M2 ORO: Advanced Integer Programming

Sophie Demassey

Mines Nantes - TASC - INRIA/LINA CNRS UMR 6241
sophie.demassey@mines-nantes.fr

September 19, 2011

Part I

Introduction

Today's lecture:

Outline

what will this class be about ?

modeling

with discrete variables and linear constraints

- some tips
- many examples

tightening

bounds with cutting-planes

- basic theory of polyedra
- specific classes of valid inequalities

solving

(mixed) integer linear programs with branch-and-bound methods

- preprocessing
- relaxations
- branching rules

splitting up

large-scale problems with LP decomposition techniques

- column-generation
- Lagrangian relaxation
- Bender's decomposition

what should you already know ?

undergraduate mathematics

- linear algebra
- logic
- proof techniques

basics of complexity

- time/space complexity
- average/worst case
- problem classes P/NP/NP-C

basics of linear programming

- linear model
- convex polyhedra, simplex,
- duality,

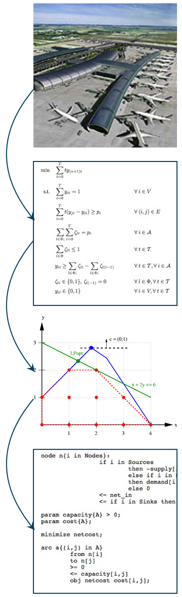
computer programming

- language(s) with objects
- debugging
- performance tests

what should you be able to do afterward ?

from the specifications of a complex problem:

- formulate a "good" mathematical model
- select and combine "appropriate" LP-based solution techniques
- use solvers and tune them for improved performance



what will happen ?

participate please !

19 sep-14 nov: 9 interactive lectures+exercises

1:	introduction	2-4:	modeling
5-6:	searching	7-8:	bounding
9:	decompositions		

every monday morning 9h30-12h30, EMN room A119-B113.

- sep 19, sep 26: room A119
- + sep 28 : wednesday morning 9h30-12h30, room A119
- oct 3, oct 10, oct 17, oct 24, nov 7, nov 14: room B113

- 3h: nov 21

readings

- L. Wolsey, Integer Programming, 1998, Wiley.
- G. Nemhauser and L. Wolsey, Integer and Combinatorial Programming, 1988, Wiley.
- C. Papadimitriou and K. Steiglitz, Combinatorial Optimization, 1982, Prentice-Hall.
- V. Chvátal, Linear Programming, 1983, Freeman.

whoami ?

http://www.emn.fr/z-info/sdemasse/

- studied Maths (commutative algebra)
- got a Ph.D. in C.S. (combinatorial optimization)
- now associate professor at École des Mines de Nantes, researcher in the Constraints group of LINA CNRS UMR 6241
- interested in hybridations of constraint programming and integer linear programming
- non native-English speaker... tell if you don't understand me !

Outline

Definitions

optimization is...

a branch of mathematics that studies real-life systems where: there is an objective to achieve and constraints to satisfy

mathematical model is...

an abstract representation of a system given as mathematical relations over a set of variables

mixed integer linear program is...

a mathematical model of an optimization problem where:
- the objective is to minimize or maximize a linear expression
- all constraints are linear equalities and inequalities
- some variables have integer or binary values

Definitions

- Mixed Integer Linear Program (MILP): some variables have integer values
- Integer Linear Program (ILP or IP): all variables have integer values
- Binary Integer Linear Program (BILP or BIP): all variables have binary values

Quizz...

programming

1. what does programming mean in mathematical programming ?

LP and convexity

1. define: convex set, convex hull, (convex) polyhedron, polytope
2. show that the set P of the feasible solutions of an LP is convex
3. show that if P is a non-empty polyhedron then one of its vertex is an optimal solution
4. cite algorithms for solving LP

Mixed Integer Linear Program (MILP)

MILP (standard form)

$$\max \sum_{j=1}^{n} c_j x_j + \sum_{k=1}^{p} h_k y_k \tag{1}$$

$$\text{s.t. } \sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{p} g_{ik} y_k \leq b_i \qquad \forall i \in \{1, 2, \ldots, m\}, \tag{2}$$

$$x_j \in \mathbb{Z}_+ \qquad \forall j \in \{1, 2, \ldots, n\}, \tag{3}$$

$$y_k \in \mathbb{R}_+ \qquad \forall k \in \{1, 2, \ldots, p\}. \tag{4}$$

instance: $c \in \mathbb{Q}^n, h \in \mathbb{Q}^p, a \in \mathbb{Q}^{m \times n}, g \in \mathbb{Q}^{m \times p}$

Mixed Integer Linear Program (MILP)

MILP (standard vectorial form)

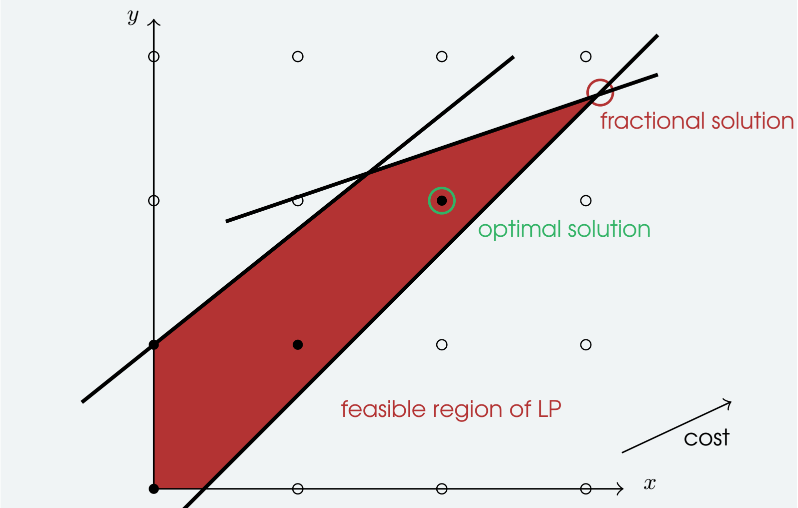
$$\max \{ \, cx + hy \mid ax + gy \leq b, x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+ \, \}$$

