimize} & x_1 & +x_2 & +x_3 & & & \\
\text{subject to} & x_1 & +2x_2 & +3x_3 & & = & 3 \\
& & -4x_2 & -9x_3 & & = & -5 \\
& & & +3x_3 & +x_4 & = & 1 \\
& x_1 & ,x_2 & ,x_3 & ,x_4 & \geq & 0
\end{array}$$

Finding an Initial BFS

- One could test different bases B to see if $A_B^{-1}b \geq 0$.

$\rightsquigarrow$ This may take a long time!

- In fact, in terms of computational complexity (which we will define later), finding a BFS is as hard as finding the optimal solution!

We will discuss an initialization method next – two-phase method.

Two-Phase Simplex Method

In the two-phase simplex method, we first solve an auxiliary problem ($\mathbb{1}$ is an all-one vector):

$$\begin{aligned} \text{minimize}_{x,y} \quad & \mathbb{1}^\top y \\ \text{subject to} \quad & Ax + y = b \\ & x, y \geq 0 \end{aligned}$$

Without loss of generality, we can assume $b \geq 0$ (otherwise, we pre-multiply that row by -1).

- Trivial BFS for this auxiliary problem: $x = 0, y = b \geq 0$.
- We can apply the Simplex method to solve it.

Theorem: Feasibility

The original problem is feasible if and only if the optimal value of the auxiliary problem is 0.

Hence, we can solve the auxiliary problem by the simplex method:

1. If the optimal value is not 0, then we can claim that the original problem is infeasible.
2. If the optimal value is 0 with solution $(x^*, 0)$, then x^* must be a BFS of the auxiliary problem. But then it must be a BFS for the original problem as well.

$\rightsquigarrow$ We can start from this point to initialize the simplex method.

If x^* is degenerate (has less then m positive entries) and still contains basic indices in the auxiliary variables, then one can pick any other columns (such that they form an independent matrix) in the non-basic part in x^* to make it a BFS for the original problem.

Procedure of the Two-Phase Method

Phase I:

1. Construct the auxiliary problem such that $b \geq 0$.
2. Solve the auxiliary problem using the simplex method.
   - If we reach an optimal solution with optimal value greater than 0, then the original problem is infeasible.
3. If the optimal value is 0 with optimal solution x^*, then we enter phase II.

Phase II:

Solve the original problem starting from the BFS x^*:

- If x^* is degenerate, then we need to supplement some indices to make it a BFS for the original problem

The Big-M Method

There is another method that can be used to solve LP without a starting BFS. Consider the following auxiliary problem:

$$\begin{array}{rc} \text{minimize} & c^\top x + M \sum_{i=1}^m y_i \\ \text{subject to} & Ax + y = b \\ & x, y \geq 0 \end{array}$$

This problem has an initial BFS $y = b \geq 0$ (assuming $b \geq 0$).

$\rightsquigarrow$ We can use the simplex method to solve it.

In the simplex procedure, pretend that M is a very large value (larger than any specified number):

- If the original problem is feasible, then the optimal solution will not involve basic indices in y.
- The two-phase approach is more common.

The Simplex Tableau

New Goal:

- We want to have a simpler implementation of the simplex method

$\rightsquigarrow$ Avoid explicit matrix inversion in the calculations!

We are going to introduce the simplex tableau, which is a practical way to implement the simplex method.

- The simplex tableau maintains a table of numbers.
- It visualizes the procedures of the simplex algorithm and facilitates the computation.
- After learning the simplex tableau, one should be able to solve small-sized linear optimization problems by hand.

Simplex Tableau

- Let B the current basis with basis matrix A_B and basis objective coefficients c_B.
- The simplex tableau is a table with the following structure:

$c^\top - c_B^\top A_B^{-1} A$	$-c_B^\top A_B^{-1} b$
$A_B^{-1} A$	$A_B^{-1} b$

We now take a closer look at what each part of the tableau means (and looks like) and how we can update the tableau efficiently in each iteration.

$c^\top - c_B^\top A_B^{-1} A$	$-c_B^\top A_B^{-1} b$
$A_B^{-1} A$	$A_B^{-1} b$

The lower part of the tableau can be viewed as a transformation of the constraint $Ax = b$ to $A_B^{-1} Ax = A_B^{-1} b$:

- This is equivalent to the original constraint.

