ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ



ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS

## M.Sc. Program in Data Science Department of Informatics

## Optimization Techniques Linear Programming – The Simplex Method

Instructor: G. ZOIS georzois@aueb.com

## **The Simplex Method**

- Designed by Dantzig (1947)
  - One of the most important algorithms of the 20<sup>th</sup> century
  - An algorithm that behaves extremely well in practice despite its exponential complexity in worst case
  - The design of the algorithm and the quest for better algorithms also contributed to building a rich theory around linear programming



# Polyhedra

- Simplex is trying to optimize a linear function over a polyhedron
- Definition: In R<sup>n</sup>, a polyhedron is defined by a set of linear inequalities on n variables

 $\mathsf{P} = \{ x : \mathsf{A}x \le b \}$ 

- Where  $x \in R^n$ ,  $b \in R^m$ , and A is an mxn matrix

• We will usually consider polyhedra in the form

 $P = \{x: Ax \le b, x \ge 0 \}$ 

- A polyhedron is
  - Infeasible, if its feasible region is empty
  - Bounded, if there exists M, such that for every x in the feasible region,  $||x||_2 \le M$
  - Unbounded, if it is not bounded

- Simplex is an algebraic procedure
- However, it is important to understand its geometric motivation
- Assume the polyhedron is non-empty and bounded
  - Then, an optimal solution always exists for any linear objective function
  - A bounded polyhedron is also called polytope
- To illustrate the geometry of simplex, we will use Example 2 from Lecture 1 as a representative example in 2 dimensions

Example 2: A polytope in R<sup>2</sup>

- Constraint boundaries: correspond to the 5 sides of the polygon
- Corner point feasible (CPF) solutions: points at the intersection of constraint boundaries
- Also called extreme point solutions or vertices of the polytope



Example 2: A polytope in R<sup>2</sup>

- Each corner point solution lies at the intersection of 2 constraint boundaries
- In 2 dimensions: how do we find each CPF solution?
  - System of 2 equations in 2 variables



- We say 2 corner point solutions are adjacent if they share 1 constraint boundary
- Here, (0, 0) and (0, 6) are adjacent,
- (0, 6) and (2, 6) are also adjacent

**X**<sub>1</sub>

Generalization to n dimensions:

In a polyhedron with n variables,

•a CPF solution is the intersection of n constraint boundaries

•How do we identify them?

- system of n equations in n variables
- Attention: make sure we have first removed "redundant" constraints
- i.e., constraints that can be implied by linear combinations of the others (otherwise the system will not have a unique solution)
- each group of n linearly independent constraints of the polyhedron yields a distinct CPF solution
- •Two CPF solutions are adjacent if they share n-1 constraint boundaries

Why are we interested in the notion of adjacent solutions?

Optimality test for linear programs:

Consider a LP with at least one optimal solution. If a CPF solution has no adjacent CPF solutions that are better, according to the objective function, then it must be an optimal solution.

- Hence, local optimality  $\Rightarrow$  global optimality
- Extremely important property
  - Also generalizes to continuous, convex functions (to be discussed in next lectures)
- In our example: (2, 6) is an optimal solution
  - (2, 6) is adjacent to (0, 6) and (4, 3)
  - None of these achieve a better value for the objective function

Outline of the simplex method from a geometric viewpoint

- Initialization: Choose an initial CPF solution
  - Usually we set all variables to 0
- Main iteration loop:
  - Apply the optimality test to the current CPF solution
  - If it is optimal stop,
  - else move to an adjacent solution that achieves the highest rate of increase in the objective function

Solving Example 2 with the simplex method

- Initialization:
  - we choose (0, 0) as the initial CPF solution
  - Optimality test: (0, 0) is not an optimal solution, there are better adjacent solutions



Solving Example 2 with the simplex method

- Iteration 1:
  - Move from (0, 0) to an adjacent solution
  - How do we pick one?
  - Choose the direction that increases the objective function at a faster rate
  - Recall:  $Z = 3x_1 + 5x_2$
  - Hence moving along the x<sub>2</sub> axis is better, stopping at (0, 6)
  - Optimality test: (0, 6) is not optimal



