CIVIL-557

Decision Aid Methodologies In Transportation

Lab IV: Integer programming and linearization

Virginie Lurkin

Transport and Mobility Laboratory (TRANSP-OR)
École Polytechnique Fédérale de Lausanne EPFL

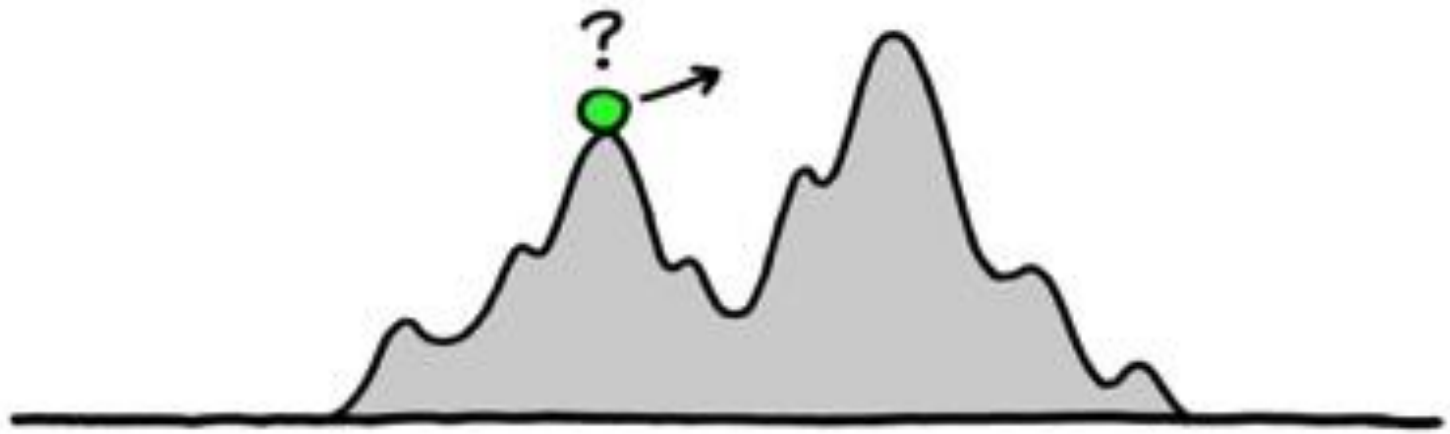
Overview

- Linearization techniques
  - Why do we linearize?
  - Absolute value
  - Big-M constraints
- Balancing storage at a container terminal
- A first simple heuristic

Linearization Techniques

Why do we linearize?

- When solving a system of linear functions, global optimality can be guaranteed.
- Some simple non-linear (convex) functions can be used (like x^2, where you can reach the top of the parabole without changing of the sense of the algorithm).
- Complex non-linear functions might get stuck in local optimum:



How to linearize absolute value?

- We have the following function:

$$min \sum_{i=1}^{6} |x_i - y_i|$$

- Two transformations exist:

$$min \sum_{i=1}^{6} \sqrt{x_i^2 - y_i^2}$$

$$\begin{cases} x_i - y_i, & if\ x_i \geq y_i \\ y_i - x_i, & if\ y_i \geq x_i \end{cases}$$

non-linear

if-condition

How to linearize absolute value?

- We want a linear version, so let's try some values for the if-condition:
  - x = 5 and y = 3, we get:
    - 5 – 3 = 2
    - 3 – 5 = -2
  - x = 3 and y = 5, we get:
    - 3 – 5 = -2
    - 5 – 3 = 2
  - We want the positive (higher) number, i.e. define new variable **z** ≥ **LHS**.
  - For the above case z is now between **2** and **infinity.**
  - To make z equal to 2, the objective function now will be to **min z** (i.e. pushing it to the smallest possible value).

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

How to linearize absolute value?

- Overall, we get the general form:

$$\min \sum_{i=1}^{6} z_i$$

$$x_i - y_i \leq z_i, \quad \text{forall } i$$

$$y_i - x_i \leq z_i, \quad \text{forall } i$$

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Big M-constraints

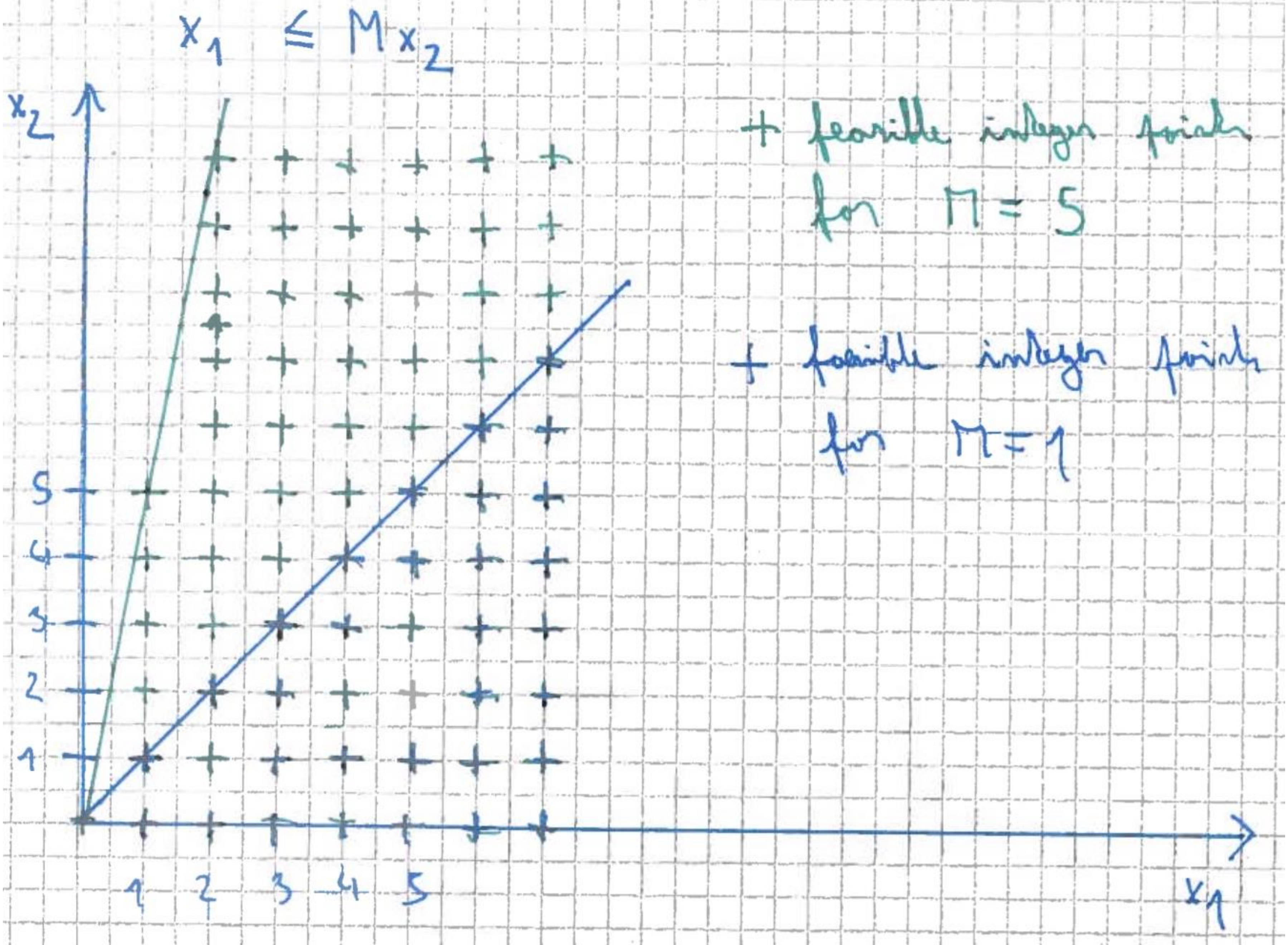
- What if we want a whole constraint to be active or not?
- Consider example:
  - Either $x_1 + x_2 + x_3 \leq 10$
  - Or $x_1 + x_2 + x_3$ is free to take any value
  - In other words, either ≤ **10** or ≤ **infinity**
  - Let's say **M = infinity**
  - Then, we can have:
    - $x_1 + x_2 + x_3 \leq 10 + M * y$
    - Where y is binary: 1 – if the constraint is free, 0 – otherwise
  - Second option:
    - $x_1 + x_2 + x_3 \leq 10 + M * (1 - y)$
    - Where y is binary: 1 – if the constraint is bounded, 0 – otherwise

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Big M-constraints

- Since the **Branch and Bound** algorithm will be branching on the **y variable**, the free option will provide a very weak bound, i.e. longer time needed for branching.
- Moreover (next slide):



$x_1 \leq M x_2$
x_2
x_1
1
2
3
4
5
+ feasible integer points for M = 5
+ feasible integer points for M = 1

Big M-constraints

- Therefore, the aim is to find as small value of M as possible.
- Let's say that the sum of the x variables in the previous example ($x_1 + x_2 + x_3 \leq 10$) is never exceeding 100, then the big M should be equal to 90 ($10 + M * y$).

