DM545/DM871

Linear and Integer Programming

Lecture 4

Exception Handling and Initialization

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Simplex: Exception Handling, Overview

Solution of an LP problem:

a. $F \neq \emptyset$ and $\nexists$ solution

b. $F \neq \emptyset$ and $\exists$ solution

   i) **one solution**

   ii) infinite solutions

c. $F = \emptyset$

Handling exceptions in the Simplex Method

1. Unboundedness
2. More than one solution
3. Degeneracies
   - benign
   - cycling
4. Infeasible starting
   Phase I + Phase II

Outline

1. Exception Handling

2. Initialization

Outline

1. Exception Handling

2. Initialization

Unboundedness

$$\begin{aligned} \max\ 2x_1 &+ x_2 \\ x_2 &\leq 5 \\ -x_1 + x_2 &\leq 1 \\ x_1, x_2 &\geq 0 \end{aligned}$$

- Initial tableau

```
|    | x1 | x2 | x3 | x4 | -z | b |
|----+----+----+----+----+----+---|
| x3 |  0 |  1 |  1 |  0 |  0 | 5 |
| x4 | -1 |  1 |  0 |  1 |  0 | 1 |
|----+----+----+----+----+----+---|
|    |  2 |  1 |  0 |  0 |  1 | 0 |
```

- x_2 entering, x_4 leaving

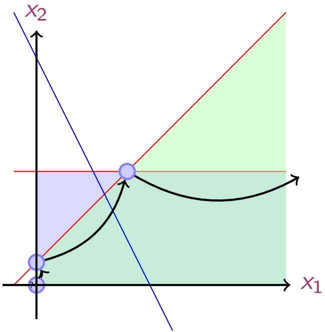
```
|             | x1 | x2 | x3 | x4 | -z |  b |
|-------------+----+----+----+----+----+----|
| II'=II-I'   |  1 |  0 |  1 | -1 |  0 |  4 |
| I'=I        | -1 |  1 |  0 |  1 |  0 |  1 |
|-------------+----+----+----+----+----+----|
| III'=III-I' |  3 |  0 |  0 | -1 |  1 | -1 |
```

$-x_1 + x_2 + x_4 = 1$, x_1 can increase without restriction, $\theta = \min\{\frac{b_i}{a_{is}} : a_{is} > 0, i = 1\dots, n\}$

- x_1 entering, x_3 leaving

```
|                | x1 | x2 | x3 | x4 | -z |   b |
|--------------+----+----+----+----+----+-----|
| I'=I          |  1 |  0 |  1 | -1 |  0 |   4 |
| II'=II+I'     |  0 |  1 |  1 |  0 |  0 |   5 |
|--------------+----+----+----+----+----+-----|
| III'=III-3I'  |  0 |  0 | -3 |  2 |  1 | -13 |
```

  x_4 was already in basis but for both I and II ($x_2 + 0x_4 = 5$), x_4 can increase arbitrarily



∞ solutions

$$
\begin{array}{lrcrcl}
\max & x_1 & + & x_2 & & \\
 & 5x_1 & + & 10x_2 & \leq & 60 \\
 & 4x_1 & + & 4x_2 & \leq & 40 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$

- Initial tableau

	x1	x2	x3	x4	-z	b
x3	5	10	1	0	0	60
x4	4	4	0	1	0	40
	1	1	0	0	1	0

- x_2 enters, x_3 leaves

	x1	x2	x3	x4	-z	b
I'=I/10	1/2	1	1/10	0	0	6
II'=II-4Ix4	2	0	-2/5	1	0	16
III'=III-I	1/2	0	-1/6	0	1	-6

- x_1 enters, x_4 leaves

```
|                 | x1 | x2 | x3   | x4   | -z |   b |
|-----------------+----+----+------+------+----+-----|
| I'=I-II'/2      |  0 |  1 | 1/5  | -1/4 |  0 |   2 |
| II'=II/2        |  1 |  0 | -1/5 | 1/2  |  0 |   8 |
|-----------------+----+----+------+------+----+-----|
| III'=III-II'/2 |  0 |  0 | 0    | -1/4 |  1 | -10 |
```

  $\mathbf{x} = (8, 2, 0, 0), z = 10$

  nonbasic variables typically have reduced costs $\neq 0$. Here x_3 has r.c. $= 0$. Let's make it enter the basis

