DM872

Math Optimization at Work

More on Modeling

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Integer Linear Programming

Linear Objective
Linear Constraints
but! integer variables

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ A\boldsymbol{x} &\leq \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned}$$

Linear Programming (LP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ A\boldsymbol{x} &\leq \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \\ \boldsymbol{x} &\ \text{ integer} \end{aligned}$$

Integer Linear Programming (ILP)

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} \\ A\boldsymbol{x} &\leq \boldsymbol{b} \\ \boldsymbol{x} &\in \{0,1\}^n \end{aligned}$$

Binary Integer Program (BIP)
0/1 Integer Programming

$$\begin{aligned} \max\ & \boldsymbol{c}^T\boldsymbol{x} + \boldsymbol{h}^T\boldsymbol{y} \\ A\boldsymbol{x} + G\boldsymbol{y} &\leq \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \\ \boldsymbol{y} &\geq \boldsymbol{0} \\ \boldsymbol{y} &\ \text{ integer} \end{aligned}$$

Mixed Integer Linear Programming (MILP)

$$\begin{aligned} \max\ & f(\boldsymbol{x}) \\ g(\boldsymbol{x}) &\leq \boldsymbol{b} \\ \boldsymbol{x} &\geq \boldsymbol{0} \end{aligned}$$

Non-linear Programming (NLP)

Outline

1. Modeling with IP, BIP, MIP

Modeling with IP, BIP, MIP

Iterate:

1. define parameters
2. define variables
3. use variables to express objective function
4. use variables to express constraints

Examples:

- problems with discrete input/output (knapsack, factory planning)
- problems with logical conditions
- combinatorial problems (sequencing, allocation, transport, assignment, partitioning)
- network problems

Variables

continuous quantities	$\in \mathbb{R}^n$
discrete quantities	$\in \mathbb{Z}^n$
decision variables	$\in \mathbb{B}^n$
indicator/auxiliary variables (for logical conditions)	$\in \mathbb{B}^n$
special ordered sets	$\in \mathbb{B}^n$
incidence vector of S	$\in \mathbb{B}^n$

Mathematical Formalism

Assignment

$$\max_{\sigma} \left\{ \sum_i c_{i,\sigma(i)} \mid \sigma : I \to J \right\}$$

Traveling Salesman Problem (TSP)

$$\min_{\pi} \left\{ \sum_i c_{i,\pi(i)} \mid \pi : \{1..n\} \to \{1..n\} \text{ and } \pi \text{ is a circuit} \right\}$$

Mathematical description
but not in linear form

Combinatorial Optimization Problem (COP)

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

Modeling in Linear Terms

Objective function and/or constraints do not appear to be linear?

- Absolute values
- Minimize the largest function value
- Maximize the smallest function value
- Constraints include variable division
- Constraints are either/or
- A variable must take one of several candidate values

Modeling: Absolute Values

$$\min \sum_{i=1}^{n} |f_i(\boldsymbol{x})|$$

$$\boldsymbol{x} \in \mathbb{R}^q$$

$$\begin{array}{lrll} \min \sum_{i=1}^{n} z_i & & & \\ \text{s.t.} & z_i & \geq f_i(\boldsymbol{x}) & i = 1..n \\ & z_i & \geq -f_i(\boldsymbol{x}) & i = 1..n \\ & z_i & \in \mathbb{R} & i = 1..n \\ & \boldsymbol{x} & \in \mathbb{R}^q & \end{array}$$

n additional variables and $2n$ additional constraints.

$$\begin{array}{lrll} \min \sum_{i=1}^{n}(z_i^+ + z_i^-) & & & \\ \text{s.t.} & f_i(\boldsymbol{x}) & = z_i^+ - z_i^- & i = 1..n \\ & z_i^+, z_i^- & \geq 0 & i = 1..n \\ & \boldsymbol{x} & \in \mathbb{R}^q & \end{array}$$

$2n$ additional variables and n additional constraints.

