Advanced Algorithms

Floriano Zini

Free University of Bozen-Bolzano
Faculty of Computer Science

Academic Year 2013-2014

Lecture 7 – Linear Programming (cont.)

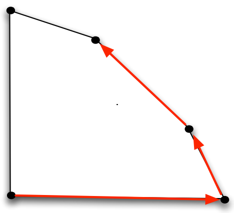
Simplex algorithm

$$\max \ \mathbf{c}^T\mathbf{x}$$
$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

- General schema

```
let v be any vertex of the feasible region
while there is a neighbor v' of v with better
objective value
      set v = v'
```

- At each iteration the simplex algorithm must
  1. check whether the current vertex v is optimal
  2. determine where to move next



Simplex algorithm (cont.)

$$\max \ \mathbf{c}^T\mathbf{x}$$
$$\mathbf{A}\mathbf{x} \leq \mathbf{b}$$
$$\mathbf{x} \geq 0$$

- Tasks 1 and 2 in the previous slide are easy if the vertex is the origin
- Why is the origin so convenient?
  - Suppose that the origin is in the feasible region
  - The origin is a vertex since it is the unique point at which the **n inequalities $\{x_1 \geq 0, \ldots, x_n \geq 0\}$ are tight**
  - The **origin is optimal iff all $c_i \leq 0$**
    - If all $c_i \leq 0$ then we can't hope a better objective value because $\mathbf{x} \geq 0$
    - If there is at least a $c_i > 0$ then we can increase the objective function by increasing the variable x_i
  - So if the origin is not optimal we can **move by increasing some x_i for which $c_i > 0$ until we hit some other constraint**
    - i.e., until **some constraints** other than **$\{x_1 \geq 0, \ldots, x_n \geq 0\}$ becomes tight**
  - At that point, we again have exactly n tight inequalities, so we are at a new vertex
  - We then **transform the coordinate systems** to move the new vertex to the origin
  - And repeat tasks 1 and 2

Simplex Algorithm - Confectionery's problem

max $13x_1 + 23x_2$

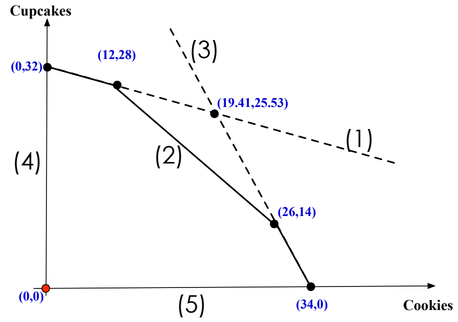
① $5x_1 + 15x_2 \leq 480$

② $4x_1 + 4x_2 \leq 160$

③ $35x_1 + 20x_2 \leq 1190$

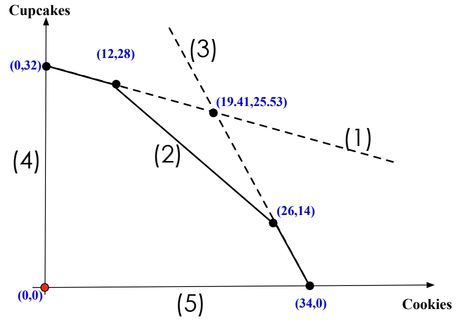
④ $x_1 \geq 0$

⑤ $x_2 \geq 0$



Simplex Algorithm - Confectionery's problem (cont.)

Current vertex:
$(x_1 = 0,\ x_2 = 0)$ origin;
(④, ⑤).
Objective value$= 0$.



max $13x_1 + 23x_2$

① $5x_1 + 15x_2 \leq 480$

② $4x_1 + 4x_2 \leq 160$

③ $35x_1 + 20x_2 \leq 1190$

④ $x_1 \geq 0$

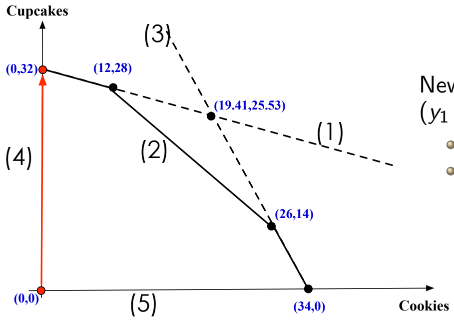
⑤ $x_2 \geq 0$

Simplex Algorithm - Confectionery's problem (cont.)