Solving Example 2 with the simplex method

- Iteration 2:
  - Move from (0, 6) to a better adjacent solution
  - Moving back is not making things better
  - Hence, only choice to move to
     (2, 6)
  - Optimality test: (2, 6) is better than (0, 6) and (4, 3), therefore, it is an optimal solution



Basic features of simplex

- It only examines CPF solutions
  - It is guaranteed that there always exists an optimal CPF solution
- Initialization: Whenever feasible, take (0, 0, ..., 0)
  - Nonnegativity constraints satisfied
  - What if the remaining constraints are violated? To be discussed again soon
- Picking the next CPF solution to visit:
  - Looking only at adjacent solutions can be easily implemented
  - The method only looks at the rate of increase in the objective function
  - Greedy local choice: we choose the direction with the best increase and stop at the adjacent solution in that direction

## **From Geometry to Algebra**

Q: How can we implement all these steps in an automated algebraic manner for any polyhedron with n variables?

- We can use the geometric viewpoint only up to n=3 variables
- For n > 3, we need a translation into precise algebraic instructions

First step: Transform the standard form into a system of linear equations

- Conversion of inequality constraints into equalities by introducing *slack variables*
- For example: consider the inequality  $x_1 \le 4$  of Example 2
- We can define the slack variable:  $x_3 = 4 x_1$
- The constraint then is converted as:

$$x_1 \le 4 \implies x_1 + x_3 = 4$$

• We can do this for all inequality constraints

### **Conversion of Example 2**

• Need 3 slack variables: x<sub>3</sub>, x<sub>4</sub>, x<sub>5</sub>

Algebraically, more convenient to work with the augmented form

Some terminology:

•Augmented solution: simply a solution for the original variables augmented by the slack variables

- For the feasible solution (3, 2), the augmented solution is (3, 2, 1, 8, 5)
- Basic Feasible (BF) solution: an augmented CPF solution
  - (0, 6) is a CPF solution in the original problem
  - (0, 6, 4, 0, 6) is the corresponding BF solution
  - From a BF solution, we can get back the CPF solution by simply omitting the slack variables
- •Understanding how BF solutions look like:
  - Example 2: 5 variables in total and 3 constraints
  - Hence, 2 degrees of freedom
  - If we set "arbitrary values" to 2 variables, then we can solve a linear system for the rest
  - In simplex: "arbitrary value" = 0

- For every BF solution:
  - We separate the variables into **basic** and **nonbasic** variables
  - Number of basic variables = m = number of constraints (excluding the nonnegativity constraints)
  - Number of nonbasic variables = n
  - Nonbasic variables are set to 0
  - Basic variables are then computed by solving the system of m linear equalities
  - The set of basic variables is referred to as the "basis" of the BF solution
- In our example:
  - (0, 0, 4, 12, 18) is a BF solution
  - Nonbasic variables:  $x_1$ ,  $x_2$ , both set to 0
  - Basis =  $\{x_3, x_4, x_5\}$
  - The values of the basis can be obtained by the constraints, after substituting  $x_1 = x_2 = 0$

- Checking adjacency of two BF solutions
  - We could check if the corresponding CPF solutions are adjacent
  - Easier way: Two BF solutions are adjacent if their bases differ only in one variable (which means that all but one of their nonbasic variables are also the same)
- Illustration:
  - Adjacent CPF solutions: (0, 0) and (0, 6)
  - Corresponding BF solutions:  $S_1 = (0, 0, 4, 12, 18)$  and  $S_2 = (0, 6, 4, 0, 6)$
  - In S<sub>1</sub>, nonbasic variables =  $\{x_1, x_2\}$ , basis =  $\{x_3, x_4, x_5\}$
  - In S<sub>2</sub>, nonbasic variables = { $x_1$ ,  $x_4$ }, basis = { $x_2$ ,  $x_3$ ,  $x_5$ }
  - Going from  $S_1$  to  $S_2$ , variable  $x_2$  switches from nonbasic to basic and variable  $x_4$  leaves the basis
- Hence: very simple way of moving from one adjacent solution to another

