Optimization Algorithms and the LP Methods

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

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

Chapters 4.2, 4.5, 5.3

Problem Classes

- Distinctions between optimization problems stem from
  - differentiable versus non-differentiable functions;
  - unconstrained versus constrained variables;
  - one-dimensional versus multi-dimensional variables;
  - convex versus non-convex minimization.
- Algorithms can be divided as
  - Finite versus convergent iterative methods: algorithms obtain a solution in a finite number of iterations; or instead that are convergent—generate a sequence of trial or approximate solutions that converge to an exact “solution.”
  - 0-order, first-order versus second-order methods: algorithms are based on just function values, or in addition first-order derivatives of functions, or in addition the second-order derivatives.

The Meaning of a "Solution"

In fact, there are several possibilities for defining what an optimal solution is. Once the definition is chosen, there must be a way of testing whether or not a given solution met the definition.

Typically, one seeks a local minimizer; or ideally, one seeks a global minimizer. But these tasks are generally harder since the validation is already difficult.

Therefore, in most cases, algorithms seek a KKT solution together with its multipliers as they can be tested effectively, either the first-order or second-order optimality conditions.

For convex optimization, a KKT solution suffices! In fact, a KKT solution may also suffice for some special nonconvex optimization with a high probability. More importantly, it seems to work in practice most of times.

Iterative Algorithms

Optimization algorithms tend to be iterative procedures.

Starting from a given point/solution $\mathbf{x}^0$, they generate a sequence $\{\mathbf{x}^k, k = 1, 2, \ldots\}$ of iterates (or trial solutions) that can be feasible or infeasible. For constrained problems, the sequence is associated with the Lagrange multiplier sequence $\{\mathbf{y}^k, k = 1, 2, \ldots\}$. Hopefully, the limit of the sequence meet the optimality conditions, and it converges fast (convergence speed).

We study algorithms that produce iterates according to well determined rules–Deterministic Algorithm rather than some random selection process–Randomized Algorithm.

The rules to be followed and the procedures that can be applied depend to a large extent on the characteristics of the problem to be solved.

Search Direction and Step Size

The iterative scheme is of the form

$$\mathbf{x}^{k+1} = \mathbf{x}^{k} + \alpha_k \mathbf{d}^{k}$$

where $\mathbf{d}^k$ is a search direction vector, both feasible and descent, and scalar α_k is called the step-size or step-length. One popular choice is $\mathbf{d}^k = -\nabla f(\mathbf{x}^k)$ – the negative (reduced-)gradient vector.

For constrained problems, we also update multipliers/dual-variables: $\mathbf{y}^{k+1} \leftarrow \mathbf{y}^{k}$ according to some rules.

The key is that once $\mathbf{x}^k$ is known, then $\mathbf{d}^k$ is chosen as some function of $(\mathbf{x}^k,\mathbf{y}^k)$, $(\mathbf{x}^{k-1},\mathbf{y}^{k-1})$… and the scalar α_k may be chosen in accordance with some line (one-dimensional optimization) search rules.

Indeed, once the search direction is chosen, the objective function can be written as $\phi(\alpha) := f(\mathbf{x}^k + \alpha\, \mathbf{d}^k)$, which is just function of α. Therefore, α is chosen such as the new iterate remains feasible and the objective is reduced the most.

Recall Descent Directions at a BFS of LP

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b, \\ & x \geq 0 \end{aligned}$$

Recall at a BFS: $A_B x_B + A_N x_N = \mathbf{b}$, with $x_B \geq 0$ and $x_N = 0$. Then we can express x_B in terms of x_N, $x_B = (A_B)^{-1} b - (A_B)^{-1} A_N x_N$.

Then, $c^T x = c^T_B x_B + c^T_N x_N = (c^T_N - c^T_B (A_B)^{-1} A_N)\, x_N + c^T_B (A_B)^{-1} b$ and increase any one variable of x_N is an extreme feasible direction.

Thus, for the BFS to be optimal, any (extreme) feasible direction must be an ascent direction, or $r_N = (c^T_N - c^T_B (A_B)^{-1} A_N) \geq 0$ is necessary and sufficient for the current BFS being optimal!

This vector $r = (c^T - y^T A)$ and $y^T = c^T_B (A_B)^{-1}$ are called the reduced cost (or reduced gradient) and shadow/dual price vectors for the current BFS. Note that reduced cost coefficients for basic variables are all zeros. If anyone of r_N is negative, then an improving (extreme) feasible direction is found by increasing the corresponding non-basic variable value.

