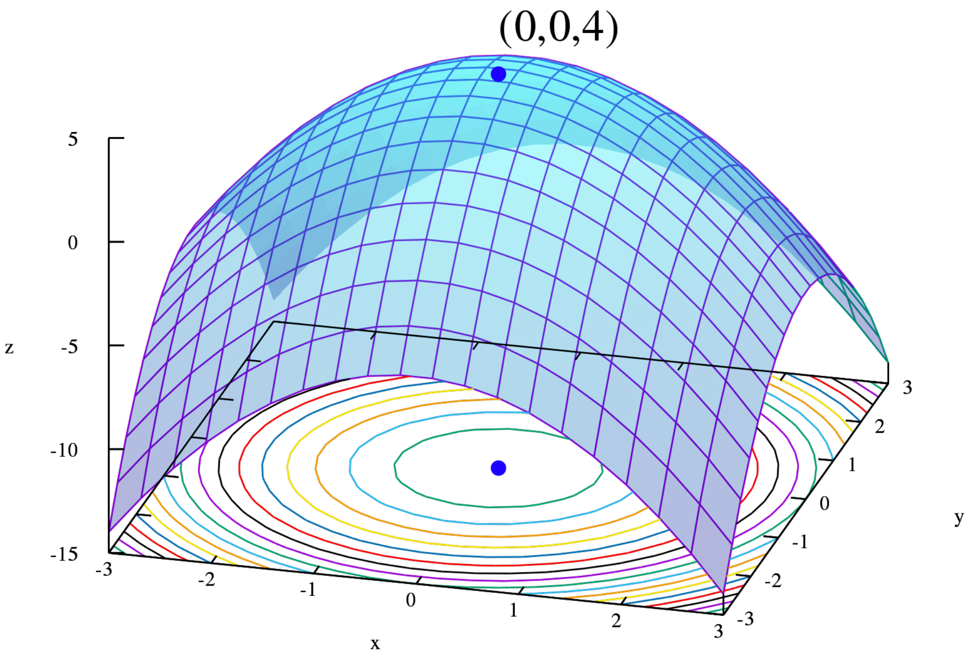
Simplex Tableaux Continued



Deepak Ajwani

School of Computer Science

Simplex Algorithm: High-Level

1. Each Simplex tableau is associated with a basic feasible solution

2. From the initial tableau, we construct a sequence of tableaus of a similar form, by gradually rewriting them according to certain rules

3. Each tableau contains the same information about the LP, only written differently

4. The desired optimal solution can be read off directly from the terminal tableau

Recap: The procedure

1. Pre-processing steps (Convert LP into the right form.)

2. Create *initial tableau*

3. While the bottom row contains a negative number, do:

    a. Choose *pivot cell* in the tableau
    b. Do *row reduction* so pivot entry is 1 and all other entries in its column are 0

4. Optimal solution is the basic feasible solution corresponding to the (basis of the) final tableau.

Questions/Comments on the Tableau method

- The process works well for maximisation problems with constraints of the type $Ax \leq b$, where $x \geq 0$ and $b \geq 0$
  - For these constraints, origin is in the feasible region
- But what if the origin is not in the feasible region?
- How do you find the first feasible solution?
  - Finding a first feasible solution of a linear program is as hard as finding the optimal solution
  - Solve a different linear program to find a feasible solution

Simplex Tableaux: Two-Phase Method

- Phase 1 (Find a feasible solution):
  - Add artificial variables to the constraints to secure a starting basic solution.
  - Find a basic solution of the resulting equations that minimises the sum of the artificial variables
  - If the minimum value of the sum is positive, the LP problem has no feasible solution. Otherwise proceed to Phase 2.
- Phase 2: Use the feasible solution from Phase 1 as a starting basic feasible solution for the original problem

Simplex Tableaux: Phase 1

Consider the following LP

Minimise $z = 4x_1 + x_2$

subject to

$$3x_1 + x_2 = 3$$

$$4x_1 + 3x_2 \geq 6$$

$$x_1 + 2x_2 \leq 4$$

$$x_1, x_2 \geq 0$$

Simplex Tableaux: Phase 1

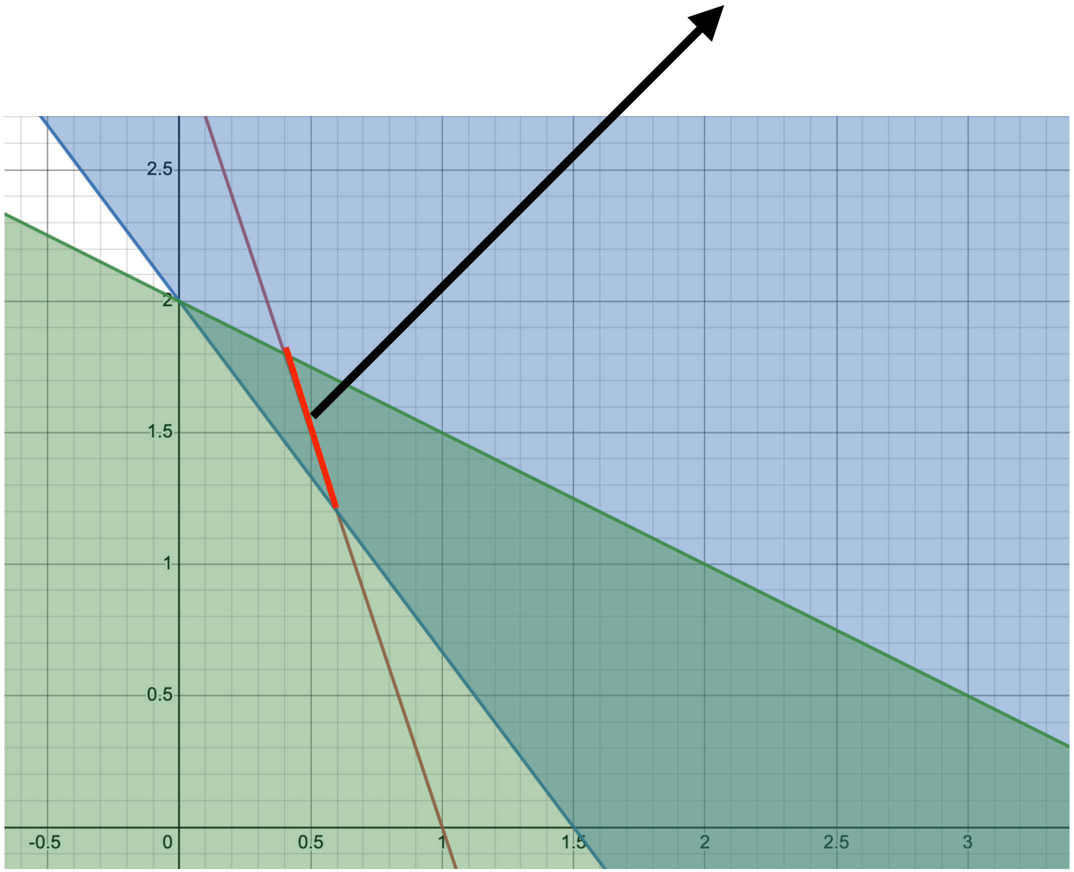
Consider the following LP

Minimise $z = 4x_1 + x_2$

subject to

$$3x_1 + x_2 = 3$$
$$4x_1 + 3x_2 \geq 6$$
$$x_1 + 2x_2 \leq 4$$
$$x_1, x_2 \geq 0$$

Feasible region lies on this line



Simplex Tableaux: Phase 1

Consider the following LP

Minimise $z = 4x_1 + x_2$

subject to

$$3x_1 + x_2 = 3$$
$$4x_1 + 3x_2 \geq 6$$
$$x_1 + 2x_2 \leq 4$$
$$x_1, x_2 \geq 0$$

Feasible region lies on this line

Origin is not a feasible solution!