Pivot 1: x_2 has a positive objective coefficient $\Rightarrow$ increase x_2 until some constraint becomes tight:

- ⑤ $x_2 \geq 0$ is released;
- ① $5x_1 + 15x_2 \leq 480$ becomes tight $\Rightarrow x_2 = \frac{480}{15} = 32$.

max $13x_1 + 23x_2$
① $5x_1 + 15x_2 \leq 480$
② $4x_1 + 4x_2 \leq 160$
③ $35x_1 + 20x_2 \leq 1190$
④ $x_1 \geq 0$
⑤ $x_2 \geq 0$



New vertex (④, ①) and has local coordinates ($y_1 = 0, y_2 = 0$), where:

- $y_1 = x_1$;
- $y_2 = 480 - 5x_1 - 15x_2 \Rightarrow$
  $x_2 = (y_2 - 480 + 5x_1)(-\frac{1}{15}) =$
  $32 - \frac{1}{3}x_1 - \frac{1}{15}y_2 = 32 - \frac{1}{3}y_1 - \frac{1}{15}y_2$

Simplex Algorithm - Confectionery's problem (cont.)

Rewrite the linear program using the new coordinates $\Rightarrow$ substitute $x_1 = y_1$ and $x_2 = 32 - \frac{1}{3}y_1 - \frac{1}{15}y_2$ in the objective function and constraints.

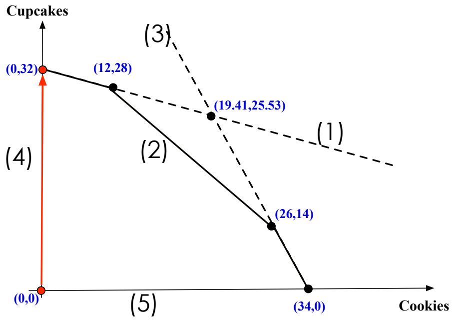
max $13x_1 + 23x_2$; max $13y_1 + 23(32 - \frac{1}{3}y_1 - \frac{1}{15}y_2)$;
max $13y_1 + 736 - \frac{23}{15}y_2 - \frac{23}{3}y_1$; max $\frac{16}{3}y_1 - \frac{23}{15}y_2 + 736$
① $5x_1 + 15x_2 \leq 480$; $5y_1 + 15(32 - \frac{1}{3}y_1 - \frac{1}{15}y_2) \leq 480$;
$5y_1 + 480 - 5y_1 - y_2 \leq 480$; $-y_2 \leq 0$; $y_2 \geq 0$
② $4x_1 + 4x_2 \leq 160$; $4y_1 + 4(32 - \frac{1}{3}y_1 - \frac{1}{15}y_2) \leq 160$;
$4y_1 + 128 - \frac{4}{15}y_2 - \frac{4}{3}y_1 \leq 160$; $\frac{8}{3}y_1 - \frac{4}{15}y_2 \leq 32$
③ $35x_1 + 20x_2 \leq 1190$; $35y_1 + 20(32 - \frac{1}{3}y_1 - \frac{1}{15}y_2) \leq 1190$;
$35y_1 + 640 - \frac{4}{3}y_2 - \frac{20}{3}y_1 \leq 1190$; $\frac{85}{3}y_1 - \frac{4}{3}y_2 \leq 550$
④ $x_1 \geq 0$; $y_1 \geq 0$
⑤ $x_2 \geq 0$; $32 - \frac{1}{3}y_1 - \frac{1}{15}y_2 \geq 0$; $\frac{1}{3}y_1 + \frac{1}{15}y_2 \leq 32$

max $\frac{16}{3}y_1 - \frac{23}{15}y_2 + 736$
① $y_2 \geq 0$
② $\frac{8}{3}y_1 - \frac{4}{15}y_2 \leq 32$
③ $\frac{85}{3}y_1 - \frac{4}{3}y_2 \leq 550$
④ $y_1 \geq 0$
⑤ $\frac{1}{3}y_1 + \frac{1}{15}y_2 \leq 32$

Simplex Algorithm - Confectionery's problem (cont.)