In the LP production example, suppose the basic variable set B = {3, 4, 5} and N={1,2}.

$$
\begin{array}{llllllll}
\min & -x_1 & -2x_2 & & & & \\
\text{s.t.} & x_1 & & +x_3 & & & = 1 \\
 & & x_2 & & +x_4 & & = 1 \\
 & x_1 & +x_2 & & & +x_5 & = 1.5 \\
 & x_1, & x_2, & x_3, & x_4, & x_5 & \geq 0.
\end{array}
$$

$$
c_N = \begin{pmatrix} -1 \\ -2 \end{pmatrix}, c_B = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, A_B = I, A_N = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix},
$$

$$
A_B^{-1} = I, y^T = (0 \quad 0 \quad 0), \quad r_N^T = (-1 \quad -2).
$$

Thus, one can increase either x_1 or x_2 to reduce the objective function value

Improving Direction and Step Size

Let us increased the value of x_2 (the entering/incoming variable) that would reduce the objective value while keep other non-basic variables unchanged. Then how much increase such that the solution stay feasible or the current basic variables remain nonnegative:

$$
\begin{array}{lllll}
 & +x_3 & & & = 1 \\
x_2 & & +x_4 & & = 1 \\
+x_2 & & & +x_5 & = 1.5
\end{array}
$$

In general, it would be the maximal possible value increase of the selected non-basic variable x_e such that $\boldsymbol{x_B} = \boxed{(\boldsymbol{A_B})^{-1}\boldsymbol{b}} - (\boldsymbol{A_B})^{-1}\boldsymbol{A_e}x_e$ remains nonnegative where $\boldsymbol{A_e}$ is the incoming column.

In this particular example, we will have

$x_2=1$, and $x_3=1, x_4=0, x_5=0.5$ (can be done by a min-ratio-test)

Then we reach a new BFS with basic variable set $B=\{3,2,5\}$ where x_2 is incoming variable and x_4 outgoing variable.

$$\begin{array}{llllllll} \textbf{min} & -x_1 & -2x_2 & & & & & \\ \textbf{s.t.} & x_1 & & +x_3 & & & = 1 \\ & & x_2 & & +x_4 & & = 1 \\ & x_1 & +x_2 & & & +x_5 & = 1.5 \end{array}$$

New basis B={3,2,5} and N={1,4}

$$A_B = \begin{array}{ccc} x_3 & x_2 & x_5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{array} \Rightarrow A^{-1}_B = \begin{array}{ccc} x_3 & x_2 & x_5 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{array} \qquad A_N = \begin{array}{cc} x_1 & x_4 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{array}$$

$\mathbf{x}^T_B=(1\ \ 1\ \ 0.5)$

$\mathbf{c}^T_B=(0\ \ -2\ \ 0)$ $\qquad\qquad$ $\mathbf{c}^T_N=(-1\ \ 0)$

$\mathbf{y}^T=\mathbf{c}^T_B A^{-1}_B=(0\ \ -2\ \ 0)$ $\qquad\qquad$ $\mathbf{r}^T_N=\mathbf{c}^T_N-\mathbf{y}^T A_N=(-1\ \ 2)$

Not optimal yet, x_1 would be incoming/entering.
Now compute $A^{-1}_B A_1=(1\ \ 0\ \ 1)^T$, and
do min-ratio-test $(1\ \ 1\ \ 0.5)./(1\ \ 0\ \ 1)_+=(1\ \ \infty\ \ 0.5)$

Thus $x_1=0.5$ (incoming) and $x_5=0$ (outgoing) with the new basis
B={3,2,1} and N={4,5} <= One can check it is Optimal

Summary of the Simplex Method

1. Initialize: with a minimization problem with respect to a BFS with basis index set B and let N denote the rest index set:

$$\mathbf{x}_B = (A_B)^{-1}\boldsymbol{b}(\geq \mathbf{0}),\ \mathbf{x}_N = \mathbf{0}$$

2. Pricing: Compute the corresponding shadow-price vector **y** and the reduced vector **r**:

$$\mathbf{y}^T = \mathbf{c}^T{}_B(A_B)^{-1} \text{ or solve } \mathbf{y}^T A_B = \mathbf{c}^T{}_B \text{, then let } \boldsymbol{r}_N = \boldsymbol{c}^T{}_N - \boldsymbol{y}^T A_N$$

and find (Dantzig or "greedy" rule): $r_e = \min_{j\in N}\{r_j\}$. (break ties arbitrarily)

3. Test of Termination: If $r_e \geq 0$, Stop -- the solution is already optimal. Otherwise select entering/incoming variable x_e If the vector $(A_B)^{-1}A_e$ contains a positive entry; If not, the objective value is unbounded -- Stop.

4. Step Sizing: Perform the Min-Ratio-Test to determine the step size:

$$\alpha = \min\{\ \mathbf{x}_B\ ./[(A_B)^{-1}A_e]_+\ \}.$$

5. Basis Update: Set $x_e = \alpha$, who is entering the basis, and elect a current basic variables with zero value (break ties arbitrarily) who becomes the outgoing variable; so that we reach a new (adjacent) BFS – Go To Step 1.

Theorem: If the reduced cost coefficient is positive for every nonbasic variable, then the optimal BFS is unique.

Two-Phase Simplex Method for LP

How to determine a starting basic feasible solution (BFS) for general LP?