Simplex Tableaux: Phase 1

LP in canonical form

Minimise $z = 4x_1 + x_2$

subject to

$$
\begin{aligned}
3x_1 + x_2 \qquad\qquad\qquad\quad &= 3 \\
4x_1 + 3x_2 - x_3 \qquad\quad\;\; &= 6 \\
x_1 + 2x_2 \qquad\qquad\quad + x_4 &= 4 \\
x_1, x_2, x_3, x_4 \geq 0 &
\end{aligned}
$$

Simplex Tableaux: Phase 1

Simplex Tableau

x_1	x_2	x_3	x_4	
3	1	0	0	3
4	3	-1	0	6
1	*2*	0	1	4
-4	-1	0	0	0

Simplex Tableaux: Two-Phase Method

- Phase 1 (Find a feasible solution):
  - **Add artificial variables to the constraints to secure a starting basic solution**
  - Find a basic solution of the resulting equations that minimises the sum of the artificial variables
  - If the minimum value of the sum is positive, the LP problem has no feasible solution. Otherwise proceed to Phase 2.
- Phase 2: Use the feasible solution from Phase 1 as a starting basic feasible solution for the original problem

Simplex Tableaux: Add Artificial Variables

We rewrite our objective function in terms of the original variables

Minimise $r = R_1 + R_2$

subject to

$$
\begin{aligned}
3x_1 + x_2 \quad\quad & +R_1 & & = 3 \\
4x_1 + 3x_2 - x_3 \quad & & +R_2 & = 6 \\
x_1 + 2x_2 \quad\quad & & & +x_4 = 4 \\
x_1, x_2, x_3, x_4, \quad & R_1, R_2 \geq 0 & &
\end{aligned}
$$

Simplex Tableaux: Add Artificial Variables

Adding Artificial Variables

Minimise $r = R_1 + R_2$

subject to

$$
\begin{aligned}
3x_1 + x_2 \quad & +R_1 & & = 3 \\
4x_1 + 3x_2 - x_3 \quad & & +R_2 & = 6 \\
x_1 + 2x_2 \quad & & & +x_4 = 4 \\
x_1, x_2, x_3, x_4, \quad & R_1, R_2 \geq 0 & &
\end{aligned}
$$

The optimal value of this objective function is 0 exactly if there exists a feasible solution of the original linear program

Simplex Tableaux: Two-Phase Method

- Phase 1 (Find a feasible solution):
  - Add artificial variables to the constraints to secure a starting basic solution
  - **Find a basic solution of the resulting equations that minimises the sum of the artificial variables**
  - If the minimum value of the sum is positive, the LP problem has no feasible solution. Otherwise proceed to Phase 2.
- Phase 2: Use the feasible solution from Phase 1 as a starting basic feasible solution for the original problem

Simplex Tableaux: Add Artificial Variables

Modify the objective function to write it in terms of original variables

Since $R_1 = 3 - 3x_1 - x_2$ and $R_2 = 6 - 4x_1 - 3x_2 + x_3$

$$r = R_1 + R_2 = 9 - 7x_1 - 4x_2 + x_3$$

Simplex Tableaux: Add Artificial Variables

Modify the objective function to write it
in terms of original variables

Since $R_1 = 3 - 3x_1 - x_2$ and $R_2 = 6 - 4x_1 - 3x_2 + x_3$

$$r = R_1 + R_2 = 9 - 7x_1 - 4x_2 + x_3$$

Our new LP for Phase 1 is thus:

Minimise $r = 9 - 7x_1 - 4x_2 + x_3$

subject to

$$\begin{array}{llll} 3x_1 + x_2 & +R_1 & & = 3 \\ 4x_1 + 3x_2 - x_3 & & +R_2 & = 6 \\ x_1 + 2x_2 & & & +x_4 = 4 \\ x_1, x_2, x_3, x_4, & R_1, R_2 \geq 0 & & \end{array}$$

Simplex Tableaux: Phase 1

Simplex Tableau

x_1	x_2	x_3	R_1	R_2	x_4	
3	1	0	1	0	0	3
4	3	-1	0	1	0	6
1	*2*	0	0	0	1	4
-7	-4	1	0	0	0	-9

Simplex Tableaux: Phase 1

We run our Simplex transformations on this:

x_1	x_2	x_3	R_1	R_2	x_4	
3	1	0	1	0	0	3
4	3	-1	0	1	0	6
1	2	0	0	0	1	4
-7	-4	1	0	0	0	-9

Simplex Tableaux: Phase 1

We run our Simplex transformations on this:

x_1	x_2	x_3	R_1	R_2	x_4	
1	1/3	0	1/3	0	0	1
4	3	-1	0	1	0	6
1	2	0	0	0	1	4
-7	-4	1	0	0	0	-9

Simplex Tableaux: Phase 1

We run our Simplex transformations on this:

x_1	x_2	x_3	R_1	R_2	x_4	
1	1/3	0	1/3	0	0	1
0	5/3	-1	-4/3	1	0	2
0	5/3	0	-1/3	0	1	3
0	-5/3	1	7/3	0	0	-2

Simplex Tableaux: Phase 1

We run our Simplex transformations on this:

x_1	x_2	x_3	R_1	R_2	x_4	
1	1/3	0	1/3	0	0	1
0	5/3	-1	-4/3	1	0	2
0	5/3	0	-1/3	0	1	3
0	-5/3	1	7/3	0	0	-2

Simplex Tableaux: Phase 1

We run our Simplex transformations on this:

x_1	x_2	x_3	R_1	R_2	x_4	
1	1/3	0	1/3	0	0	1
0	1	-3/5	-4/5	3/5	0	6/5
0	5/3	0	-1/3	0	1	3
0	-5/3	1	7/3	0	0	-2

Simplex Tableaux: Phase 1

We run our Simplex transformations on this:

x_1	x_2	x_3	R_1	R_2	x_4	
1	0	1/5	3/5	-1/5	0	3/5
0	1	-3/5	-4/5	3/5	0	6/5
0	0	1	1	-1	1	1
0	0	0	1	1	0	0

Simplex Tableaux: Two-Phase Method

- Phase 1 (Find a feasible solution):
  - Add artificial variables to the constraints to secure a starting basic solution
  - Find a basic solution of the resulting equations that minimises the sum of the artificial variables
  - **If the minimum value of the sum is positive, the LP problem has no feasible solution. Otherwise proceed to Phase 2.**
- Phase 2: Use the feasible solution from Phase 1 as a starting basic feasible solution for the original problem

Simplex Tableaux: Phase 1

x_1	x_2	x_3	R_1	R_2	x_4	
1	0	1/5	3/5	-1/5	0	3/5
0	1	-3/5	-4/5	3/5	0	6/5
0	*0*	1	1	-1	1	1
0	0	0	1	1	0	0

Optimal solution for Phase 1 LP is

$$x_1 = 3/5,\ x_2 = 6/5,\ x_3 = 0,\ x_4 = 1,\ R_1 = 0,\ R_2 = 0$$

Minimum value of the sum is 0

$\Longrightarrow$ Original LP has a feasible solution

We proceed to Phase 2

Simplex Tableaux: Phase 1

Optimal solution for Phase 1 LP is

$$x_1 = 3/5,\ x_2 = 6/5,\ x_3 = 0,\ x_4 = 1,\ R_1 = 0,\ R_2 = 0$$

Minimum value of the sum is 0

Simplex Tableaux: Two-Phase Method

- Phase 1 (Find a feasible solution):
    - Add artificial variables to the constraints to secure a starting basic solution
    - Find a basic solution of the resulting equations that minimises the sum of the artificial variables
    - If the minimum value of the sum is positive, the LP problem has no feasible solution. Otherwise proceed to Phase 2.
- **Phase 2: Use the feasible solution from Phase 1 as a starting basic feasible solution for the original problem**

Simplex Tableaux: Phase 2

We get the tableau for Phase 2 as follows:

- Removing columns corresponding to artificial variables
- Reverting back to original LP objective

x_1	x_2	x_3	R_1	R_2	x_4	
1	0	1/5	3/5	-1/5	0	3/5
0	1	-3/5	-4/5	3/5	0	6/5
0	0	1	1	-1	1	1
0	0	0	1	1	0	0

Simplex Tableaux: Phase 2

We get the tableau for Phase 2 as follows:

- Removing columns corresponding to artificial variables
- Reverting back to original LP objective

x_1	x_2	x_3	x_4	
1	0	1/5	0	3/5
0	1	-3/5	0	6/5
0	*0*	1	1	1
-4	-1	0	0	0

Simplex Tableaux: Phase 2

Our Phase 2 LP is thus:

Minimise $z = 4x_1 + x_2$

subject to

$$
\begin{aligned}
x_1 + (1/5)x_3 &= 3/5 \\
x_2 - (3/5)x_3 &= 6/5 \\
x_3 + x_4 &= 1 \\
x_1, x_2, x_3, x_4 &\geq 0
\end{aligned}
$$

Simplex Tableaux: Phase 2

In terms of non-basic variable, we can reformulate the objective function as follows

Minimise $z = \dfrac{18}{5} - \dfrac{1}{5}x_3$

subject to

$$
\begin{aligned}
x_1 + (1/5)x_3 &= 3/5 \\
x_2 - (3/5)x_3 &= 6/5 \\
x_3 + x_4 &= 1 \\
x_1, x_2, x_3, x_4 &\geq 0
\end{aligned}
$$

Simplex Tableaux: Phase 2

Our new tableau looks as follows:

x_1	x_2	x_3	x_4	
1	0	1/5	0	3/5
0	1	-3/5	0	6/5
0	*0*	1	1	1
0	0	-1/5	0	-18/5

Simplex Tableaux: Phase 2

After running Simplex transformations, the final tableaux looks as follows:

x_1	x_2	x_3	x_4	
1	0	0	-1/5	2/5
0	1	0	3/5	9/5
0	0	1	1	1
0	0	0	1/5	-17/5

Simplex Tableaux: Phase 2

After running Simplex transformations, the final tableaux looks as follows:

x_1	x_2	x_3	x_4	
1	0	0	-1/5	2/5
0	1	0	3/5	9/5
0	*0*	1	1	1
0	0	0	1/5	-17/5

Optimal Solution: $x_1 = 2/5,\ x_2 = 9/5,\ x_3 = 1,\ x_4 = 0$

Optimal value: 17/5

Simplex Tableaux: Phase 2

Optimal Solution: $x_1 = 2/5,\ x_2 = 9/5,\ x_3 = 1,\ x_4 = 0$

Optimal value: 17/5

Special Cases in the Simplex Method

- What if the solution is infeasible?
- What if the solution is unbounded?
- What about alternative optima?
- What about degeneracy?

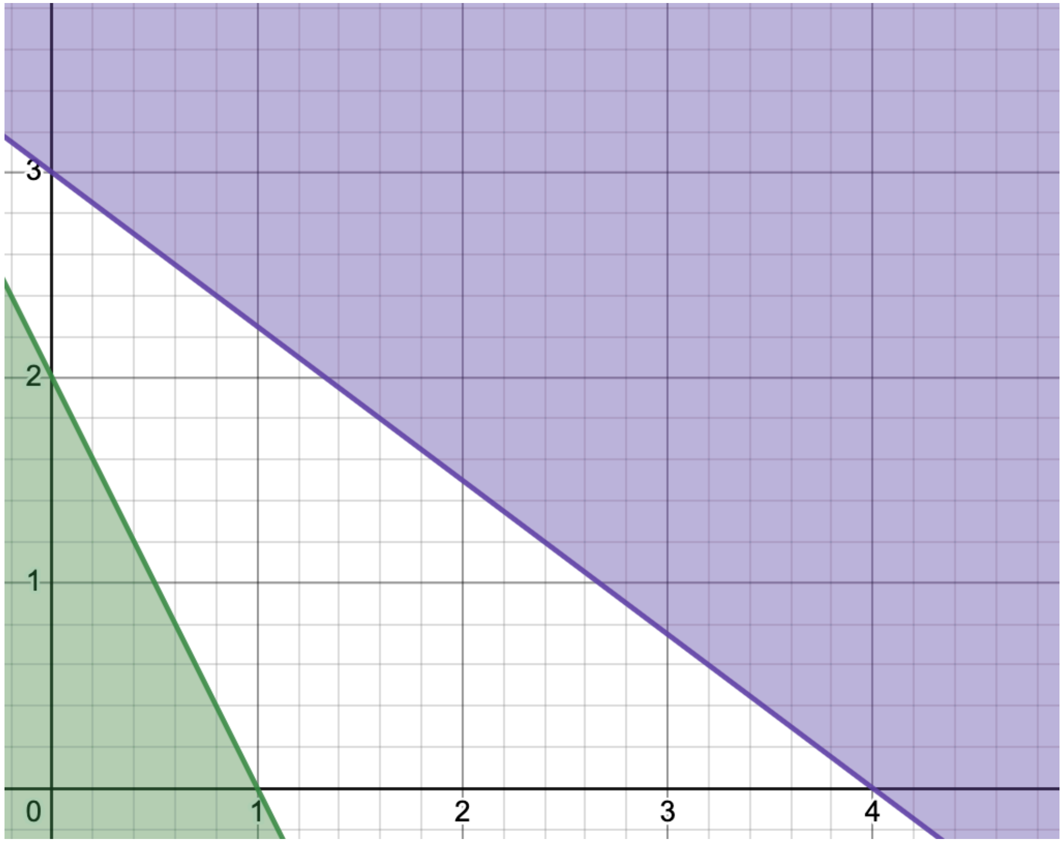
Simplex: Dealing with Infeasibility

- What if the solution is infeasible?

Maximise $z = 3x_1 + 2x_2$

subject to

$$2x_1 + x_2 \leq 2$$
$$3x_1 + 4x_2 \geq 12$$
$$x_1, x_2 \geq 0$$



Simplex: Dealing with Infeasibility

- What if the solution is infeasible?
  - Objective of Phase 1 solution of LP will be greater than 0
    - Some artificial variables will have to be non-zero to satisfy the constraints

Simplex: Dealing with Unbounded Solution

- What if the solution is unbounded?
  - Some variable(s) may be increased indefinitely without violating any of the constraints
  - Associated objective function value may also be unbounded in this case

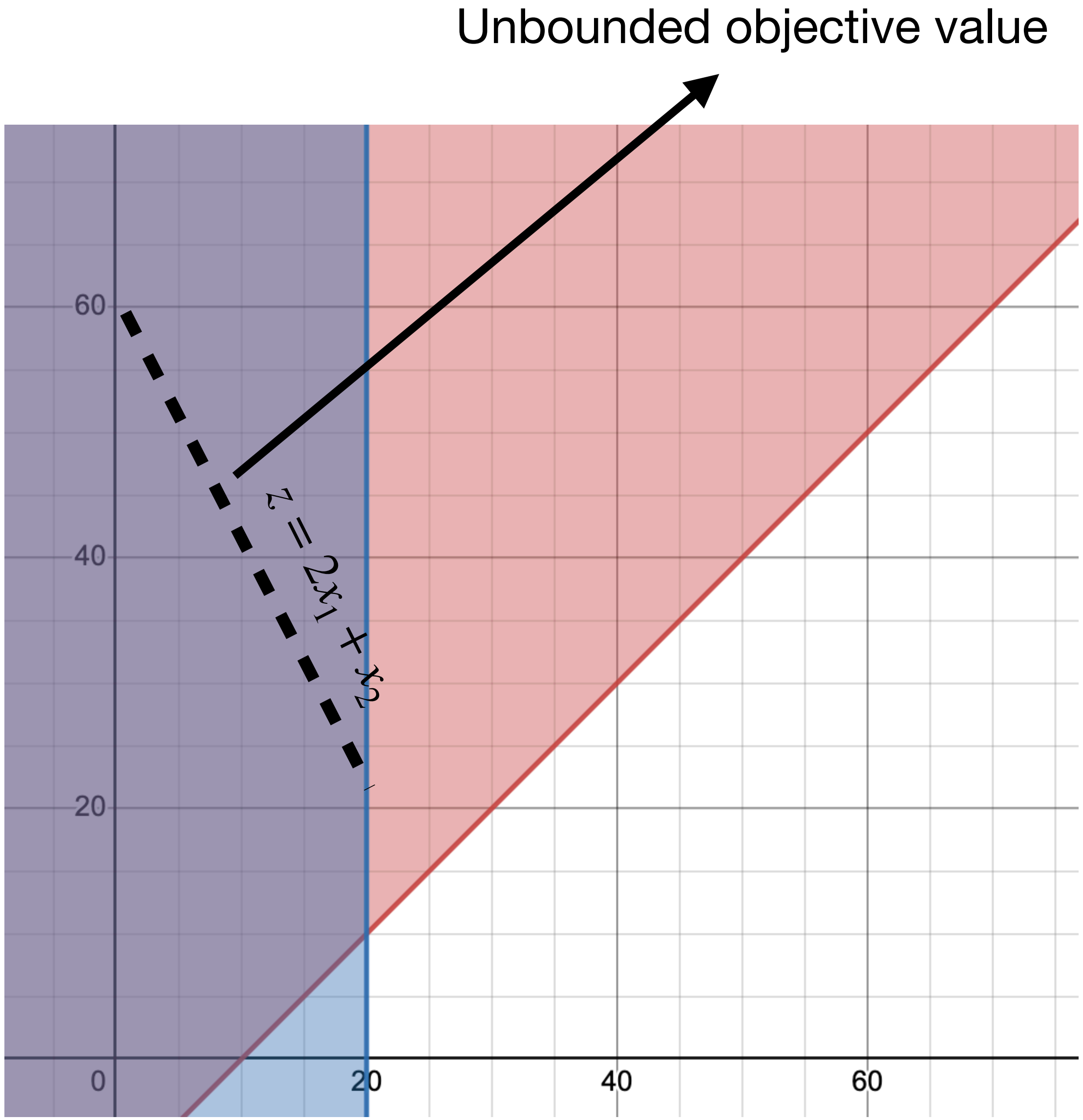
Simplex: Dealing with Unbounded Solution

