Data-driven Decision Making

Integer optimization, Location problems

João Pedro Pedroso

2024/2025

Last class: Duality in linear optimization

Example: primal problem

$$
\begin{array}{lrrrr}
\text{maximize} & 15x_1 + & 18x_2 + & 30x_3 & \\
\text{subject to:} & 2x_1 + & x_2 + & x_3 \leq & 60 \\
 & x_1 + & 2x_2 + & x_3 \leq & 60 \\
 & & & x_3 \leq & 30 \\
 & x_1, & x_2, & x_3 \geq & 0
\end{array}
$$

Example: dual problem

$$
\begin{array}{lrrrr}
\text{minimize} & 60\pi_1 + & 60\pi_2 + & 30\pi_3 & \\
\text{subject to:} & 2\pi_1 + & \pi_2 & \geq & 15 \\
 & \pi_1 + & 2\pi_2 & \geq & 18 \\
 & \pi_1 + & \pi_2 + & \pi_3 \geq & 30 \\
 & \pi_1, & \pi_2, & \pi_3 \geq & 0
\end{array}
$$

Last class: Duality: economic interpretation

- Optimal value of the dual variable associated to a constraint: shadow price
  - impact in the optimum of a unit change in the right hand side
- Reduced cost associated to a decision variable:
  - impact in the optimum when a variable x which is 0 in the optimum is increased to 1
  - also, amount by which an objective function coefficient would have to improve for the variable to become positive value in the optimal solution

Last class: linear optimization + duality

- slack variable: difference between the right- and the left-hand sides of a constraint
  - in AMPL: `.slack` attribute for a constraint
    → *e.g.*, `display Capacity.slack;`
- dual variable
  - in AMPL: `.dual` attribute for a constraint
    → *e.g.*, `display Capacity.dual;`
  - or, simply the constraint name → *e.g.*, `display Capacity;`
- reduced cost
  - in AMPL: `.rc` attribute for a variable
    → *e.g.*, `x.rc;`

Last class: synthesis of duality theory

- Weak duality:
  - objective for any dual-feasible solution *"is better"* that the objective of any primal-feasible solution
- Strong duality theorem:
  - optimum (objective) of the dual is equal to the optimum of the primal
- Complementary slackness: at the optimum, for every constraint:
  - if the slack is positive, then the shadow price is zero
  - if the shadow price is positive, then the slack is zero
  - *note:* any solution that satisfies the complementary slackness is optimal

Integer optimization

Integer optimization: puzzle example

Adding the number of heads of cranes, turtles and octopuses totals 32, and the number of legs sums to 80. What is the minimum number for turtles plus octopuses?

- Many real-world optimization problems require solutions composed of integers (instead of real numbers)
- Answer to this puzzle: meaningful if solution has integer values only
- Formalizing as an optimization problem $\rightarrow$ formulation
  - variables:
    - $x \rightarrow$ number of cranes
    - $y \rightarrow$ number of turtles
    - $z \rightarrow$ number of octopuses
  - constraints:
    - number of heads is 32
      $x + y + z = 32$
    - number of legs is 80
      $2x + 4y + 8z = 80$
  - objective: minimize the number of turtles and octupuses minimize $y + z$

Complete description

$$
\begin{array}{lrrrcr}
\text{minimize} & & y + & z & & \\
\text{subject to:} & x + & y + & z & = & 32 \\
 & 2x + & 4y + & 8z & = & 80 \\
 & x, & y, & z & \geq & 0
\end{array}
$$

Solve it

```
var x >=0;
var y >=0;
var z >=0;

minimize obj: y + z;

subject to
C1: x + y + z = 32;
C2: 2*x + 4*y + 8*z = 80;
```

Solution?

Solution?

```
ampl: display x, y, z;
1    x          29.3333
2    y          0
3    z          2.66667
;
```

How to fix it?