- x_3 enters, x_2 leaves

```
|                 | x1 | x2 | x3 | x4   | -z |   b |
|-----------------+----+----+----+------+----+-----|
| I'=5I           |  0 |  5 |  1 | -5/4 |  0 |  10 |
| II'=II+I'/5     |  1 |  1 |  0 | 4    |  0 |  10 |
|-----------------+----+----+----+------+----+-----|
| III'=III        |  0 |  0 |  0 | -1/4 |  1 | -10 |
```

  $\mathbf{x} = (10, 0, 10, 0), z = 10$

  There are 2 optimal solutions $\rightsquigarrow$ all their convex combinations are optimal solutions (from the proof of the fundamental theorem of LP) $\rightsquigarrow$

$$\mathbf{x} = \sum_i \alpha_i \mathbf{x}_i$$

$$\alpha_i \geq 0$$

$$\sum_i \alpha_i = 1$$

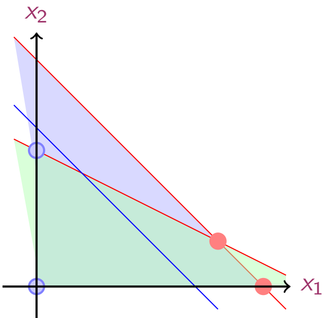
$$\mathbf{x}_1^T = [8, 2, 0, 0]$$

$$\mathbf{x}_2^T = [10, 0, 10, 0]$$

$$\alpha_1 = \alpha$$

$$\alpha_2 = 1 - \alpha$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \alpha \begin{bmatrix} 8 \\ 2 \\ 0 \\ 0 \end{bmatrix} + (1 - \alpha) \begin{bmatrix} 10 \\ 0 \\ 10 \\ 0 \end{bmatrix}$$



$$x_1 = 8\alpha + 10(1 - \alpha)$$

$$x_2 = 2\alpha$$

$$x_3 = 10(1 - \alpha)$$

$$x_4 = 0$$

Degeneracy

$$
\begin{array}{rrrrl}
\max & & & x_2 & \\
& -x_1 & + & x_2 & \leq 0 \\
& x_1 & & & \leq 2 \\
& & & x_1, x_2 & \geq 0
\end{array}
$$

- Initial tableau

```
|    | x1 | x2 | x3 | x4 | -z | b |
|----+----+----+----+----+----+---|
| x3 | -1 |  1 |  1 |  0 |  0 | 0 |
| x4 |  1 |  0 |  0 |  1 |  0 | 2 |
|----+----+----+----+----+----+---|
|    |  0 |  1 |  0 |  0 |  1 | 0 |
```

  $b_i = 0$ (one basic var. is zero) might lead to cycling

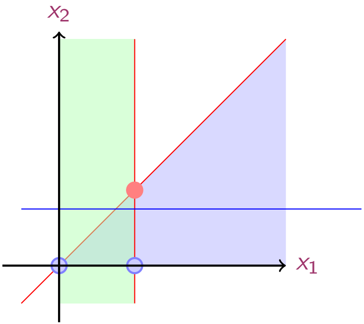
- degenerate pivot step: not improving, the entering variable stays at zero

```
|   | x1 | x2 | x3 | x4 | -z | b |
|---+----+----+----+----+----+---|
|   | -1 |  1 |  1 |  0 |  0 | 0 |
|   |  1 |  0 |  0 |  1 |  0 | 2 |
|---+----+----+----+----+----+---|
|   |  1 |  0 | -1 |  0 |  1 | 0 |
```

- now nondegenerate:

```
|   | x1 | x2 | x3 | x4 | -z |  b |
|---+----+----+----+----+----+----|
|   |  0 |  1 |  0 |  1 |  0 |  2 |
|   |  1 |  0 |  0 |  1 |  0 |  2 |
|---+----+----+----+----+----+----|
|   |  0 |  0 | -1 | -1 |  1 | -2 |
```

$x_1 = 2, x_2 = 2, z = 2$



$\geq n+1$ constraints meet at a vertex

Def: An improving variable is one with positive reduced cost

Def: A degenerate iteration is one in which the objective function does not increase.
Def: The simplex method cycles if the same tableau appears in two iterations.

Degenerate conditions may appear often in practice but cycling is rare. (see compendium for the smallest possible example)

Theorem

If the simplex fails to terminate, then it must cycle.

Proof:

- there is a finite number of basis and simplex chooses to always increase the cost
- hence the only situation for not terminating is that a basis must appear again and iterations in between are degenerate. Two tabelaux with the same basis are the same (related to uniqueness of basic solutions)

Pivot Rules

Some pivoting rules can prevent the occurrence of cycling alltogether.