- What if the solution is unbounded?

Maximise $z = 2x_1 + x_2$

subject to

$$x_1 - x_2 \leq 10$$
$$2x_1 \leq 40$$
$$x_1, x_2 \geq 0$$



Simplex: Dealing with Unbounded Solution

Maximise $z = 2x_1 + x_2$

subject to

$$x_1 - x_2 \leq 10$$
$$2x_1 \leq 40$$
$$x_1, x_2 \geq 0$$

x_1	x_2	x_3	x_4	
1	-1	1	0	10
2	0	0	1	40
-2	-1	0	0	0

Simplex: Dealing with Unbounded Solution

Maximise $z = 2x_1 + x_2$

subject to

$$x_1 - x_2 \leq 10$$
$$2x_1 \leq 40$$
$$x_1, x_2 \geq 0$$

x_1	x_2	x_3	x_4	
1	-1	1	0	10
2	0	0	1	40
-2	-1	0	0	0

Simplex: Dealing with Unbounded Solution

We will reach a tableau where a variable will have a negative z coefficient, but ***all*** the constraint coefficients are ≤ 0

x_1	x_2	x_3	x_4	
1	-1	1	0	10
2	0	0	1	40
-2	-1	0	0	0

Simplex: Dealing with Unbounded Solution

- A variable will have a negative z coefficient
  - An increase in its value will increase the objective value
- All the constraint coefficients are ≤ 0
  - It's value can be increased indefinitely without violating any of the constraints

Simplex: Dealing with Unbounded Solution

- Once we reach a tableau where a variable will have a negative z coefficient, but ***all*** the constraint coefficients are ≤ 0, we can conclude that the LP is unbounded

- If the LP is unbounded, we will reach such a tableau

Simplex: Dealing with Alternative Optima

- What if there are alternative optimal solutions?
  - We would like to find them as it gives more options to the decision makers
    - Choose from many solutions without experiencing deterioration in the objective value

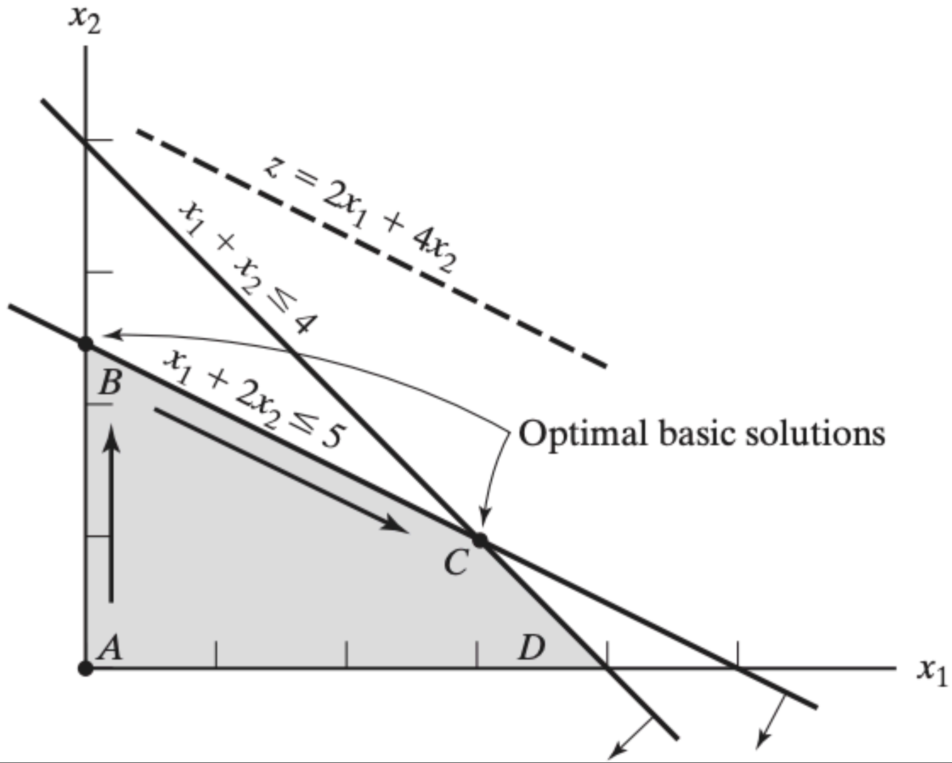
Simplex: Dealing with Alternative Optima

- What if there are alternative optimal solutions?
  - This happens when the objective function is parallel to a nonredundant binding constraint
    - Any point on the line BC is optimal

Maximise $z = 2x_1 + 4x_2$

subject to

$$x_1 + 2x_2 \leq 5$$
$$x_1 + x_2 \leq 4$$
$$x_1, x_2 \geq 0$$



Simplex: Dealing with Alternative Optima

- What if there are alternative optimal solutions?
  - Existence of alternative can be detected in the optimal tableau by examining the z-equation coefficients of the ***non***basic variables
  - The zero coefficient of a nonbasic variable indicates that it can be made basic, altering the values of the basic variables without changing the value of z

Simplex: Dealing with Alternative Optima

x_1	x_2	x_3	x_4	
1	2	1	0	5
1	1	0	1	4
-2	-4	0	0	0

Simplex: Dealing with Alternative Optima

- The zero coefficient of a nonbasic variable indicates that it can be made basic, altering the values of the basic variables without changing the value of z

x_1	x_2	x_3	x_4	
1/2	1	1/2	0	5/2
1/2	0	-1/2	1	3/2
0	0	2	0	10

Optimal Solution: $x_1 = 0,\ x_2 = 5/2,\ x_3 = 0,\ x_4 = 3/2$

Simplex: Dealing with Alternative Optima

- The zero coefficient of a nonbasic variable indicates that it can be made basic, altering the values of the basic variables without changing the value of z