- We need to add conditions to force the variables to have integer values $\rightarrow$ integrality constraints

$$
\begin{array}{lrrrcr}
\text{minimize} & & y + & z & & \\
\text{subject to:} & x + & y + & z & = & 32 \\
 & 2x + & 4y + & 8z & = & 80 \\
 & x, & y, & z & \geq & 0, \textbf{integer}
\end{array}
$$

- Linear optimization problems requiring variables to be integers: integer optimization problems

AMPL

```
var x >=0, integer;
var y >=0, integer;
var z >=0, integer;

minimize obj: y + z;

subject to
C1: x + y + z = 32;
C2: 2*x + 4*y + 8*z = 80;
```

AMPL

```
var x >=0, integer;
var y >=0, integer;
var z >=0, integer;

minimize obj: y + z;

subject to
C1: x + y + z = 32;
C2: 2*x + 4*y + 8*z = 80;
```

```
ampl: display x, y, z;
x = 28
y = 2
z = 2
```

Remarks

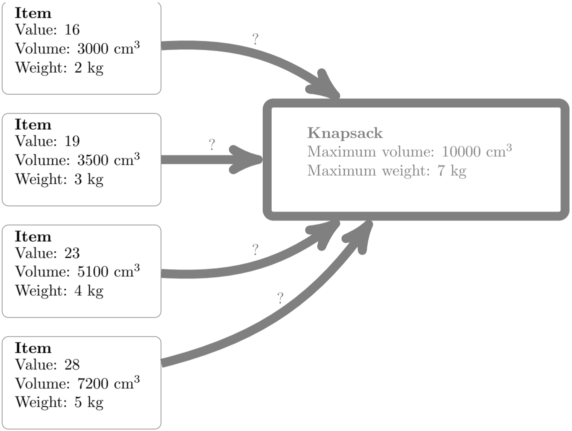
- For small integer optimization problems like this, the answer can be quickly found
  $x = 28, y = 2, z = 2$
- Meaning: there are 28 cranes, 2 turtles and 2 octopuses
- Completely different of the continuous version
  - in general, we cannot guess the value of an integer solution from the continuous model
- Usually, integer-optimization problems are much harder to solve than linear-optimization problems

Multi-Constrained Knapsack Problem

Multi-Constrained Knapsack Problem

- Knapsack problem:
  - fill up a knapsack of certain capacity
  - items taken from a given set
  - aim: take collection with maximum value

Knapsack example



- knapsack's volume 10,000 cm^3
- maximum weight: 7 kg
- four items:
  - weight 2, 3, 4, 5
  - volume 3000, 3500, 5100, 7200
  - value 16, 19, 23, 28
- how to fill the knapsack with items such that the total value is maximum?

Mathematical optimization model

- Variables:
  - $x_j = 1$ if item j is taken
  - $x_j = 0$ otherwise, $\quad j = 1, \ldots 4$
- Constraints:
  - total weight cannot exceed 7 kg
  - total volume cannot exceed 10,000 cm³
- Model

$$
\begin{array}{lrrrrcr}
\text{maximize} & 16x_1 + & 19x_2 + & 23x_3 + & 28x_4 & & \\
\text{subject to:} & 2x_1 + & 3x_2 + & 4x_3 + & 5x_4 & \leq & 7 \\
 & 30x_1 + & 35x_2 + & 51x_3 + & 72x_4 & \leq & 100 \\
 & x_1, & x_2, & x_3, & x_4 & \in & \{0, 1\}
\end{array}
$$

AMPL model

$$
\begin{array}{lrrrrcr}
\text{maximize} & 16x_1 + & 19x_2 + & 23x_3 + & 28x_4 & & \\
\text{subject to:} & 2x_1 + & 3x_2 + & 4x_3 + & 5x_4 & \leq & 7 \\
 & 30x_1 + & 35x_2 + & 51x_3 + & 72x_4 & \leq & 100 \\
 & x_1, & x_2, & x_3, & x_4 & \in & \{0,1\}
\end{array}
$$

```
set J;    # items
set I;    # constraints
param v {J};  # value of each item
param b {I};  # limit on each constraing
param a {I,J};  # "weight" of object j in dimension i
var x{J} binary;

maximize z: sum {j in J} v[j] * x[j];

subject to
C {i in I}: sum {j in J} a[i,j] * x[j] <= b[i];
```