One technique is constructing a so-called Phase I Problem, and uses the Simplex Method itself to solve the Phase I LP problem for which a starting BFS is known, and for which an optimal basic solution is a BFS for the original LP problem if it's feasible. For example, for the standard equality form with the right-hand-side nonnegative, the Phase-I problem is

$$\min \quad z_1+z_2+\ldots+z_m, \text{ s.t. } \quad A\mathbf{x}+\mathbf{z}=\mathbf{b},\ (\mathbf{x},\mathbf{z}) \geq \mathbf{0}.$$

If Phase I results in the discovery of a BFS for the original problem, then we can initiate Phase II wherein the Simplex Method is applied to the solving the original problem.

The combination of Phases I and II gives rise to the Two-Phase Simplex Method.

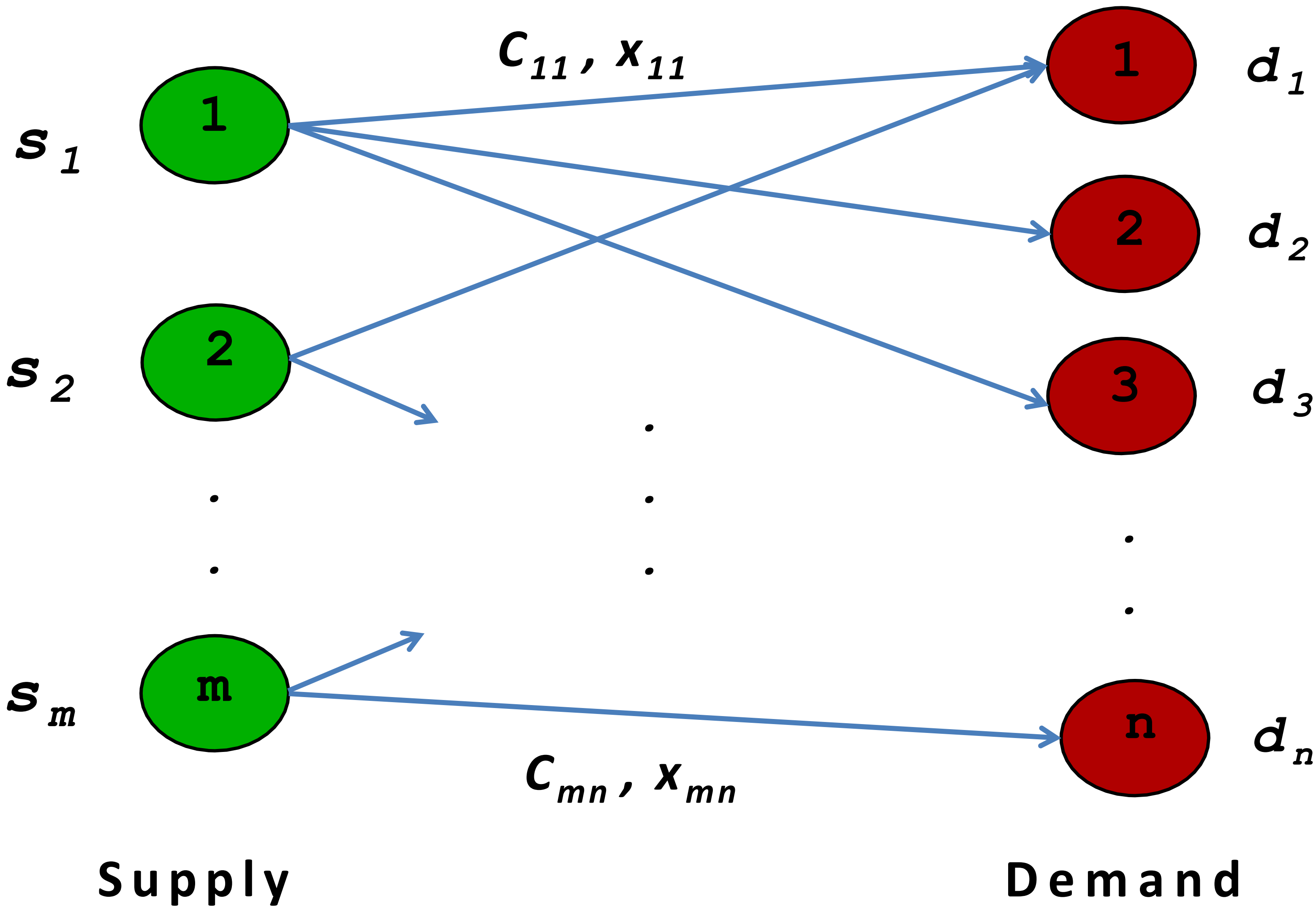
The Transportation Simplex Method

$$\begin{array}{lll}
\min & \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} & \\
\text{s.t.} & \sum_{j=1}^{n} x_{ij} = s_i, & \forall\, i = 1,\ldots,m \\
 & \sum_{i=1}^{m} x_{ij} = d_j, & \forall\, j = 1,\ldots,n \\
 & x_{ij} \geq 0, & \forall\, i, j
\end{array}$$

Assume that the total supply equal the total demand. Thus, exactly one equality constraint is redundant.

At each step the simplex method attempts to send units along a route that is unused (non-basic) in the current BFS, while eliminating one of the routes that is currently being used (basic).