Writing $A = [A_B, A_N]$, we have

$$A_B^{-1} A = [I, A_B^{-1} A_N].$$

- This block must contain an identity matrix.

When the basis is B, the current basic feasible solution is

$$x = [x_B; x_N] = [A_B^{-1} b; 0].$$

Therefore, the lower right corner yields the current BFS.

The term

$$c^\top - c_B^\top A_B^{-1} A$$

is the reduced cost at this basis.

- Recall that the reduced costs for basic variables are zero. Therefore, this block contains zeros for the basic indices

Lastly, the term

$$-c_B^\top A_B^{-1} b = -c_B^\top x_B$$

is the negative of the objective value at this basis.

Simplex Tableau

Thus, the simplex tableau should look like (after reordering the columns in $[B; N]$):

$0_m^\top$	$c_N^\top - c_B^\top A_B^{-1} A_N$	$-c_B^\top x_B$
I_m	$A_B^{-1} A_N$	x_B

Here, $0_m \in \mathbb{R}^m$ is a vector of m zeros and I_m is the m-dimensional identity matrix.

This form of an LP is called the canonical form:

- The constraint matrix for the basic variables (not necessarily the first m columns) is an identity matrix.
- The objective function part for the basic variables is zero.

In the canonical form, it is easy to see whether the current BFS is optimal or not:

- ⇝ Consider the reduced costs in the top row of the tableau.
  - If in a simplex tableau, the numbers in the top row are all nonnegative, then we have reached optimality!

-1	-2	0	0	0	0
1	0	1	0	0	100
0	2	0	1	0	200
1	1	0	0	1	150

  - Are we in an optimal solution? ⇝ No.

Simplex Tableau

The simplex tableau will maintain a canonical form for each iteration until we reach optimality:

- For now we assume that we already have a canonical form to start with (we have discussed how to find the initial basic solution).

Task: We want to map each algebraic step of the simplex method to an operation on the simplex tableau. Here are the key steps:

1. Compute the reduced costs.
2. Choose the incoming basic index.
3. Compute θ^* and choose the outgoing basic index.
4. Update the tableau with the new basis.

We call the procedure of transforming from one canonical form (one set of basis) to another one pivoting.

In the canonical form, the reduced costs are simply the coefficients in the top row.

- We can choose any column with negative reduced cost to be the incoming basic index.

Consider the example (production plan):

-1	-2	0	0	0	0
1	0	1	0	0	100
0	2	0	1	0	200
1	1	0	0	1	150

We can choose either the first or second column as the entering basic index (if we use Bland's rule, then we choose the first one).

Pivoting Step II: θ^* and Outgoing Basis

Assume we have chosen column j as the incoming basis.

We need to make sure that the next BFS is still feasible (≥ 0). The step size θ^* was determined via:

$$\theta^* = \min_{d_i < 0, i \in B} -x_i / d_i$$

where x_i is the ith entry of the basic solution and $d_B = -A_B^{-1} A_j$.

In the simplex tableau, this is equivalent to

$$\theta^* = \min_i \left\{ \frac{\bar{b}_i}{\bar{A}_{ij}} : \bar{A}_{ij} > 0 \right\}$$

where $\bar{b}$ is the lower right column and $\bar{A}$ is the lower left part of the tableau.

- This is called the Minimal Ratio Test (MRT).

Pivoting Step II: θ^* and Outgoing Basis

If $\bar{A}_{ij} \leq 0$ for all i, then the problem is unbounded. ($\rightsquigarrow d \geq 0$).

Otherwise, assume index i achieves the minimum in:

$$\theta^* = \min_i \left\{ \frac{\bar{b}_i}{\bar{A}_{ij}} : \bar{A}_{ij} > 0 \right\}$$

- Then the column in the current basis whose ith element is 1 is the outgoing basis. We call row i the pivot row.

Example: If we choose column 1 to be the incoming basis:

B	-1	-2	0	0	0	0
3	1	0	1	0	0	100
4	0	2	0	1	0	200
5	1	1	0	0	1	150

Then, MRT will choose the third column to be the outgoing basis.