- solution $\in \{ \, (x, y) \mid x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+ \, \}$
- feasible solution $\in \{ \, (x, y) \mid ax + gy \leq b, x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+ \, \}$
- optimal solution $\in \operatorname{argmax}\{cx + hy \mid ax + gy \leq b, x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+\}$
- fractional solution $\in \operatorname{argmax}\{cx + hy \mid ax + gy \leq b, x \in \mathbb{R}^n_+, y \in \mathbb{R}^p_+\}$
- optimum $= \max \{ \, cx + hy \mid ax + gy \leq b, x \in \mathbb{Z}^n_+, y \in \mathbb{R}^p_+ \, \}$

Usually, solving a MILP is to find an optimal solution
Sometimes, one may be only interested in the optimum value or in a *good* feasible solution

Mixed Integer Linear Program (MILP)

IP with two variables $x, y \in \mathbb{Z}_+$ (graphic representation)



Modeling with integers

many optimization problems are combinatorial:

- look for an object from a finite or countably infinite set
  ex: an integer (*number of trains*), a set (*part of staff*), a permutation (*order of tasks*), a graph (*path in a network*)
- the physical entities are indivisible and must be associated to discrete variables
  ex: train, employee, task, arc
- the constraints are nonlinear and can be modeled using 0-1 (binary) variables
  ex: yes/no decisions (*employee x is assigned or not*), logical conditions (*if train x is scheduled then rail track y is occupied a time t*), nonlinear cost (*set-up costs of tasks*)

Modeling with integers

a combinatorial optimization problem is...

formally defined by

$$min_{S \subseteq N} \{ \sum_{j \in S} c_j \mid S \in \mathcal{F} \}$$

given:

- a finite set N
- a set $\mathcal{F}$ of subsets of N
- a vector of weights $c \in \mathbb{R}^N$

- $\mathcal{P}(N)$ is the space of solutions (the search space)
- $\mathcal{F}$ is the set of feasible solutions of the COP

$\longrightarrow$ many combinatorial problems can properly be modeled as MILP

How hard is integer programming ?

how hard is LP ?

LP$\in \mathcal{P}$

- the ellipsoid algorithm is polynomial-time
- the simplex algorithm is practical (even if non-polynomial)

how hard is MILP ?

MILP is $\mathcal{NP} - complete$ in the $strong$ sense

- no known (pseudo)polynomial-time algorithm for solving MILP
- no known general practical algorithm for large MILP

LP relaxation

IP is much harder than LP ?

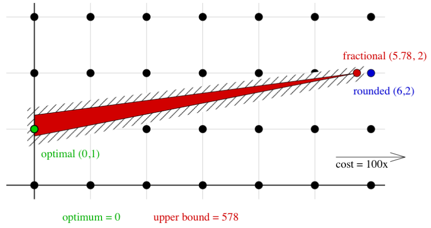
Why not solve the LP relaxation and round the solutions to the closest integer ?

- solving the associated LP relaxation results in a upper bound to the optimal solution to the MILP (for maximization)

Why not solve the LP relaxation and round the solutions ?

because, in many cases:

- the optimum value of the LP relaxation is far from that of the IP
- one need a solution and not only the optimum value:
  - rounding an optimal solution of LP gives an integer solution but not necessarily a feasible solution of IP
  - rounding may result in a feasible solution far from optimal



- often even worse for BIP: $(0.5, 0.5, \ldots)$ can be a fractional solution but gives no information

Quizz...

Rouding the fractional solution

1. draw and solve the following IP and its LP relaxation

$$\begin{aligned} \max\ & x + 0.64y \\ \text{s.t. } & 50x + 31y \leq 250 \\ & 3x - 2y \geq -4 \\ & x, y \in \mathbb{Z}_+ \end{aligned}$$

Then, how to solve a MILP ?

with the simplex (or any LP) algorithm !

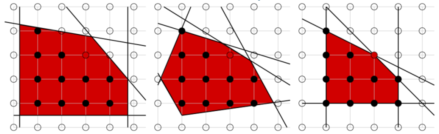
- alone (if the fractional solution is integer)
- with cutting-planes (to shrink the LP polyhedron)
- with branch-and-bound (intelligent enumeration of the solutions)
- with decomposition (separation of the subproblems)

these are generic algorithms, they can be combined, they can be adapted to specific classes of problems

BUT no general algorithm is known that is practical for any large MILP

Then, how to solve a MILP ?

- we will see these methods in the later lectures
- first we need to know how to model as MILP
- a typical MILP can have many formulations
- not all formulations are created equal



- then we need to find the **good** formulations

Quizz...

combinatorial

1. what is the nature of LP: a continuous or combinatorial optimization problem ?
2. describe a MILP where a fractional solution is an optimal solution
3. prove that solving the associated LP relaxation results in a upper bound to the optimal solution to the MILP (for maximization)

Part II

Modeling with Integers (I)

Outline

Modeling with discrete variables

combinatorial problems

In many optimization problems, some physical entities are indivisible and one need to find:

- count: the number of elements in a finite discrete set
- selection: one (best) element out of a finite discrete set
- order: a permutation of a list of elements
- schedule: a timed permutation
- graph: a subtructure in a given graph

Modeling with discrete variables

combinatorial problems

- indivisible entities must be associated to discrete variables
- mapping: discrete $\leftrightarrow$ non-negative integer $\mathbb{Z}_+$
- feasibility and optimality conditions can often be modeled as linear functions of the discrete variables

Examples of classes of combinatorial problems...

