CS 524: Introduction to Optimization
Lecture 21 : Constraint logic calculus

Michael Ferris

Computer Sciences Department
University of Wisconsin-Madison

October 23, 2023

Review

- Turn on indicator:
- Suppose $f(x) \leq \mathcal{M}$ for all $x \in X$, and $\delta \in \{0, 1\}$
- Then

$$f(x) > 0 \Rightarrow \delta = 1$$

is implied by the inequality (linear if f is):

$$f(x) \leq \mathcal{M}\delta$$

- Turn on constraint:
- Assume $g(x) \leq \mathcal{M}$ for all $x \in X$
- Then

$$\delta = 1 \Rightarrow g(x) \leq 0$$

is implied by the inequality (linear if g is):

$$g(x) \leq \mathcal{M}(1 - \delta)$$

Light Bulb Calculus

- See Williams, Chapter 9 for further information.
- In the above, we showed rules for indicator variables (often denoted δ) to
  - Imply that a constraint holds
  - Be implied by a constraint holding
- We now consider rules that allow us to apply logical conditions to these indicator variables.
- Having logical conditions is a powerful modeling tool

Constraint Logic Programming

Binary variables δ_i represent statements P_i as indicators:

$$\delta_i = \begin{cases} 1 & \text{if statement } P_i \text{ is true} \\ 0 & \text{if statement } P_i \text{ is false} \end{cases}$$

P_i could be "do project i" or "$f(x) \leq 0$"
δ_i is an indicator variable for whether the statement is true or false.
Standard boolean algebra notation for connectives between statements:

$\vee$	means 'or'
$\wedge$	means 'and'
$\neg$	means 'not'
$\rightarrow$	means 'implies'
$\leftrightarrow$	means 'if and only if'
$\veebar$	means 'exclusive or'

Other connectives such as "nor" or "nand" are also used in the literature.

General Statement	MIP Constraint
at least k out of n are true	$\sum_{i=1}^{n} \delta_i \geq k$
$P_1 \vee P_2 \vee \cdots P_n$	$\sum_{i=1}^{n} \delta_i \geq 1$
exactly k out of n are true	$\sum_{i=1}^{n} \delta_i = k$
at most k out of n are true	$\sum_{i=1}^{n} \delta_i \leq k$
if at least k out of n are true then $n+1$ is true	$\delta_{n+1} \geq \frac{\sum_{i=1}^{n} \delta_i - k + 1}{n-k+1}$
$(P_1 \wedge \cdots P_k) \rightarrow (P_{k+1} \vee \cdots P_n)$	$\sum_{i=1}^{k} (1 - \delta_i) + \sum_{i=k+1}^{n} \delta_i \geq 1$

Fall Into the...

- GAP: Generalized Assignment Problem

- We have a set $I = \{1, 2, \ldots, m\}$ of machines
- and a set $J = \{1, 2, \ldots, n\}$ of jobs that must be performed on the machines.
- Each machine $i \in I$ has a capacity of b_i units of work
- Each job $j \in J$ requires a_{ij} units of work to be completed if it is scheduled on machine i.
- All jobs must be assigned to exactly one machine.
- Suppose there is a fixed cost h_i of assigning any jobs to machine i

GAP Models

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} h_i z_i$$

$$\sum_{j \in J} a_{ij} x_{ij} \leq b_i \qquad \forall i \in I$$

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J (onemachperjob)$$

z_i is 1 if machine i is on

$$x_{ij}, z_i \in \{0, 1\} \qquad \forall i \in I, j \in J$$

Fixed Cost Logic

- Logic 1:

$$\sum_{j \in J} x_{ij} > 0 \Rightarrow z_i = 1 \quad \forall i \in I$$

- Turn on indicator:

$$\sum_{j \in J} x_{ij} \leq \mathcal{M} z_i \quad \forall i \in I$$

  - vub_eq_1: $\mathcal{M} = |J| \quad (\forall i \in I)$
  - vub_eq_2: $\mathcal{M}_i = |\{j : j \text{ may be assigned to } i\}|$

- Logic 2:

$$x_{ij} > 0 \Rightarrow z_i = 1 \quad \forall i \in I, \forall j \in J$$

- Turn on indicator: vub_eq_3

$$x_{ij} \leq z_i \quad \forall i \in I, \ \forall j \in J$$

More Fixed Cost Logic

- Logic 3:

$$\sum_{j \in J} a_{ij} x_{ij} > 0 \Rightarrow z_i = 1 \quad \forall i \in I$$

- Turn on indicator: `vub_eq_4`

$$\sum_{j \in J} a_{ij} x_{ij} \leq \mathcal{M} z_i \quad \forall i \in I$$

  - $\mathcal{M}_i = b_i$, so can replace capacity constraints

---

Which is Best?!?!?