Example – Continued

B	-1	-2	0	0	0	0
3	1	0	1	0	0	100
4	0	2	0	1	0	200
5	1	1	0	0	1	150

What if we choose column 2 to be the incoming basis?

- The MRT will choose the fourth column as outgoing basis.
- The MRT finds that the second row achieves the minimum ratio. Then we choose the basis whose second row element is 1 to be the outgoing basis, which is column 4.

Terminologies

- We call the incoming column the pivot column.
- We call the row that achieves the MRT the pivot row (determines the outgoing basis).
- The intersection element of the pivot column and the pivot row is called the pivot element.

B	-1	-2	0	0	0	0
3	1	0	1	0	0	100
4	0	2	0	1	0	200
5	1	1	0	0	1	150

Assume we have determined the incoming and outgoing basis (pivot element $\bar{A}_{ij}$).

We perform the following two steps:

1. Divide each element in the pivot row by the pivot element.
2. Add proper multiples (could be negative) of the pivot row (after the first step) to each other rows, including the top row of objective coefficients, such that all other elements in the pivot column become zeros (including the top row).

- Both operations include the right-hand-side column of $\bar{b}$.

Afterwards, the new pivot column should be $(0; ...; 0; 1; 0; ...; 0)$ with 1 at the pivot row.

- The new resulting tableau will still be in a canonical form, however, with the new basis.

Simplex Method in the Tableau

We have shown how to get from one canonical form to another, we then repeat this procedure until we reach optimality.

- When choosing the incoming and the outgoing basis, we use the smallest index rule.
- This will guarantee that the simplex iterations will terminate in a finite number of steps.

We also attach the index of the basis to the left of the tableau to indicate the current basis (just for clarity).

Example: Step I

Consider the production example. The initial simplex tableau is ($B = \{3, 4, 5\}$):

B						
B	-1	-2	0	0	0	0
3	1	0	1	0	0	100
4	0	2	0	1	0	200
5	1	1	0	0	1	150

We use the smallest index rule. The pivot column (incoming basis) is the first column, the pivot row is the first row (outgoing basis is column 3), the pivot element is 1 (in yellow).

- Divide the pivot row by the pivot element.
- Add proper multiples of row 1 to other rows (including the top row) such that all other elements in the new pivot column (except the pivot element) become zero.

Example: Step II

The tableau becomes:

B	0	-2	1	0	0	100
1	1	0	1	0	0	100
4	0	2	0	1	0	200
5	0	1	-1	0	1	50

- It is not optimal since there is one negative reduced cost.
- The only choice for the pivot column is column 2.
- Use MRT, the pivot row should be row 3 (outgoing basis is column 5).

We apply the same procedure:

- Add $2\times$ row 3 to the very top row, and $-2\times$ row 3 to the second row in the constraint.

Example: Step II

The tableau becomes:

B						
B	0	-2	1	0	0	100
1	1	0	1	0	0	100
4	0	2	0	1	0	200
5	0	1	-1	0	1	50

- It is not optimal since there is one negative reduced cost.
- The only choice for the pivot column is column 2.
- Use MRT, the pivot row should be row 3 (outgoing basis is column 5).

We apply the same procedure:

- Add $2\times$ row 3 to the very top row, and $-2\times$ row 3 to the second row in the constraint.

The tableau becomes:

B						
B	0	-2	1	0	0	100
1	1	0	1	0	0	100
4	0	2	0	1	0	200
5	0	1	-1	0	1	50

- It is not optimal since there is one negative reduced cost.
- The only choice for the pivot column is column 2.
- Use MRT, the pivot row should be row 3 (outgoing basis is column 5).

We apply the same procedure:

- Add $2\times$ row 3 to the very top row, and $-2\times$ row 3 to the second row in the constraint.

Example: Step III

The tableau becomes:

B						
B	0	0	-1	0	2	200
1	1	0	1	0	0	100
4	0	0	2	1	-2	100
2	0	1	-1	0	1	50

- It is still not optimal since there is one negative reduced cost.
- The only choice for the pivot column is column 3.
- Use MRT, the pivot row should be row 2 (outgoing basis is column 4).

