MATH 3172 3.0
Combinatorial Optimization

Workshop 5

Jacques Nel

April 2, 2020

A quick note about revision:

This is the 3^{rd} revision of this assignment.

1 A-4 Cane sugar production

This problem is taken from[1].

1.1 Parameters

Let $W = \{1, \ldots, m\}$ enumerate $m = 11$ wagons or lots, and $S = \{1, \ldots, n\}$ enumerate n time slots. The refinery has $k = 3$ equivalent processing lines. n timeslots are required to process m wagons where n is given by $n = \texttt{ceil}(m/k) = 4$. Each lot $w \in W$ has an associated hourly loss Δ_w and remaining lifespan l_w until total loss. Furthermore, a single lot takes $D = 2$ hours to process on any given production line.

1.2 Decision variable

Let $\boldsymbol{x} = [x_{ws}] \in \{0, 1\}^{m \times n}$ where for $w \in W$ and $s \in S$,

$$x_{ws} = \begin{cases} 1 & \text{lot } w \text{ is processed in slot } s \\ 0 & \text{otherwise} \end{cases}.$$

1.3 Model

We seek to minimize the loss in raw material resulting from fermentation due to delayed processing of a lot. The model is

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) = \sum_{w \in W} \sum_{s \in S} s d \Delta_w x_{ws} && \text{(1a)} \\ \text{subject to} \quad & \sum_{s \in S} x_{ws} = 1, \forall w \in W, && \text{(1b)} \\ & \sum_{w \in W} x_{ws} \leq k, \forall s \in S, && \text{(1c)} \\ & \sum_{s \in S} s x_{ws} \leq l_w / d, \forall w \in W && \text{(1d)} \end{aligned}$$

The objective function eq. (1a) is the total loss in raw material resulting from delayed processing summed over all lots and wagons. All lots must be assigned to exactly one slot as enforced by eq. (1b). Next, eq. (1c) guarantees that at most $k = 3$ lots can be processed in any one timeslot. Finally, eq. (1d) ensures that a lot is processed before its total loss occurs. Observe that total loss of a lot occurs after l_w/d slots.

[1] C. Guéret, C. Prins, M. Sevaux, *Applications of optimization with Xpress-MP.* Paris: Dash Optimization Ltd., 2007. Page 74.

1.4 Results

The optimal solution results in a loss of $f(\boldsymbol{x}^*) = 1620$ kg with the following time slot assignments:

Table 1.1: Optimal time slot allocations for each lot

Slot 1	**Slot 2**	**Slot 3**	**Slot 4**
lot 3	lot 1	lot 10	lot 2
lot 6	lot 5	lot 8	lot 4
lot 7	lot 8		

Note: Column j is generated with $\{w : x_{wj} = 1, \forall w \in \mathrm{W}\}$.

2 A-6 Production of electricity

This problem is taken from[2].

2.1 Parameters

Let $T = \{1, \ldots, n\}$ enumerate $n = 7$ time periods (of varying length) and $P = \{1, \ldots, m\}$ enumerate $m = 4$ generator types. For a time period $t \in T$ let l_t denote the length of time period in hours, and let d_t denote the forecasted power demand given by d_t.

Table 2.1: Length and forecasted demand of time periods

Period t	**Length** l_t	**Demand** d_t
1	6	1.2×10^4
2	3	3.2×10^4
3	3	2.5×10^4
4	2	3.6×10^4
5	4	2.5×10^4
6	4	3.0×10^4
7	2	1.8×10^4

For each generator type $p \in P$, there are a_p units available. Each unit has a minimum base power output θ_p (if it is running) and can scale up to a maximum output denoted ψ_p, but incurring additional operating cost.

Table 2.2: Number of available units and power output capacity

Type	**Num. available** a_p	**Min. output** θ_p	**Max output** ψ_p
1	10	7.5×10^2	1.75×10^3
2	4	1.0×10^3	1.5×10^3
3	8	1.2×10^3	2.0×10^3
4	3	1.8×10^3	3.5×10^3

[2] C. Guéret, C. Prins, M. Sevaux, *Applications of optimization with Xpress-MP*. Paris: Dash Optimization Ltd., 2007. Page 78.