So far we chose an arbitrary improving variable to enter. Rules for breaking ties in selecting entering improving variables (more important than selecting leaving variables)

- Largest Coefficient: the improving var with largest coefficient in last row of the tableau. Original Dantzig's rule, can cycle

- Largest increase: absolute improvement: $\text{argmax}_j\{c_j\theta_j\}$ computationally more costly

- Steepest edge the improving var that if entering in the basis moves the current basic feasible sol in a direction closest to the direction of the vector $\mathbf{c}$ (ie, maximizes the cosine of the angle between the two vectors):

$$\mathbf{a}\cdot\mathbf{b} = \|\mathbf{a}\|\,\|\mathbf{b}\|\cos\theta \quad \Longrightarrow \quad \max_{\mathbf{x}_{\text{new}}} \frac{\mathbf{c}^T(\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}})}{\cancel{\|\mathbf{c}\|}\,\|\,\mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}\,\|}$$

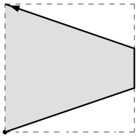
- Bland's rule (smallest-subscript rule) chooses the improving var with the lowest index and, if there are more than one leaving variable, the one with the lowest index.
  Prevents cycling but is slow (no smart choice for entering variable)

- Random edge select var uniformly at random among the improving ones

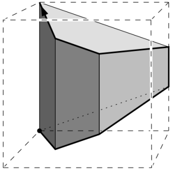
- Perturbation method: perturb values of b_i terms to avoid $b_i = 0$, which must occur for cycling.
  To avoid cancellations: $0 < \epsilon_m \ll \epsilon_{m-1} \ll \cdots \ll \epsilon_1 \ll 1$
  It affects the choice of the leaving variable
  Can be shown to be the same as lexicographic method, which prevents cycling

Efficiency of Simplex Method

- Trying all points is $\approx 4^m$
- In practice between $2m$ and $3m$ iterations
- Klee and Minty 1978 constructed an example that requires $2^n - 1$ iterations in $\mathbb{R}^n$:



$n = 2$

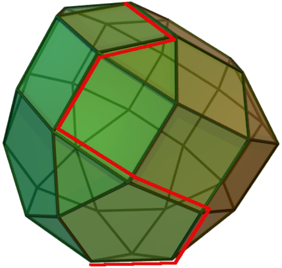


$n = 3$

- random shuffle of indexes + lowest index for entering + lexicographic for leaving: expected iterations $< e^{C\sqrt{n \ln n}}$

Efficiency of Simplex Method

- unknown if there exists a pivot rule that leads to polynomial time.
- Clairvoyant's rule: shortest possible sequence of steps
  Hirsh conjecture $O(n-d)$ for an n-facet polytope in d-dimensional Euclidean space but best known $n^{1+\ln n}$



- smoothed complexity: slight random perturbations of worst-case inputs
  D. Spielman and S. Teng (2001), *Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time*
  $O(max(n^5 \log^2 m, n^9 log^4 n, n^3 \sigma^{-4}))$

Outline

Initial Infeasibility

$$\begin{array}{rrcrcl} \max & x_1 & - & x_2 & & \\ & x_1 & + & x_2 & \leq & 2 \\ & 2x_1 & + & 2x_2 & \geq & 5 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

$$\begin{array}{rrcrcrcrcl} \max & x_1 & - & x_2 & & & & & & \\ & x_1 & + & x_2 & + & x_3 & & & = & 2 \\ & -2x_1 & - & 2x_2 & & & + & x_4 & = & -5 \\ & & & & & & x_1, x_2, & x_3, x_4 & \geq & 0 \end{array}$$

- Initial tableau

```
|    | x1 | x2 | x3 | x4 | -z |  b |
|----+----+----+----+----+----+----|
| x3 |  1 |  1 |  1 |  0 |  0 |  2 |
| x4 | -2 | -2 |  0 |  1 |  0 | -5 |
|----+----+----+----+----+----+----|
|    |  1 | -1 |  0 |  0 |  1 |  0 |
```

$\rightsquigarrow$ we do not have an initial basic feasible solution!!

In general finding any feasible solution is difficult as finding an optimal solution, otherwise we could do binary search

Auxiliary Problem (I Phase of Simplex)

We introduce auxiliary variables:

$$
\begin{array}{rrrrrrl}
w^* = \max & -x_5 \equiv \min x_5 & & & & & \\
& x_1 + x_2 + x_3 & & & & = & 2 \\
& 2x_1 + 2x_2 & & - x_4 & + x_5 & = & 5 \\
& & & x_1, x_2, x_3, x_4, x_5 & & \geq & 0
\end{array}
$$

if $w^* = 0$ then $x_5 = 0$ and the two problems are equivalent
if $w^* > 0$ then not possible to set x_5 to zero.