Transportation and Supply Chain Network



The Transportation Data Table

	1	2	3	4	Supply
1	12	13	4	6	500
2	6	4	10	11	700
3	10	9	12	4	800
Demand	400	900	200	500	2000

Transportation Simplex Method: Phase I

1. Start with the cell in the northwest corner cell

2. Allocate as many units as possible, consistent with the available supply and demand.

3. Move one cell to right if there is remaining supply; otherwise, move one cell down.

4. goto Step 2.

				500
				700
				800
400	900	200	500	

North-West Corner Method: Compute a BFS

400				100
				700
				800
0	900	200	500	

North-West Corner Method: Compute a BFS

400	100			0
				700
				800
0	800	200	500	

North-West Corner Method: Compute a BFS

400	100			0
	700			0
				800
0	100	200	500	

North-West Corner Method: Compute a BFS

400	100			0
	700			0
	100			700
0	0	200	500	

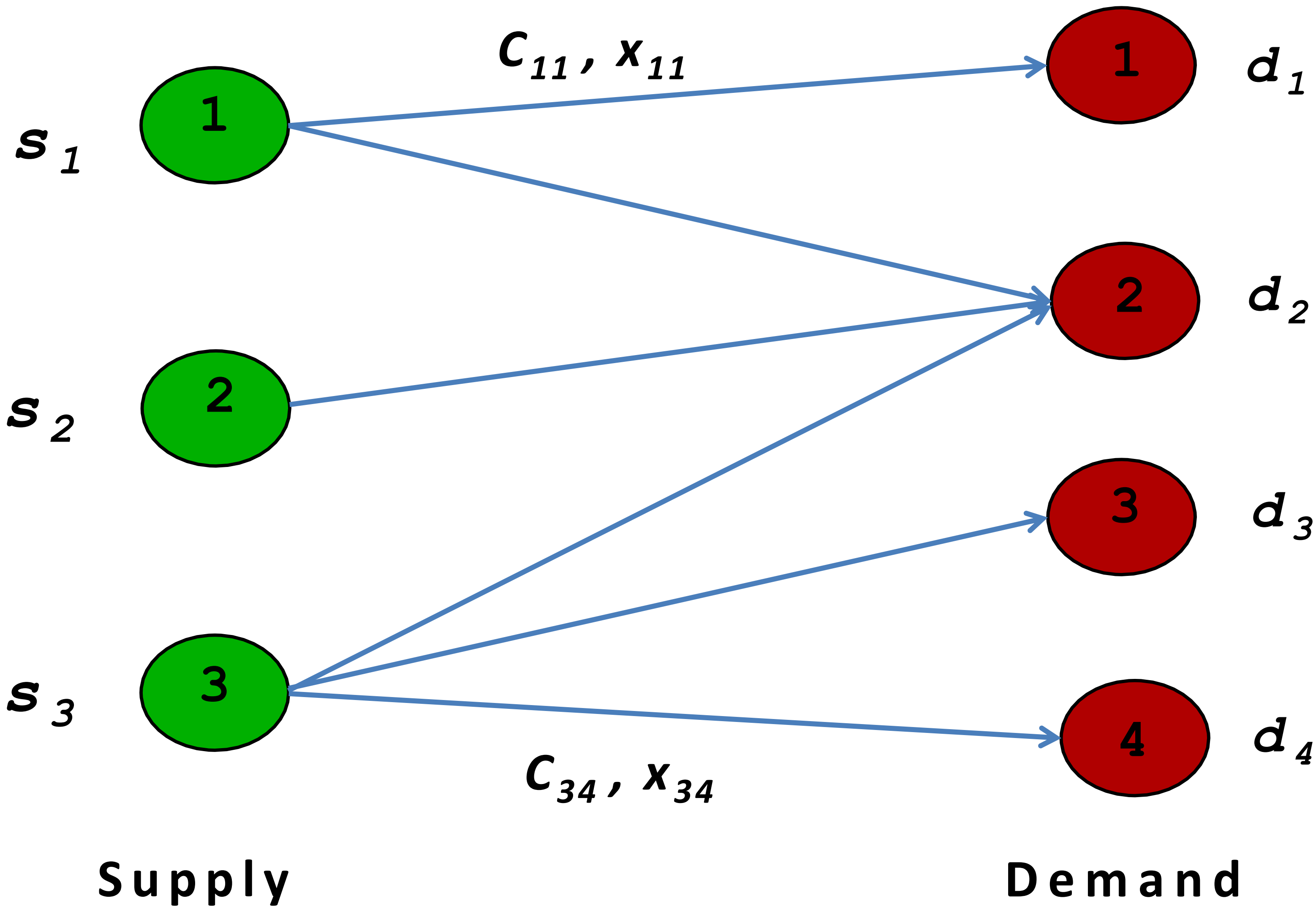
North-West Corner Method: Compute a BFS

400	100			0
	700			0
	100	200		500
0	0	0	500	

North-West Corner Method: Compute a BFS

400	100			0
	700			0
	100	200	500	0
0	0	0	0	

A BFS as a "Tree" Structure in the Network



(Tailored) Transportation Simplex Method: Phase II

1. Determine the shadow prices (for each supply side u_i and each demand side v_j) from every USED cell (basic variable)

$$\mathbf{y}^{\mathbf{T}} = \mathbf{c}^{\mathbf{T}}{}_{B}(A_B)^{-1} \Rightarrow \mathbf{y}^{\mathbf{T}} A_B = \mathbf{c}^{\mathbf{T}}{}_{B} \Rightarrow u_i + v_j = c_{ij}$$

