CO327 Deterministic OR Models (2021-Spring)

Tutorial 2. Consolidation examples

Formulate IP, Knapsack problems, Assignment problems

Instructor: Andersen Ang

Combinatorics and Optimization, U.Waterloo, Canada

msxang@uwaterloo.ca, where $\mathbf{x} = \lfloor \pi \rfloor$

Homepage: angms.science

First draft: December 30, 2020

Last update: May 26, 2021

Integer Program (IP)

- IP in standard form

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ & \mathbf{x} \geq 0 \\ & \mathbf{x} \in \mathbb{N}^n \end{aligned}$$

- Let's see some modeling / formulation examples.

Investment problem: study time investment

- You are a student, you have 168hours (7 days) before the final exam. You have 5 options, study the 3 courses: C1, C2, and C3, eat or sleep. Each action has an effect on your grade, your MP (mental energy) and your HP (physical energy):

Action	C1	C2	C3	eat	sleep
Grade	+5	+7	+4	0	-1
MP	-8	-11	-6	-1	+15
HP	-5	-8	-3	+15	0

Table: Effect of the action per hour.

  Furthermore,
    - You die if either your MP is below 10 or your HP is below 0.
    - You die if you sleep less than 40 hours.
    - You also need at least 20 grade points for each subject to pass.
- Your start with 50 MP and 20 HP, and zero grade.
- Your goal is to decide how to spend your time so as to maximize your grade while passing all the subjects and not die. Formulate this as an integer program.

Solution

- Let x_1, x_2, x_3 be the time (number of hours) spend on C1, C2 and C3, respectively.
- Let x_4 and x_5 be the time (number of hours) spend on eat and sleep, respectively.
- Integer constraint: $x_i \in \mathbb{N} = \{1, \dots\}, i = 1, 2, \dots, 5$
- Lower bound time constraint: $x_i \geq 0, i = 1, 2, \dots, 5$
- Upper bound time constraint: $\sum_i x_i \leq 168$
- MP constraint: $-8x_1 - 11x_2 - 6x_3 - x_4 + 15x_5 + 50 \geq 10$
- HP constraint: $-5x_1 - 8x_2 - 3x_3 + 15x_4 + 20 \geq 0$
- Sleep hour constraint: $x_5 \geq 40$.
- Per-course passing constraint: $5x_1 \geq 20$, $7x_2 \geq 20$, $4x_3 \geq 20$.
- Goal: maximize total grade $5x_1 + 7x_2 + 4x_3$

Difference between IP and BIP

- You have 14000$ and there are 4 investment opportunities.

Investment	1	2	3	4
Deposit	5000	7000	4000	3000
Present value	8000	11000	6000	4000

  Your goal is to decide which investments to place your money so as to maximize the total present value. Formulate it as an IP.

- IP sol.: let $x_1 = 1$ be "amount of investment 1" made, then the IP is

$$\begin{aligned} \max_{x_1,\dots,x_4} \quad & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{s.t.} \quad & 5x_1 + 7x_2 + 4x_3 + 3x_4 \leq 14 \\ & x_i \geq 0, x_i \in \mathbb{N}, i = 1, 2, 3, 4. \end{aligned}$$

  Important note: this LP means you can invest multiple portions of the same investment.

- BIP sol.: if $x_i \in \mathbb{N}$ becomes $x_i \in \{0, 1\}$ constraint, it means multiple investment of the same investment is not allowed.

Capital budgeting: single period and multi-period

- The previous investment problem is called "single-period capital budgeting". A more general problem is multi-period capital budgeting.

Investment	1	2	3	4
Deposit month 1	5000	7000	0	3000
Deposit month 2	5000	7000	4000	4000
Deposit month 3	0	7000	2000	0
Present value	8000	11000	6000	4000

where 0 means "free". It can mean the project finish (or haven't started) in that month.

- In this case, by inspection, we do not have enough money to buy multiple portion of the same investment → better use BIP

$$\begin{array}{rl} \max\limits_{x_1,\ldots,x_4} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{s.t.} & 5x_1 + 7x_2 + \qquad +3x_4 \leq 14 \\ & 5x_1 + 7x_2 + 4x_3 + 4x_4 \leq 14 \\ & \qquad 7x_2 + 2x_3 \qquad\quad \leq 14 \\ & x_i \in \{0,1\}, i = 1,2,3,4. \end{array}$$

Knapsack

- You are planning "what to bring for hiking".