AMPL data

```
param: J:   v :=
       1 16
       2 19
       3 23
       4 28;
param: I:   b :=
       1   7
       2 100;
param a :
            1  2  3  4   :=
      1     2  3  4  5
      2    30 35 51 72;
```

Solution

```
x [*] :=
1   0
2   1
3   1
4   0
;
```

- Solution found:
  - take items 2 and 3, leave 1 and 4
  - total value: 42
- Next: briefly sketch how this solution is found

Branch-and-bound

Branch-and-bound

- In many cases solutions must have integer values
  - knapsack problem: 1 to include an item, 0 otherwise
- Linear optimization cannot be used directly for such cases
  - variables assume *continuous* values in linear optimization
- If we may add constraints $0 \leq x_j \leq 1$, for all j, but linear optimization may give fractional values for the solution
- Solving integer-optimization models is much harder than linear optimization
- Systematic approach for solving an integer-optimization model: $\rightarrow$ branch-and-bound

Branch-and-bound

- Use linear optimization for solving a relaxed model
  - drop integer requirements on the variables
    $\rightarrow$ linear-optimization relaxation
- If relaxation's solution is integer $\rightarrow$ optimum
- Otherwise:
  - systematically subdivide the problem into two subproblems
  - exclude the previous solution from both of them

Example

- Let us use the previous model
- For each variable, replace `binary` with

$$0 \leq x_j \leq 1$$

- Linear optimization $\rightarrow$ solution:

```
x [*] :=
1  1
2  1
3  0.5
4  0
;
```

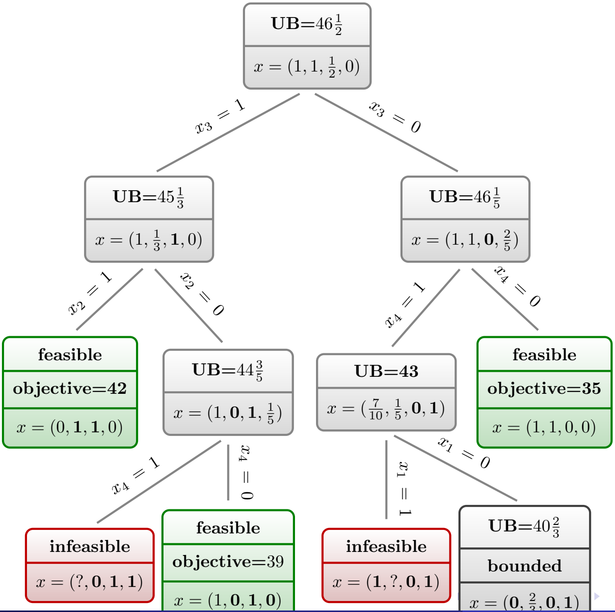
$\rightarrow$ infeasible

- Objective value: 46.5, upper bound to the optimum (why?)
  (in minimization problems: *lower bound*)

Branching

- subproblem 1: $x_3 \leq 0$ ($\rightarrow$ `x3 = 0`)
- subproblem 2: $x_3 \geq 1$ ($\rightarrow$ `x3 = 1`)

Solution



UB=$46\frac{1}{2}$
$x = (1, 1, \frac{1}{2}, 0)$
$x_3 = 1$
$x_3 = 0$
UB=$45\frac{1}{3}$
$x = (1, \frac{1}{3}, \mathbf{1}, 0)$
UB=$46\frac{1}{5}$
$x = (1, 1, \mathbf{0}, \frac{2}{5})$
$x_2 = 1$
$x_2 = 0$
$x_4 = 1$
$x_4 = 0$
feasible
objective=42
$x = (0, \mathbf{1}, \mathbf{1}, 0)$
UB=$44\frac{3}{5}$
$x = (1, \mathbf{0}, \mathbf{1}, \frac{1}{5})$
UB=43
$x = (\frac{7}{10}, \frac{1}{5}, \mathbf{0}, \mathbf{1})$
feasible
objective=35
$x = (1, 1, 0, 0)$
$x_4 = 1$
$x_4 = 0$
$x_1 = 1$
$x_1 = 0$
infeasible
$x = (?, \mathbf{0}, \mathbf{1}, \mathbf{1})$
feasible
objective=39
$x = (1, \mathbf{0}, \mathbf{1}, \mathbf{0})$
infeasible
$x = (\mathbf{1}, ?, \mathbf{0}, \mathbf{1})$
UB=$40\frac{2}{3}$
bounded
$x = (\mathbf{0}, \frac{2}{3}, \mathbf{0}, \mathbf{1})$