One can always set $v_n = 0$ by viewing the last demand constraint redundant. Then do back-substitution…

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=$
	700 4			0 $u_2=$
	100 9	200 12	500 4	0 $u_3=$
0 $v_1=$	0 $v_2=$	0 $v_3=$	0 $v_4=0$	

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=$
	700 4			0 $u_2=$
	100 9	200 12	500 4	0 $u_3=4$
0 $v_1=$	0 $v_2=$	0 $v_3=$	0 $v_4=0$	

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=$
	700 4			0 $u_2=$
	100 9	200 12	500 4	0 $u_3=4$
0 $v_1=$	0 $v_2=$	0 $v_3=8$	0 $v_4=0$	

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=$
	700 4			0 $u_2=$
	100 9	200 12	500 4	0 $u_3=4$
0 $v_1=$	0 $v_2=5$	0 $v_3=8$	0 $v_4=0$	

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=$
	700 4			0 $u_2=-1$
	100 9	200 12	500 4	0 $u_3=4$
0 $v_1=$	0 $v_2=5$	0 $v_3=8$	0 $v_4=0$	

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=8$
	700 4			0 $u_2=-1$
	100 9	200 12	500 4	0 $u_3=4$
0 $v_1=$	0 $v_2=5$	0 $v_3=8$	0 $v_4=0$	

Step 1: Compute Shadow Prices

400 12	100 13			0 $u_1=8$
	700 4			0 $u_2=-1$
	100 9	200 12	500 4	0 $u_3=4$
0 $v_1=4$	0 $v_2=5$	0 $v_3=8$	0 $v_4=0$	

Transportation Simplex Method: Phase II

1. Determine the shadow prices (for each supply side u_i and each demand side v_j) from every USED cell (basic variable)

$$\mathbf{y}^T = \mathbf{c}^T{}_B(A_B)^{-1} \Rightarrow \mathbf{y}^T A_B = \mathbf{c}^T{}_B \Rightarrow u_i + v_j = c_{ij}$$

One can always set $v_n = 0$ by viewing the last demand constraint redundant; then do back-substitution…

2. Calculate the reduced costs for the UNUSED cells (non-basic variable)

$$r_N = c^T{}_N - y^T A_N \Rightarrow r_{ij} = c_{ij} - u_i - v_j$$

If the reduced cost for every unused cell is nonnegative, then STOP: declare OPTIMAL

Step 2: Compute Reduced Costs

400 12	100 13	4	6	500 $u_1=8$
6	700 4	10	11	700 $u_2=-1$
10	100 9	200 12	500 4	800 $u_3=4$
400 $v_1=4$	900 $v_2=5$	200 $v_3=8$	500 $v_4=0$	2000

$$r_{ij} = c_{ij} - u_i - v_j$$

Step 2: Compute Reduced Costs

400 12 0	100 13 0	4 -12	6 -2	500 $u_1=8$
6 3	700 4 0	10 3	11 12	700 $u_2=-1$
10 2	100 9 0	200 12 0	500 4 0	800 $u_3=4$
400 $v_1=4$	900 $v_2=5$	200 $v_3=8$	500 $v_4=0$	2000

Reduced costs are computed in RED

Transportation Simplex Method: Phase II

1. Determine the shadow prices (for each supply side u_i and each demand side v_j) from every USED cell (basic variable)

$$\mathbf{y}^{\mathbf{T}} = \mathbf{c}^{\mathbf{T}}{}_B(A_B)^{-1} \Rightarrow \mathbf{y}^{\mathbf{T}} A_B = \mathbf{c}^{\mathbf{T}}{}_B \Rightarrow u_i + v_j = c_{ij}$$

One can always set $v_n = 0$ by viewing the last demand constraint redundant; then do back-substitution…

2. Calculate the reduced costs for the UNUSED cells (non-basic variable)

$$r_N = c^T{}_N - y^T A_N \Rightarrow r_{ij} = c_{ij} - u_i - v_j$$

If the reduced cost for every unused cell is nonnegative, then STOP: declare OPTIMAL

3. Select an unused cell with the most negative reduced cost as in-coming. Using a minRT, chain-reaction-cycle, determine the max units (α) that can be allocated to the in-coming cell and adjust the allocation appropriately. Update the values of the new set of USED (basic) cells (a new BFS).