x_1	x_2	x_3	x_4	
0	1	1	-1	1
1	0	-1	2	3
0	0	2	0	10

Optimal Solution: $x_1 = 3,\ x_2 = 1,\ x_3 = 0,\ x_4 = 0$

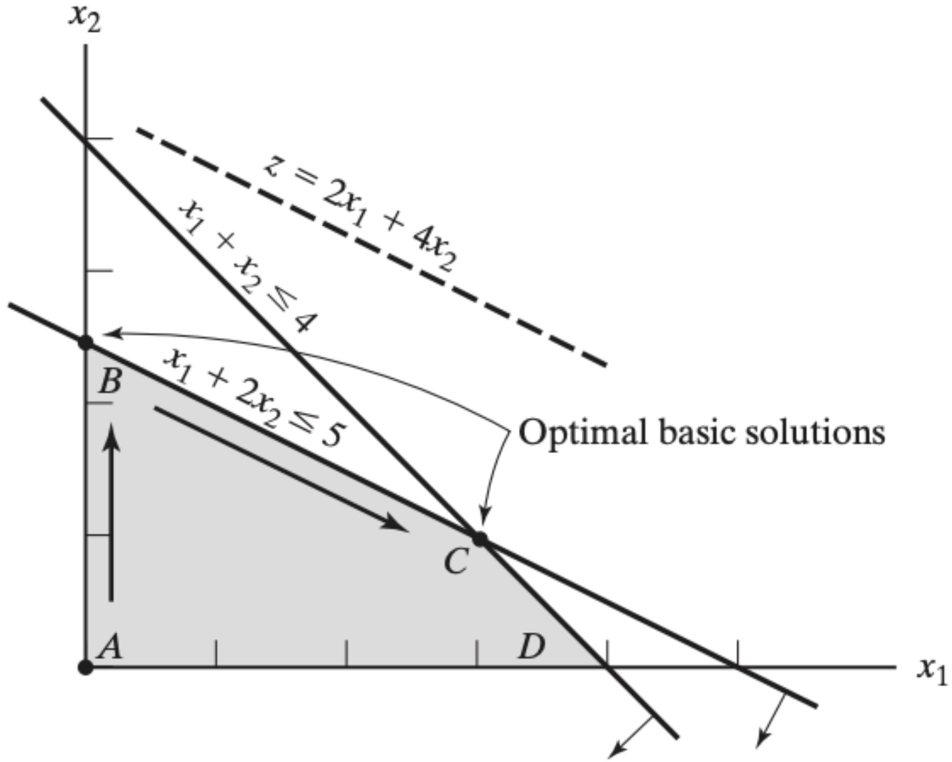
Simplex: Dealing with Alternative Optima

Optimal Solution 1: $x_1 = 0,\ x_2 = 5/2$

Optimal Solution 2: $x_1 = 3,\ x_2 = 1$

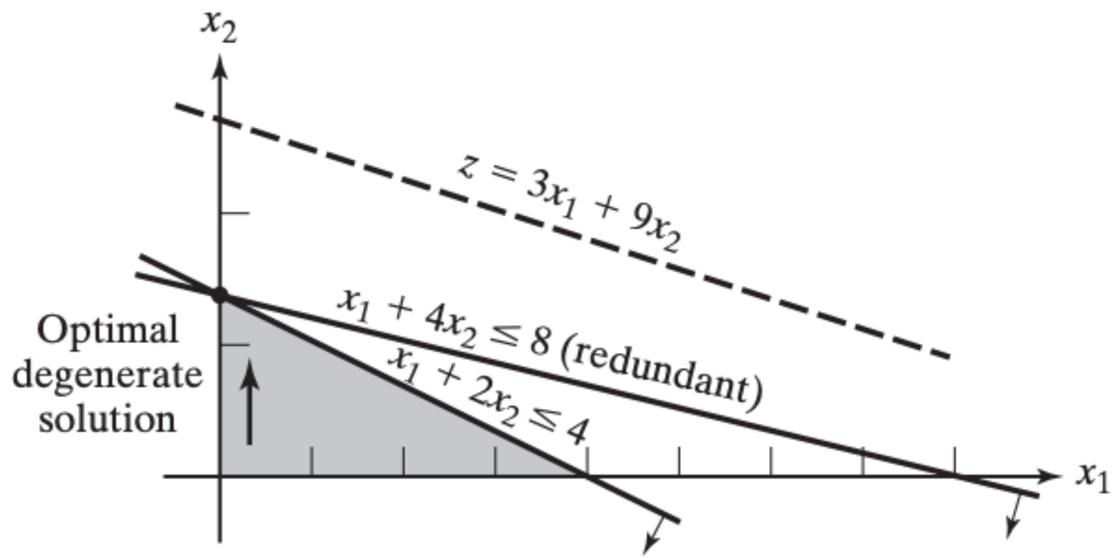
Note that these are two optimal vertices

All points on the line joining these vertices will also be optimal



Simplex Algorithm: Degeneracy

- What is degeneracy?
  - Multiple feasible basis in the Simplex tableau correspond to the same basic feasible solution (vertex)
  - Happens when the value of some basic variables is zero
    - All nonbasic variables have value zero
  - Degeneracy can occur only if we have more than n of the bounding hyperplanes meeting at a vertex in a n dimensional space

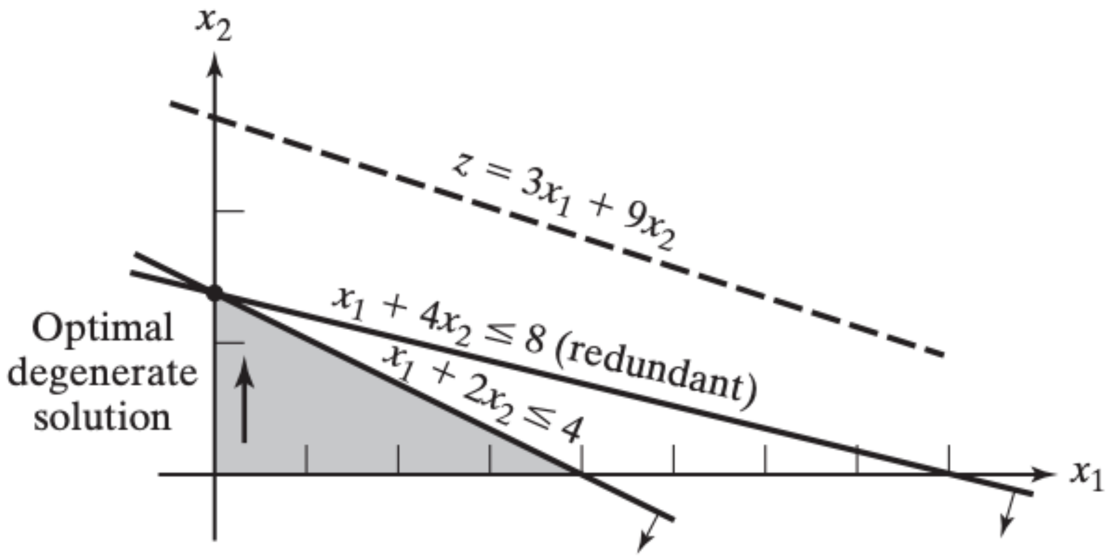


Simplex Algorithm: Degeneracy

Maximise $z = 3x_1 + 9x_2$

subject to

$$x_1 + 4x_2 \leq 8$$
$$x_1 + 2x_2 \leq 4$$
$$x_1, x_2 \geq 0$$



- Three constraints pass through the optimum point ($x_1 = 0,\ x_2 = 2$) in a two-dimensional problem
- This means that one of the constraints is redundant
- Redundancy means that an associated constraint can be removed without changing the solution space

Simplex Algorithm: Degeneracy

- When deciding which variable leaves the basis, a tie for the minimum ratio may occur and can be broken arbitrarily
- When this happens, at least one basic variable will be zero in the next iteration and the new solution is said to be degenerate

x_1	x_2	x_3	x_4	
1	4	1	0	8
1	2	0	1	4
-3	-9	0	0	0

Simplex Algorithm: Degeneracy

- When deciding which variable leaves the basis, a tie for the minimum ratio may occur and can be broken arbitrarily
- When this happens, at least one basic variable will be zero in the next iteration and the new solution is said to be degenerate

x_1	x_2	x_3	x_4	
1/4	1	1/4	0	2
1/2	0	-1/2	1	0
-3/4	0	9/4	0	18

Simplex Algorithm: Degeneracy

- When deciding which variable leaves the basis, a tie for the minimum ratio may occur and can be broken arbitrarily
- When this happens, at least one basic variable will be zero in the next iteration and the new solution is said to be degenerate

x_1	x_2	x_3	x_4	
0	1	1/2	-1/2	2
1	0	-1	2	0
0	0	3/2	3/2	18

Simplex Algorithm: Degeneracy and Cycling

- Degeneracy can lead to cycling
- The simplex method enters a repetitive sequence of iterations, never improving the objective value and never satisfying the optimality condition
- The tableau cycles between the same set of solutions

Does the Simplex Algorithm Terminate?