Current vertex:
$(y_1 = 0,\ y_2 = 0)$ origin;
(④, ①).
Objective value$= 736$.

max $\frac{16}{3}y_1 - \frac{23}{15}y_2 + 736$
① $y_2 \geq 0$
② $\frac{8}{3}y_1 - \frac{4}{15}y_2 \leq 32$
③ $\frac{85}{3}y_1 - \frac{4}{3}y_2 \leq 550$
④ $y_1 \geq 0$
⑤ $\frac{1}{3}y_1 + \frac{1}{15}y_2 \leq 32$

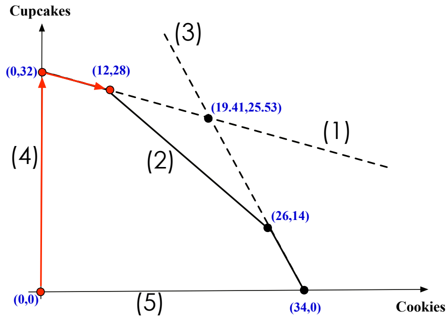


Simplex Algorithm - Confectionery's problem (cont.)

Pivot 2: y_1 has a positive objective coefficient $\Rightarrow$ increase y_1 until some constraint becomes tight:

- ④ $y_1 \geq 0$ is released;
- ② $\frac{8}{3}y_1 - \frac{4}{15}y_2 \leq 32$ becomes tight $\Rightarrow y_1 = 32 \cdot \frac{3}{8} = 12$.

max $\frac{16}{3}y_1 - \frac{23}{15}y_2 + 736$
① $y_2 \geq 0$
② $8/3*y_1 - 4/15*y_2 \leq 32$
③ $\frac{85}{3}y_1 - \frac{4}{3}y_2 \leq 550$
④ $y_1 \geq 0$
⑤ $\frac{1}{3}y_1 + \frac{1}{15}y_2 \leq 32$



New vertex (②, ①) and has local coordinates $(z_1 = 0, z_2 = 0)$, where:

- $z_2 = y_2$
- $z_1 = 32 - \frac{8}{3}y_1 + \frac{4}{15}y_2 \Rightarrow$
  $y_1 = (z_1 - 32 - \frac{4}{15}y_2) \cdot (-\frac{3}{8}) =$
  $12 - \frac{3}{8}z_1 + \frac{1}{10}y_2 = 12 - \frac{4}{8}z_1 + \frac{1}{10}z_2$;

Simplex Algorithm - Confectionery's problem (cont.)

Rewrite the linear program using the new coordinates ⇒ substitute $y_2 = z_2$ and $y_1 = 12 - \frac{3}{8}z_1 + \frac{1}{10}z_2$ in the objective function and constraints.

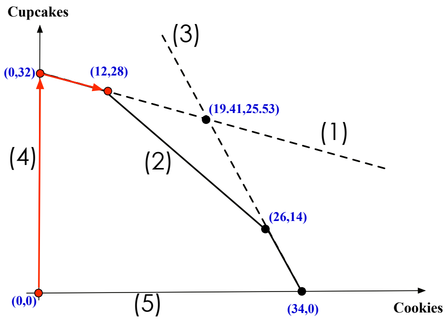
max $-2z_1 - z_2 + 800$
① $z_2 \geq 0$
② $z_1 \geq 0$
③ $-\frac{85}{8}z_1 + \frac{9}{2}z_2 \leq 210$
④ $\frac{3}{8}z_1 - \frac{1}{10}z_2 \leq 12$
⑤ $-\frac{1}{8}z_1 + \frac{1}{10}z_2 \leq 28$

max $\frac{16}{3}y_1 - \frac{23}{15}y_2 + 736$; max $\frac{16}{3} \cdot (12 - \frac{3}{8}z_1 + \frac{1}{10}z_2) - \frac{23}{15}z_2 + 736$;
max $64 - 2z_1 + \frac{8}{15}z_2 - \frac{23}{15}z_2 + 736$; max $-2z_1 - y_2 + 800$
① $y_2 \geq 0$; $z_2 \geq 0$
② $\frac{8}{3}y_1 - \frac{4}{15}y_2 \leq 32$; $\frac{8}{3} \cdot (12 - \frac{3}{8}z_1 + \frac{1}{10}z_2) - \frac{4}{15}z_2 \leq 32$;
$32 - z_1 + \frac{4}{15}z_2 - \frac{4}{15}z_2 \leq 32$; $-z_1 \leq 0$; $z_1 \geq 0$
③ $\frac{85}{3}y_1 - \frac{4}{3}y_2 \leq 550$; $\frac{85}{3} \cdot (12 - \frac{3}{8}z_1 + \frac{1}{10}z_2) - \frac{4}{3}y_2 \leq 550$;
$340 - \frac{85}{8}z_1 + \frac{17}{6}z_2 - \frac{4}{3}y_2 \leq 550$; $-\frac{85}{8}z_1 + \frac{9}{2}z_2 \leq 210$
④ $y_1 \geq 0$; $12 - \frac{3}{8}z_1 + \frac{1}{10}z_2 \geq 0$; $\frac{3}{8}z_1 - \frac{1}{10}z_2 \leq 12$
⑤ $\frac{1}{3}y_1 + \frac{1}{15}y_2 \leq 32$; $\frac{1}{3} \cdot (12 - \frac{3}{8}z_1 + \frac{1}{10}z_2) + \frac{1}{15}z_2 \leq 32$;
$4 - \frac{1}{8}z_1 + \frac{1}{30}z_2 + \frac{1}{15}z_2 \leq 32$; $-\frac{1}{8}z_1 + \frac{1}{10}z_2 \leq 28$