Big M-constraints

- Let's consider additional example:

An optimization model requires the two following conditions on inequalities (1)-(4) to hold simultaneously:

- Condition 1: at least two of the inequalities (1)-(4) must hold
- Condition 2: if both inequalities (1) and (2) hold, then inequality (3) must hold,

Reformulate inequalities (1)-(4) to represent these conditions,

$$
\begin{gathered}
5x_1 + 3x_2 + 3x_3 - x_4 \leq 10 \\
2x_1 + 5x_2 - x_3 + 3x_4 \leq 10 \\
-x_1 + 3x_2 + 5x_3 + 3x_4 \leq 10 \\
3x_1 - x_2 + 3x_3 + 5x_4 \leq 10 \\
x_1, x_2, x_3, x_4 \geq 0
\end{gathered}
$$

Big M-constraints

- Step #1 mapping binary decision to the state of constraints.
  - 2 options: y = 1 constraint active or y = 1 constraint inactive (free)
  - Option 1:

$$
\begin{aligned}
5x_1 + 3x_2 + 3x_3 - x_4 &\leq 10 + M(1 - y_1) \\
2x_1 + 5x_2 - x_3 + 3x_4 &\leq 10 + M(1 - y_2) \\
-x_1 + 3x_2 + 5x_3 + 3x_4 &\leq 10 + M(1 - y_3) \\
3x_1 - x_2 + 3x_3 + 5x_4 &\leq 10 + M(1 - y_4) \\
y_2 + y_2 + y_3 + y_4 &\geq 2 \\
y_1 + y_2 &\leq 1 + y_3 \\
x_1, x_2, x_3, x_4 &\geq 0 \\
y_1, y_2, y_3, y_4 &\in \{0, 1\}
\end{aligned}
$$

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Big M-constraints

- Step #1 mapping binary decision to the state of constraints.
  - 2 options: y = 1 constraint active or y = 1 constraint inactive (free)
  - Option 2:

$$
\begin{aligned}
5x_1 + 3x_2 + 3x_3 - x_4 &\leq 10 + M(y_1) \\
2x_1 + 5x_2 - x_3 + 3x_4 &\leq 10 + M(y_2) \\
-x_1 + 3x_2 + 5x_3 + 3x_4 &\leq 10 + M(y_3) \\
3x_1 - x_2 + 3x_3 + 5x_4 &\leq 10 + M(y_4) \\
y_2 + y_2 + y_3 + y_4 &\leq 2 \\
(1 - y_1) + (1 - y_2) &\leq 1 + (1 - y_3) \\
x_1, x_2, x_3, x_4 &\geq 0 \\
y_1, y_2, y_3, y_4 &\in \{0, 1\}
\end{aligned}
$$

Big M-constraints

- Step #2 further conditioning on the constraints.
  - Part I – at least 2 inequalities must hold (works for both options same way):

$$
\begin{aligned}
5x_1 + 3x_2 + 3x_3 - x_4 &\leq 10 + M(y_1) \\
2x_1 + 5x_2 - x_3 + 3x_4 &\leq 10 + M(y_2) \\
-x_1 + 3x_2 + 5x_3 + 3x_4 &\leq 10 + M(y_3) \\
3x_1 - x_2 + 3x_3 + 5x_4 &\leq 10 + M(y_4) \\
\boxed{y_2 + y_2 + y_3 + y_4 \leq 2} \\
(1 - y_1) + (1 - y_2) &\leq 1 + (1 - y_3) \\
x_1, x_2, x_3, x_4 &\geq 0 \\
y_1, y_2, y_3, y_4 &\in \{0, 1\}
\end{aligned}
$$

Big M-constraints

- Step #2 further conditioning on the constraints.
  - Part 2 – if (1) and (2) hold, 3 must hold (i.e. if (1) holds and (2) does not, (3) is free to be 0 or 1!).
    - Option 1:

$$
\begin{aligned}
5x_1 + 3x_2 + 3x_3 - x_4 &\leq 10 + M(1 - y_1) \\
2x_1 + 5x_2 - x_3 + 3x_4 &\leq 10 + M(1 - y_2) \\
-x_1 + 3x_2 + 5x_3 + 3x_4 &\leq 10 + M(1 - y_3) \\
3x_1 - x_2 + 3x_3 + 5x_4 &\leq 10 + M(1 - y_4) \\
y_2 + y_2 + y_3 + y_4 &\geq 2 \\
y_1 + y_2 &\leq 1 + y_3 \\
x_1, x_2, x_3, x_4 &\geq 0 \\
y_1, y_2, y_3, y_4 &\in \{0, 1\}
\end{aligned}
$$