## Final step before running simplex:

• It becomes convenient to also treat the objective function as another equality constraint



- No need for a slack variable since we have equality to begin with
- We will not really treat Z as a new variable

### Initialization:

- We need to choose an initial BF solution
  - In our example, setting  $x_1 = 0$  and  $x_2 = 0$  is feasible
  - Augmented solution: (0, 0, 4, 12, 18)
  - Hence, initial basis =  $\{x_3, x_4, x_5\}$ , nonbasic variables:  $x_1, x_2$
- Optimality test:
  - Initial value of the objective function: Z = 0
  - Recall Z =  $3x_1 + 5x_2$ , expressed as a function of the nonbasic variables
  - Coefficient for each nonbasic variable: rate of improvement for Z, if that variable were to be increased.
  - Here the rates of improvement are positive, which means the current BF solution is not optimal
  - Hence: simple way to answer the optimality test

Iteration 1:

- We need to determine the direction of movement towards an adjacent BF solution
  - Coefficient of  $x_2$  in Z > coefficient of  $x_1$
  - We pick  $x_2$  as the variable to increase
  - Variable x<sub>2</sub> will enter the basis (referred to as the entering basic variable)
- How much shall we increase x<sub>2</sub>?
  - For as long as we do not violate the constraints!

(1) 
$$x_1 + x_3 = 4 \implies x_3 = 4$$
  
(2) $2x_2 + x_4 = 12 \implies x_4 = 12 - 2x_2$   
(3) $3x_1 + 2x_2 + x_5 = 18 \implies x_5 = 18 - 2x_2$ 

And now use the nonnegativity constraints!

Iteration 1:

(1)  $x_1 + x_3 = 4 \implies x_3 = 4$ (2) $2x_2 + x_4 = 12 \implies x_4 = 12 - 2x_2$ (3) $3x_1 + 2x_2 + x_5 = 18 \implies x_5 = 18 - 2x_2$ 

 $x_{3} \ge 0 \implies \text{no upper bound on } x_{2}$   $x_{4} \ge 0 \implies 12 - 2x_{2} \ge 0 \implies x_{2} \le 6$  $x_{5} \ge 0 \implies 18 - 2x_{2} \ge 0 \implies x_{2} \le 9$ 

- We pick the minimum value implied by the upper bounds
- Increasing x<sub>2</sub> beyond the value of 6 would result in an infeasible solution
- Hence, we stop at x<sub>2</sub> = 6

- Can we arrive at  $x_2 = 6$  with a more automated way?
- Minimum Ratio Test:
  - For each constraint, divide the constant term by the coefficient of x<sub>2</sub>
  - The minimum such ratio tells us how much to increase  $x_2$
- (1)  $x_1 + x_3 = 4$   $\Rightarrow$  ratio = 4/0 = + $\infty$  (0 coefficient of  $x_2$ )
- (2)  $2x_2 + x_4 = 12$   $\Rightarrow$  ratio = 12/2 = 6
- (3)  $3x_1 + 2x_2 + x_5 = 18 \implies ratio = 18/2 = 9$
- Setting x<sub>2</sub> = 6 makes variable x<sub>4</sub> drop to 0
- x<sub>4</sub> is called the *leaving basic variable*

#### **Iteration 1:**

• Summarize what we have done so far:

	Initial BF solution	New BF solution
Nonbasic variables	$x_1 = 0, x_2 = 0$	$x_1 = 0, x_4 = 0$
Basis	x <sub>3</sub> = 4, x <sub>4</sub> = 12, x <sub>5</sub> = 18	x <sub>2</sub> = 6, x <sub>3</sub> = ?, x <sub>5</sub> = ?