Modeling scheduling problems

scheduling

- decide when to commit resources between a variety of possible tasks
- tasks are partially ordered and associated to shared resources
- order and date their execution
- production process (jobs/machines), computing (processes/processors), project management
- discrete variables:
  - assign a task to a time period (ex: starting time of a job)
  - count the load of a resource at each time
  - model the relative order of two tasks

Modeling timetabling problems

Timetabling

- decide how to commit resources between a variety of possible tasks
- tasks are associated to time periods
- coordinate the resources to execute a task
- assign a valid sequence of tasks to each resource
- transport (flight/airplane), workplace (shift/nurse), school (class/teacher& room)
- discrete variables:
  - assign a task to a resource (ex: shift of a nurse)
  - model the in/compatibility of 2 resources at some time
  - model a sequence of tasks
  - assign a sequence of tasks to a resource

Modeling routing problems

Routing

- selecting paths in a network along which to send network traffic
- different travelling measures: cost, load, distance, time
- assign vehicles to paths
- capacity, precedence between nodes, time windows, etc.
- discrete variables:
  - assign a node to a vehicle
  - model the relative order of 2 nodes in a path
  - model a path
  - count the cost or weight of a path
  - assign a path to a vehicle

Modeling placement problems

packing and geometric placement

- assign items to containers and place them in 1D, 2D or 3D without overlapping
- different packing measures: profit, size, weight
- minimize the number of containers or the total gap, maximize the profit of the packed items
- some items are incompatible or must be at a given distance
- the lenghts on some dimension can be cumulated
- manufacturing (cut of paper, steel), transport and stock, location and sizing, puzzles
- discrete variables:
  - assign an item to a container
  - assign a container to a set of items
  - count the load of a container
  - assign an item to a coordinate

Modeling with discrete variables

some other classes of combinatorial problems:

network design, configuration, geometrical tiling, satisfiability, frequency allocation, phylogenetic, coloring, localization, motion planning,...

Many real-world problems are composed of subproblems of the preceding classes, ex:

- job scheduling and human resources
- vehicle routing and packing 3D

Outline

Modeling with binary variables

boolean condition as a 0-1 variable

- decision: *is item j selected ?*
- assignment: *is item j assigned to value i ?*
- value indicator: *is variable x positive (or $x \geq \alpha$) ?*
- condition indicator: *does constraint c hold ?*

Modeling with binary variables

logical/numeric condition as a linear combination of 0-1 variables

- (exclusive) disjunction: *(either) δ_1 or δ_2*
- dependency: *if δ_1 then δ_2*
- exclusive alternative: *exactly 1 out of n*
- counter or bound: *exactly/at least/at most k out of n*

Modeling with binary variables

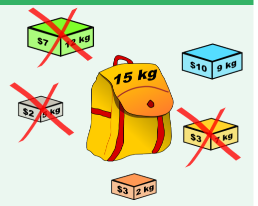
non-linear value functions with 0-1 variables

- set-up value: $f(x) =$ either $a(x)$ or $a(x) + b$
- discrete values: $f(x) = f_i$ if $x = i$
- piecewise linear: $f(x)$ linear on $[a_{i-1}, a_i]$

Modeling yes/no decision

Example: Integer Knapsack Problem

Given n items with associated *values* c_j and *weights* w_j and a knapsack of capacity K; find a subset of items of maximum value to pack (the total weight does not exceed K).



- *is item j selected ?*
  - 0-1 decision variable:

$$\boxed{x_j = 1 \text{ iff } j \text{ is selected.}}$$

Modeling yes/no decision

Integer Knapsack Problem

$$\max \sum_{j=1}^{n} c_j x_j \tag{5}$$

$$\text{s.t.} \sum_{j=1}^{n} w_j x_j \leq K \tag{6}$$

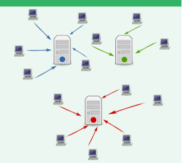
$$x_j \in \{0,1\} \qquad j = 1, \ldots, n \tag{7}$$

- $x_j = 1$ iff item j is selected

Modeling multiple choice (1 out of n)

Example: Weighted Matching Problem

Given n tasks and m resources associated to preferences c_{ij} for assigning task i to resource j; find an assignment of tasks to resources that maximize the sum of the preferences.



- *is task i assigned to resource j?*
  - 0-1 decision variable: $x_{ij} = 1$ iff i is assigned to j
- *i is assigned to exactly one resource j.*
  - constraint:

$$x_{ij} = 1 \iff x_{ik} = 0, \ \forall k \neq j$$

  - linearized as:

$$\boxed{\sum_{j=1}^{m} x_{ij} = 1}$$

Modeling multiple choice (1 out of n)

Weighted Matching Problem

$$\max \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \tag{8}$$

$$\text{s.t. } \sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1,\dots,n \tag{9}$$

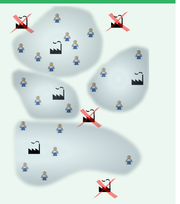
$$x_{ij} \in \{0,1\} \qquad i = 1,\dots,n;\ j = 1,\dots,m \tag{10}$$

- $x_{ij} = 1$ iff task i is assigned to resource j

Modeling dependency (if-then)

Example: Uncapacitated Facility Location Problem

Given n potential facility locations and m customers to serve from one facility, associated to costs c_j for opening facility j and d_{ij} for serving customer i from facility j; find a subset of locations to open facilities that minimizes the cost (opening and service).



- *is facility j opened ?*
  - 0-1 decision variable: $x_j = 1$ iff j is open
- *is customer i served from facility j? i is served from exactly one j.*
  - 0-1 decision variable: $y_{ij} = 1$ iff i is served from j
  - constraint: $\sum_{j=1}^{m} y_{ij} = 1$
- *if customer i is served from facility j then j is open.*
  - constraint:

$$y_{ij} = 1 \implies x_j = 1$$

  - linearized as:

$$\boxed{y_{ij} \le x_j}$$

Modeling dependency (if-then)

Uncapacitated Facility Location Problem

$$\min \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n}\sum_{i=1}^{m} d_{ij}y_{ij} \tag{11}$$

$$\text{s.t. } \sum_{j=1}^{n} y_{ij} = 1 \qquad i = 1,\dots,m \tag{12}$$

$$y_{ij} \le x_j \qquad j = 1,\dots,n;\ i = 1,\dots,m \tag{13}$$

$$x_j \in \{0,1\} \qquad j = 1,\dots,n \tag{14}$$

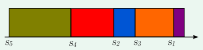
$$y_{ij} \in \{0,1\} \qquad j = 1,\dots,n;\ i = 1,\dots,m \tag{15}$$

- $x_j = 1$ iff facility j is opened
- $y_{ij} = 1$ iff customer i is served from facility j

Modeling exclusive disjunction (either-or)

Example: Scheduling Problem $1||C_{max}$

Find a minimal makespan schedule of n tasks with durations $p_i \in \mathbb{Z}_+^*$ on one machine, i.e. no two tasks are executed simultaneously.