Branch-and-bound tree

- Nodes $\rightarrow$ subproblems $\rightarrow$ linear-optimization relaxation
- Edges $\rightarrow$ decisions $\rightarrow$ reduce some variable's domain
- Leaves:
  - if feasible $\rightarrow$ candidate solution
  - otherwise, can be neglected
- Order of visit: important for keeping the tree small
- At any time: best found solution is call the incumbent solution
- Bounding:
  - neglect nodes whose upper bound is inferior to the incumbent
- When there are not more nodes to expand:
  $\rightarrow$ incumbent is the optimum

Different approach: cutting plane method

- Branch-and-bound: tree search, creating alternative subproblems
- Cutting planes:
  - when a solution is not integer, add a constraint removing it
  - general approch: Gomori cuts $\rightarrow$ approch a feasible solution
  - often, *problem-specific*
    - *e.g.*, for the TSP (will be seen later)

Practical notes

- For difficult problems, there may be limitations on the size of the problems that can be tackled by the solver
- A large number of interesting, real-world problems can be solved successfully
- In other situations, the solver cannot find the optimal solution
  - but it may find a solution close to the optimum within reasonable time
  - in many applications, this is enough for practical implementation

Example

A call center requires different numbers of full-time employees on different days of the week. The number of full-time employees required on each day is the following.

Day	Number of employees
Monday	17
Tuesday	13
Wednesday	15
Thursday	19
Friday	14
Saturday	16
Sunday	11

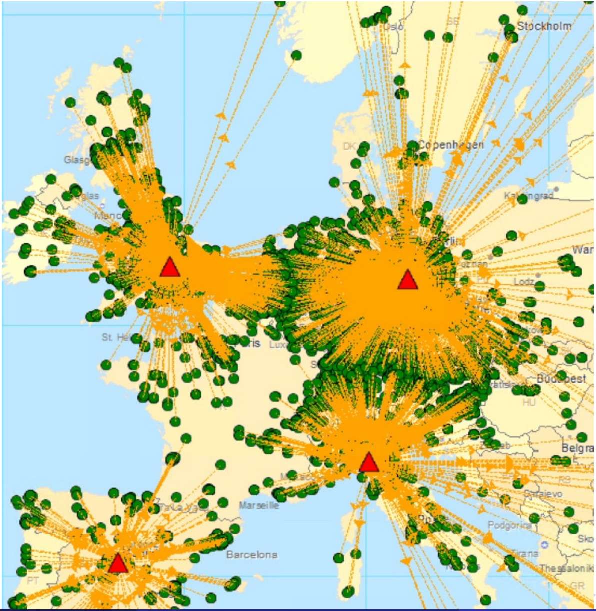
Union rules state that each full-time employee must work five consecutive days and then receive two days off. For example, an employee who works Monday to Friday must be off on Saturday and Sunday. The call center wants to meet its daily requirements using only full-time employees.

Example

1. Formulate a linear optimization problem that the call center can use to minimize the number of full-time employees who must be hired. Solve it and analyze its solution.
2. Consider the corresponding integer optimization problem. Solve it and analyze its solution.

Location problems

Facility Location Problems



Facility location problems

- Classical optimization problem for determining the sites for factories and warehouses
  - choosing the best among potential sites
  - subject to constraints requiring that demands are served by established facilities
  - objective: select facility sites in order to minimize costs
- Typical cost structure:
  - part proportional to distances from demand points to serving facilities
  - part related to opening facilities
- Facilities may have limited capacities for serving
  - capacitated
  - uncapacitated
- We will see several formulations and analyse their performance