- Final step of Iteration 1:
  - Convert the system of equations according to the new basis
  - Express the objective function in terms of the new nonbasic variables
  - Compute the missing values in the new BF solution (for  $x_3$  and  $x_5$ )

Iteration 1: Initial constraints

- (0)  $Z 3x_1 5x_2 = 0$
- (1)  $x_1 + x_3 = 4$
- (2)  $2x_2 + x_4 = 12$
- $(3) \quad 3x_1 + 2x_2 + x_5 = 18$
- We need x<sub>2</sub> to disappear from (0), (1) and (3)
- Start with row (2): row (2)  $/2 \implies x_2 + 1/2 x_4 = 6$
- We can then
  - multiply a row by a constant
  - Add/subtract multiples of a row to/from another row
  - For example: row (0) := row (0) + 5  $\cdot$  row (2)

#### **Iteration 1:**

Final set of constraints at the end of the iteration

- (0)  $Z 3x_1 + 5/2 x_4 = 30$ (1)  $x_1 + x_3 = 4$ (2)  $x_2 + 1/2 x_4 = 6$ (3)  $3x_1 - x_4 + x_5 = 6$
- Procedure for obtaining the new form of the constraints: the Gauss-Jordan method
- Hence, assignment of values in the new BF solution:
  - $x_1 = 0, x_4 = 0, x_2 = 6, x_3 = 4, x_5 = 6$
- Optimality test:
  - $Z = 30 + 3x_1 5/2x_4$ , positive coefficient for  $x_1 \Rightarrow$  not optimal
  - Hence, we need to move to an adjacent BF solution

- Which variable should now enter the basis?
  - Unique choice: Coefficient of  $x_1$  is the only positive coefficient in Z
  - Variable x<sub>1</sub> is now the new entering basic variable
- How much shall we increase x<sub>1</sub>?
  - Apply the Minimum Ratio Test
  - Set  $x_1 := 2$  due to equation (3)
- Which variable exits the basis?
  - Again, from the Minimum Ratio Test, x<sub>5</sub> will be the leaving variable
- New basis: {x<sub>1</sub>, x<sub>2</sub>, x<sub>3</sub>}
  - Nonbasic variables:  $x_4 = x_5 = 0$
  - New BF solution: (2, 6, 2, 0, 0)

#### **Iteration 2:**

Substituting using the Gauss-Jordan method:

- (0)  $Z + 3/2 x_4 + x_5 = 36$ (1)  $x_3 + 1/3 x_4 - 1/3 x_5 = 2$ (2)  $x_2 + \frac{1}{2} x_4 = 6$ (3)  $x_1 - \frac{1}{3} x_4 + \frac{1}{3} x_5 = 2$
- Optimality test:
  - $Z = 36 3/2 x_4 x_5$
  - There is no direction of improvement, increasing  $x_4$  or  $x_5$  will decrease the objective function
  - Current BF solution is optimal
  - Solution of the original linear program:  $x_1 = 2$ ,  $x_2 = 6$  and Z = 36

- So far we have managed to transform our geometric intuition into an algebraic procedure
- Operations used pretty simple
- Nevertheless, we can make the process even more automated
- Simplex tableau: A tabular representation of the constraints and the current BF solution
- All we need to know: the basis and the coefficients in each row

### Algebraic form vs tableau:

Let us revisit the initialization in our Example:



#### Corresponding tableau form

	Basis	Coefficients						
		Z	<b>x</b> <sub>1</sub>	<b>x</b> <sub>2</sub>	<b>X</b> <sub>3</sub>	<b>x</b> <sub>4</sub>	<b>X</b> 5	side
	Z	1	-3	-5	0	0	0	0
rows = number of	<b>X</b> <sub>3</sub>	0	1	0	1	0	0	4
	<b>x</b> <sub>4</sub>	0	0	2	0	1	0	12
	<b>X</b> 5	0	3	2	0	0	1	18