Modeling: Minimax

Minimize the largest of a number of function values:

$$\min \quad \max\{f_1(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\}$$

- Introduce an auxiliary variable z:

$$\begin{aligned} \min \quad & z \\ \text{s. t. } & f_1(\boldsymbol{x}) \leq z \\ & f_2(\boldsymbol{x}) \leq z \end{aligned}$$

Modeling: Divisions

Constraints include variable division:

- Constraint of the form

$$\frac{a_1x + a_2y + a_3z}{d_1x + d_2y + d_3z} \leq b$$

  - Rearrange:

$$a_1x + a_2y + a_3z \leq b(d_1x + d_2y + d_3z)$$

    which gives:

$$(a_1 - bd_1)x + (a_2 - bd_2)y + (a_3 - bd_3)z \leq 0$$

Later we will see linear-fractional programming transformed into equivalent linear programs (Charnes and Cooper)

Modeling: "Either/Or Constraints"

In conventional mathematical models, the solution must satisfy all constraints. Suppose that your constraints are "either/or":

$$a_1x_1 + a_2x_2 \leq b_1 \qquad \text{or}$$
$$d_1x_1 + d_2x_2 \leq b_2$$

Introduce new variable $y \in \{0, 1\}$ and a large number M:

$$a_1x_1 + a_2x_2 \leq b_1 + My \qquad \text{if } y = 0 \text{ then this is active}$$
$$d_1x_1 + d_2x_2 \leq b_2 + M(1 - y) \qquad \text{if } y = 1 \text{ then this is active}$$

Modeling: “Either/Or Constraints”

Binary integer programming allows to model alternative choices:

- Eg: 2 feasible regions, ie, disjunctive constraints, not possible in LP.
  introduce y auxiliary binary variable and M, a big number:

$$
\begin{aligned}
Ax &\leq b + My && \text{if } y = 0 \text{ then this is active} \\
A'x &\leq b' + M(1 - y) && \text{if } y = 1 \text{ then this is active}
\end{aligned}
$$

Modeling: "Either/Or Constraints"

Generally:

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1m}x_m &\leq d_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2m}x_m &\leq d_2 \\
&\vdots \\
a_{m1}x_1 + a_{N2}x_2 + a_{N3}x_3 + \ldots + a_{Nm}x_m &\leq d_N
\end{aligned}
$$

Exactly K of the N constraints must be satisfied.
Introduce binary variables $y_1, y_2, \ldots, y_N$ and a large number M

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1m}x_m &\leq d_1 + My_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2m}x_m &\leq d_2 + My_2 \\
&\vdots \\
a_{m1}x_1 + a_{N2}x_2 + a_{N3}x_3 + \ldots + a_{Nm}x_m &\leq d_N + My_N
\end{aligned}
$$

$$y_1 + y_2 + \ldots y_N = N - K$$

K of the y-variables are 0, so K constraints must be satisfied

Modeling: "Either/Or Constraints"

At least $K \leq N$ of $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$, $i = 1, \ldots, N$ must be satisfied

introduce y_i, $i = 1, ..., N$ auxiliary binary variables

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i + My_i, \qquad i = 1..N$$

$$\sum_{i} y_i \leq N - K$$

Modeling: "Possible Constraints Values"

A constraint must take on one of N given values:

$$
\begin{gathered}
a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m = d_1 \text{ or} \\
a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m = d_2 \text{ or} \\
\vdots \\
a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m = d_N
\end{gathered}
$$

Introduce binary variables $y_1, y_2, \ldots, y_N$:

$$a_1x_1 + a_2x_2 + a_3x_3 + \ldots + a_mx_m = d_1y_1 + d_2y_2 + \ldots d_Ny_N$$

$$y_1 + y_2 + \ldots y_N = 1$$

Logical Conditions

x binary
y integer
z continuous

Linking constraints $\quad z \in \mathbb{R}, x \in \mathbb{B}$

$$\text{if } z = 0 \text{ then } x = 0, \text{ if } z > 0 \text{ then } x = 1 \quad \rightsquigarrow \quad z - Mx \leq 0$$
$$x = 1 \Longrightarrow z \geq m \quad \rightsquigarrow \quad z - mx \geq 0$$