See `gap.gms` - probably too small example to determine (possibly 4 + 2)

An issue

- What about $f(x) \geq 0 \Rightarrow \delta = 1$?
- This requires thought from the modeler and is not always doable
- Idea is to find $\epsilon > 0$ so that any x that satisfies $f(x) + \epsilon > 0$ also satisfies $f(x) \geq 0$ and conversely, and then apply our "Turn on indicator" to $f(x) + \epsilon$
- Example: suppose x_1 and x_2 are binary, then

$$f(x) = x_1 + x_2 - 1 \geq 0$$

is equivalent to

$$x_1 + x_2 > 0, \text{ that is } f(x) + 1 > 0$$

so $\epsilon = 1$ will work here. Several similar examples will be used below.

Some Additional Problems on GAP

1. If you use k or more machines, then you must pay a penalty cost of λ
2. If you operate machine one or two, then you may not operate both machines 3 and 5
3. If you operate both machine 1 and machine 3, then you may use no more than 50% of the capacity of machine 5
4. Each job $j \in J$ has a duration (or length) d_j. Minimize makespan.

GAP 1

- If you use k or more machines, then you must pay a penalty cost of λ

- Need (earlier) "turn on indicator" logic for z_i
- Model $\sum_{i \in I} z_i \geq k \Rightarrow \delta_1 = 1$
- Add $\lambda\delta_1$ to objective function

- Use $f(z) = \sum_{i \in I} z_i - (k-1)$ so $f(z) > 0$ is equivalent to $\sum_{i \in I} z_i \geq k$. Then $f(z) \leq (|I| - k + 1)$, so "Turn on indicator" leads to:

$$\sum_{i \in I} z_i - (k-1) \leq (|I| - k + 1)\delta_1$$

GAP 2

- If you operate machine one or two, then you may not operate both machines 3 and 5

- Split the modeling of $z_1 + z_2 \geq 1 \Rightarrow z_3 + z_5 \leq 1$ in two:
  - $z_1 + z_2 \geq 1 \Rightarrow \delta_2 = 1$
  - $\delta_2 = 1 \Rightarrow z_3 + z_5 \leq 1$

Gap 2, cont.

- First implication is equivalent (by binary z_i) to $z_1 + z_2 > 0 \Rightarrow \delta_2 = 1$ so can use "Turn on indicator" with $\mathcal{M} = 2$:

$$z_1 + z_2 \leq 2\delta_2$$

(Note: could also model (better) as $\delta_2 \geq z_1, \delta_2 \geq z_2$)

- Second implication is the "Turn on constraint" (with $g(z) = z_3 + z_5 - 1$ bounded by $\mathcal{M} = 1$ resulting in:

$$z_3 + z_5 - 1 \leq (1 - \delta_2).$$

GAP 3

- If you operate both machine 1 and machine 3, then you may use no more than 50% of the capacity of machine 5

- Model $z_1 + z_3 \geq 2 \Rightarrow \sum_{j \in J} a_{5j} x_{5j} \leq 0.5 b_5$ as
  - $z_1 + z_3 > 1 \Rightarrow \delta_3 = 1$
  - $\delta_3 = 1 \Rightarrow \sum_{j \in J} a_{5j} x_{5j} \leq 0.5 b_5$

GAP 3, Cont.

- Bound (on f) for $z_1 + z_3 > 1 \Rightarrow \delta_3 = 1$ has $\mathcal{M} = 1$ leading to:

$$z_1 + z_3 - 1 \leq \delta_3$$

- Bound (on g) for $\delta_3 = 1 \Rightarrow \sum_{j \in J} a_{5j} x_{5j} \leq 0.5 b_5$ has $\mathcal{M} = b_5 - 0.5 b_5 = 0.5 b_5$ leading to:

$$\sum_{j \in J} a_{5j} x_{5j} - 0.5 b_5 \leq 0.5 b_5 (1 - \delta_3)$$

or equivalently:

$$\sum_{j \in J} a_{5j} x_{5j} + 0.5 b_5 \delta_3 \leq b_5$$

GAP 4

- Each job $j \in J$ has a duration (or length) d_j. Minimize makespan.

- MINIMAX again. (No integer variables needed)
- Let $t \geq \max_{i \in I} \{\sum_{j \in J} d_j x_{ij}\}$.

$$\min t$$
$$t \geq \sum_{j \in J} d_j x_{ij} \quad \forall i \in I$$