Simplex Algorithm - Confectionery's problem (cont.)

Current vertex:
$(z_1 = 0,\ z_2 = 0)$ origin;
(②, ①).
Objective value$= 800$.

max $-2z_1 - z_2 + 800$
① $z_2 \geq 0$
② $z_1 \geq 0$
③ $-\frac{85}{8}z_1 + \frac{9}{2}z_2 \leq 210$
④ $\frac{3}{8}z_1 - \frac{1}{10}z_2 \leq 12$
⑤ $-\frac{1}{8}z_1 + \frac{1}{10}z_2 \leq 28$



All the coefficients of the objective function are negative ➔

STOP!! Maximum found!

Simplex Algorithm - Confectionery's problem (cont.)

- How to compute the optimal solution point?
- Solve the system of equations (1), (2) of the original LP

max $13x_1 + 23x_2$

① $5x_1 + 15x_2 \leq 480$

② $4x_1 + 4x_2 \leq 160$

③ $35x_1 + 20x_2 \leq 1190$

④ $x_1 \geq 0$

⑤ $x_2 \geq 0$

$$\begin{array}{l} 5x_1 + 15x_2 = 480 \\ 4x_1 + 4x_2 = 160 \end{array} \Rightarrow \begin{array}{l} 5x_1 + 15x_2 = 480 \\ x_1 = \frac{1}{4} \cdot (160 - 4x_2) \end{array}$$

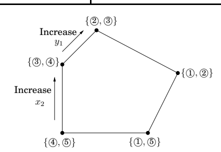
$$\begin{array}{l} 5x_1 + 15x_2 = 480 \\ x_1 = 40 - x_2 \end{array} \Rightarrow \begin{array}{l} 5 \cdot (40 - x_2) + 15x_2 = 480 \\ x_1 = 40 - x_2 \end{array}$$

$$\begin{array}{l} 200 - 5x_2 + 15x_2 = 480 \\ x_1 = 40 - x_2 \end{array} \Rightarrow \begin{array}{l} 10x_2 = 280 \\ x_1 = 40 - x_2 \end{array}$$

$$\begin{array}{l} x_2 = 28 \\ x_1 = 12 \end{array}$$

Simplex Algorithm – Another problem

Initial LP: $\max\ 2x_1 + 5x_2$ $2x_1 - x_2 \leq 4$ ① $x_1 + 2x_2 \leq 9$ ② $-x_1 + x_2 \leq 3$ ③ $x_1 \geq 0$ ④ $x_2 \geq 0$ ⑤	*Current vertex:* {④, ⑤} (origin). *Objective value:* 0. *Move:* increase x_2. ⑤ is released, ③ becomes tight. Stop at $x_2 = 3$. New vertex {④, ③} has local coordinates (y_1, y_2): $y_1 = x_1, \quad y_2 = 3 + x_1 - x_2$
Rewritten LP: $\max\ 15 + 7y_1 - 5y_2$ $y_1 + y_2 \leq 7$ ① $3y_1 - 2y_2 \leq 3$ ② $y_2 \geq 0$ ③ $y_1 \geq 0$ ④ $-y_1 + y_2 \leq 3$ ⑤	*Current vertex:* {④, ③}. *Objective value:* 15. *Move:* increase y_1. ④ is released, ② becomes tight. Stop at $y_1 = 1$. New vertex {②, ③} has local coordinates (z_1, z_2): $z_1 = 3 - 3y_1 + 2y_2, \quad z_2 = y_2$
Rewritten LP: $\max\ 22 - \frac{7}{3}z_1 - \frac{1}{3}z_2$ $-\frac{1}{3}z_1 + \frac{5}{3}z_2 \leq 6$ ① $z_1 \geq 0$ ② $z_2 \geq 0$ ③ $\frac{1}{3}z_1 - \frac{2}{3}z_2 \leq 1$ ④ $\frac{1}{3}z_1 + \frac{1}{3}z_2 \leq 4$ ⑤	*Current vertex:* {②, ③}. *Objective value:* 22. *Optimal:* all $c_i < 0$. Solve ②, ③ (in original LP) to get optimal solution $(x_1, x_2) = (1, 4)$.