Step 3: Chain Reaction Cycle

400	100	+α		0
	700			0
	100	200	500	0
0	0	0	0	

Step 3: Chain Reaction Cycle

400	100	+α		0
	700			0
	100	200 -α	500	0
0	0	0	0	

Step 3: Chain Reaction Cycle

400	100	+α		0
	700			0
	100 +α	200 -α	500	0
0	0	0	0	

Step 3: Chain Reaction Cycle

400	100 $-\alpha$	$+\alpha$		0
	700			0
	100 $+\alpha$	200 $-\alpha$	500	0
0	0	0	0	

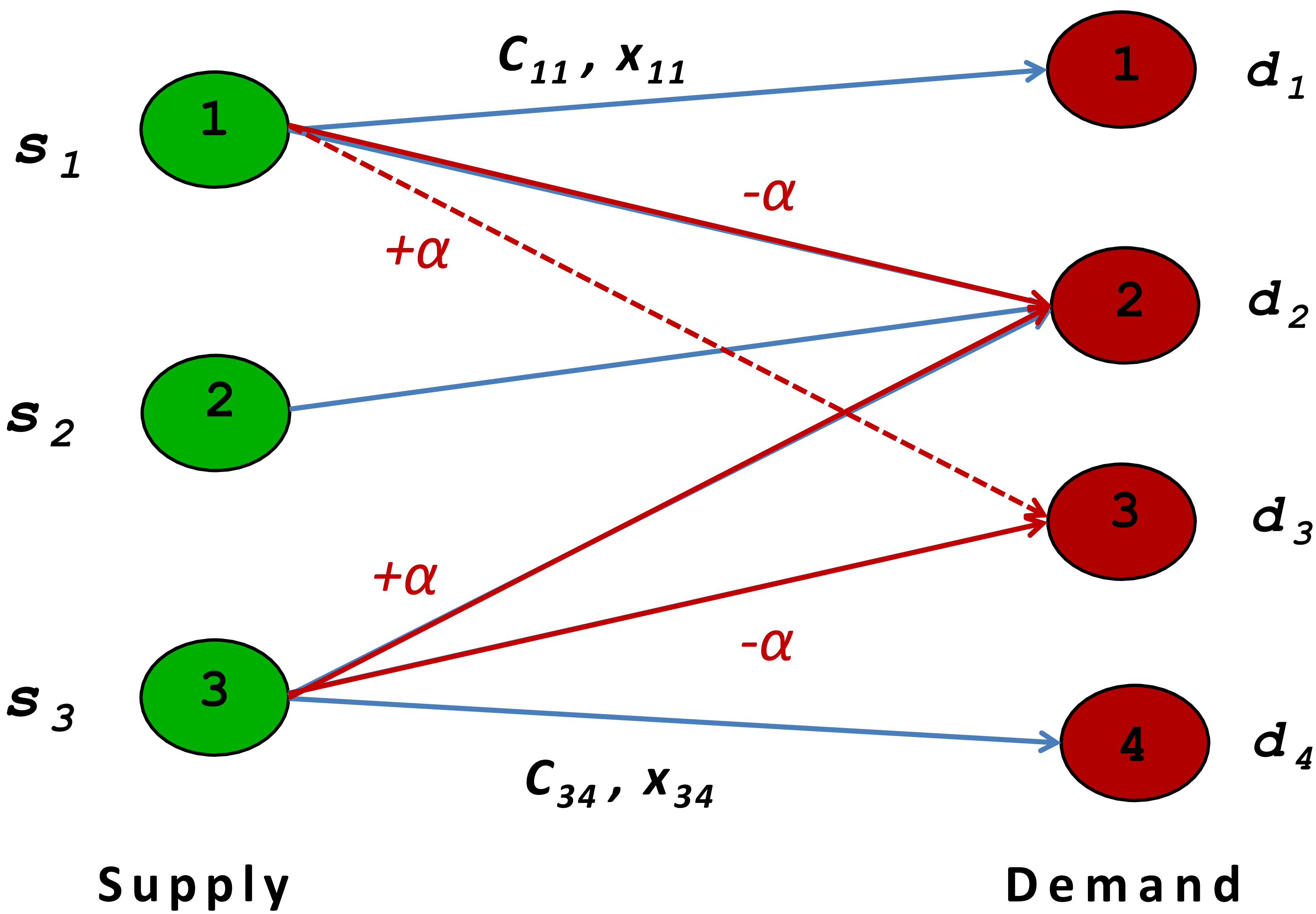
$$\alpha = 100$$

Step 3: Chain Reaction Cycle

400	100 13 -α	4 +α		0
	700			0
	100 9 +α	200 12 -α	500	0
0	0	0	0	

α = 100, and the cost is reduced by 1200

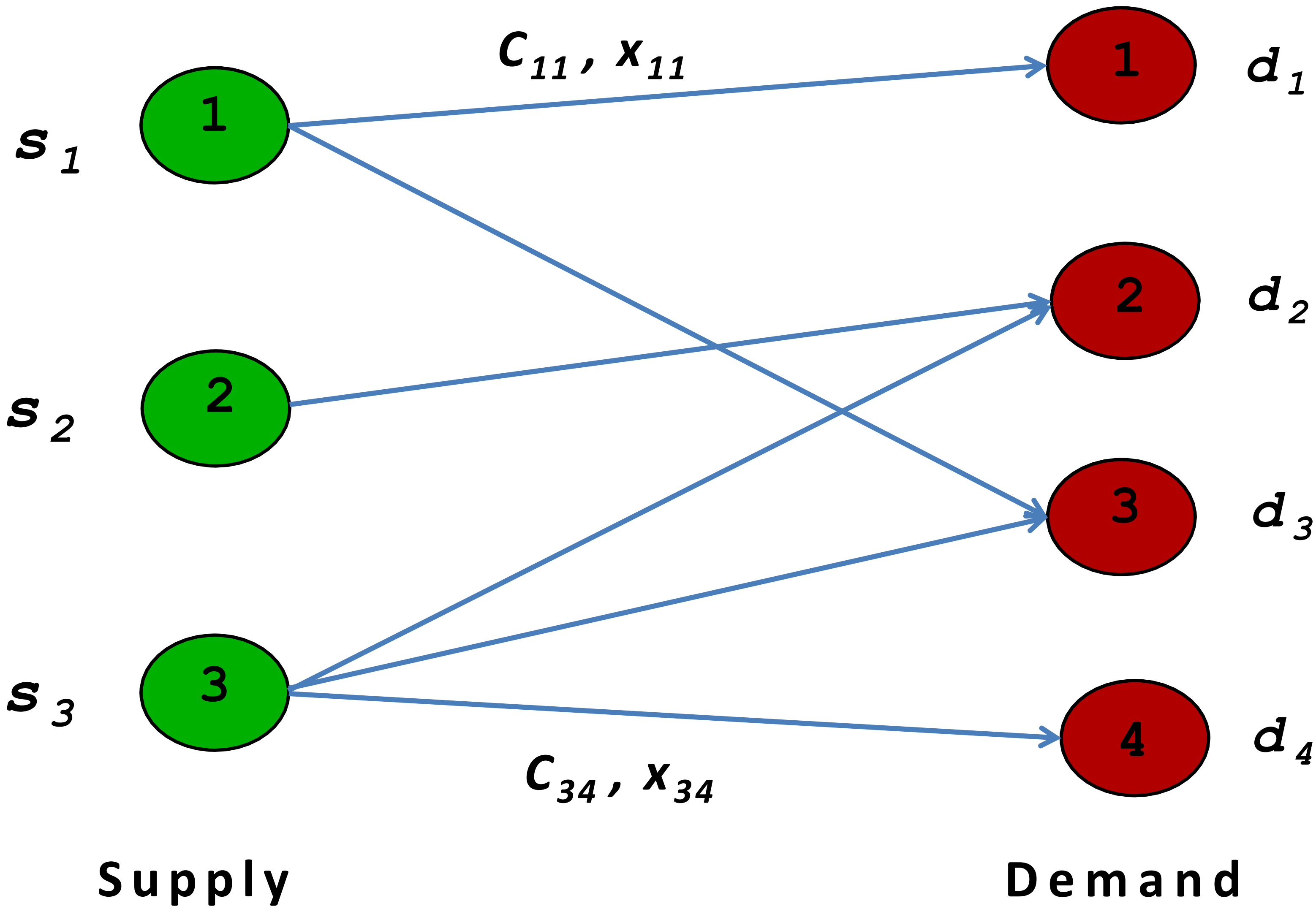
Find the Cycle on the "Tree" Structure



Step 3: Update to the New BFS

400		100		0
	700			0
	200	100	500	0
0	0	0	0	

A New "Tree" Structure in the Network



Transportation Simplex Method: Phase II

1. Determine the shadow prices (for each supply side u_i and each demand side v_j) from every USED cell (basic variable)

$$\mathbf{y}^{\mathbf{T}} = \mathbf{c}^{\mathbf{T}}{}_B(A_B)^{-1} \Rightarrow \mathbf{y}^{\mathbf{T}} A_B = \mathbf{c}^{\mathbf{T}}{}_B \Rightarrow u_i + v_j = c_{ij}$$

One can always set $v_n = 0$ by viewing the last demand constraint redundant; then do back-substitution…

2. Calculate the reduced costs for the UNUSED cells (non-basic variable)

$$r_N = c^T{}_N - y^T A_N \Rightarrow r_{ij} = c_{ij} - u_i - v_j$$

If the reduced cost for every unused cell is nonnegative, then STOP: declare OPTIMAL

3. Select an unused cell with the most negative reduced cost as in-coming. Using the minRT, chain-reaction-cycle, determine the max units (α) that can be allocated to the in-coming cell and adjust the allocation appropriately. Update the values of the new set of USED (basic) cells (a new BFS).

Go to Step 1

Some Issues for the Simplex Method

In theory, one can select any in-coming nonbasic variable as long as its reduced cost is negative.

For maximization problem you really don't need to transform it into a minimization problem: goal is to make reduced profit non-positive

If one of its basic variable has value 0, then BFS is degenerate. Thus, the maximum amount increase, α, equals 0. You may pretend it's $\varepsilon > 0$ but arbitrarily small and continue the transformation process.

Generally, a special care needs to be taken for degenerate cases to avoid possible cycling, that is, no progress can be made and the method never reaches an optimal corner solution.