- Can it end up in a cycle?
  - Objective function value is generally non-decreasing, but what about degenerate vertices with same objective function value
  - Depends on the pivot rules for leaving and entering variables
  - There are lexicographic rules for pivot selection that prevent cycling (not used in practice as they are very slow)
- There are finite number of vertices in the feasible polytope
- If we are not cycling, we visit each vertex at most once with each basis
- So eventually, the algorithm will run out of vertices and terminate

Complexity of LP

- Simplex algorithm (as formulated by Dantzig) can take exponential time in the worst-case
  - Not clear if there is a variant with polynomial time

- Efficient in practice despite its exponential worst-case complexity
  - Led to the development of other measures of complexity
    - Polynomial-time **average-case complexity** under various probability distributions
    - **Smoothed Analysis**: Study the behaviour of worst-case scenarios under small perturbation.
      - The running time of the simplex method on input with noise is polynomial in the number of variables and the magnitude of the perturbations

Are there polynomial time algorithms for LP?

- Yes, there are.
  - Ellipsoid Algorithm
    - Asymptotically better than Simplex, but in practice much worse
  - Interior Point Methods
    - Polynomial time and comparable to Simplex in practice
    - For some LPs, Simplex is faster and for others interior point methods are the winners

What does polynomial time mean?

- Polynomial: Number of bit operations is polynomial in the number of bits required to represent the input
  - Bit size of integer i is $\langle i \rangle = \lceil \log_2(|i| + 1) \rceil + 1$
  - Bit size of rational number $r = \dfrac{p}{q}$ is $\langle p \rangle + \langle q \rangle$
  - Bit size of LP is $\langle A \rangle + \langle b \rangle + \langle c \rangle$ for rational A, b and c
- Note that we are counting bit operations, we are not considering a single arithmetic operation as one step
  - Addition of two k bit integers requires at least k steps

What does polynomial time mean?

- Strongly Polynomial: Number of arithmetic operations is polynomial in the number of variables and constraints, independent of the bit representation of the input

- No strongly polynomial algorithm is known for LP and finding one is a major open problem

Poly-time LP algorithms

- Khachiyan's Algorithm -- Ellipsoid Method

Доклады Академии наук СССР
1979. Том 244, № 5

УДК 519.95

МАТЕМАТИКА

Л. Г. ХАЧИЯН

ПОЛИНОМИАЛЬНЫЙ АЛГОРИТМ В ЛИНЕЙНОМ ПРОГРАММИРОВАНИИ

(Представлено академиком А. А. Дородницыным 4 X 1978)

Рассмотрим систему из $m \geqslant 2$ линейных неравенств относительно $n \geqslant 2$ вещественных переменных $x_1, \ldots, x_j, \ldots, x_n$

$$a_{i1}x_1 + \ldots + a_{in}x_n \leqslant b_i, \quad i = 1, 2, \ldots, m, \tag{1}$$

с целыми коэффициентами a_{ij}, b_i. Пусть

$$L = \left[\sum_{i,j=1}^{m,n} \log_2(|a_{ij}|+1) + \sum_{i=1}^{m} \log_2(|b_i|+1) + \log_2 nm \right] + 1 \tag{2}$$

есть длина входа системы, т. е. число символов 0 и 1, необходимых для записи (1) в двоичной системе счисления.

Mathematical Programming Study 14 (1981) 61–68.
North-Holland Publishing Company

KHACHIYAN'S ALGORITHM FOR LINEAR PROGRAMMING*

Peter GÁCS and Laszlo LOVÁSZ
Computer Science Department, Stanford University, Stanford, CA 94305, U.S.A.

Received 10 October 1979

L.G. Khachiyan's algorithm to check the solvability of a system of linear inequalities with integral coefficients is described. The running time of the algorithm is polynomial in the number of digits of the coefficients. It can be applied to solve linear programs in polynomial time.

Key Words: Linear Programming, Inequalities, Complexity, Polynomial Algorithms.

0. Introduction

L.G. Khachiyan [1, cf. also 2, 3] published a polynomial-bounded algorithm to solve linear programming. These are some notes on this paper. We have ignored his considerations which concern the precision of real computations in order to make the underlying idea clearer; on the other hand, proofs which are missing from his paper are given in Section 2. Let

$$a_i x < b_i \quad (i = 1, \ldots, m,\ a_i \in Z^n,\ b_i \in Z) \tag{1}$$

be a system of *strict* linear inequalities with integral coefficients. We present an algorithm which decides whether or not (1) is solvable, and yields a solution if it is. Define

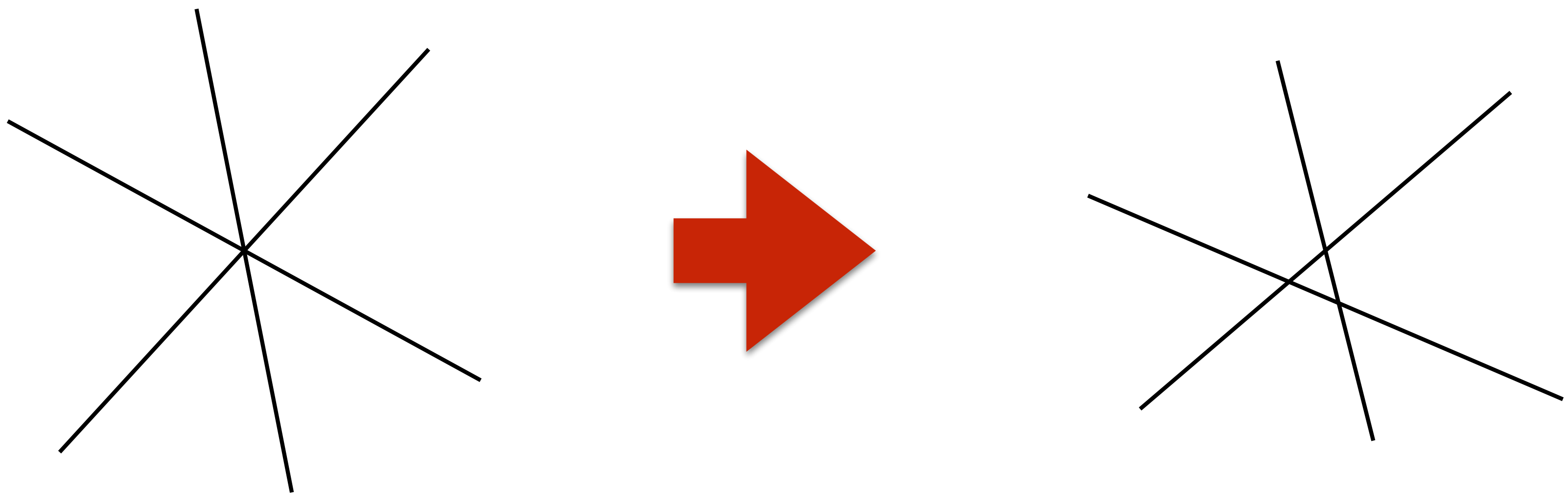
$$L = \sum_{i,j} \log(|a_{ij}|+1) + \sum_i \log(|b_i|+1) + \log nm + 1.$$

L is a lower bound on the space needed to state the problem.