Starting a generator unit of type $p \in P$ incurs a startup cost λ_p. Running the generator incurs a fixed cost per hour μ_p. Additional scalable output, on top of the base output, incurs a hourly cost ν_p that is proportional to the additional output.

Figure 1: Various costs associated with generator types

Type	**Start cost** λ_p	**Run cost** μ_p	**Add. cost** ν_p
1	5000	2250	2.7
2	1600	1800	2.2
3	2400	3750	1.8
4	1200	4800	3.8

2.2 Decision variables

Suppose $p \in P$ and $t \in T$. Let $0 \le x_{pt} \in \mathbb{Z}$ denote the number of generators of type p started in period t, and $0 \le y_{pt} \in \mathbb{Z}$ be the number of generators of type p running in period t. Finally, let $0 \le z_{pt} \in \mathbb{R}$ denote the additional power generated by unit of type p during period t.

To simplify notation, let $\boldsymbol{x} = [x_{pt}] \in \mathbb{Z}^{m \times n}, \boldsymbol{y} = [y_{pt}] \in \mathbb{Z}^{m \times n}$, and $\boldsymbol{z} = [z_{pt}] \in \mathbb{R}^{m \times n}$.

2.3 Model

$$\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}{\text{minimize}} \quad & \sum_{t \in T} \sum_{p \in P} \lambda_p x_{pt} + l_t \left(\mu_p y_{pt} + \nu_p z_{pt} \right) && \text{(2a)} \\
\text{subject to} \quad & x_{p1} \ge y_{p1} - y_{pn}, \forall p \in P, && \text{(2b)} \\
& x_{pt} \ge y_{pt} - y_{p(t-1)}, \forall p \in P, 1 < t \in T, && \text{(2c)} \\
& z_{pt} \le \left(\psi_p - \theta_p \right) y_{pt}, \forall (p,t) \in P \times T, && \text{(2d)} \\
& \sum_{p \in P} \theta_p y_{pt} + z_{pt} \ge d_t, \forall t \in T, && \text{(2e)} \\
& \sum_{p \in P} \psi_p y_{pt} \ge 1.2 d_t, \forall t \in T, && \text{(2f)} \\
& y_{pt} \le a_p, \forall (p,t) \in P \times T, && \text{(2g)} \\
& x_{pt} \ge 0, \forall (p,t) \in P \times T, && \text{(2h)} \\
& y_{pt} \ge 0, \forall (p,t) \in P \times T, && \text{(2i)} \\
& z_{pt} \ge 0, \forall (p,t) \in P \times T && \text{(2j)}
\end{aligned}$$

The cost function eq. (2a) is simply the startup cost, running cost, and additional power cost summed over all decision variables. The number of generators started for $t = 1$ is related to the number of generators running by eq. (2b). This relationship depends on the numbers generators running at the end period $t = n$. The next family of constraints eq. (2c) is similar to the above, but deals with the relationship for $1 < t \le n$.

Additional power output z_{pt} is bounded by the difference between the maximum and the base output, ie. $\psi_p - \theta_p$, as expressed by eq. (2d). Next, eq. (2e) ensures that the total ouput of all generator units meets forecasted demand d_t for all $t \in T$. Furthermore, a 20% safety buffer is required at all times. Equation (2f) ensures that, if demand were to suddenly spike, a minimum of 20% of d_t of additional capacity can instantly be made available, by increasing additional output z_{pt} up its maximum $\psi_p - \theta_p$.

The family of constraints eq. (2g) simply places an upper bound on the number of units running, in a given period, equal to the given available number of units a_p of each type. Finally, eq. (2h), eq. (2i), and eq. (2j) simply enforce the canonical non-negativity of x_{pt}, y_{pt}, and z_{pt} respectively.