Logical conditions and $0 - 1$ variables

$$\begin{aligned}
X_1 \vee X_2 &\iff x_1 + x_2 \geq 1 \\
X_1 \wedge X_2 &\iff x_1 = 1, x_2 = 1 \\
\neg X_1 &\iff x_1 = 0 \text{ or } (1 - x_1 = 1) \\
X_1 \rightarrow X_2 &\iff x_1 - x_2 \leq 0 \\
X_1 \leftrightarrow X_2 &\iff x_1 - x_2 = 0
\end{aligned}$$

Examples

- $(X_A \vee X_B) \rightarrow (X_C \vee X_D \vee X_E)$

$$\begin{array}{ll} x_A + x_B \geq 1 & x_C + x_D + x_E \geq 1 \\ x_A + x_B \geq 1 \Longrightarrow x = 1 & x = 1 \Longrightarrow x_C + x_D + x_E \geq 1 \\ x_A + x_B - 2x \leq 0 & x_C + x_D + x_E \geq x \end{array}$$

- Disjunctive constraints in scheduling (see few slides back)
  job i must either preceed or follow job j

- Constraint: $x_1 x_2 = 0$

  1) replace $x_1 x_2$ by x_3
  2) $x_3 = 1 \iff x_1 = 1, x_2 = 1$

$$\begin{aligned} -x_1 \quad\quad + x_3 &\leq 0 \\ - x_2 + x_3 &\leq 0 \\ x_1 + x_2 - x_3 &\leq 1 \end{aligned}$$

- $z \cdot x, \qquad z \in \mathbb{R}, x \in \mathbb{B}$

  1) replace zx by z_1
  2) impose:

$$
\begin{aligned}
x = 0 &\iff z_1 = 0 \\
x = 1 &\iff z_1 = z
\end{aligned}
$$

$$
\begin{aligned}
z_1 - Mx &\leq 0 \\
-z + z_1 &\leq 0 \\
z - z_1 + Mx &\leq M
\end{aligned}
$$

- Special ordered sets of type $1/2$ (for continuous or integer vars):
  SOS1: set of vars within which exactly one must be non-zero
  SOS2: set of vars within which at most two can be non-zero. The two variables must be adjacent in the ordering

- separable programming and piecewise linear functions (next 5 slides)

Separable Programming

- Separable functions: sum of functions of single variables:

$$x_1^2 + 2x_2 + e^{x^3} \quad \text{YES}$$

$$x_1 x_2 + \frac{x_2}{x_1 + 1} + x_3 \quad \text{NO}$$

  (actually, some non-separable can also be made separable:
  1. $x_1 x_2$ by y
  2. relate y to x_1 and x_2 by:

$$\log y = \log x_1 + \log x_2$$

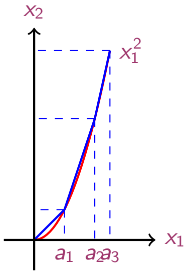
  needs care if x_1 and x_2 close to zero.)

- non-linear separable functions can be approximated by piecewise linear functions (valid for both constraints and objective functions)

Convex Non-linear Functions

- We can model convex non-linear functions by piece-wise linear functions and LP

$$
\begin{array}{rrrrl}
\min & x_1^2 & -\ 4x_1 & -\ 2x_2 & \\
 & x_1 & +\ x_2 & \leq 4 & \\
 & 2x_1 & +\ x_2 & \leq 5 & \\
 & -x_1 & +\ 4x_2 & \geq 2 & \\
 & & x_1, x_2 & \geq 0 &
\end{array}
$$



- LP Formulation

$$
\begin{aligned}
&x = \lambda_0 a_0 + \lambda_1 a_1 + \lambda_2 a_2 + \lambda_3 a_3\\
&y = \lambda_0 f(a_0) + \lambda_1 f(a_1) + \lambda_2 f(a_2) + \lambda_3 f(a_3)\\
&\textstyle\sum_{i=0}^{3} \lambda_i = 1\\
&\lambda_i \geq 0 \qquad i = 0, \ldots, 3\\
&\text{at most two adjacent } \lambda_i \text{ can be non zero} \qquad (*)
\end{aligned}
$$

- To model (*) which are SOS2 we would need binary indicator variables and hence BIP as in next slide.

