Integer Linear programming

Formulation

Hugues Talbot

Laboratoire CVN

April 13, 2018

Outline

Integers vs. Real numbers

- So far we have only seen LP with constraints and variables that are real numbers (variables must be positive for the simplexe).
- One way to *extend* the application field is to require that some or all aspects of the problem deal only with integers.
- In the case where all constraints and all variables are integers, these problems are called integer programs (IP)
- In the case where only some variables or some constraints are expressed in integers, this is called a mixed program (MP, MIP or sometimes MILP).

Example

- Maximize $z = x_1 + x_2$
- with

$$\begin{aligned} -2x_1 + \ 2x_2 &\geq \ 1 \\ -8x_1 + 10x_2 &\leq 13 \end{aligned}$$

- and $x_1, x_2 \geq 0$
- With the simplex, the real number optimum is $\{x_1 = 4, x_2 = 9/2\}$.
- If the x_i are constrainted to be integers, the optimum is $\{x_1 = 1, x_2 = 2\}$ (very different !)
- How can we move from the real to the integer solution ? Using the closest integer that meet the constraints is not realistic, it might be very far from the optimum.

IP problems categories

1. Problems with discrete I/O: production of objects, etc ;
2. Problems with logic conditions: addition of variables with extra constraints. (for exemple : if product A is made, then also make products B or C...)
3. Combinatorial problems: sequences, ressource allocation, timetables, all NP problems can be formulated as IP.
4. Non-linear problems can often be formulated as IP. Useful when the feasible region is non-convex.
5. Network and graph problems, e.g. graph coloring.

Remarks on IP problems

- Many models fall into the second category (linear pb. with extra logic conditions), thus many problems are LP with only some integer variables added.
- IP formulations are useful (with experience) for modeling a huge class of problems. Perhaps contrary to expectations, restricting conditions leads to a larger class of problems that can be modeled ;
- However, modeling a problem is not solving it. In general IP solving is NP-hard and so solvers have non-polynomial complexity
- The complexity of the solution often depends on the formulation!

Decisions variables

- Among integer variables are Boolean variables that can only take value 0 or 1.
- Such variables are often used in MP for representing décisions : implications, links between variables, etc.
- E.g; decision variables and indicator variables.

Example: a selection problem

- Let there be 4 possible choices, each necessitating resources and yielding an outcome:

choice	resources	yield
1	5	16
2	7	22
3	4	12
4	3	8

- We only have 14 resources units available.
- How to maximise the yield ?
- Extentions :
  1. We must make at most two choices
  2. Choice 2 is only available if Choix 1 is also taken.

Choice problem : formulation

- We use boolean variables (0 and 1), for instance

$$x_j, j \in \{1, \ldots, 4\} = \begin{cases} 1 & \text{Choice taken} \\ 0 & \text{otherwise} \end{cases}$$

- We maximize $z = 16x_1 + 22x_2 + 12x_3 + 8x_4$
- Subject to

$$5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14$$

- Optional constraints :
  1. $x_1 + x_2 + x_3 + x_4 \leq 2$
  2. $x_1 \geq x_2 \Leftrightarrow -x_1 + x_2 \leq 0$

Indicator variables

- We suppose that x models a (real) quantity of some ingredient to be included in some mix/recipe. We wish to distinguish $x = 0$ from the case $x > 0$.
- We introduce the variable δ, which is worth 1 only when $x > 0$, with the constraint

$$x - M\delta \leq 0,$$

  with M a known coefficient representing an upper bound for x.
- with this (linear) constraint, we do have $x > 0 \Rightarrow \delta = 1$.