- Initial tableau

```
|    | x1 | x2 | x3 | x4 | x5 | -z | -w | b |
|----+----+----+----+----+----+----+----+---|
|    |  1 |  1 |  1 |  0 |  0 |  0 |  0 | 2 |
|    |  2 |  2 |  0 | -1 |  1 |  0 |  0 | 5 |
| z  |  1 | -1 |  0 |  0 |  0 |  1 |  0 | 0 |
|----+----+----+----+----+----+----+----+---|
| w  |  0 |  0 |  0 |  0 | -1 |  0 |  1 | 0 |
```

Keep z always in basis

- we reach a canonical form simply by letting x_5 enter the basis:

```
|       | x1 | x2 | x3 | x4 | x5 | -z | -w | b |
|-------+----+----+----+----+----+----+----+---|
|       |  1 |  1 |  1 |  0 |  0 |  0 |  0 | 2 |
|       |  2 |  2 |  0 | -1 |  1 |  0 |  0 | 5 |
| z     |  1 | -1 |  0 |  0 |  0 |  1 |  0 | 0 |
|-------+----+----+----+----+----+----+----+---|
| IV+II |  2 |  2 |  0 | -1 |  0 |  0 |  1 | 5 |
```

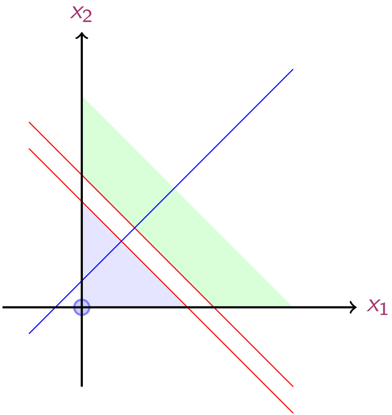
  now we have a basic feasible solution!

- x_1 enters, x_3 leaves

```
|        | x1 | x2 | x3 | x4 | x5 | -z | -w |  b |
|--------+----+----+----+----+----+----+----+----|
|        |  1 |  1 |  1 |  0 |  0 |  0 |  0 |  2 |
| II-2I' |  0 |  0 | -2 | -1 |  1 |  0 |  0 |  1 |
| III-I' |  0 | -2 | -1 |  0 |  0 |  1 |  0 | -2 |
|--------+----+----+----+----+----+----+----+----|
| IV-2I' |  0 |  0 | -2 | -1 |  0 |  0 |  1 |  1 |
```

  $w^* = -1$ then no solution with $x_5 = 0$ exists then no feasible solution to initial problem

$$
\begin{aligned}
\max \quad x_1 \; - \; x_2 & \\
x_1 \; + \; x_2 & \leq 2 \\
2x_1 \; + \; 2x_2 & \geq 5 \\
x_1, x_2 & \geq 0
\end{aligned}
$$



Initial Infeasibility - Another Example

$$
\begin{array}{rrrrl}
\max & x_1 & - & x_2 & \\
 & x_1 & + & x_2 & \leq 2 \\
 & 2x_1 & + & 2x_2 & \geq 2 \\
 & & & x_1, x_2 & \geq 0
\end{array}
$$

$$
\begin{array}{rrrrrrrl}
\max & x_1 & - & x_2 & & & & \\
 & x_1 & + & x_2 & + x_3 & & & = 2 \\
 & 2x_1 & + & 2x_2 & & - & x_4 & = 2 \\
 & & & & & & x_1, x_2, x_3, x_4 & \geq 0
\end{array}
$$

Auxiliary problem (I phase):

$$
\begin{array}{rrrrrrl}
w = \max & -x_5 \equiv \min x_5 & & & & & \\
 & x_1 & + & x_2 & + x_3 & & = 2 \\
 & 2x_1 & + & 2x_2 & & - x_4 + x_5 & = 2 \\
 & & & & & x_1, x_2, x_3, x_4, x_5 & \geq 0
\end{array}
$$

- Initial tableau

	x1	x2	x3	x4	x5	-z	-w	b
	1	1	1	0	0	0	0	2
	2	2	0	-1	1	0	0	2
z	1	-1	0	0	0	1	0	0
w	0	0	0	0	-1	0	1	0

  $\rightsquigarrow$ we do not have an initial basic feasible solution.