Capacitated facility location problem

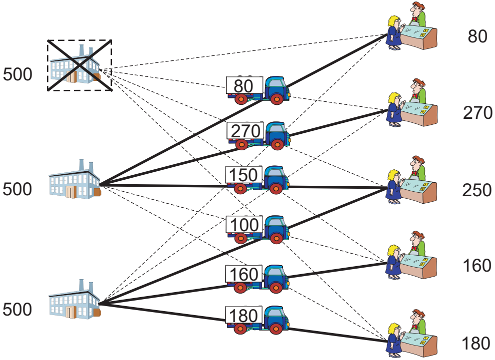
- total demand that each facility may satisfy is limited
- modeling:
  - demand satisfaction
  - capacity constraints

Example

- A company has three potential sites for installing its facilities and five demand points
- Each site j has a yearly activation cost f_j
  - an annual leasing expense incurred for using it
  - independently of the volume it serves
- Volume is limited to a given maximum yearly amount M_j
- Transportation cost c_{ij} per unit served from facility j to demand point i

Customer i	1	2	3	4	5		
Annual demand d_i	80	270	250	160	180		
Facility j	cost c_{ij}					f_j	M_j
1	4	5	6	8	10	1000	500
2	6	4	3	5	8	1000	500
3	9	7	4	3	4	1000	500

Example



Customer i	1	2	3	4	5		
Annual demand d_i	80	270	250	160	180		
Facility j	c_{ij}					f_j	M_j
1	4	5	6	8	10	1000	500
2	6	4	3	5	8	1000	500
3	9	7	4	3	4	1000	500

Formulation

- Customers $i \in I = \{1, 2, \ldots, n\}$
- Sites for facilities $j \in J = \{1, 2, \ldots, m\}$
- Variables:
  - $x_{ij} \geq 0 \rightarrow$ amount served from facility j to demand point i
  - $y_j = 1$ if a facility is established at location j, 0 otherwise

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{subject to:} & \displaystyle\sum_{j \in J} x_{ij} = d_i & \forall i \in I \\
& \displaystyle\sum_{i=1}^{n} x_{ij} \leq M_j y_j & \forall j \in J \\
& x_{ij} \leq d_i y_j & \forall i \in I,\ j \in J \\
& x_{ij} \geq 0 & \forall i \in I,\ j \in J \\
& y_j \in \{0, 1\} & \forall j \in J
\end{array}
$$

- Objective: minimize activation costs + transportation costs
- First constraints: demand satisfaction
- Second constraints: quantity served from each facility
  - 0 if not activated
  - facility's capacity if activated
- Third constraints: variable upper bounds
  - redundant, but yield tighter linear optimization relaxation

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{subject to:} & \displaystyle\sum_{j \in J} x_{ij} = d_i & \forall i \in I \\
& \displaystyle\sum_{i=1}^{n} x_{ij} \leq M_j y_j & \forall j \in J \\
& x_{ij} \leq d_i y_j & \forall i \in I,\ j \in J \\
& x_{ij} \geq 0 & \forall i \in I,\ j \in J \\
& y_j \in \{0,1\} & \forall j \in J
\end{array}
$$

AMPL model

```
set I;
set J;
param f {J};
param c {I, J};
param d {I};
param M {J};

var x {I, J} >=0;
var y {J} binary;

subject to
Demand {i in I}: sum {j in J} x[i,j] = d[i];
Supply {j in J}: sum {i in I} x[i,j] <= M[j] * y[j];
Bounds {i in I, j in J}: x[i,j] <= d[i] * y[j];

minimize cost: sum {j in J} f[j] * y[j] +
               sum {i in I, j in J} c[i,j] * x[i,j];
```

AMPL data

```
data;
param: J:  M    f :=  # defines set "J" and param "M" and "f"
       1 500  1000
       2 500  1000
       3 500  1000 ;
param: I:  d :=  # defines set "I" and param "d"
       1  80
       2 270
       3 250
       4 160
       5 180;
param c (tr) :   # (tr) --> transposed
            1  2  3  4  5 :=
      1     4  5  6  8  10
      2     6  4  3  5  8
      3     9  7  4  3  4 ;
```