Item	1	2	$\cdots$	m
Value	c_1	c_2	$\cdots$	a_m
Weight	a_1	a_1	$\cdots$	a_m

- You want to maximize the total value of the items you bring, while keeping the total weight under a limit b.

$$\begin{aligned} \max_{x_1,\ldots,x_m} \quad & c_1x_1 + c_2x_2 + \cdots + c_mx_m \\ \text{s.t.} \quad & a_1x_1 + a_2x_2 + \cdots + a_mx_m \le b \\ & x_i \in \{0,1\}, i = 1, 2, \cdots, m. \end{aligned}$$

- BIP: you either bring item i ($x_i = 1$, you can only bring 1) or leave it ($x_i = 0$).
- LIP: you either bring item i ($x_i > 0$, $x_i =$ number of items to bring) or leave it ($x_i = 0$).
- The "standard form" of Knapsack problem: all a_i, b_i, c_i are nonnegative.

Transform IP to Knapsack problem .. (1/3)

$$\begin{aligned}\max_{x_1,\ldots,x_4} \quad & 8x_1 + 11x_2 - 6x_3 - 4x_4 \\ \text{s.t.} \quad & 5x_1 - 7x_2 - 4x_3 + 3x_4 \leq 14 \\ & x_i \in \{0,1\}, i = 1, 2, \cdots, 4.\end{aligned}$$

- For x_3, both c_3 and a_3 are negative. Perform the transformation $x_3 = 1 - y_3$, this gives

$$\begin{aligned}\max_{x_1,x_2,y_3,x_4} \quad & 8x_1 + 11x_2 - 6(1-y_3) - 4x_4 \\ \text{s.t.} \quad & 5x_1 - 7x_2 - 4(1-y_3) + 3x_4 \leq 14 \\ & x_i \in \{0,1\}, i = 1,2,4, y_3 \in \{0,1\}\end{aligned} \Longrightarrow \begin{aligned}\max_{x_1,x_2,y_3,x_4} \quad & 8x_1 + 11x_2 + 6y_3 - 4x_4 - 6 \\ \text{s.t.} \quad & 5x_1 - 7x_2 + 4y_3 + 3x_4 \leq 18 \\ & x_i \in \{0,1\}, i = 1,2,4, y_3 \in \{0,1\}\end{aligned}$$

Transform IP to Knapsack problem .. (2/3)

- Now we have the IP

$$
\begin{aligned}
\max_{x_1,x_2,y_3,x_4} \quad & 8x_1 + 11x_2 + 6y_3 - 4x_4 - 6 \\
\text{s.t.} \quad & 5x_1 - 7x_2 + 4y_3 + 3x_4 \leq 18 \\
& x_i \in \{0,1\}, i = 1,2,4, y_3 \in \{0,1\}
\end{aligned}
$$

- For x_4, it has negative value $c_4 = -4$ while positive weight $a_4 = +3$. Meaning that picking it will only decrease the total value of knapsack, hence $x_4 = 0$ in the optimal solution: this item is being rejected in the planning, and disappear in the IP

$$
\begin{aligned}
\max_{x_1,x_2,y_3} \quad & 8x_1 + 11x_2 + 6y_3 - 6 \\
\text{s.t.} \quad & 5x_1 - 7x_2 + 4y_3 \leq 18 \\
& x_i \in \{0,1\}, i = 1,2, y_3 \in \{0,1\}
\end{aligned}
$$

Transform IP to Knapsack problem .. (3/3)

- Now we have the IP

$$\begin{aligned} \max_{x_1,x_2,y_3} \quad & 8x_1 + 11x_2 + 6y_3 - 6 \\ \text{s.t.} \quad & 5x_1 - 7x_2 + 4y_3 \leq 18 \\ & x_i \in \{0,1\}, i = 1,2, y_3 \in \{0,1\} \end{aligned}$$

- For x_2, it has positive value $c_2 = +11$ while negative weight $a_2 = -7$, so selecting it will "magically" increase the knapsack capacity and gain net value, so it is a must to select it $\rightarrow x_2 = 1$: this item is being accepted in the planning, and disappear in the IP

$$\begin{aligned} \max_{x_1,y_3} \quad & 8x_1 + 11 + 6y_3 - 6 \\ \text{s.t.} \quad & 5x_1 - 7 + 4y_3 \leq 18 \\ & x_1 \in \{0,1\}, y_3 \in \{0,1\} \end{aligned} \implies \begin{aligned} \max_{x_1,y_3} \quad & 8x_1 + 6y_3 + 5 \\ \text{s.t.} \quad & 5x_1 + 4y_3 \leq 25 \\ & x_1 \in \{0,1\}, y_3 \in \{0,1\} \end{aligned}$$

Multidimensional and general integer Knapsack problem

- Multidimensional 0-1 Knapsack problem

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \in \{1,0\}^m \end{aligned} \quad , \quad \mathbf{c} \ge 0, \mathbf{A} \ge 0, \mathbf{b} \ge 0$$

- General multidimensional Knapsack problem

$$\begin{aligned} \max_{\mathbf{x}} \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{A}\mathbf{x} \le \mathbf{b} \\ & \mathbf{x} \in \mathbb{N}^m \end{aligned} \quad , \quad \mathbf{c} \ge 0, \mathbf{A} \ge 0, \mathbf{b} \ge 0$$

- Similar to "all LP can be transformed to canonical form or standard form", all BIP (0-1 IP) can be transformed to Knapsack form.