2.4 Results

The optimal solution was found with a total operating cost of $f(\boldsymbol{x}^*, \boldsymbol{y}^*, \boldsymbol{z}^*) = \$1,456,810$.

Table 2.3: Optimal power generation schedule for 4 generator types over 7 planning periods

Type		**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	No. used	3	4	4	7	3	3	3
	Tot. output	2250	4600	3000	8600	2250	2600	2250
	Add. output	0	1600	0	3350	0	350	0
2	No. used	4	4	4	4	4	4	4
	Tot. output	5750	6000	4200	6000	4950	6000	5950
	Add. output	1750	2000	200	2000	950	2000	1950
3	No. used	2	8	8	8	8	8	4
	Tot. output	4000	16000	16000	16000	16000	16000	8000
	Add. output	1600	6400	6400	6400	6400	6400	3200
4	No. used	0	3	1	3	1	3	1
	Tot. output	0	5400	1800	5400	1800	5400	1800
	Add. output	0	0	0	0	0	0	0

Note: The above table was generated from the solution values x^*_{pt}, y^*_{pt}, and z^*_{pt} with `a-6_report.py`.

3 C-2 Production of drinking glasses

This problem is taken from[3].

3.1 Parameters

Let $W = \{1, \ldots, n\}$ enumerate $n = 12$ week-long planning periods, and let $P = \{1, \ldots, m\}$ enumerate the $m = 6$ product variants. Each product $p \in P$ has a predicted demand d_{pt} during week $t \in W$ as given in table 3.1.

Table 3.1: Predicted weekly demand for each product variant

Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
V1	20	22	18	35	17	19	23	20	29	30	28	32
V2	17	19	23	20	11	10	12	34	21	23	30	12
V3	18	35	17	10	9	21	23	15	10	0	13	17
V4	31	45	24	38	41	20	19	37	28	12	30	37
V5	23	20	23	15	10	22	18	30	28	7	15	10
V6	22	18	20	19	18	35	0	28	12	30	21	23

Each product variant $p \in P$ has an associated basic production cost λ_p and an inventory storage cost μ_p incurred on product in inventory over given period. Production requires a known amount worker labour time δ_p, machine time π_p, and production area γ_p. In every period there is a limited amount of worker time Δ, available machine time Π, and production area Γ. Lastly, at the start of planning period there exists volume I_p of item p in the inventory, and it is required that there is F_p of item p at the end of the planning period in the inventory. All parameters are given in table 3.2.

[3] C. Guéret, C. Prins, M. Sevaux, *Applications of optimization with Xpress-MP*. Paris: Dash Optimization Ltd., 2007. Page 106.

Table 3.2: Given costs, production resources, and inventory of product variants

	prod. cost	**inv. cost**	**init. stock**	**fin. stock**	**labour**	**mach. time**	**area**
V1	100	25	50	10	3	2	4
V2	80	28	20	10	3	1	5
V3	110	25	0	10	3	4	5
V4	90	27	15	10	2	8	6
V5	200	10	0	10	4	11	4
V6	150	20	10	10	4	9	9

3.2 Decision variables

For a given product $p \in P$ and week $t \in W$ let $0 \leq x_{pt} \in \mathbb{Z}$ denote the production volume. Also let $0 \leq y_{pt} \in \mathbb{Z}$ denote the amount of product stored in the inventory at the end of period t.

To simplify notation let $\boldsymbol{x} = [x_{pt}] \in \mathbb{Z}^{m \times x}$ and $\boldsymbol{y} = [y_{pt}] \in \mathbb{Z}^{m \times x}$.