- *starting time of task j ?*
  - integer variable: $s_j \in \mathbb{Z}_+^*$
- *is task j preceding task i ?*
  - 0-1 indicator variable: $x_{ij} = 1$ iff $s_j - s_i \ge p_i$
  - linearized constraint (let $M = \sum_{j=1}^{n} p_j$):

$$\boxed{s_j - s_i \ge M x_{ij} + (p_i - M)}$$

- *either i precedes j or j precedes i.*
  - constraint: $x_{ij} = 1 \iff x_{ji} = 0$
  - linearized as:

$$\boxed{x_{ij} + x_{ji} = 1}$$

- note: $x_{ij} = 1 \Leftarrow s_j - s_i \ge p_i$ is a consequence of the two

Modeling dependency (either-or)

Scheduling Problem $1||C_{max}$

$$
\begin{aligned}
& \min s_{n+1} & & (16)\\
\text{s.t. } & s_{n+1} \geq s_j + p_j & j = 1,\dots,n & \quad (17)\\
& s_j - s_i \geq M x_{ij} + (p_i - M) & i,j = 1,\dots,n & \quad (18)\\
& x_{ij} + x_{ji} = 1 & i,j = 1,\dots,n; i < j & \quad (19)\\
& s_j \in \mathbb{Z}_+ & j = 1,\dots,n+1 & \quad (20)\\
& x_{ij} \in \{0,1\} & i,j = 1,\dots,n & \quad (21)
\end{aligned}
$$

- s_j starting time of task j
- $x_{ij} = 1$ iff task i is preceding task j

Quizz...

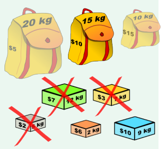
scheduling problem

1. what is the complexity of the preceding scheduling problem ?
2. propose a method to solve it
3. is it necessary to set variables s integer ?

Modeling discrete values

Example: Knapsack Problem with Variable Capacity

A knapsack is available in m different capacities K_i (increasing) at different values C_i (decreasing: it is preferable to carry a smal weight); find the packing of n items with values c_j and weights w_j in one knapsack that maximizes the total value of the pack (items+sack)



- *is item j selected ?*
  - 0-1 decision variable: $x_j = 1$ iff j is selected
- *is configuration (K_i, C_i) chosen ? choose exactly one.*
  - 0-1 decision variable: $y_i = 1$ iff i is chosen
  - the knapsack cost is $\sum_{i=1}^m C_i y_i$ and capacity $\sum_{i=1}^m K_i y_i$
  - with constraint: $\sum_{i=1}^m y_i = 1$

Special Ordered Set of type 1 (SOS1)

a sequence $\{y_1, \dots, y_m\}$ of positive variables in which no more than one member is non-zero in any feasible solution.

Modeling discrete values

Knapsack Problem with Variable Capacity

$$
\begin{aligned}
& \max \sum_{i=1}^{n} C_i y_i + \sum_{j=1}^{m} c_j x_j & & (22)\\
\text{s.t. } & \sum_{j=1}^{m} w_j x_j \leq \sum_{i=1}^{n} K_i y_i & & (23)\\
& \sum_{i=1}^{n} y_i = 1 & & (24)\\
& x_j \in \{0,1\} & j = 1,\dots,m & \quad (25)\\
& y_i \in \{0,1\} & i = 1,\dots,n & \quad (26)
\end{aligned}
$$

- $x_j = 1$ iff item j is selected
- $y_i = 1$ iff configuration (K_i, C_i) is selected

Modeling discrete values

Knapsack Problem with Variable Capacity

$$\max \sum_{i=1}^{n} C_i y_i + \sum_{j=1}^{m} c_j x_j \tag{22}$$

$$\text{s.t. } \sum_{j=1}^{m} w_j x_j \leq \sum_{i=1}^{n} K_i y_i \tag{23}$$

$$SOS1(y_1, \ldots, y_m) \tag{24}$$

$$x_j \in \{0,1\} \qquad j = 1, \ldots, m \tag{25}$$

$$y_i \in \{0,1\} \qquad i = 1, \ldots, n \tag{26}$$

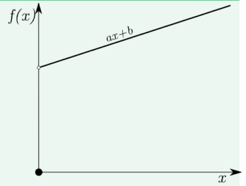
Most solvers allow you to specify SOS1
and implement a specialized solution algorithm (branching)

Modeling a set-up value

Example: linear cost function with set-up

$$f : [0, U] \subseteq \mathbb{R}_+ \to \mathbb{R}_+$$

$$f(x) = \begin{cases} 0 & \text{if } x = 0 \\ ax + b & \text{if } 0 < x \leq U \end{cases}$$



- *is variable x positive ?*
  - 0-1 decision variable: $\delta = 1$ iff $x > 0$
  - linearized constraint:

    $$\boxed{\delta \leq x \leq U\delta}$$

  - the cost becomes the linear function:

    $$\boxed{f(x) = ax + b\delta}$$

Quizz...