Variants of linear programming

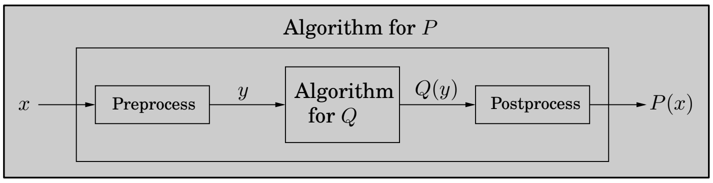
- A general linear program has many degrees of freedom

1. It can be either a maximization or a minimization problem
2. Its constraints can be equations and/or inequalities
3. The variables are often restricted to be nonnegative, but they can also be unrestricted in sign

- We will now show that these various LP options can all be **reduced** to one another via simple **transformations**

Reductions

- We want to solve **problem *P***
- We already have an **algorithm** that solves **problem *Q***
- If any **subroutine** for ***Q*** can also be used to **solve *P***, we say ***P* reduces to *Q***
- Often, *P* is solvable by a single call to *Q*'s subroutine
  - any instance *x* of *P* can be transformed into an instance *y* of *Q* such that *P(x)* can be deduced from *Q(y)*



Variants of linear programming (cont.)

- To turn a maximization problem into a minimization (or vice versa), multiply the coefficients of the objective function by -1

  $max\ c_1*x_1 + c_2*x_2 + \ldots + c_n*x_n \rightarrow min\ -c_1*x_1 - c_2*x_2 - \ldots - c_n*x_n$

- To turn an inequality constraint like $\sum_{i=1}^{n} a_i x_i \leq b$ into an equation, introduce a new **slack variable *s*** and use $\sum_{i=1}^{n} a_i x_i + s = b \quad s \geq 0$
  - vector $(x_1, \ldots, x_n)$ satisfies the original inequality constraint if and only if there is some $s \geq 0$ for which it satisfies the new equality constraint
- To change an equality constraint into inequalities rewrite $a*x=b$ as the equivalent pair of constraints $a*x \leq b$ and $a*x \geq b$
- To deal with a variable x that is unrestricted in sign
  - Introduce two nonnegative variables, $x^+, x^- \geq 0$
  - Replace x, wherever it occurs in the constraints or the objective function, by $x^+ - x^-$

Standard form of Linear Programs

- **Any LP** (maximization or minimization, with both inequalities and equations, and with both nonnegative and unrestricted variables) **can be reduced into** an much more constrained LP that we call the **standard form**
  - the **variables** are all **nonnegative**
  - the **constraints** are all **equations**
  - the **objective function** is to be **minimized**

Standard version

$$\min\ c_1x_1 + c_2x_2 + \ldots + c_nx_n$$
$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2$$
$$\ldots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m$$
$$x_1, x_2, \ldots, x_n \geq 0$$

Matrix version

$$\min\ c^T x$$
$$Ax = b$$
$$x \geq 0$$

Standard form of Linear Programs (cont.)

- **Example 1**

$$\begin{aligned} \max\ & x_1 + 6x_2 \\ & x_1 \leq 200 \\ & x_2 \leq 300 \\ & x_1 + x_2 \leq 400 \\ & x_1, x_2 \geq 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} \min\ & -x_1 - 6x_2 \\ & x_1 + s_1 = 200 \\ & x_2 + s_2 = 300 \\ & x_1 + x_2 + s_3 = 400 \\ & x_1, x_2, s_1, s_2, s_3 \geq 0 \end{aligned}$$

- **Example 2**
  - Confectionary's problem in standard form

Original formulation:

x_1:cookies, x_2: cupcakes

$$\begin{aligned} & \max\ 13x_1 + 23x_2 \\ & 5x_1 + 15x_2 \leq 480 \\ & 4x_1 + 4x_2 \leq 160 \\ & 35x_1 + 20x_2 \leq 1190 \\ & x_1, x_2 \geq 0 \end{aligned}$$