We apply the same procedure again:

- Divide row 2 by 2, then add $1\times$ row 2 to the very top row, add $-1\times$ row 2 to the first row in the constraint, add $1\times$ row 2 to the last row.

Example: Step III

The tableau becomes:

B	0	0	-1	0	2	200
1	1	0	1	0	0	100
4	0	0	2	1	-2	100
2	0	1	-1	0	1	50

- It is still not optimal since there is one negative reduced cost.
- The only choice for the pivot column is column 3.
- Use MRT, the pivot row should be row 2 (outgoing basis is column 4).

We apply the same procedure again:

- Divide row 2 by 2, then add $1\times$ row 2 to the very top row, add $-1\times$ row 2 to the first row in the constraint, add $1\times$ row 2 to the last row.

Example: Step III

The tableau becomes:

B						
B	0	0	-1	0	2	200
1	1	0	1	0	0	100
4	0	0	2	1	-2	100
2	0	1	-1	0	1	50

- It is still not optimal since there is one negative reduced cost.
- The only choice for the pivot column is column 3.
- Use MRT, the pivot row should be row 2 (outgoing basis is column 4).

We apply the same procedure again:

- Divide row 2 by 2, then add $1\times$ row 2 to the very top row, add $-1\times$ row 2 to the first row in the constraint, add $1\times$ row 2 to the last row.

Example: Step IV

The tableau becomes:

B						
B	0	0	0	1/2	1	250
1	1	0	0	-1/2	1	50
3	0	0	1	1/2	-1	50
2	0	1	0	1/2	0	100

All the reduced costs are positive now:

- Thus it is optimal!
- The optimal solution is $(50, 100, 50, 0, 0)$ with optimal value -250.

Another Example

Consider the linear optimization problem:

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

First, we write down the standard form:

$$\begin{array}{lccccccc} \text{minimize} & -10x_1 & -12x_2 & -12x_3 & & & & \\ \text{s.t.} & x_1 & +2x_2 & +2x_3 & +s_1 & & & = 20 \\ & 2x_1 & +x_2 & +2x_3 & & +s_2 & & = 20 \\ & 2x_1 & +2x_2 & +x_3 & & & +s_3 & = 20 \\ & x_1, & x_2, & x_3, & s_1, & s_2, & s_3 & \geq 0 \end{array}$$

We write down the initial tableau ($B = \{4, 5, 6\}$):

B							
	-10	-12	-12	0	0	0	0
4	1	2	2	1	0	0	20
5	2	1	2	0	1	0	20
6	2	2	1	0	0	1	20

- This is also in a canonical form already.

By the smallest index rule, we choose column 1 to enter the basis. By the minimum ratio test, we have two candidates to leave the basis: 5th column (row 2) or 6th column (row 3).

By the smallest index rule again, we choose 5th column to exit (pivot row is row 2):

- Divide each element in row 2 by 2.
- Add $10\times$ new row 2 to the top row, $-1\times$ new row 2 to the first constraint row, and $-2\times$ new row 2 to the last row.

Simplex Algorithm: Step I

We write down the initial tableau ($B = \{4, 5, 6\}$):

B							
B	-10	-12	-12	0	0	0	0
4	1	2	2	1	0	0	20
5	2	1	2	0	1	0	20
6	2	2	1	0	0	1	20

- This is also in a canonical form already.

By the smallest index rule, we choose column 1 to enter the basis. By the minimum ratio test, we have two candidates to leave the basis: 5th column (row 2) or 6th column (row 3).

By the smallest index rule again, we choose 5th column to exit (pivot row is row 2):

- Divide each element in row 2 by 2.
- Add $10\times$ new row 2 to the top row, $-1\times$ new row 2 to the first constraint row, and $-2\times$ new row 2 to the last row.

Simplex Algorithm: Step I

We write down the initial tableau ($B = \{4, 5, 6\}$):

B							
B	-10	-12	-12	0	0	0	0
4	1	2	2	1	0	0	20
5	2	1	2	0	1	0	20
6	2	2	1	0	0	1	20

- This is also in a canonical form already.

By the smallest index rule, we choose column 1 to enter the basis. By the minimum ratio test, we have two candidates to leave the basis: 5th column (row 2) or 6th column (row 3).