Intervals

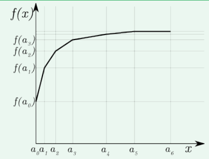
1. represent graphically the set of solutions defined by: $y \in \{0,1\}$, $x \in \{0, \ldots, U\}$, $y = 1 \iff x > 0$ and deduce a linearization
2. idem for $y = 1 \iff x > a$ with $0 < a < U$
3. idem for $y = 1 \iff a < x \leq b$ with $0 < a < b < U$

Modeling a piecewise linear function

- useful to approximate some real cost functions

Example: piecewise linear cost function

Let real numbers $a_0 < a_1 < \cdots < a_n$ and a cost function $f : [a_0, a_n] \subseteq \mathbb{R}_+ \to \mathbb{R}_+$ that is continuous and linear on each interval $[a_{i-1}, a_i]$



- convexity of (f on) each interval: $x \in [a_{i-1}, a_i] \implies \exists \lambda_{i-1}, \lambda_i > 0$ st

$$(\lambda_{i-1} + \lambda_i = 1) \quad \wedge \quad (x = \lambda_{i-1} a_{i-1} + \lambda_i a_i) \quad \wedge \quad (f(x) = \lambda_{i-1} f(a_{i-1}) + \lambda_i f(a_i))$$

- $\boxed{x = \sum_i \lambda_i a_i}$ with one or two consecutive nonzero λ_j's

Special Ordered Set of type 2 (SOS2)

a sequence $\{\lambda_0, \ldots, \lambda_n\}$ of positive variables in which not more than two consecutive members may be nonzero in a feasible solution.

Modeling a piecewise linear function

Example: piecewise linear cost function

$$\min \sum_{i=0}^{n} \lambda_i f(a_i) \tag{27}$$
$$\text{s.t. } x = \sum_{i=0}^{n} \lambda_i a_i \tag{28}$$
$$SOS2(\lambda_0, \dots, \lambda_n) \tag{29}$$
$$\sum_{i=0}^{n} \lambda_i = 1 \tag{30}$$
$$\lambda_i \geq 0 \qquad i = 0, \dots, n \tag{31}$$
$$x \in S \tag{32}$$

Even if variables are fractional, this is an integer program !

Quizz...

Question

1. linearize the following problem:

$$SOS2(\lambda_0, \dots, \lambda_n) \tag{33}$$
$$0 \leq \lambda_i \leq U_i \qquad i = 0, \dots, n \tag{34}$$

Modeling with binary variables: Outline 1

boolean condition as a 0-1 variable

indicator	linearization
$\delta = 1 \iff x > 0, x \in \mathbb{Z}_+$	$\delta \leq x \leq U\delta$
$\delta = 1 \iff x > a$	$(a+\epsilon)\delta \leq x \leq a + (U-a)\delta$
$\delta = 1 \iff a \leq x < b$	need 2 indicators
$\delta = 1 \implies Ay \geq a$	$Ay \geq m + (a-m)\delta$
$\delta = 0 \implies Ay \geq a$	$Ay \geq m + (a-m)(1-\delta)$
$\delta = 1 \implies Ay < b$	$Ay \leq M + (b+\epsilon-M)\delta$
$Ay \geq b \implies \delta = 0$	$Ay \leq M + (b+\epsilon-M)\delta$

with $x \in [0, U] \subseteq \mathbb{R}_+$, $Ay \in [m, M] \subseteq \mathbb{R}$ and indicator $\delta \in \{0, 1\}$

- It is often not necessary to enforce equivalence ($\iff$) between the indicator and the condition: one-sense implication ($\implies$) can be enough (and simplier)

Modeling with binary variables: Outline 2

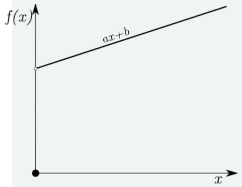
logical/numeric condition as a linear combination of 0-1 variables

condition	example	linearization
exclusive disjunction	*either c or $\bar{c}$*	$\delta = 1 \iff c$
exclusive disjunction	*either c_1 or c_2*	$\delta_1 + \delta_2 = 1$
disjunction	*c_1 or c_2*	$\delta_1 + \delta_2 \geq 1$
dependency	*if c_1 then c_2*	$\delta_2 \geq \delta_1$
exclusive alternative	*exactly 1 out of n*	$\sum_{i=1}^n \delta_i = 1$
counter	*exactly k out of n*	$\sum_{i=1}^n \delta_i = k$
bound	*at least k out of n*	$\sum_{i=1}^n \delta_i \geq k$
bound	*at most k out of n*	$\sum_{i=1}^n \delta_i \leq k$

Modeling with binary variables: Outline 3

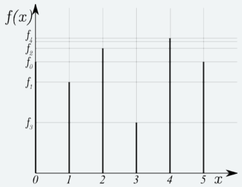
non-linear value function with 0-1 variables

set-up value:
$f(x) = a(x) \vee a(x) + b$



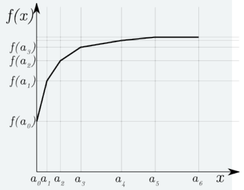
$f(x) = a(x) + b\delta$

discrete values:
$f(x) = f_i$ if $x = i$



$f(x) = \sum_i \delta_i f_i$
with $SOS1(\delta_i)$

piecewise linear:
linear on $[a_{i-1}, a_i]$



$f(x) = \sum_i \delta_i f(a_i)$
with $SOS2(\delta_i)$

Modeling with discrete variables
Modeling with binary variables
logical/numeric condition
non-linear value functions
outline

How to model as a MILP

Given an problem instance

- identify and differentiate data (constant) and unknowns (variables)
- define a set of variables
- use these variables to define the feasibility constraints and the objective function (even non-linear)
- define an additional or alternative set of variables to model non-linear or missing constraints
- ensure that the feasible solutions correspond to this model

It is not always easy to define a first set of variables:

- identify the class of your problem and look for the literature to find usual formulations
- for COP, identify the groud set N and use an incidence vector $x \in \{0,1\}^N$: $x_j = 1$ iff $j \in S$

Sophie Demassey, TASC, EMN-INRIA/LINA
Université de Nantes / M2 ORO / Advanced Integer Programming / L2: modeling

Part III

Improving Models

Outline

alternative MILP

a discrete set of solutions can be modeled as different IPs

having either:

- the same variables but different constraints
- different variables and constraints

how to choose between two models ?

criteria for good models

a *good* model

- small-size: polynomial number of constraints/variables
- flexible: about slight changes of the problem
- easy LP: the LP relaxation is solved fast
- structured: call for decomposition
- effective LP: the LP relaxation value is close to the optimum

These criteria may be complementary, ex:

- small $\Longrightarrow$ easy LP

or contradictory, ex:

- easy LP / effective LP
- small / structured

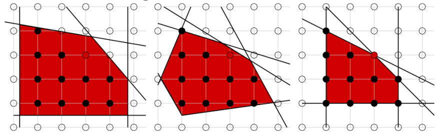
criteria for good models

a *good* model

- small-size: polynomial number of constraints/variables
- flexible: about slight changes of the problem
- easy LP: the LP relaxation is solved fast
- structured: call for decomposition
- effective LP: the LP relaxation value is close to the optimum

formulation strength

Let $X \subseteq \mathbb{Z}^n$ a discrete set, e.g. the feasible solution set of an IP.



formulation strength

- a formulation for X is any polyhedron $P \subseteq \mathbb{R}^n$ such that $X = P \cap \mathbb{Z}^n$
- $conv(X)$ is the strongest formulation of X:
  - the optimum of a linear program over $conv(X)$ is at an extreme point
  - the extreme points of $conv(X)$ all lie in X
  - $max\{cx \mid x \in X\} = max\{cx \mid x \in conv(X)\}$
  - the second program can be solved in polynomial time

characterizing $conv(X)$ is as hard as solving the IP

explicit convex hull

There are a number of problems for which $conv(X)$ can easily be characterized

- assignment problem
- spanning tree problem
- matching problem

totally unimodular matrix

$$(P) = \max\{\ cx \mid Ax \leq b, x \in \mathbb{Z}^n_+\ \}$$

- basic feasible solutions of the LP relaxation $(\bar{P})$ take the form: $\bar{x} = (\bar{x}_B, \bar{x}_N) = (B^{-1}b, 0)$ where B is a square submatrix of (A, I_m)
- Cramer's rule: $B^{-1} = B^*/det(B)$ where B^* is the adjoint matrix (made of products of terms of B)
- Proposition: if (P) has integral data (A, b) and if $det(B) = \pm 1$ then $\bar{x}$ is integral

Definition

A matrix A is totally unimodular (TU) if every square submatrix has determinant $+1$, -1 or 0.

Proposition

If A is TU and b is integral then any optimal solution of $(\bar{P})$ is integral.

totally unimodular matrix

How to recognize TU ?

Sufficient condition

A matrix A is TU if

- all the coefficients are $+1$, -1 or 0
- each column contains at most 2 non-zero coefficient
- there exists a partition (M_1, M_2) of the set M of rows such that each column j containing two non zero coefficients satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.

Proposition

A is TU $\iff$ A^t is TU $\iff$ (A, I_m) is TU
where A^t is the transpose matrix, I_m the identitiy matrix

Quizz: easy problems

Capacitated Transhipment Problem

Given a digraph $G = (V, A)$ with either (positive) demand or (negative) supply b_i at each node $i \in V$, arc capacities h_{ij}, and unit flow costs c_{ij} for all $(i, j) \in A$. Find a feasible integer flow that satisfies all the demands at minimum cost.

1. model as an IP
2. analyze this problem

Answer

Capacitated Transhipment Problem

$$
\begin{aligned}
\min \ & \sum_{(i,j)\in A} c_{ij} x_{ij} & \\
\text{s.t.} \ & \sum_{j \in \delta^+(i)} x_{ij} - \sum_{j \in \delta^-(i)} x_{ij} = b_i & i \in V \\
& x_{ij} \leq h_{ij} & (i,j) \in A \\
& x_{ij} \in \mathbb{Z}_+ & (i,j) \in A
\end{aligned}
$$

- $x_{ij} = 1$ is the flow value on arc $(i, j) \in A$
- $\delta^+(i)$ and $\delta^-(i)$ are the sets of successors and predecessors of $i \in V$, resp.

Answer

Capacitated Transhipment Problem

- The matrix is $\begin{pmatrix} M \\ I_A \end{pmatrix}$ with $M \in \{0, 1, -1\}^{V \times A}$ the incidence matrix of the graph and I_A the identity matrix
- each column x_{ij} in M has coefficients:

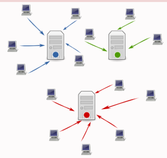
$$
a_{rij} = \begin{cases} 1 & \text{in row } r = i \\ -1 & \text{in row } r = j \\ 0 & \text{in all other rows r} \end{cases}
$$

- the rows of M can be partitioned as $M_1 = M$ and $M_2 = \emptyset$ such that: $\sum_{r \in M_1} a_{rij} - \sum_{r \in M_2} a_{rij} = (1 + (-1)) - 0 = 0$ for all column x_{ij}
- M is TU then the matrix of the IP is TU
- if demands b and capacities h are all integral then any optimum network flow is integral.

Quizz: easy problems

Weighted Matching Problem

Given n tasks and m resources associated to preferences c_{ij} for assigning task i to resource j; find an assignment of tasks to resources that maximizes the sum of the preferences.



Assignment problem

Special case where $n = m$ and each resource is assigned to exactly one task.