Image Source: https://co-at-work.zib.de/slides/Montag_14.9/Berthold_Theory_II.pdf

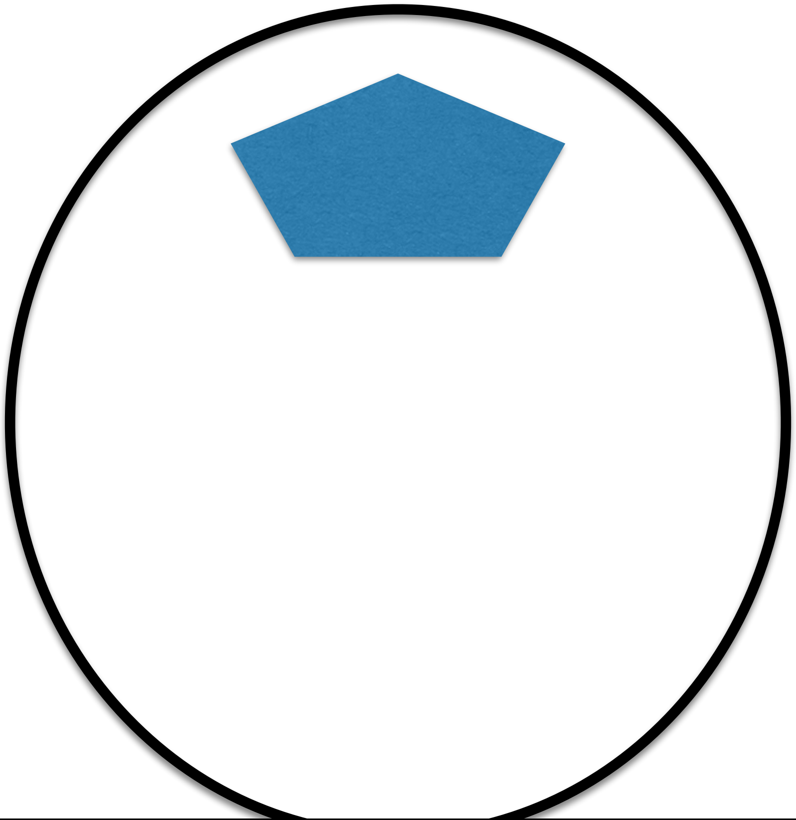
Ellipsoid Algorithm

- Finds whether there exists a feasible solution or not
- First Step: Relax the constraints such that the feasible space has positive volume



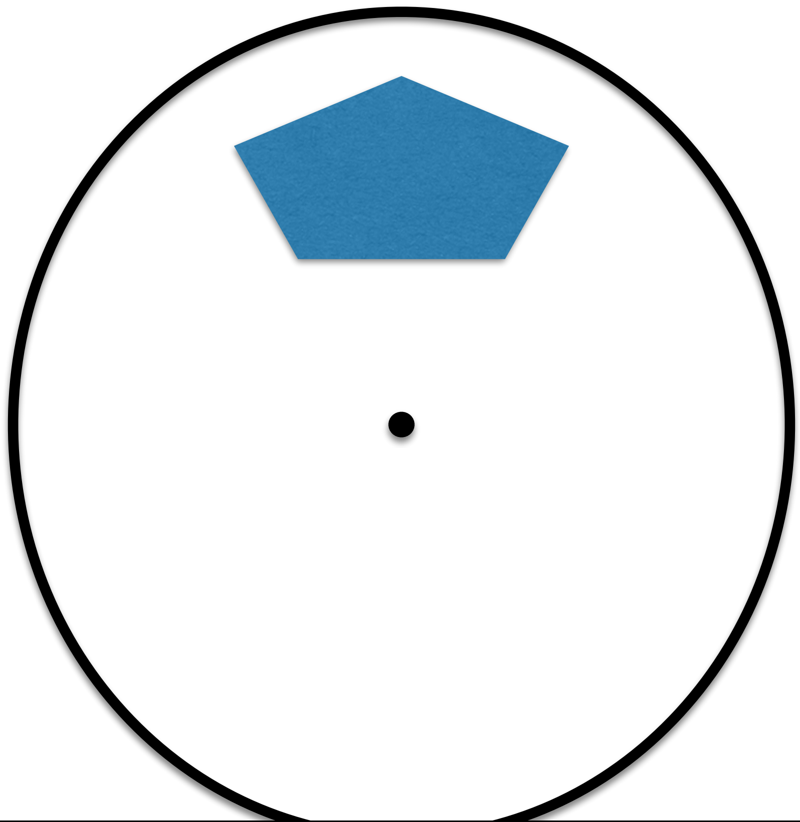
Ellipsoid Algorithm

- Second Step: Bound the feasible set in a large ellipse/ball



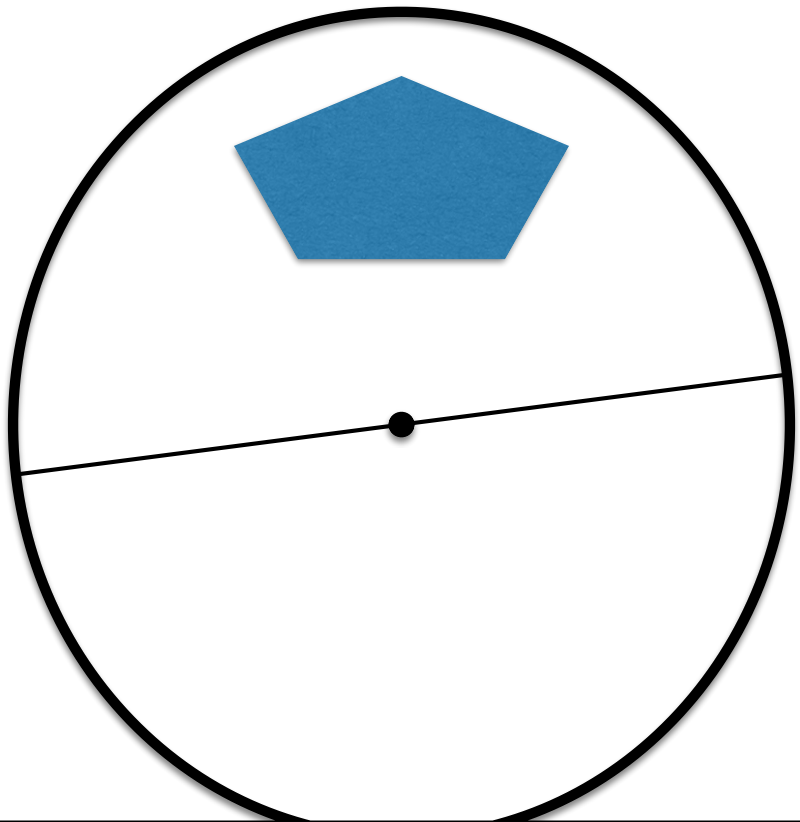
Ellipsoid Algorithm

- Third Step: Consider the centre of ellipsoid. If it is a feasible solution, done.
- Otherwise, find a separating hyperplane of the centre from the feasible set.



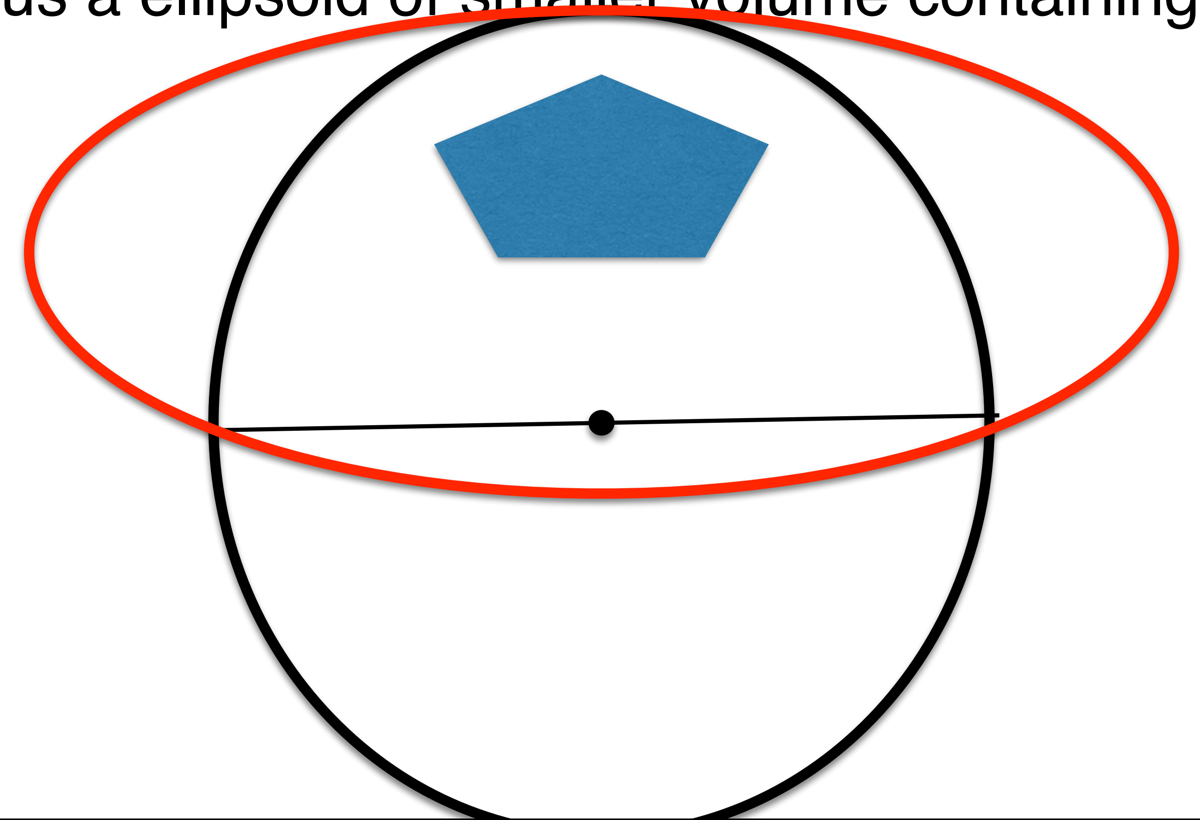
Ellipsoid Algorithm

- Third Step: Consider the centre of ellipsoid. If it is a feasible solution, done.
- Otherwise, find a separating hyperplane of the centre from the feasible set.