By the smallest index rule again, we choose 5th column to exit (pivot row is row 2):

- Divide each element in row 2 by 2.
- Add $10\times$ new row 2 to the top row, $-1\times$ new row 2 to the first constraint row, and $-2\times$ new row 2 to the last row.

Simple Algorithm: Step II

Then the tableau becomes:

B							
B	0	-7	-2	0	5	0	100
4	0	3/2	1	1	-1/2	0	10
1	1	1/2	1	0	1/2	0	10
6	0	1	-1	0	-1	1	0

Column 2 is the pivot column. By MRT, the pivot row is row 3.

- Here we encounter a degenerate case with minimal ratio 0.
- It means that in this pivoting, we can not strictly improve the objective value.
- But we can still proceed as normal (no cycle will occur if we use the Bland's rule):
  - We add $7\times$ row 3 to the top row, $-3/2\times$ row 3 to the first constraint row and $-1/2\times$ row 3 to the second constraint row.

Simple Algorithm: Step II

Then the tableau becomes:

B							
B	0	-7	-2	0	5	0	100
4	0	3/2	1	1	-1/2	0	10
1	1	1/2	1	0	1/2	0	10
6	0	1	-1	0	-1	1	0

Column 2 is the pivot column. By MRT, the pivot row is row 3.

- Here we encounter a degenerate case with minimal ratio 0.
- It means that in this pivoting, we can not strictly improve the objective value.
- But we can still proceed as normal (no cycle will occur if we use the Bland's rule):
  - We add $7\times$ row 3 to the top row, $-3/2\times$ row 3 to the first constraint row and $-1/2\times$ row 3 to the second constraint row.

Then the tableau becomes:

B							
B	0	-7	-2	0	5	0	100
4	0	3/2	1	1	-1/2	0	10
1	1	1/2	1	0	1/2	0	10
6	0	1	-1	0	-1	1	0

Column 2 is the pivot column. By MRT, the pivot row is row 3.

- Here we encounter a degenerate case with minimal ratio 0.
- It means that in this pivoting, we can not strictly improve the objective value.
- But we can still proceed as normal (no cycle will occur if we use the Bland's rule):
  - We add $7\times$ row 3 to the top row, $-3/2\times$ row 3 to the first constraint row and $-1/2\times$ row 3 to the second constraint row.

Then the tableau becomes:

B							
	0	0	-9	0	-2	7	100
4	0	0	5/2	1	1	-3/2	10
1	1	0	3/2	0	1	-1/2	10
2	0	1	-1	0	-1	1	0

We choose column 3 to enter the basis. By MRT, the pivot row is row 1 (column 4 leaving basis).

- We multiply 2/5 to each number in row 1, then add $9\times$ row 1 to the top row, $-3/2\times$ row 1 to the second constraint row and $1\times$ row 1 to the last row.

Then the tableau becomes:

B							
B	0	0	-9	0	-2	7	100
4	0	0	5/2	1	1	-3/2	10
1	1	0	3/2	0	1	-1/2	10
2	0	1	-1	0	-1	1	0

We choose column 3 to enter the basis. By MRT, the pivot row is row 1 (column 4 leaving basis).

- We multiply 2/5 to each number in row 1, then add $9\times$ row 1 to the top row, $-3/2\times$ row 1 to the second constraint row and $1\times$ row 1 to the last row.

Then the tableau becomes:

B							
	0	0	-9	0	-2	7	100
4	0	0	5/2	1	1	-3/2	10
1	1	0	3/2	0	1	-1/2	10
2	0	1	-1	0	-1	1	0

We choose column 3 to enter the basis. By MRT, the pivot row is row 1 (column 4 leaving basis).

- We multiply $2/5$ to each number in row 1, then add $9\times$ row 1 to the top row, $-3/2\times$ row 1 to the second constraint row and $1\times$ row 1 to the last row.

Then the tableau becomes:

B	0	0	0	18/5	8/5	8/5	136
3	0	0	1	2/5	2/5	-3/5	4
1	1	0	0	-3/5	2/5	2/5	4
2	0	1	0	2/5	-3/5	2/5	4

This is optimal since all reduced costs are non-negative. The optimal solution is $(4, 4, 4, 0, 0, 0)$ with optimal value -136.

Questions?