Degeneracy in BFS

400				400
0 Basic	700			700
	200	200	500	900
400	900	200	500	

You may find $\alpha = 0$ in this case,
but you continue the update…

Cycling Example

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

Initially, the basic variables are $\{x_5, x_6\}$ and it is a degenerate BFS. The simplex method 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

Cycle-Broken Rules

Double Smallest Rule: among the nonbasic variables with the negative reduced cost coefficients, select one with the smallest index to enter; among the basic variables with the smallest ratio, select one with the smallest index to exit

Random Selection Rule: among the nonbasic variables with the negative reduced cost coefficients, randomly select one to enter; among the basic variables with the smallest ratio, randomly select one to exit.

Worst-Case Convergence Speed

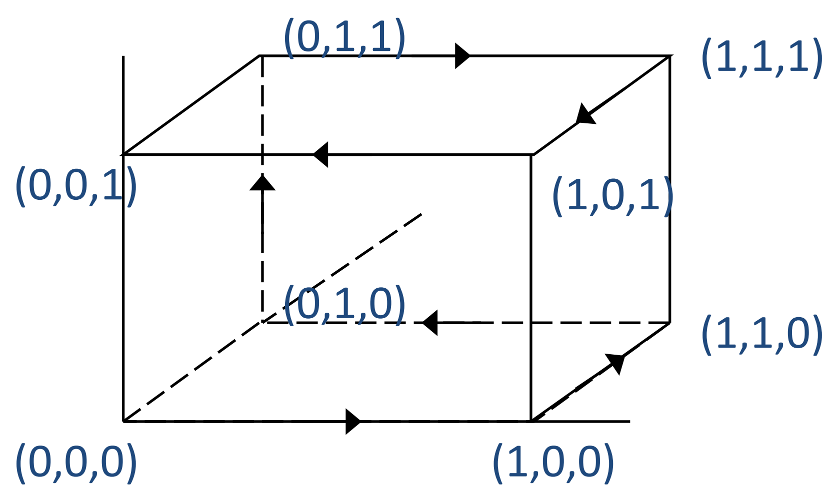
$$
\begin{array}{llllll}
\max & 9x_1 & +3x_2 & +x_3 & & \\
\text{s.t.} & & & & & \\
 & x_1 & & & \leq & 1 \\
 & 6x_1 & +x_2 & & \leq & 9 \\
 & 18x_1 & +6x_2 & +x_3 & \leq & 81 \\
 & x_1 & x_2 & x_3 & \geq & 0
\end{array}
$$

The Simplex method converges in finite number of steps.

The optimal solution is $x_3 = 81$ and $x_1 = x_2 = 0$.

The simplex method, using the greedy rule, needs $2^3 - 1$ steps to reach the optimal solution.

One can extend this (Klee-Mindy) example with n variables such that the simplex method with the greedy rule needs 2^n-1 steps to convergence.

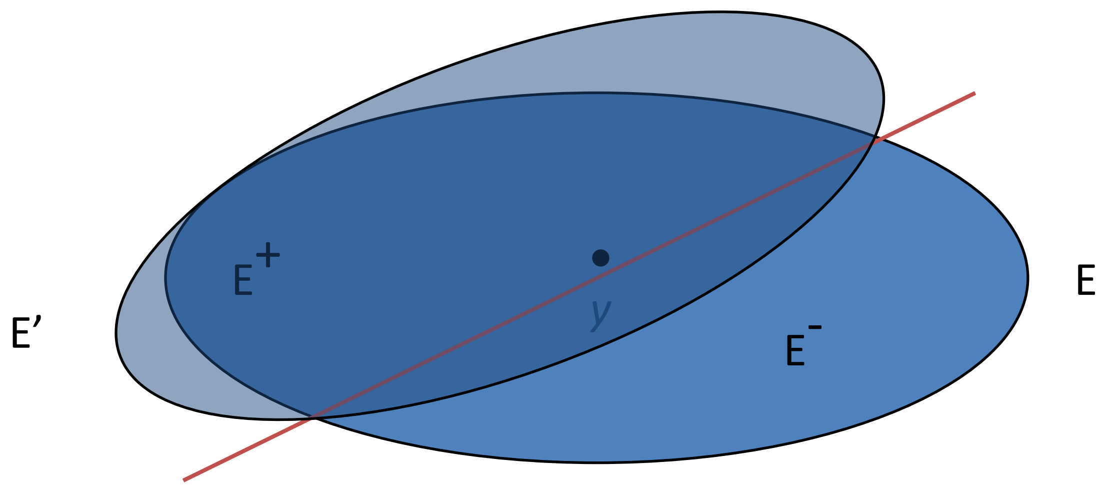


Hamiltonian Path on a 3-Cube

The Ellipsoid Method

- Let y be the center of an ellipsoid E. Through it we place a hyper-plane and divide E into two half-ellipsoids, say E^+ and E^-.
- Compute the min-volume ellipsoid E' that contains E^+.

$$V(E') \leq e^{-.5/(n+1)} V(E).$$



The Ellipsoid Method (continued)

- First polynomial-time algorithm for Linear Programming