Ellipsoid Algorithm

- Third Step: Consider the centre of ellipsoid. If it is a feasible solution, done.
- Otherwise, find a separating hyperplane of the centre from the feasible set.
- That gives us a ellipsoid of smaller volume containing the feasible set



Ellipsoid Algorithm

- Keep repeating till either you find a feasible solution or the volume of the ellipsoid reduces below the volume of feasible solution that you had ensured in the first step

- Running time: Polynomial
  - $O((m + n^2)n^5 \log nU)$ where
  - n is the number of variables
  - m is the number of constraints
  - U is the numerical size of coefficients

Ellipsoid Algorithm

- Better worst-case performance than Simplex
- Is typically slower in practice
  - Degree of the polynomial is quite high
  - Simplex method is slow only on artificially constructed linear programs, which it never encounters in practice
  - Ellipsoid method seldom behaves better than in the worst case
- Can be useful in situations where it is difficult to enumerate all the constraints, but it is easy to find
  - Whether a point is in feasible space
  - A separating oracle to separate a point from feasible space

Poly-time LP algorithms

- Karmarkar's Algorithm -- Barrier Method
- Poly-time with practical impact

[11] **Patent Number: 4,744,028**

[45] **Date of Patent: May 10, 1988**

[54] METHODS AND APPARATUS FOR EFFICIENT RESOURCE ALLOCATION

[75] Inventor: **Narendra K. Karmarkar**, Somerset, N.J.

[73] Assignee: **American Telephone and Telegraph Company, AT&T Bell Laboratories**, Murray Hill, N.J.

[21] Appl. No.: **725,342**

[22] Filed: **Apr. 19, 1985**

[51] **Int. Cl.[4]** **G06F 15/20; H04Q 3/66; H04M 7/00**

[52] **U.S. Cl.** **364/402**

[58] **Field of Search** 364/402; 379/113, 221; 340/524

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,364,115	12/1982	Asai	364/765
4,479,176	10/1984	Grimshaw	369/168
4,481,600	11/1984	Asai	364/765

Karmarkar at Bell Labs: an equation to find a new way through the maze

Folding the Perfect Corner

A young Bell scientist makes a major math breakthrough

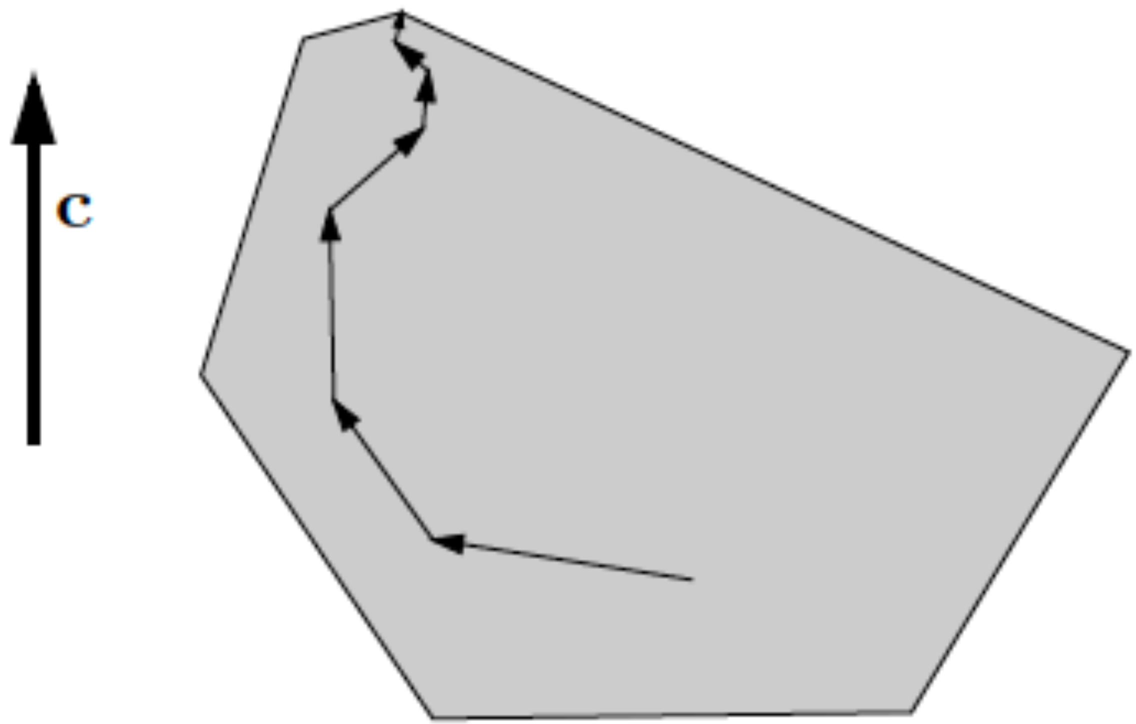
Every day 1,200 American Airlines jets crisscross the U.S., Mexico, Canada and the Caribbean, stopping in 110 cities and bearing over 80,000 passengers. More than 4,000 pilots, copilots, flight personnel, maintenance workers and baggage carriers are shuffled among the flights; a total of 3.6 million gal. of high-octane fuel is burned. Nuts, bolts, altimeters, landing gears and the like must be checked at each destination. And while performing these scheduling gymnastics, the company must keep a close eye on costs, projected revenue and profits.

Like American Airlines, thousands of companies must routinely untangle the myriad variables that complicate the efficient distribution of their resources. Solving such monstrous problems requires the use of an abstruse branch of mathematics known as linear programming. It is the kind of math that has frustrated theoreticians for years, and even the fastest and most powerful computers have had great difficulty juggling the bits and pieces of data. Now Narendra Karmarkar, a 28-year-old Indian-born mathematician at Bell Laboratories in Murray Hill, N.J., after only a years' work has cracked the puzzle of linear programming by devising a new algorithm, a step-by-step mathematical formula. He has translated the procedure into a program that should allow computers to track a greater combination of tasks than ever before and in a fraction of the time.

Unlike most advances in theoretical mathematics, Karmarkar's work will have an immediate and major impact on the real world. "Breakthrough is one of the most abused words in science," says Ronald Graham, director of mathematical sciences at Bell Labs. "But this is one situation where it is truly appropriate."

Before the Karmarkar method, linear equations could be solved only in a cumbersome fashion, ironically known as the simplex method, devised by Mathematician George Dantzig in 1947. Problems are conceived of as giant geodesic domes with thousands of sides. Each corner of a facet on the dome

Interior Point Methods



$$f_\mu(\mathbf{x}) = \mathbf{c}^T\mathbf{x} + \mu \cdot \sum_{i=1}^{m} \ln\left(b_i - \mathbf{a}_i\mathbf{x}\right),$$

We keep decreasing μ till we get very close to the boundary

Summary

- Learning how to find a first feasible solution in Simplex algorithm
- Learning how to deal with infeasibility, unboundedness, alternative optima and degeneracy in Simplex algorithm
- We don't know if there is a variant of Simplex algorithm that runs in polynomial time in the worst-case
    - It is very fast in practice
- Ellipsoid method and some interior point methods are polynomial time in terms of the bit complexity and bit operations