Big M-constraints

- Step #2 further conditioning on the constraints.
  - Part 2 – if (1) and (2) hold, 3 must hold (i.e. if (1) holds and (2) does not, (3) is free to be 0 or 1!).
    - Option 2:

$$
\begin{aligned}
5x_1 + 3x_2 + 3x_3 - x_4 &\leq 10 + M(y_1) \\
2x_1 + 5x_2 - x_3 + 3x_4 &\leq 10 + M(y_2) \\
-x_1 + 3x_2 + 5x_3 + 3x_4 &\leq 10 + M(y_3) \\
3x_1 - x_2 + 3x_3 + 5x_4 &\leq 10 + M(y_4) \\
y_2 + y_2 + y_3 + y_4 &\leq 2 \\
\boxed{(1 - y_1) + (1 - y_2) \leq 1 + (1 - y_3)} \\
x_1, x_2, x_3, x_4 &\geq 0 \\
y_1, y_2, y_3, y_4 &\in \{0, 1\}
\end{aligned}
$$

Balancing Container Storage

TRANSP-OR

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

During the Lab…

Consider a container terminal with a storage yard consisting of **100 blocks**, each with storage space to hold **600 containers**, numbered serially 1 to 100. For $i = 1$ to 100, during a particular 4-hour planning period, if no newly arriving containers in this period are sent to block i for storage, the number of stored containers in each block is:

1	320	14	220	27	100	40	119	53	181	66	336	79	155	92	182
2	157	15	372	28	183	41	43	54	233	67	411	80	360	93	96
3	213	16	101	29	99	42	71	55	414	68	280	81	360	94	301
4	96	17	212	30	505	43	219	56	30	69	115	82	290	95	121
5	413	18	251	31	99	44	363	57	333	70	200	83	350	96	278
6	312	19	86	32	254	45	98	58	427	71	117	84	157	97	372
7	333	20	79	33	330	46	500	59	251	72	284	85	116	98	119
8	472	21	295	34	279	47	413	60	83	73	263	86	141	99	282
9	171	22	138	35	300	48	259	61	144	74	477	87	82	100	310
10	222	23	343	36	150	49	182	62	404	75	431	88	116		
11	439	24	281	37	340	50	391	63	76	76	297	89	99		
12	212	25	372	38	221	51	360	64	84	77	380	90	78		
13	190	26	450	39	79	52	447	65	196	78	327	91	220		

Simplex solver

The terminal estimates that in this period **15166 new containers** will be unloaded from docked vessels and dispatched to the storage yard for storage.

Add-ons

- Some containers are transferring perishable goods and therefore, need to be in a refrigarated state (file `Containers.dat`, 1 – if a container is having such need). Several blocks in the yard have the refrigarating facility (file `Refrigarators.dat`, 1 – if a block offers such facility).
- To further prevent heavy traffic in the yard area, 2 adjacent blocks cannot exceed 75% of their capacity at the same time.

Assignment #3 – Part A

- Solve the basic model given to you in Lecture 2.
- Solve the model with the refrigaration add-on only.
- Solve the model with the adjacency add-on only.
- Solve the complete model (basic + all add-ons at the same time).
- Report the value of objective function for all the models. Since the LP formulation of the basic model gives fractional solutions, re-run it as an IP (i.e. x variables are integers). Keep the IP settings for all subsequent models. Are the x values changing among the different models?
- What should be the value of big-M for the adjacency constraints?

Assignment #3 – Part B

- Implement the following small heurisitc for the container storage problem

  1. Rearrange blocks so that a_i increases with i
  2. Determine x_i in increasing order of i using:

$$x_1 = \min\{N, \max\{0, A \times \mathrm{F} - a_1\}\}$$
$$x_i = \min\{\max\{0, A \times \mathrm{F} - a_i\}, N - \sum_{r=1}^{i-1} x_r\} \quad \forall i \geq 2$$

- Compare the solution of the exact method and the heuristic on the different instances
- How would you adapt the heuristic to consider the refrigaration and/or adjacency constraints?

TRANSP-OR

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Assignment #3 – Part B

- Note: you can implement the heuristic with your favourite software (e.g. matlab), but if you want to do it with OPL, you need to include your code within an execute command as follows

```
int N=...;
int B=...;
int A=...;
int a[1..100]=...;
range block = 1..B;
float F = (N+sum (i in block)
a[i])/(A*B);
float x[block];
float y[block];
float q[block];

execute heuristic{
writeln("Heuristic solution");
// your code
}
```

Assignment #3

- Send a single file containing all the answers including your code (format is up to you) AND mathematical formulations of the add-ons.
- Group/Individual work
- Send your reports to virginie.lurkin(at)epfl.ch
- Send your reports by 8:00 P.M. next Monday