Indicator variables: followup

- For the opposite implication, ($x = 0 \Rightarrow \delta = 0$), this is a little harder. This implication is equivalent to $\delta = 1 \Rightarrow x > 0$.
- We introduce a less severe notion : $\delta = 1 \Rightarrow x > m$, with m an acceptable minimum level under which we consider x unused (this is application-dependent). An equivalent constraint is then

$$x - m\delta \geq 0.$$

Fixed load problem

- Example : unit of energy production
- This is a general case featuring production costs and marginal costs.
- Generated power is either $P_i = 0$, or $P_i^m \leq P_i \leq P_i^M$.
- with a cost:

$$C_i = \begin{cases} 0 & \text{Production unit is stopped} \\ a_i + b_i P_i & \text{Otherwise} \end{cases}$$

- We wish to produce a certain power level at least cost.
- The cost is here non-linear and even non-continuous, obviously an LP cannot represent the problem well
- How can we at least formulate the problem

Fixed load problem as an IP

- We introduce the variable

$$x_i = \begin{cases} 0 & \text{Unit } i \text{ stopped} \\ 1 & \text{Otherwise} \end{cases}$$

- Constraints are

$$x_i P_i^m \leq P_i \leq x_i P_i^M$$

- with

$$C_i = a_i x_i + b_i P_i$$

  (and not $C_i = a_i x_i + b_i P_i x_i$, which would not be linear).
- Note : condition $P_i = 0, x_i = 1$ is feasible but has a higher cost compared to the case $P_i = 0, x_i = 0$, and so cannot be part of the optimum.

Assignment problems

- We are considering n tasks assigned to n persons ;
- only one task is assigned to a person ;
- the yield corresponding to task i associated to person j is given by a matrix C_{ij} ;
- we wan to maximize the yield ;
- Formulation ?
- Extended formulation: case when the number of persons is greater than the number of tasks.

Assignment formulation

- We introduce the variables x_{ij}

$$x_{ij} = \begin{cases} 1 \text{ task i assigned to person j} \\ 0 \text{ otherwise} \end{cases}$$

- constraints :

$$\begin{aligned} \textstyle\sum_{i=1}^{n} x_{ij} &= 1 (\text{only one task assigned to } j) \\ \textstyle\sum_{j=1}^{n} x_{ij} &= 1 (\text{each task assigned only once}) \end{aligned}$$

- Maximize

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

Extended assignment problem

- If the number of persons $>$ number of tasks, i.e n persons, m tasks, $n > m$:
  $\sum_{i=1}^{m} x_{ij} \leq 1, j = 1, \ldots, n$
- Sum of the tasks assigned to person j is ≤ 1 :
  $\sum_{j=1}^{n} x_{ij}, i = 1, \ldots, m$
- Sum of persons assigned to task i is $= 1$: we add some slack variables :
  $s_j + \sum_{i=1}^{m} x_{ij} = 1, j = 1, \ldots, n$
- Maximize the same profit

$$z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij}$$

Another way to formulate the problem: transport problem

We will see how to formulate and solve such problems efficiently in the last two lectures

- assignment problems
- integer constraints
- max value of x_{ij} is 1, min = 0, thus a boolean problem.

Convert Boolean algebra problems into linear algebra IP with Boolean variables

- We introduce

$$\delta_i = \begin{cases} 1 & \text{if } x_i \text{ is true} \\ 0 & \text{else} \end{cases}$$

- $$\begin{aligned} x_i \text{ true} &\leftrightarrow \delta_i = 1 \\ x_i \text{ false} &\leftrightarrow \delta_i = 0 \end{aligned}$$

Boolean operations

$\sim$	negation
$\wedge$	logical AND
$\vee$	logical OR
$\Rightarrow$	Implication
$\Leftrightarrow$	equivalence
$\oplus$	exclusive OR
T	TRUE
F	FALSE

Some operators

$$
\begin{aligned}
x_1 \vee x_2 = T &\Leftrightarrow \delta_1 + \delta_2 \geq 1 \\
x_1 \wedge x_2 = F &\Leftrightarrow \delta_1 = 1, \delta_2 = 1 \\
\sim x_1 = 1 &\Leftrightarrow \delta_1 = 0 \\
x_1 \Rightarrow x_2 &\Leftrightarrow \delta_2 \geq \delta_1 \\
x_1 \Leftrightarrow x_2 &\Leftrightarrow \delta_1 = \delta_2 \\
x_1 \oplus x_2 = T &\Leftrightarrow \delta_1 + \delta_2 = 1
\end{aligned}
$$

