COMP9334: Capacity Planning of Computer Systems and Networks

Optimisation (5):
Power of binary variables

Integer Programming - What have you seen?

- A recurrent theme is to use integer programming to make *binary decisions*
- Examples of binary decisions
  - Week 8A: Grid computing problem
    - Choose a particular grid computing company or not
  - Week 8B: Routing of flows
    - Should the flow be routed on a link or not?
  - Week 9A: Placement problem
    - Should a location be chosen or not?

This week’s lecture: Power of binary variables

- Not only for making yes-or-no type of decisions, binary variables can be used to capture many other requirements
  - Restricted range of values
  - Either-or constraints
  - If-then constraints
  - Piecewise linear functions

Restricted range of values

- Some variables can only take certain values
  - E.g. network links can only be of capacity 155 Mbps, 466 Mbps, 622 Mbps, etc
- If decision variable x can only take values from $\{a_1, a_2, ..., a_m\}$, this can be modeled by using an additional set of binary decision variables

$$y_i = \begin{cases} 1 & \text{if } a_i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Restricted range of values

- Then, the above requirement can be captured by

$$\begin{aligned} x &= \sum_{i=1}^{m} a_i y_i \\ \sum_{i=1}^{m} y_i &= 1 \\ y_i &\in \{0, 1\} \end{aligned}$$

- E.g. if $a_1 = 155, a_2 = 466, a_3 = 622$, we have
  - $y_1 = 1 \;\Rightarrow\; y_2 = y_3 = 0 \;\Rightarrow\; x = 155$
  - $y_2 = 1 \;\Rightarrow\; y_1 = y_3 = 0 \;\Rightarrow\; x = 466$
  - $y_3 = 1 \;\Rightarrow\; y_1 = y_2 = 0 \;\Rightarrow\; x = 622$

Either-or constraints

- A Cloud computing service provider offers 3 different packages with different speed and cost for each package. You can buy any cycles from any package but the deal requires that
  - # cycles from Package 1 $+$ # cycles from Package 2 $\geq$ 10000, or,
  - # cycles from Package 2 $+$ # cycles from Package 3 $\geq$ 50000
    - At least one of these two inequalities must hold,
      but not necessarily both
- Let $w_i =$ number of cycles to be bought from Package i

Either-or constraints (cont.)

- The above requirement can be captured by using an additional binary decision variable p

$$\begin{aligned} w_1 + w_2 &\geq 10000p \\ w_2 + w_3 &\geq 50000(1-p) \\ p &\in \{0,1\} \\ w_i &\geq 0, \quad i = 1,2,3 \end{aligned}$$

Case 1: $p = 0$, we have

$$\begin{aligned} w_1 + w_2 &\geq 0 \leftarrow \text{Trivially satisfied} \\ w_2 + w_3 &\geq 50000 \\ w_i &\geq 0, \quad i = 1,2,3 \end{aligned}$$

Case 2: $p = 1$, we have

$$\begin{aligned} w_1 + w_2 &\geq 10000 \\ w_2 + w_3 &\geq 0 \leftarrow \text{Trivially satisfied} \\ w_i &\geq 0, \quad i = 1,2,3 \end{aligned}$$

Either-or constraints (cont.)

- In general, if one of the following two constraints must be satisfied

$$\begin{aligned}\sum_{i=1}^{n} a_{1,i} x_i &\geq b_1 \\ \sum_{i=1}^{n} a_{2,i} x_i &\geq b_2\end{aligned}$$

where $a_{j,i}$ are given parameters, $x_i (\geq 0)$ are decision variables, b_j's are constants, then the either-or constraints can be modelled by

$$\begin{aligned}\sum_{i=1}^{n} a_{1,i} x_i &\geq b_1 p \\ \sum_{i=1}^{n} a_{2,i} x_i &\geq b_2 (1-p) \\ p &\in \{0, 1\}\end{aligned}$$

If-then constraints

- We may want to impose if-then constraints, e.g.

$$\text{if } x_1 + x_2 > 1, \text{ then } y \geq 4$$

where x_1, x_2 are binary variables, and $0 \leq y \leq 10$

- The above if-then constraint can be captured by using an additional binary decision variable p

$$\begin{aligned} x_1 + x_2 - 1 &\leq 1 - p \\ -y + 4 &\leq 4p \\ p &\in \{0, 1\} \end{aligned}$$

If-then constraints (cont.)

- To understand how this works, consider the two cases:
  - Case 1: If $x_1 + x_2 > 1$ holds
    - Since $x_1 + x_2 > 1$, $x_1 + x_2 - 1 > 0$
    - Since p can only be 1 or 0, the inequality constraint $x_1 + x_2 - 1 \leq 1 - p$ forces p to be 0
    - Since $p = 0$, from the inequality constraint $-y + 4 \leq 4p$, we have $y \geq 4$ which is the condition that we want to impose when $x_1 + x_2 > 1$ holds
  - Case 2: If $x_1 + x_2 > 1$ does not hold
    - In this case, since $x_1 + x_2 - 1 \leq 0$, p can be either 0 or 1
    - If $p = 0$, the inequality constraint $-y + 4 \leq 4p$ becomes $y \geq 4$
    - If $p = 1$, the inequality constraint $-y + 4 \leq 4p$ becomes $y \geq 0$
    - Thus, p can be chosen such that there is no restriction on the value of y

If-then constraints (cont.)