Pacie	Coefficients										
Dasis	Z	<b>x</b> <sub>1</sub>	<b>X</b> <sub>2</sub>	X <sub>3</sub>	<b>X</b> 4	<b>X</b> 5	side				
Z	1	-3	-5	0	0	0	0				
Х <sub>3</sub>	0	1	0	1	0	0	4				
<b>x</b> <sub>4</sub>	0	0	2	0	1	0	12				
<b>X</b> 5	0	3	2	0	0	1	18				

- For notational convenience: treat Z also as a basic variable
- Optimality test in a tableau:
  - We have reached an optimal solution when the coefficients in row (0) are all nonnegative

Pasia		Coefficients										
Dasis	Z X1		<b>X</b> <sub>2</sub>	X <sub>3</sub>	<b>x</b> <sub>4</sub>	<b>X</b> 5	side					
Z	1	-3	-5	0	0	0	0					
X <sub>3</sub>	0	1	0	1	0	0	4					
<b>x</b> <sub>4</sub>	0	0	2	0	1	0	12					
<b>X</b> 5	0	3	2	0	0	1	18					

- Which variable should enter the basis?
  - The nonbasic variable with the most negative coefficient in row (0), hence x<sub>2</sub>
  - Column of x<sub>2</sub>: pivot column
- Minimum Ratio Test
  - How do we run it?
  - Information we need is the right side column and the column of  $\boldsymbol{x}_2$

Pacia		Coefficients										
Dasis	Z	<b>X</b> <sub>1</sub>	<b>X</b> <sub>2</sub>	X <sub>3</sub>	<b>X</b> 4	<b>X</b> 5	side					
Z	1	-3	-5	0	0	0	0					
<b>X</b> <sub>3</sub>	0	1	0	1	0	0	4					
<b>x</b> <sub>4</sub>	0	0	2	0	1	0	12					
<b>X</b> 5	0	3	2	0	0	1	18					

- Outcome of the Minimum Ratio Test
  - Minimum achieved at row of  $x_4$
  - Leaving variable: x<sub>4</sub>, i.e., the basic variable corresponding to that row
  - Row of  $x_4$ : the **pivot row**
  - Intersection of pivot row and pivot column: pivot element (=2 in this iteration)

		Right					
Z	Z x <sub>1</sub>		X <sub>3</sub>	<b>x</b> <sub>4</sub>	<b>X</b> 5	side	
1	-3	-5	0	0	0	0	
0	1	0	1	0	0	4	
0	0	2	0	1	0	12	12/2 = 6
0	3	2	0	0	1	18	18/2 = 9
	Z 1 0 0	Z       x1         1       -3         0       1         0       0         0       3	Coeff         Z $X_1$ $X_2$ 1       -3       -5         0       1       0         0       2       2         0       3       2	Coefficients         Z $X_1$ $X_2$ $X_3$ 1       -3       -5       0         0       1       0       1         0       2       0       0         0       3       2       0	CoefficientsZ $X_1$ $X_2$ $X_3$ $X_4$ 1-3-50001010020103200	CoefficientsZ $X_1$ $X_2$ $X_3$ $X_4$ $X_5$ 1-3-500001010002010032001	CoefficientsRight sideZ $X_1$ $X_2$ $X_3$ $X_4$ $X_5$ T-3-50000O101004O0201012O3200118

- Final step: Gauss-Jordan method to get the new tableau
  - Divide first the pivot row by the pivot element
  - This makes the coefficient of  $x_2$  equal to 1 in the pivot row
  - Then we can add/subtract appropriate multiples of the pivot row to the other rows (just as in the algebraic description of simplex)

Pacie	Coefficients										
Dasis	Z	<b>x</b> <sub>1</sub>	<b>x</b> <sub>2</sub>	X <sub>3</sub>	<b>X</b> 4	<b>X</b> 5	side				
Z	1	-3	0	0	5/2	0	30				
Х <sub>3</sub>	0	1	0	1	0	0	4				
<b>x</b> <sub>2</sub>	0	0	1	0	1/2	0	6				
<b>X</b> 5	0	3	0	0	-1	1	6				