Assignment problem

- You have 3 machines and you have 3 jobs to finish. The following table the time for each machine to finish a job:

	Job 1	Job 2	Job 3
Machine 1	14	5	8
Machine 2	2	12	6
Machine 3	7	8	3

  Your want to minimize the total time needed to complete the 3 jobs. Formulate this as a IP.

Decision variable is not always just a matrix

	Job 1	Job 2	Job 3
Machine 1	14	5	8
Machine 2	2	12	6
Machine 3	7	8	3

If you let $x_{ij} \in \{0, 1\}$ meaning: "assign machine i to job j?", then the decision variable $\mathbf{x}$ now is a matrix $\mathbf{X}$:

	Job 1	Job 2	Job 3
Machine 1	x_{11}	x_{12}	x_{13}
Machine 2	x_{21}	x_{22}	x_{23}
Machine 3	x_{31}	x_{32}	x_{33}

Since cost $\mathbf{c}$ always has the same dimension as $\mathbf{x}$, so $\mathbf{c}$ is also a matrix now

	Job 1	Job 2	Job 3
Machine 1	c_{11}	c_{12}	c_{13}
Machine 2	c_{21}	c_{22}	c_{23}
Machine 3	c_{31}	c_{32}	c_{33}

The objective function

	Job 1	Job 2	Job 3
Mac1	x_{11}	x_{12}	x_{13}
Mac2	x_{21}	x_{22}	x_{23}
Mac3	x_{31}	x_{32}	x_{33}

	Job 1	Job 2	Job 3
Mac1	c_{11}	c_{12}	c_{13}
Mac2	c_{21}	c_{22}	c_{23}
Mac3	c_{31}	c_{32}	c_{33}

- Scalar summation form

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

which is a linear function.

- By definition of trace, we have a compact matrix notation

$$\mathrm{Tr}\left(\mathbf{C}^\top \mathbf{X}\right)$$

which is also by definition, the inner product between two matrices

$$\langle \mathbf{C}, \mathbf{X} \rangle.$$

For comparison, the inner product between two vectors: $\langle \mathbf{c}, \mathbf{x} \rangle = \mathbf{c}^\top \mathbf{x}$.

The constraint

	J1	J2	J3		
M1	x_{11}	x_{12}	x_{13}	$\leftarrow$	M1 go to which J
M2	x_{21}	x_{22}	x_{23}		
M3	x_{31}	x_{32}	x_{33}		
	$\uparrow$				
	J1 goes to which M				

- Each machine need to go to one job

$$\sum_j x_{ij} = 1, \ i = 1, 2, 3$$

  or each row of $\mathbf{X}$ sum to 1.

- Each job need to go to one M

$$\sum_i x_{ij} = 1, \ j = 1, 2, 3$$

  or each column of $\mathbf{X}$ sum to 1.

The assignment problems

- Scalar form

$$
\begin{aligned}
\min_{x_{ij}} \quad & \sum_{ij} c_{ij} x_{ij} \\
\text{s.t.} \quad & \sum_{i} x_{ij} = 1, j = 1, ..., m \\
& \sum_{j} x_{ij} = 1, i = 1, ..., m \\
& x_{ij} \in \{0, 1\}
\end{aligned}
$$

- Matrix form

$$
\begin{aligned}
\min_{\mathbf{X}} \quad & \langle \mathbf{C}, \mathbf{X} \rangle \\
\text{s.t.} \quad & \mathbf{X}^\top \mathbf{1} = \mathbf{1} \\
& \mathbf{X} \mathbf{1} = \mathbf{1} \\
& \mathbf{X} \in \{0, 1\}^{m \times m}
\end{aligned}
$$

What does the solution looks like

- Given a square matrix $\mathbf{X} \in \mathbb{R}^{m \times m}$, if all its row sum to 1 and all its columns sum to 1, $\mathbf{X}$ is called a *doubly stochastic matrix*.
- Given a square matrix $\mathbf{X} \in \{0, 1\}^{m \times m}$, if all its row sum to 1 and all its columns sum to 1, $\mathbf{X}$ is called an *elementary matrix*.
- Elementary matrices $\subset$ doubly stochastic matrices
- The solution of the assignment problem will be an elementary matrix.
  Proof by contraction.

Summary

- Formulate problem into integer programs / binary integer programs
- Knapsack problem and transform IP to Knapsack problem standard form
- Assignment problem

Next: assignment 2.

End of document