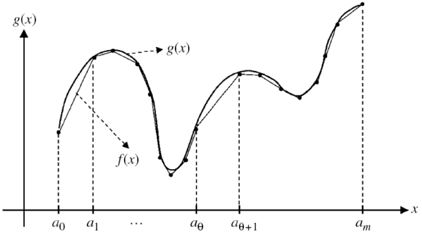
- However since the problem is convex, an optimal solution lies on the borders of the functions and hence we can skip introducing the binary variables and relax (*)

Non-convex Functions

Piece-wise Linear Functions

- non-convex functions require indicator variables and IP formulation

$$g(x) = \sum_j g_j(x) \quad g_j \text{ non linear}$$



$f(x)$ is a piecewise linear function of $g(x)$

- approximated by $f(x)$ piecewise linear in the disjoint intervals $[a_i, b_i]$
- convex hull formulation (convex combination of points)

$$\bigcup_{i\in I} \begin{pmatrix} x = & \lambda_i a_i + \mu_i b_i \\ y = & \lambda_i f(a_i) + \mu_i f(b_i) \\ \lambda_i + \mu_i = 1 & \lambda_i, \mu_i \geq 0 \end{pmatrix}$$

Remember how we modeled disjunctive polyhedra...

(cntd)

- using indicator variables δs we obtain the BIP formulation:

$$
\begin{aligned}
& x = \textstyle\sum_{i \in I} (\lambda_i a_i + \mu_i b_i) \\
& y = \textstyle\sum_{i \in I} (\lambda_i f(a_i) + \mu_i f(b_i)) \\
& \lambda_i + \mu_i = \delta_i \quad \forall i \in I \\
& \textstyle\sum_{i \in I} \delta_i = 1 \\
& \lambda_i, \mu_i \geq 0 \quad \forall i \in I \\
& \delta_i \in \{0, 1\} \quad \forall i \in I
\end{aligned}
$$

the δs are SOS1.

Good/Bad Models

- Number of variables: sometimes it may be advantageous increasing if they are used in search tree.

  $0-1$ var have specialized algorithms for preprocessing and for branch and bound. Hence a large number solved efficiently. Good using.
  Binary expansion:

  $$0 \leq y \leq u$$

  $$y = x_0 + 2x_1 + 4x_2 + 8x_3 + \ldots + 2^r x_r \qquad r = \log_2 u$$

- Making explicit good variables for branching:

  $$\sum_j a_j x_j \leq b$$

  $$\sum_j a_j x_j + u = b$$

  u may be a good variable to branch (u is relaxed in LP but must be integer as well)

- Symmetry breaking:
  Eg machine maintenance (in FPMM) $y_j \in \mathbb{Z}$ vs $x_j \in \mathbb{B}$

- Difficulty of LP models depends on number of constraints:

$$\min \sum_t |a_t z_t - b_t|$$

$$\max \sum_t z'_t$$
$$z'_t \geq a_t z_t - b_1$$
$$z'_t \geq b_t - a_t z_t$$

$$\max \sum_t z_t^+ - z_t^-$$
$$z_t^+ - z_t^- = a_t z_t - b_t$$

more variables but less constraints

- With IP it might be instead better increasing the number of constraints.

- Make big M as small as possible in IP (reduces feasible region possibly fitting it to convex hull).

Practical Tips

- Units of measure: check them!
  all data should be scaled to stay in $0.1 - 10$
  some software does this automatically but better to have control over things

- Write few lines of text describing what the equations express and which are the variables, give examples on the problem modeled.

- Try the model on small simple example that can be checked by hand.

- Be diffident of infeasibility and unboundedness, double check.

- Estimate the potential size.
  If IP problem large and no structure then it might be hard.
  If TUM then solvable with very large size
  If other structure, eg, packing, covering also solvable with large size

- Check the output of the solver and understand what is happening

- If all fails resort to heuristics