Combinations of logical and continuous variables

- Logical variable = indicator (e.g. open/closed, cold/hot)
- case $x_i = [f(x_i) \leq 0]$, so let $\delta_i = 1$ if $x_i = T$.
- $$\begin{array}{l} \delta_i = 1 \text{ if } f(x_i) \leq 0 \\ \delta_i = 0 \text{ if } f(x_i) \geq \varepsilon \text{ (precision)} \end{array}$$
- We consider
$$\begin{array}{rl} M & = \max_{x_i}(f(x_i)) \\ m & = \min_{x_i}(f(x_i)) \end{array},$$
$M > 0, m < 0.$

Algebraic equations

$$f(x_i) \leq M(1 - \delta_i)$$
$$f(x_i) \geq \varepsilon + (m - \varepsilon)\delta_i$$

1. if $\delta_i = 1$, then $f(x_i) \leq 0, f(x_i) \geq m$.
2. if $\delta_i = 0$, then $f(x_i) \leq M, f(x_i) \geq \varepsilon$.
3. if $f(x_i) \geq 0$, then $\delta_i = 0$ (see proof on blackboard)
4. if $f(x_i) \leq 0$, then $\delta_i = 1$ (same)

Product of Boolean variables

- Consider the non-linear product $\delta_3 = \delta_1 \delta_2$, this leads to non-linear constraints.
- To transform it into a series of linear constraints, we write

$$\delta_3 \leq \delta_1, \qquad \delta_3 \leq \delta_2$$
$$\delta_3 \geq \delta_1 + \delta_2 - 1$$

we verify equivalence :

1. $\delta_1 = 0, \delta_2 = 0 \Rightarrow \delta_3 \leq 0 \Rightarrow \delta_3 = 0$
2. $\delta_1 = 1, \delta_2 = 0 \Rightarrow \delta_3 \leq 0, \delta_3 \geq 0 \Rightarrow \delta_3 = 0$
3. same for $\delta_1 = 0, \delta_2 = 1$
4. $\delta_1 = 1, \delta_2 = 1 \Rightarrow \delta_3 \leq 1, \delta_3 \geq 1 \Rightarrow \delta_3 = 1$
5. inversely : $\delta_3 = 1 \Rightarrow \delta_1 \geq 1, \delta_2 \geq 1, \delta_1 + \delta_2 \leq 2$, therefore $\delta_1 = \delta_2 = 1$.
6. $\delta_3 = 0 \Rightarrow \delta_1 \geq 0, \delta_2 \geq 0, \delta_1 + \delta_2 \leq 1$ so we have at most one of δ_1 or δ_2 equal to zero.

Product of a binary variable and a continuous variable

- Consider the non-linear product of tow variables, one being continuous and the other Boolean: $\delta f(x)$, i.e :

$$y = \begin{cases} 0 & \text{if } \delta = 0 \\ f(x) & \text{if } \delta = 1 \end{cases}$$

- We set the following constraints :

$$\begin{cases} y \leq M\delta \\ y \geq m\delta \\ y \leq f(x) - m(1-\delta) \\ y \geq f(x) - M(1-\delta) \end{cases}$$

Continuous-Boolean product, next

1. Let $\delta = 0$, therefore $y \leq 0, y \geq 0$, so $y = 0$.
   Also $f(x) - M \leq y \leq f(x) - m$, therefore $f(x) - M \leq 0$ and $f(x) - m \geq 0$, which is non-critical.
2. Let $\delta = 1$, then

$$\left.\begin{array}{l} y \leq M \text{ (non-critical)} \\ y \geq m \text{ (non-critical)} \\ y \leq f(x) \\ y \geq f(x) \end{array}\right\} y = f(x)$$

3. The opposite way is also correct : $y = f(x) \Leftrightarrow \delta = 1$.

Applications

- Dynamical and logic systems
  Example:

$$X(t+1) = \begin{cases} 0.8X(t) + U(t) & \text{if } X(t) \geq 0 \\ -0.8X(t) + U(t) & \text{if } X(t) < 0 \end{cases}$$

  with $-10 \leq X \leq 10$, $-1 \leq U \leq 1$.
- Piecewise affine (linear) systems (e.g. income tax optimization).
- Finite-state automata.