**New tableau** 

## End of Iteration 1:

- Optimality test:
  - There exists a negative coefficient in row (0)
  - Hence, we need to go to the next iteration

Decia			Right					
DdSIS	Z	<b>x</b> <sub>1</sub>	<b>x</b> <sub>2</sub>	<b>X</b> 3	<b>X</b> 4	<b>X</b> 5	side	
Z	1	-3	0	0	5/2	0	30	
<b>X</b> <sub>3</sub>	0	1	0	1	0	0	4	4/1 = 4
<b>x</b> <sub>2</sub>	0	0	1	0	1/2	0	6	
Х <sub>5</sub>	0	3	0	0	-1	1	6	6/3 = 2

- Which variable should enter the basis?
  - Only variable  $x_1$  has a negative coefficient in row (0)
  - **Pivot column:** The column of  $x_1$
- Minimum Ratio Test
  - Variable  $x_5$  is the leaving variable
  - **Pivot row:** The row of  $x_5$

Basis	Pasis			Right					
	Dasis	Z	<b>x</b> <sub>1</sub>	<b>x</b> <sub>2</sub>	<b>X</b> <sub>3</sub>	<b>X</b> 4	<b>X</b> 5	side	
	Z	1	0	0	0	3/2	1	36	
	X <sub>3</sub>	0	0	0	1	1/3	-1/3	2	New tableau
	<b>x</b> <sub>2</sub>	0	0	1	0	1/2	0	6	
	<b>x</b> <sub>1</sub>	0	1	0	0	-1/3	1/3	2	

## End of Iteration 2:

- Optimality test
  - No negative coefficient in the row of Z
  - Hence we stop at the current BF solution (2, 6, 2, 0, 0)
  - Optimal solution to the original problem:  $x_1 = 2$ ,  $x_2 = 6$

# Summary: Geometric, algebraic and tableau form

We have seen 3 different ways of thinking about the same algorithm

#### • Geometric view:

- This is how the algorithm was inspired
- Useful only for 2 or 3 dimensions

### • Algebraic description

More convenient for learning the logic of the algorithm esp. in higher dimensions

### • Tableau form

- Equivalent to the algebraic form in terms of operations performed
- However, it organizes the data in a more compact form
- Allows for better automation

Some issues that may arise

### • During the execution:

- Many choices for the entering variable
- Many choices for the leaving variable
- No leaving variable

## • At initialization:

- Difficulty in finding an initial feasible solution to begin with

### • At termination:

Multiple optimal solutions

### • Many choices for the entering variable:

- No problem, make an arbitrary choice
- An optimal solution will be reached eventually
- Hard to know in advance which one is the best choice

## •Many choices for the leaving variable:

- -This may cause problems
- -All such variables will become 0 at the end of the iteration
- -Hence, we will have some basic variables with a 0 value

-Such solutions are called *degenerate* 

- -They may not allow Z to increase in the next iteration
- -The algorithm may get trapped in a loop where some variables enter and exit the basis repeatedly and Z gets stuck at the same value

-Bland's rule: If there are multiple candidate leaving variables, always choose the variable with the smallest index

-Also: rarely been observed in practice, almost safe to ignore this

### •No leaving basic variable:

- This means that the entering variable can be increased indefinitely
- The increase does not yield any negative values to the current basic variables
- In the tableau form: all coefficients in pivot column are negative or 0 (except first row)
- Conclusion: The problem is unbounded, optimal solution is  $+\infty$
- Maybe a mistake has occured in the initial formulation of the problem

- Multiple optimal solutions:
  - If there are multiple optimal solutions, there are at least 2 optimal CPF solutions
  - Any convex combination of these CPF solutions is also an optimal solution
  - In some problems we may only care to identify one optimal solution and stop
  - If we care to find more optimal CPF solutions:
    - Run more iterations of simplex after we found the first optimal solution
    - Choose a nonbasic variable with zero coefficient in the row of Z as the entering variable
    - There exists such a variable whenever there are multiple optimal solutions