Formulation for capacitated case

- Customers $i \in I = \{1, 2, \ldots, n\}$
- Sites for facilities $j \in J = \{1, 2, \ldots, m\}$
- Variables:
  - $x_{ij} \geq 0 \rightarrow$ amount served from facility j to demand point i
  - $y_j = 1$ if a facility is established at location j, 0 otherwise

$$
\begin{array}{lll}
\text{minimize} & \displaystyle\sum_{j \in J} f_j y_j + \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} & \\
\text{subject to:} & \displaystyle\sum_{j \in J} x_{ij} = d_i & \forall i \in I \\
& \displaystyle\sum_{i=1}^{n} x_{ij} \leq M_j y_j & \forall j \in J \\
& x_{ij} \leq d_i y_j & \forall i \in I,\ j \in J \\
& x_{ij} \geq 0 & \forall i \in I,\ j \in J \\
& y_j \in \{0, 1\} & \forall j \in J
\end{array}
$$

Weak and strong formulations

Weak and strong formulations

- Consider the facility location problem with no capacity
  - any quantity may can be produced at each site
  - uncapacitated facility location problem
- One way of modeling: set M very large in

$$\sum_{i \in I} x_{ij} \leq M_j y_j \quad \forall j \in J$$

- Notice that we may omit constraints $x_{ij} \leq d_i y_j, \quad \forall i \in I, j \in J$
  - Removing them: problem becomes difficult to solve, especially as size increases $\rightarrow$ big M pitfall.

Big M

$$\sum_{i \in I} x_{ij} \leq M_j y_j \quad \forall j \in J$$

- Idea: model "if we do not activate a warehouse, we cannot transport from there"
- Parameter M represents a large enough number
  - constraint should be binding if $y_j = 0$
  - it shouldn't be active otherwise

Example (Winstons' book "Operations Research")

Because of excessive pollution on the Momiss River, the state of Momiss is going to build pollution control stations. Three sites (1, 2, and 3) are under consideration. Momiss is interested in controlling the pollution levels of two pollutants (1 and 2). The state legislature requires that at least 80,000 tons of pollutant 1 and at least 50,000 tons of pollutant 2 be removed from the river. The relevant data for this problem are shown below. Formulate an integer optimization problem to minimize the cost of meeting the state legislature's goals.

Site	Cost of building station ($)	Cost of treating 1 ton water ($)	Amount removed (ton) per ton of water	
			Pollutant 1	Pollutant 2
1	100000	20	0.40	0.30
2	60000	30	0.25	0.20
3	40000	40	0.20	0.25

Big M

$$\sum_{i \in I} x_{ij} \leq M_j y_j \quad \forall j \in J$$

- Idea: model *"if we do not activate a warehouse, we cannot transport from there"*
- Parameter M represents a large enough number
  - constraint should be binding if $y_j = 0$
  - it shouldn't be active otherwise
- However:
  - large values for M do disturb the model in practice

Modeling tip

A large number M must be set to a value as small as possible

- Whenever possible, it is better not to use a large number
- If its use is necessary, choose a number that is as small as possible, as long as the formulation is correct.
- Using large numbers, as $M = 9999999$, is unthinkable, except for very small instances.

Exercise

Adapt the previous model to use the *minimum value* for M

Remarks

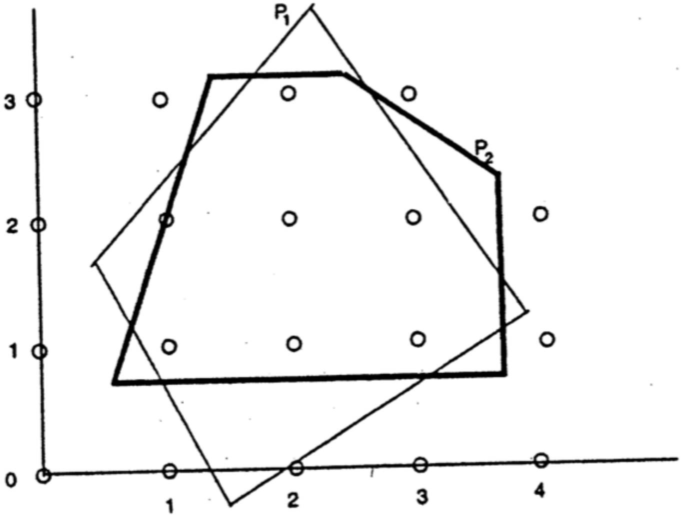
- For the uncapacitated facility location problem:
  - correct formulation: set $M =$ total amount demanded
- However, it is possible to improve the formulation:
  - adding $x_{ij} \leq d_i y_j$
- what formulation should we use?
  - answer depends on the particular case
  - in general stronger formulations are recommended.
    - strength $\rightarrow$ defined in terms of the linear optimization relaxation