3.3 Model

$$
\begin{aligned}
\underset{\boldsymbol{x},\boldsymbol{y}}{\text{minimize}} \quad & f(\boldsymbol{x},\boldsymbol{y}) = \sum_{p \in P} \sum_{t \in W} \lambda_p x_{pt} + \mu_p y_{pt} && \text{(3a)} \\
\text{subject to} \quad & y_{p1} = I_p + x_{pt} - d_{pt}, \forall p \in P, && \text{(3b)} \\
& y_{pt} = y_{p(t-1)} + x_{pt} - d_{pt}, \forall p \in P, t \in W, && \text{(3c)} \\
& y_{pn} = F_p, \forall p \in P,, && \text{(3d)} \\
& \sum_{p \in P} \delta_p x_{pt} \leq \Delta, \forall t \in W,, && \text{(3e)} \\
& \sum_{p \in P} \pi_p x_{pt} \leq \Pi, \forall t \in W,, && \text{(3f)} \\
& \sum_{p \in P} \gamma_p x_{pt} \leq \Gamma, \forall t \in W,, && \text{(3g)} \\
& x_{pt} \geq 0 \forall p \in P, t \in W,, && \text{(3h)} \\
& y_{pt} \geq 0 \forall p \in P, t \in W, && \text{(3i)}
\end{aligned}
$$

We seek to minimize total production cost. Equation (3a) is a cost function which simply sums the total production and storage costs over all decision variables.

Equation (3b) and eq. (3c) states that the inventory at time t is equal the previous inventory plus the product minus the demand. Equation (3b) makes special consideration for the initial inventory I_p. Equation (3d) ensures that the final inventory for product p is equal to F_p at the end of the planning period.

Equation (3e), eq. (3f) and eq. (3g) ensures that limited production factors: worker time capacity Δ, machine time capacity Π, and production area Γ constraints are respected. For example, production of product p during period t requires $\pi_p x_{pt}$ machine hours, the total of which shall not exceed Π for the given period.

Lastly eq. (3h) and eq. (3i) are simply the cononical non-negativity constraints on both decision variables x_{pt} and y_{pt}.

3.4 Results

A optimal solution $(\boldsymbol{x}^*, \boldsymbol{t}^*)$ is found with a total production cost of $f(\boldsymbol{x}^*, \boldsymbol{y}^*) = \$186,076$.

Table 3.3: Production and storage quantities for each product type

	Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	Prod.	0	0	11	34	29	7	23	21	29	29	29	41
	Store	30	8	1	0	12	0	0	1	1	0	1	10
2	Prod.	7	21	14	17	11	10	12	34	21	23	30	22
	Store	10	12	3	0	0	0	0	0	0	0	0	10
3	Prod.	18	35	17	11	8	21	23	15	10	0	13	27
	Store	0	0	0	1	0	0	0	0	0	0	0	10
4	Prod.	16	45	24	38	41	20	20	36	29	11	31	46
	Store	0	0	0	0	0	0	1	0	1	0	1	10
5	Prod.	47	16	34	14	23	24	43	0	26	4	0	0
	Store	24	20	31	30	43	45	70	40	38	35	20	10
6	Prod.	14	17	20	18	18	35	1	27	12	49	28	7
	Store	2	1	1	0	0	0	1	0	0	19	26	10

Note: I choose to solve this as an integer programming model. This is the reason why my results differ slightly from the book. The above table simply states the optimal solution values for x^*_{pt} and y^*_{pt} for all $p \in P$ and $t \in W$.

4 D-5 Cutting sheet metal

This problem is taken from[4].

4.1 Parameters

Let $S = \{1, \dots, n\}$ enumerate $n = 4$ different sizes, ie. $\{\mathsf{36x50}, \mathsf{24x36}, \mathsf{20x60}, \mathsf{18x30}\}$. Also, let $P = \{1, \dots, m\}$ enumerate m different cutting patterns. For $s \in S$ and $p \in P$ let c_{sp} denote number of pieces of size s yielded by pattern p. The values of c_{sp} are given by table 4.1.

Table 4.1: Yields of various cutting patterns

Pattern	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
36x50	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
24x36	2	1	0	2	1	0	3	2	1	0	5	4	3	2	1	0
20x60	0	0	0	2	2	2	1	1	1	1	0	0	0	0	0	0
18x30	0	1	3	0	1	3	0	2	3	5	0	1	3	5	6	8

Each cut size $s \in S$ has a given demand $\boldsymbol{d} = (d_s : s \in S)^T = (8, 13, 5, 15)^T$. Finaly, each pattern has equivalent cost $\kappa = 1$, or simply the cost of each sheet of raw material.