- Difficulty in finding an initial feasible solution:
  - What if the all-0 solution is not feasible? How do we start simplex then?
  - This can happen when some coefficients b<sub>i</sub> are negative
  - Strategy: Define an auxiliary problem so that
    - It is easy to find an initial feasible solution in the auxiliary problem
    - The optimal solution of the auxiliary can tell us whether there exists a feasible solution in our original problem
  - 2-phase simplex method:
    - First run simplex on the auxiliary problem
    - See whether we can identify an initial basic feasible solution from the optimal solution of the auxiliary problem
    - If yes, run simplex on our original problem

- Difficulty in finding an initial feasible solution:
  - There are various ways to define the auxiliary problem
  - Illustration:

 $\begin{array}{ll} \max Z = c^{\mathsf{T}} \, x & \min \, 1^{\mathsf{T}} y \\ \text{s.t.} & \text{s.t.} \\ \sum_{j} a_{ij} x_{j} \leq b_{i} & \Longrightarrow & \sum_{j} a_{ij} x_{j} + y_{i} = b_{i} \, , \, i = 1, ..., n \\ x_{i} \geq 0, \, i = 1, ..., n & x_{i} \geq 0, \, y_{i} \geq 0, \, i = 1, \, ..., n \end{array}$ 

- The auxiliary problem always has a feasible solution
  - Set original variables to 0, and y equal to b.
- The original problem has a feasible solution if and only if the optimal of the auxiliary is 0

# Other variants in implementing Simplex

- The revised simplex method
  - Based on exploiting fast matrix operations
  - Each iteration requires solving 2 systems of linear equations
  - But these are not solved from scratch
  - Only small updates based on the solution from previous iteration
- The dual simplex method
  - Applying simplex to the dual linear program
  - But essentially working with the primal
  - Useful tool for sensitivity analysis
- Many other variations have also been suggested over the years...

# **Complexity of Simplex**

- •Extremely well-behaved in practice
- Empirically, number of iterations in simplex looks proportional to number of constraints, e.g., usually no more than 3m
- Can we have a good theoretical upper bound on the number of iterations?
- NO! There are examples that need an exponential (2<sup>n</sup>) number of iterations, discovered first by [Klee, Minty '72]
- Despite that, it is still one of the preferred algorithms for solving linear programs!

## **Other Algorithms**

•The ellipsoid method: The first polynomial time algorithm

- By [Kachiyan '79], however not well behaved in practice

### •Interior point methods: also polynomial time algorithms

- First conceived by Karmarkar [1984]
- Main ideas:

6

(0, 0)

- Again keep moving from a feasible solution to a better one
- But this time, we move along solutions in the interior of the polytope
- The current solution keeps getting closer and closer to a vertex of the polytope



# **Other Algorithms**

### Types of interior point methods

## •Affine scaling algorithms

- One of the simplest interior point algorithms
- Based on approximating polyhedra by "ellipsoids"
- Optimizing over ellipsoids in each iteration
- Non-linear problems but solvable with closed form solutions

## Potential reduction algorithms

- Do not measure progress by the increase in the objective function
- Instead use a non-linear potential function

### Path following algorithms

- Transforms the initial problem into an unconstrained problem (or a problem with equality constraints)
- Incorporates the inequality constraints " $x_i \ge 0$ " into the objective function (logarithmic barrier function)
- Solves the resulting non-linear problem with Newton's method

# **Simplex vs Interior Point Algorithms**

### • Comparisons

- In theory: interior point methods are polynomial time algorithms (for any n and m), simplex may need exponential time
- In practice: average case complexity of simplex very low compared to worst case
- One iteration of interior point methods needs much more computation time than in simplex to decide the next feasible solution
- But: as the number of constraints increases, interior point methods do not need much more iterations
- Number of iterations in simplex may increase rapidly as we increase the number of variables and constraints
  - Interior point methods go through the internal part of the polytope
  - Adding more constraints reduces the feasible region, by adding more constraint boundaries
  - Hence, for problems with many thousands of constraints, interior point methods seem to be the best hope