Answer

Assignment Problem

- it is a special case of the transshipment (min-cost flow) problem where the graph is bipartite: $V = Tasks \cup Resources$, $A = Tasks \times Resources$, capacities are all equal to 1, demands are 1 at a node-$task$ and -1 at a node-$resource$

other special minimum cost flow problems:

- the shortest $s-t$ path problem
- the maximum $s-t$ flow problem

Answer

Weighted Matching Problem

$$\max \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \tag{35}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{ij} = 1 \qquad i = 1,\dots,n \tag{36}$$

$$x_{ij} \leq 1 \qquad i = 1,\dots,n;\ j = 1,\dots,m \tag{37}$$

$$x_{ij} \in \mathbb{Z}_+ \qquad i = 1,\dots,n;\ j = 1,\dots,m \tag{38}$$

- the matrix $\begin{pmatrix} M \\ I_{n\times m} \end{pmatrix}$ is TU since each column x_{ij} of M has at most one non-zero coefficient:

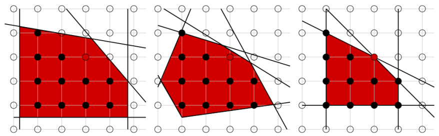
$$a_{rij} = \begin{cases} 1 & \text{in row } r = i \\ 0 & \text{in all other rows r} \end{cases}$$

Outline

stronger formulation: formal definition

A typical IP is not easy:

$$P \subseteq conv(X) \subsetneq \bar{P}$$



- the closer $\bar{P}$ is to $conv(X)$ (in the direction of c), the better

Definition

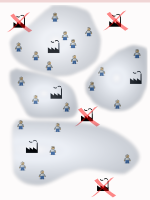
Given two formulations P_1 and P_2 of a same ILP,

- P_1 is at least as strong as P_2 if $P_1 \subseteq P_2$
- P_1 is stronger than P_2 if $P_1 \subsetneq P_2$
- P_1 is ideal or integral if $P_1 = conv(X)$

Quizz: strengthening constraints

Uncapacitated Facility Location Problem

Given n potential facility locations and m customers to serve from one facility, associated to costs c_j for opening facility j and d_{ij} for serving customer i from facility j; find a subset of locations to open facilities that minimizes the cost (opening and service).



1. model this problem as a BIP
2. derive a second model by aggregating the dependency constraints
3. compare the size and the strength of the two models

Answer: disaggregated model

Uncapacitated Facility Location Problem

$$\begin{aligned}
\min\ & \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n}\sum_{i=1}^{m} d_{ij} y_{ij} & \\
\text{s.t. } & \sum_{j=1}^{n} y_{ij} = 1 & i = 1,\dots,m \\
& y_{ij} \le x_j & j = 1,\dots,n;\ i = 1,\dots,m \\
& x_j \in \{0,1\}, y_{ij} \in \{0,1\} & j = 1,\dots,n;\ i = 1,\dots,m
\end{aligned}$$

- $x_j = 1$ iff facility j is opened, $y_{ij} = 1$ iff customer i is served from j
- size: $O(nm)$ in model 1

Answer: aggregated model

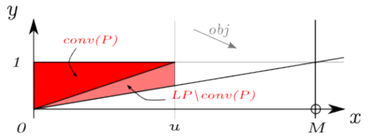
Uncapacitated Facility Location Problem

$$\begin{aligned}
\min\ & \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n}\sum_{i=1}^{m} d_{ij} y_{ij} & \\
\text{s.t. } & \sum_{j=1}^{n} y_{ij} = 1 & i = 1,\dots,m \\
& \sum_{i=1}^{m} y_{ij} \le m x_j & j = 1,\dots,n \\
& x_j \in \{0,1\}, y_{ij} \in \{0,1\} & j = 1,\dots,n;\ i = 1,\dots,m
\end{aligned}$$

- size: $O(nm)$ in model 1 and $O(n+m)$ in model 2
- but model 1 is stronger than model 2 (*prove it*):
- ex: $m = n = 40$, standard LP-based B&B solving times: 2 seconds with model 1, >14h with model 2

Avoid Big M's

- $P = \{(x,y) \in \mathbb{R}_+ \times \mathbb{Z}_+ \mid x \leq My, x \leq u, y \in \{0,1\}\}$
- $\bar{P} = \{(x,y) \in \mathbb{R}_+ \times \mathbb{R}_+ \mid x \leq My, x \leq u, y \leq 1\}$
- $conv(P) = \{(x,y) \in \mathbb{R}_+ \times \mathbb{R}_+ \mid x \leq uy, y \leq 1\}$



tightening big M

- if $M = u$ then $\bar{P} = conv(P)$
- smaller is the big M, stronger is the formulation

Quizz: adding constraints

Perfect Matching Problem

Given a set of n people that need to be paired in teams of two, a cost c_{ij} for each team composed of 2 persons i and j, the problem is to minimize the cost pairing over all teams.

- the problem can be represented by the complete graph $\mathbb{K}_n$
- nodes $i \in N$ represent people, edges $e \in E$ represent pairings

Answer

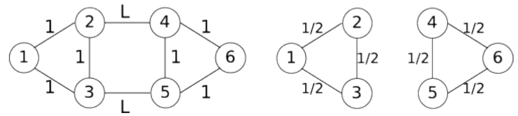
Perfect Matching Problem

$$\begin{aligned} \min \ & \sum_{e \in E} c_e x_e & \\ \text{s.t.} \ & \sum_{e \in E \mid i \in e} x_e = 1 & i = 1, \dots, n \\ & x_e \in \{0,1\} & e \in E \end{aligned}$$

- $x_e = 1$ iff the endpoints of e are matched
- *note that the matrix is usually not TU*

Valid inequality for matching

- this formulation can be extremely weak, example:



- the optimum of the instance above-left has value $L + 2$
- the LP relaxation optimum (above-right) has value 3
- we can add inequality:

$$x_{24} + x_{35} \geq 1$$

- it is a valid inequality: satisfied by every feasible solution
- it is a cutting-plane: satisfied by every feasible solution, but violated by the current fractional solution

Odd-set inequalities

- The cutset of a subset S of nodes:

$$\delta(S) = \{\{i,j\} \in E \mid i \in S, j \notin S\}$$

- if $|S|$ is odd then every perfect matching contains at least one edge from $\delta(S)$
- hence each odd cutset induces a possible valid inequality:

$$\sum_{e \in \delta(S)} x_e \geq 1$$

- Edmonds' theorem tells that the model with all odd set inequalities is integral
- but its size is exponential
- the valid inequalities can be generated on the fly: cutting-planes algorithm

Quizz: changing variables (1)

Uncapacitated Lot-Sizing Problem (ULS)

The problem is to decide a production plan for a n-period horizon for a single product. The basic model can be viewed as having data:
f_t the fixed cost of producing in period t;
p_t the unit production cost in period t;
h_t the unit storage cost in period t;
d_t the demand in period t.
The production plan must satisfy the demand with a minimum cost.