- In general, the if-then constraint

$$\text{if } f(x_1, x_2, \ldots, x_n) > 0, \text{ then } g(x_1, x_2, \ldots, x_n) \geq 0$$

can be modeled by

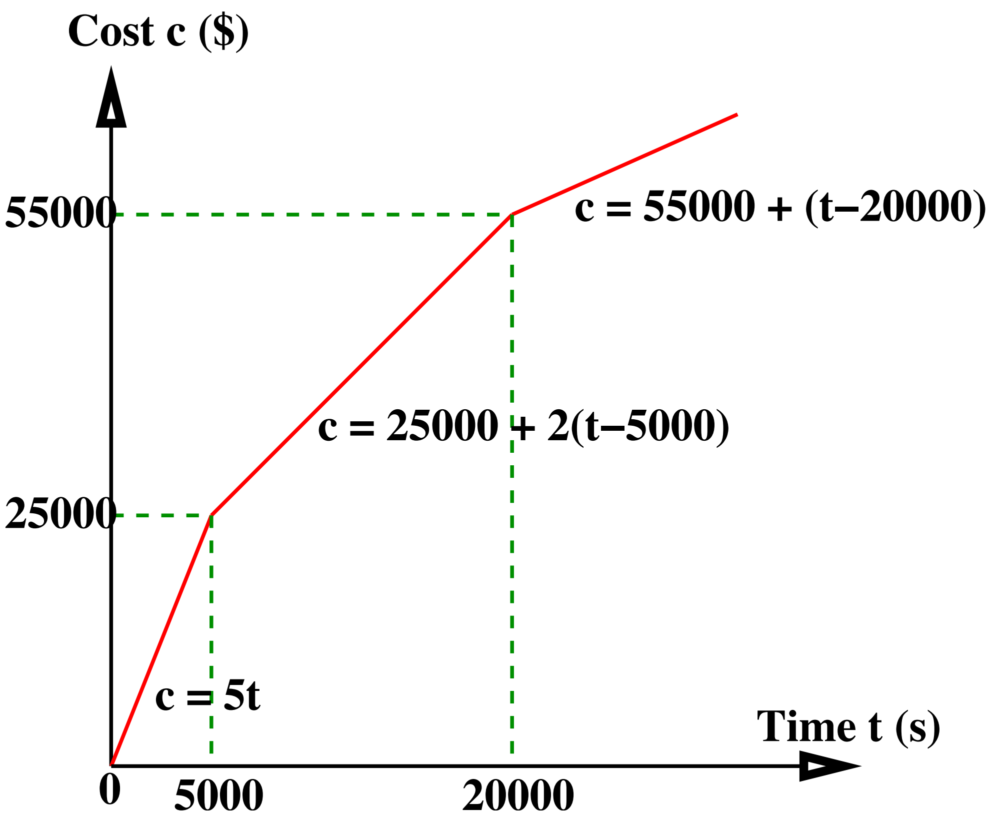
$$\begin{aligned} f(x_1, x_2, \ldots, x_n) &\leq M_1(1-p) \\ -g(x_1, x_2, \ldots, x_n) &\leq M_2\, p \end{aligned}$$

where p is a binary variable, M_1 and M_2 are constants chosen large enough such that $f(x_1, x_2, \ldots, x_n) \leq M_1$ and $-g(x_1, x_2, \ldots, x_n) \leq M_2$ hold for all possible choices of $x_1, x_2, \ldots, x_n$

Piecewise linear functions

- We can use binary variables to model piecewise linear functions
- Example: A Cloud computing service provider may use a progressive charging scheme
  - 5 dollars/sec for the first 5,000 sec
  - 2 dollars/sec for the next 15,000 sec
  - 1 dollar/sec thereafter

Piecewise linear functions (cont.)



- Decision variables

$$y_i = \begin{cases} 1 & \text{if segment } i \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

- Segment 1: $0 \le t \le 5000$, cost = $5t$
- Segment 2: $5000 \le t \le 20000$, cost = $2t$ + 15000
- Segment 3: $20000 \le t$, cost = t + 35000

Piecewise linear functions (cont.)

- We have
  - $y_1 = 1 \Rightarrow 0 \leq t \leq 5000$ and cost = $5t$
  - $y_2 = 1 \Rightarrow 5000 \leq t \leq 20000$ and cost = $2t$ + 15000
  - $y_3 = 1 \Rightarrow 20000 \leq t$ and cost = t + 35000
  - $y_1 + y_2 + y_3 = 1$
- We can rewrite these as
  - $0 \leq ty_1 \leq 5000y_1$
  - $5000y_2 \leq ty_2 \leq 20000y_2$
  - $20000y_3 \leq ty_3$
  - cost = $y_1(5t) + y_2(2t + 15000) + y_3(t + 35000)$
  - $y_1 + y_2 + y_3 = 1$
- Problem: non-linear constraints

Piecewise linear functions (cont.)

- Define $t_i = ty_i$ for $i = 1, 2, 3$
  - cost = $5t_1 + 2t_2 + 15000y_2 + t_3 + 35000y_3$
  - $0 \leq t_1 \leq 5000y_1$
  - $5000y_2 \leq t_2 \leq 20000y_2$
  - $20000y_3 \leq t_3 \leq My_3$
  - $y_1 + y_2 + y_3 = 1$
  - $t = t_1 + t_2 + t_3$
- Note
  - t_i is non-zero if the corresponding $y_i = 1$
  - M is a sufficiently large number to enforce
$$y_3 = 0 \Rightarrow t_3 = 0 \quad \text{and} \quad t_3 \geq 20000 \Rightarrow y_3 = 1$$
  - This is a non-standard expression
  - An alternative expression can be found in Winston Chapter 9

Integer programming and optimisation: Summary

- What you have learnt
  - How to formulate integer programming problems
  - How to solve them using AMPL
  - Examples of using integer programming for network design and analysis
- There are a lot more to learn but this will give you a starting point ...

References

- Advanced formulation of integer programming problems
  - Winston, “Operations Research”, Section 9.2
- Network flow problems
  - Ahuja et al, “Network Flows”, Sections 1.2