4.2 Decision variable

Let $0 \leq x_p \in \mathbb{Z}$ denote the number of times pattern p is used. To simplify notation let $\boldsymbol{x} = (x_p : p \in P)$.

[4] C. Guéret, C. Prins, M. Sevaux, *Applications of optimization with Xpress-MP*. Paris: Dash Optimization Ltd., 2007. Page 134.

4.3 Model

$$\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) = \sum_{p \in P} x_p \kappa && \text{(4a)}\\
\text{subject to} \quad & \sum_{p \in P} c_{sp} x_p \geq d_s, \forall s \in S, && \text{(4b)}\\
& x_p \geq 0, \forall p \in P && \text{(4c)}
\end{aligned}$$

We seek to minimize the total cost which is given by eq. (4a). This is simply the total number of sheets of raw material used. Equation (4b) is the family of demand constraints, which guarantee that demand is met for each size $s \in S$. Finaly eq. (4c) is simply the canonical non-negativity constraint on the decision variable x_p.

4.4 Results

An optimal solution is found which uses 11 sheets of raw material to satisfy demand, with a cost function value of $f(\boldsymbol{x}^*) = 11$. The following quantities of each pattern are used:

`pattern 1` = 3, `pattern 3` = 5, `pattern 4` = 2, `pattern 7` = 1, and all others are unused. These values are simply the optimal non-zero values of the decision variable $x_p^*, \forall p \in$ P.

5 F-1 Flight connections at a hub

This problem is taken from[5].

5.1 Parameters

Let $P = \{1, \ldots, n\}$ enumerate both $n = 6$ incoming flights and then n outgoing flights. For $i \in P$ and $j \in P$, let μ_{ij} denote the number of passengers arriving on flight i from origin i seeking to continue on to destination j given by table 5.1.

Table 5.1: Arriving passengers and destinations

City	1	2	3	4	5	6
1	35	12	16	38	5	2
2	25	8	9	24	6	8
3	12	8	11	27	3	2
4	38	15	14	30	2	9
5	-	9	8	25	10	5
6	-	-	-	14	6	7

5.2 Decision variable

Let $x_{ij} \in \{0, 1\}$ indicate that aircraft from origin $i \in P$ travels to destination $j \in P$ for next flight when $x_{ij} = 1$. To simplify notation let $\boldsymbol{x} = [x_{ij}] \in \{0, 1\}^{n \times n}$.

[5] C. Guéret, C. Prins, M. Sevaux, *Applications of optimization with Xpress-MP*. Paris: Dash Optimization Ltd., 2007. Page 157.

5.3 Model

We seek to minimize the number of passengers requiring to disembark and transfer to another plane for their next flight. In other words, we wish to maximize the number of passengers staying on their arriving aircraft.

$$\begin{aligned}
\underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) = \sum_{i \in P} \sum_{j \in P} \mu_{ij} x_{ij} && \text{(5a)} \\
\text{subject to} \quad & \sum_{j \in P} \mu_{ij} x_{ij} = 1, \forall i \in P, && \text{(5b)} \\
& \sum_{i \in P} \mu_{ij} x_{ij} = 1, \forall j \in P. && \text{(5c)}
\end{aligned}$$

5.4 Results

The optimal solution has a total of $f(\boldsymbol{x}^*) = 112$ passengers remaining on their arrival flights for the remainder of their journeys. The following assignment of aircraft to destination, minimizes passenger inconvenience:

Bordeaux → London 38

Clermon-Ferrand → Bern 8

Marseille → Brussels 11

Nantes → Berlin 38

Nice → Rome 10

Toulouse → Vienna 7

Note: The above mapping is simply constructed by the permutation matrix given by $\boldsymbol{x}^*$.

Refer to `f-1_report.py` for implementation details.