Natural formulation of ULS

Uncapacitated Lot-Sizing Problem

$$\begin{aligned}
\min \; & \sum_{t=1}^{n} f_t y_t + \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t & \\
\text{s.t. } & s_{t-1} + x_t = d_t + s_t & t = 1, \dots, n \\
& x_t \leq M y_t & t = 1, \dots, n \\
& y_t \in \{0,1\} & t = 1, \dots, n \\
& s_t, x_t \geq 0 & t = 1, \dots, n \\
& s_0 = 0 &
\end{aligned}$$

- x_t is the amount produced in period t
- s_t the stock at the end of period t
- $y_t = 1$ iff production occurs in period t.
- $M = \sum_{t=1}^{n} d_t$ is a weak upper bound on x_t

Extended formulation of ULS

- let z_{it} model the amount produced in period i designated to satisfy demand in period $t \geq i$
- the production lot x_i is splitted into smaller pieces:

$$x_i = \sum_{t=i}^{n} z_{it}$$

- with tighter dependency constraints:

$$z_{it} \leq d_t y_i \qquad i = 1, \dots, n; t = i, \dots, n$$

- demands can now be modeled as:

$$\sum_{i=1}^{t} z_{it} = d_t \qquad t = 1, \dots, n$$

Extended formulation of ULS

Uncapacitated Lot-Sizing Problem

$$\min \sum_{t=1}^{n} f_t y_t + \sum_{i=1}^{n}\sum_{t=i}^{n} p_i z_{it} + \sum_{i=1}^{n}\sum_{t=i+1}^{n}\sum_{j=i}^{t-1} h_j z_{it}$$

$$\begin{aligned}
\text{s.t. } & \sum_{i=1}^{t} z_{it} = d_t && t = 1,\ldots,n \\
& z_{it} \le d_t y_i && i = 1,\ldots,n; t = i,\ldots,n \\
& y_t \in \{0,1\} && t = 1,\ldots,n \\
& z_{it} \ge 0 && i = 1,\ldots,n; t = i,\ldots,n
\end{aligned}$$

- z_{it} is the amount produced in period i to satisfy the demand in period t
- $y_t = 1$ iff production occurs in period t
- we can prove that this formulation is integral

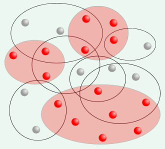
Selecting from a set

Set-{Partitioning, Covering, Packing}

Given a ground set N of items and a set $\mathcal{J} \subseteq 2^N$ of subsets of N, select subsets in $\mathcal{J}$ such that each item i appears in

$\begin{cases} \text{exactly one} & \text{(partitioning)} \\ \text{at least one} & \text{(covering)} \\ \text{at most one} & \text{(packing)} \end{cases}$ selected subset.

Define $a_{ij} = 1$ iff item i belongs to subset $j \in \mathcal{J}$.



Selecting from a set

Set Covering Problem

$$\min \sum_{s \in \mathcal{S}} c_s x_s \qquad (39)$$

$$\text{s.t. } \sum_{s \in \mathcal{S}} a_{is} x_s \ge 1 \qquad i = 1,\ldots,n \qquad (40)$$

$$x_s \in \{0,1\} \qquad s \in \mathcal{S} \qquad (41)$$

- $a_{is} = 1$ iff item i belongs to s (constant)
- $x_s = 1$ iff set s is selected (variable)

Quizz: changing variables (2)

Bin-Packing Problem (1D-BP)

m items have to be packed each into one of the n available containers. Each item j has a weight w_j and all containers have the same capacity c. The problem is to find a packing in a minimum number of containers so as the total weights of the items in each container does not exceed the capacity.

Compact formulation of 1D-BP

Bin-Packing Problem

$$\begin{aligned}
\min\ & \sum_{i=1}^{n} y_i \\
\text{s.t.}\ & \sum_{j=1}^{m} w_j x_{ij} \leq c y_i && i = 1,\dots,n \\
& \sum_{i=1}^{n} x_{ij} = 1 && j = 1,\dots,m \\
& x_{ij} \in \{0,1\} && i = 1,\dots,n;\ j = 1,\dots,m \\
& y_i \in \{0,1\} && i = 1,\dots,n
\end{aligned}$$

- $y_i = 1$ if container i is used
- x_{ij} if item j is packed in container i

another formulation of 1D-BP

- Consider N the set of items and $\mathcal{S} \subseteq 2^N$ defined by:

$$s \in \mathcal{S} \iff s \neq \emptyset \wedge \sum_{j \in s} w_j \leq c$$

- 1D-BP is to select subsets in $\mathcal{S}$ such that each item j appears in exactly one subset.

Set-partitioning formulation

$$\begin{aligned}
\min\ & \sum_{s \in \mathcal{S}} x_s \\
\text{s.t.}\ & \sum_{s \in \mathcal{S}} a_{js} x_s = 1 && j = 1,\dots,n \\
& x_s \in \{0,1\} && s \in \mathcal{S}
\end{aligned}$$

set-partitioning formulation of 1D-BP

- this model is stronger than the compact model
- it requires to pre-compute $\mathcal{S}$: all the patterns filling a container
- this model may contain an exponential number of variables
- the variables can be generated on the fly: column-generation algorithm

Outline

contrast with LP

- in LP, the same problem can also have multiple formulations
- in LP, conventional wisdom is that bigger formulations take longer to solve
- in IP, the size of the formulation does not determine how difficult the IP is (ex: 1D-BP, ULS)
- decomposition methods allow to handle exponential-size models: progressive cutting-planes/column generation
- the main criterium for a good IP model is the quality of the LP relaxation