Definition: Strong and Weak Formulations

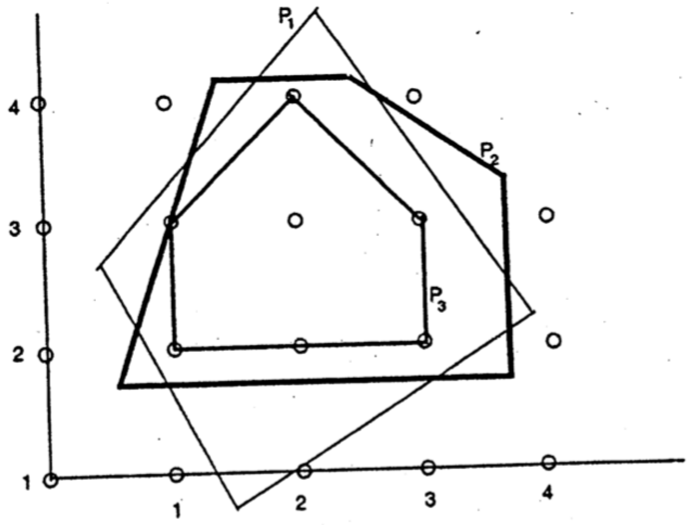
- For some problem: two formulations A and B
- Linear optimization relaxation:
  - exclude integrality constraints
- Let feasible regions be P_A and P_B
- If $P_A \subset P_B \rightarrow$ formulation A is stronger than B
  - B is weaker than A

Intuitively, if P_A is tighter than P_B, the bound obtained by the relaxation is closer to the optimum of the integer problem

Different formulations



Ideal formulation



Case study: facility location problem

- Consider formulations:
  - $A \rightarrow$ using constraints $x_{ij} \leq d_i y_j$
  - $B \rightarrow$ using only $\sum_{j=1}^{m} x_{ij} \leq \left(\sum_{i=1}^{n} d_i\right) y_j$
- A is stronger than B
- Let us check it:
  - $P_A, P_B \rightarrow$ feasible regions of A and B
  - constraints for B are obtained by adding those of A
  - hence :$P_A \subseteq P_B$

Remarks

- Truly stronger formulation: either
  - $P_A \subset P_B$
  - verify that the solution of the linear relaxation of B is not included in P_A
- "Is it always preferable to use a stronger formulation?"
  - no theoretical answer
  - part of the mathematical modeling art $\rightarrow$ guidance next

Understanding effect of stronger formulations

- Often, stronger formulations require more constraints or variables
  - previous example:
    - strong formulation: nm constraints
    - weak requires only n
- Time for solving the linear relaxation:
  - depends on the number of constraints and variables
  - likely to be longer for stronger formulation
- Trade-off between
  - shorter times for solving relaxations in weaker formulations
  - longer computational times for branch-and-bound

Guideline: *as the size of the enumeration tree grows very rapidly when the scale of the problem increases, even if the number of constraints and variables becomes larger, stronger formulations are usually preferable*

- This class:
  - Optimization problems with integer variables
    - formulation
    - how these problems are solved
- Next class: Integer optimization
  - more location problems
  - optimization in graphs

Bibliography

- Operations research
  Wayne L. Winston
  ISBN: 9780534423629
- AMPL: A Modeling Language for Mathematical Programming
  R Fourer, DM Gay, and BW Kernighan
  Second edition, ISBN 0-534-38809-4
  Available in the AMPL documentation (`http://www.ampl.com`)
- Mathematical Optimization: Solving Problems using Python and Gurobi
  M Kubo, JP Pedroso, M Muramatsu, and A Rais
  Kindaikagakusha, Tokyo (2012)