A few problems in combinatorial optimization

1. General IP problem

$$\begin{aligned} \min \; & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \leq \mathbf{b} \\ & x_i \text{ integers} \end{aligned}$$

2. Special case: Boolean programming

$$\begin{aligned} \min \; & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \leq \mathbf{b} \\ & x_i = \text{ 0 or 1}, \mathbf{x} \in \mathcal{B} \end{aligned}$$

Typical problems - I

3. Integer Knapsack: only one constraint

$$\max \ \mathbf{C}^T\mathbf{x}$$
$$\mathbf{a}^T\mathbf{x} \ \leq b$$
$$\mathbf{x} \in \mathcal{B}$$

Note : $\mathbf{a} > 0$, else, $x'_i = 1 - x_i$.

4. Multiple Knapsack

$$\max \ \mathbf{C}^T\mathbf{x}$$
$$\mathbf{A}\mathbf{x} \ \leq \mathbf{b}$$
$$\mathbf{x} \in \mathcal{B}$$

Note : $\forall i, j, A_{ij} \geq 0$.

Typical problems - II

5. Coupling (a.k.a. set matching)

$$\begin{aligned} \max \ & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \leq \mathbf{1} \text{ (vector of 1s)} \\ & \mathbf{x} \in \mathcal{B} \end{aligned}$$

   Note : $\forall i, j, A_{ij} \in \mathcal{B}, \mathbf{x} \in \mathcal{B}$.

6. Set covering

$$\begin{aligned} \min \ & \mathbf{C}^T\mathbf{x} \\ & \mathbf{Ax} \geq \mathbf{1} \text{ (vector of 1s)} \\ & \mathbf{x} \in \mathcal{B} \end{aligned}$$

   Note : $\forall i, j, A_{ij} \in \mathcal{B}, \mathbf{x} \in \mathcal{B}$.

Typical problems - III

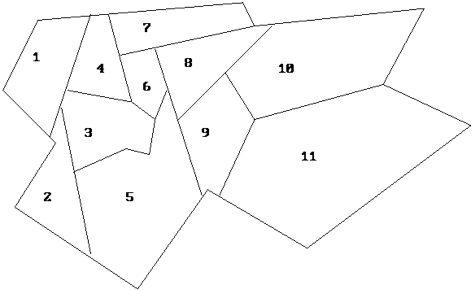
7. Set partitionning

$$\begin{aligned} \min \ & \mathbf{C}^T\mathbf{x} \\ & \mathbf{A}\mathbf{x} = \mathbf{1} \text{ (vector of 1s)} \\ & \mathbf{x} \in \mathcal{B} \end{aligned}$$

Note : $\forall i, j, A_{ij} \in \mathcal{B}, x \in \mathcal{B}$.
Special case: assignment problems

Example of a set covering problem

Let there be some city, composed of the following neighborhoods (”arrondissements”)



We are tasked with building a number of hospitals that will take care of medical emergencies in their own and adjacent neighborhoods. We must minimize the number of hospitals to be built.

Formulation of the hospitals problem

- Let there be a Boolean variable x_i associated with each neighborhood. If x_i is 1, this means a hospital will be built in neighborhood i, and none will be built in this neighborhood if it is zero.
- Here is a possible formulation

$\min \sum_i x_i$, subject to:

$$
\begin{array}{l}
x_1 + x_2 + x_3 + x_4 \geq 1 \\
x_1 + x_2 + x_3 + x_5 \geq 1 \\
x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \\
x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \\
x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \geq 1 \\
x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \\
x_4 + x_6 + x_7 + x_8 \geq 1 \\
x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \geq 1 \\
x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
x_8 + x_9 + x_{10} + x_{11} \geq 1 \\
x_9 + x_{10} + x_{11} \geq 1
\end{array}
$$

- We need to “cover” the set of neighborhoods optimally: a solution is given by $x_3 = x_8 = x_9 = 1$ and the rest 0. A solution with three hospitals is not unique but is optimal.

Conclusion

- Many problems can be modelled by integer or mixed programming (IP or MP).
- E.g: logic problems, assignment, set problems, etc.
- IP and MP are powerful tools for modelling.
- To solve these problems, we need to answer these questions:
  - existence of a solution ?
  - unicity ?
  - algorithm for a resolution ?
- We shall see that IP and MP are generally NP-hard, so difficult to solve exactly.
- There exists to principal methods for solving them: the Gomory plane-cutting method and Branch-and-bound.