- set in canonical form:

	x1	x2	x3	x4	x5	-z	-w	b
	1	1	1	0	0	0	0	2
	2	2	0	-1	1	0	0	2
z	1	-1	0	0	0	1	0	0
IV+II	2	2	0	-1	0	0	1	2

- x_1 enters, x_5 leaves

	x1	x2	x3	x4	x5	-z	-w	b
	0	0	1	1/2	-1/2	0	0	1
	1	1	0	-1/2	1/2	0	0	1
z	0	-2	0	1/2	-1/2	1	0	-1
w	0	0	0	0	-1	0	1	0

  $w^* = 0$ hence $x_5 = 0$ we have a starting feasible solution for the initial problem.

- (II phase) We keep only what we need:

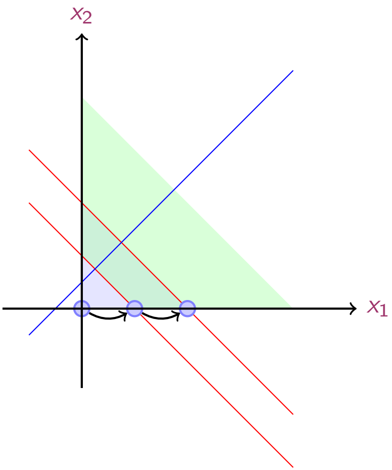
```
|    | x1 | x2 | x3 | x4   | -z |  b |
|----+----+----+----+------+----+----|
|    |  0 |  0 |  1 | 1/2  |  0 |  1 |
|    |  1 |  1 |  0 | -1/2 |  0 |  1 |
|----+----+----+----+------+----+----|
| z  |  0 | -2 |  0 | 1/2  |  1 | -1 |
```

-

```
|    | x1 | x2 | x3 | x4 | -z |  b |
|----+----+----+----+----+----+----|
|    |  0 |  0 |  2 |  1 |  0 |  2 |
|    |  1 |  1 |  1 |  0 |  0 |  2 |
|----+----+----+----+----+----+----|
| z  |  0 | -2 | -1 |  0 |  1 | -2 |
```

Optimal solution: $x_1 = 2, x_2 = 0, x_3 = 0, x_4 = 2$, $z = 2$.

$$
\begin{array}{rrcrcl}
\max & x_1 & - & x_2 & & \\
 & x_1 & + & x_2 & \leq & 2 \\
 & 2x_1 & + & 2x_2 & \geq & 2 \\
 & & & x_1, x_2 & \geq & 0
\end{array}
$$



In Dictionary Form

$$\begin{array}{rrrrrr} \max & x_1 & - & x_2 & & \\ & x_1 & + & x_2 & \leq & 2 \\ & 2x_1 & + & 2x_2 & \geq & 5 \\ & & & x_1, x_2 & \geq & 0 \end{array}$$

$$\begin{array}{rcrcrcr} x_3 & = & 2 & - & x_1 & - & x_2 \\ x_4 & = & -5 & + & 2x_1 & + & 2x_2 \\ \hdashline z & = & x_1 & + & x_2 & & \end{array}$$

sol. infeasible

We introduce corrections of infeasibility

$$\begin{array}{rrrrrrr} \max & -x_0 \equiv \min x_0 & & & & & \\ & x_1 & + & x_2 & & & \leq 2 \\ & 2x_1 & + & 2x_2 & - & x_0 & \geq 5 \\ & & & x_1, x_2, x_0 & \geq & 0 & \end{array}$$

$$\begin{array}{rcrcrcrcr} x_3 & = & 2 & - & x_1 & - & x_2 & & \\ x_4 & = & -5 & + & 2x_1 & + & 2x_2 & + & x_0 \\ \hdashline z & = & & & & & & - & x_0 \end{array}$$

It is still infeasible but it can be made feasible by letting x_0 enter the basis
which variable should leave?
the most infeasible: the var with the b term whose negative value has the largest magnitude

Simplex: Exception Handling, Summary

Solution of an LP problem:

a. $F \neq \emptyset$ and $\nexists$ solution

b. $F \neq \emptyset$ and $\exists$ solution

   i) **one solution**

   ii) infinite solutions

c. $F = \emptyset$

Handling exceptions in the Simplex Method

1. Unboundedness
2. More than one solution
3. Degeneracies
   - benign
   - cycling
4. Infeasible starting
   Phase I + Phase II