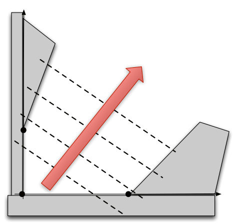
Standard form:
- translate the objective function
- add slack variable to convert each inequality to an equality

$$\begin{aligned} & \min\ -13x_1 - 23x_2 \\ & 5x_1 + 15x_2 + s_1 = 480 \\ & 4x_1 + 4x_2 + s_2 = 160 \\ & 35x_1 + 20x_2 + s_3 = 1190 \\ & x_1, x_2, s_1, s_2, s_3 \geq 0 \end{aligned}$$

Simplex Algorithm - pay attention

There are cases when an LP does not have an optimum

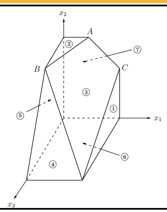
- **Unfeasibility**
  - the constraints are so tight that it is impossible to satisfy all of them
    - For instance: $x \leq 1, x \geq 2$
- **Unboundedness**
  - The constraints are so loose that it is possible to achieve arbitrarily high objective values
    - For instance: $\max x_1 + x_2\ x_1, x_2 \geq 0$



Simplex Algorithm - pay attention (cont.)

Degeneracy

- In the polyhedron on the right, vertex *B* is **degenerate**
- **Geometrically**, this means that it is the **intersection of more than *n=3* faces** of the polyhedron: (2), (3), (4), and (5)
- **Algebraically**, it means that if we choose any one of four sets of three inequalities {(2),(3),(4)}, {(2),(3),(5)}, {(2),(4),(5)}, {(3),(4),(5)} and solve the corresponding system of three linear equations, we'll get the same solution: (0, 300, 100)
- **Serious problem**: simplex may return a suboptimal degenerate vertex because all its neighbors are identical to it and thus have no better objective
- If we modify simplex so that it detects degeneracy and continues to hop from vertex to vertex despite lack of any improvement in the cost, it may end up looping forever
- One way to fix this is by a **perturbation**
  - Change each b_i by a tiny random amount to $b_i \pm \varepsilon_i$
  - This doesn't change the essence of the LP since the ε_i's are tiny, but it has the effect of differentiating between the solutions of the linear systems
  - To see why geometrically, imagine that the four planes (2),(3),(4),(5) were shaken a little
  - Vertex *B* splits into two vertices, very close to one another



$$\max \; x_1 + 6x_2 + 13x_3$$

$$\begin{aligned} x_1 &\le 200 & ① \\ x_2 &\le 300 & ② \\ x_1 + x_2 + x_3 &\le 400 & ③ \\ x_2 + 3x_3 &\le 600 & ④ \\ x_1 &\ge 0 & ⑤ \\ x_2 &\ge 0 & ⑥ \\ x_3 &\ge 0 & ⑦ \end{aligned}$$

Simplex Algorithm - pay attention (cont.)

The starting vertex

- How do we find a vertex at which to start simplex if the origin is not in the feasible region?
- In general it is find by applying the simplex algorithm to a modified version of the original program
- The possible results of this application are
  - a basic feasible solution is found
  - the feasible region is empty: the linear program in infeasible
- If the starting point does exist, the simplex algorithm is applied on the original problem starting from the found starting point

Simplex Algorithm - running time

- Let *n* be the number of variables and *m* be the number of inequality constraints
- Suppose the current vertex is *v*
  - by definition it is the unique point at which *n* inequality constraints are satisfied with equality
- Each of its neighbors shares *n-1* of these inequalities
  - *So v* has at most *n**m* neighbors
- The cost for choosing the most promising neighbor is *O(m*n)*
- The total number of iteration is at most $\binom{m+n}{n}$ (the maximum number of vertices) that is exponential in *n*
- Thus, the simplex is **exponential!**
- **However, real-world scenarios do not originate exponential problems and this makes simplex very valuable and widely used**

Integer linear programming

- The **optimum solution** of a LP might turn out to be **fractional**
- This number is usually **rounded** to the upper or lower integer in order to make sense
  - the overall **cost increases**
- In **many examples** rounding is **unlikely** to **affect things too much**
- **But** there are **LPs** where **rounding has to be made very carefully** in order to end up with an integer solution of reasonable quality
- In general, there is a **tension** between **the ease of obtaining fractional solutions** and the **desirability of integer ones**
- Finding the optimum integer solution of an LP is an important but very hard problem, called **integer linear programming**