MATH 3172 3.0
Combinatorial Optimization

Workshop 5

Jacques Nel

March 27, 2020

A quick note about revision:

This assignment has been revised from the first version. The following changes have been made.

1. Parameters and variables are presented in a numbered list, instead of in hard to read paragraphs.

2. Parameter and variable names have been changed to multi-letter names to better convey meaning. Single-letter names were initially used before because they require less work to typeset in LaTeX.

3. Model constraints are now accompanied by a short statement to describe their purpose.

4. Parameter tables have been omitted. They do not add any meaning to the presentation of the models. You can refer to the accompanying source code files for this information.

5. Results are presented in a cleaner manner. A short description is given to explain how results are derived from the output of the solver, when it is not trivial.

1 A-4 Cane sugar production

This problem is taken from[1].

1.1 Parameters

1. $m = 11$: number of wagons or lots
2. $k = 3$: number of processing lines
3. $n = \texttt{ceil}(m/k) = 4$: number of time slots
4. $\text{WAGONS} = \{0, \ldots, m-1\}$: enumerates $m = 11$ wagons
5. $\text{SLOTS} = \{0, \ldots, n-1\}$: enumerates $n = \texttt{ceil}(m/k)$ time slots
6. LOSS_w : hourly loss in kg for lot $w \in \text{WAGONS}$
7. LIFE_w : lifespan of lot $w \in \text{WAGONS}$
8. $\text{DUR} = 2$: processing time in hours per lot

1.2 Decision variable

$\text{process}_{ws} \in \{0, 1\}$: binary decision variable; indicates lot $w \in \text{WAGONS}$ is processed in slot $s \in \text{SLOTS}$, when equal to 1.

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

$$\text{minimize} \sum_{w \in \text{WAGONS}} \sum_{s \in \text{SLOTS}} (s+1) \cdot \text{DUR} \cdot \text{LOSS}_w \cdot \text{process}_{ws}$$

Subject to following constraints:

$$\sum_{s \in \text{SLOTS}} \text{process}_{ws} = 1, \forall w \in \text{WAGONS} \qquad \text{(each lot assigned to 1 slot)}$$

$$\sum_{w \in \text{WAGONS}} \text{process}_{ws} \leq k, \forall s \in \text{SLOTS} \qquad \text{(limit lots per slot)}$$

$$\sum_{s \in \text{SLOTS}} (s+1) \cdot \text{process}_{ws} \leq \text{LIFE}_w/\text{DUR}, \forall w \in \text{WAGONS}$$

(lot must be processed before total loss)

[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 1620 kg with the following time slot assignments:

Table 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 $\{w : \text{process}_{wj} = 1, \forall w \in \text{WAGONS}\}$

2 A-6 Production of electricity

This problem is taken from[2].

2.1 Parameters

1. $n = 7$: number of time periods
2. $\text{TIME} = \{0, \dots n-1\}$: enumerates time periods
3. $m = 4$: number of generator types
4. $\text{TYPES} = \{0, \dots m-1\}$: enumerates generator types
5. LEN_t : length of time period $t \in \text{TIME}$
6. DEM_t : forecasted power demand at time period $t \in \text{TIME}$
7. PMIN_p : minimum base power output of generator type $p \in \text{TYPES}$ if it is running
8. PMAX_p : maximum power output of generator type $p \in \text{TYPES}$
9. CSTART_p : startup cost of generator type $p \in \text{TYPES}$
10. CRUN_p : hourly running cost of generator type $p \in \text{TYPES}$
11. CADD_p : additional cost for scalable output on top of base power for type $p \in \text{TYPES}$
12. AVAIL_p : number of available generators of type $p \in \text{TYPES}$

2.2 Decision variables

Let $(p, t) \in \text{TYPES} \times \text{TIME}$,

1. $0 \leq \text{start}_{pt} \in \mathbb{Z}$: number of generators of type p started in period t
2. $0 \leq \text{work}_{pt} \in \mathbb{Z}$: number of generators of type p running in period t
3. $0 \leq \text{padd}_{pt} \in \mathbb{R}$: additional scalable output from generator type p in period t

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

2.3 Model

We seek to minimize the total operating cost.

$$\text{minimize} \sum_{t \in \text{TIME}} \sum_{p \in \text{TYPES}} \text{CSTART}_p \cdot \text{start}_{pt} + \text{LEN}_t \left(\text{CRUN}_p \cdot \text{work}_{pt} + \text{CADD}_p \cdot \text{padd}_{pt} \right)$$

Subject to the following constraints:

$$\text{start}_{p0} \geq \text{work}_{p0} - \text{work}_{p(n-1)}, \forall p \in \text{TYPES} \qquad \text{(relation for started and working generators for first period)}$$

$$\text{start}_{pt} \geq \text{work}_{pt} - \text{work}_{p(t-1)}, \forall p \in \text{TYPES}, 0 < t \in \text{TIMES} \qquad \text{(same as above; for other periods)}$$

$$\text{padd}_{pt} \leq (\text{PMAX}_p - \text{PMIN})\, \text{work}_{pt}, \forall (p,t) \in \text{TYPES} \times \text{TIME} \qquad \text{(limit maximum power output)}$$

$$\sum_{p \in \text{TYPES}} \text{PMIN}_p \cdot \text{work}_{pt} + \text{padd}_{pt} \geq \text{DEM}_t, \forall t \in \text{TIME} \qquad \text{(demand is satisfied)}$$

$$\sum_{p \in \text{TYPES}} \text{PMAX}_p \cdot \text{work}_{pt} \geq 1.2 \cdot \text{DEM}_t, \forall t \in \text{TIME} \qquad \text{(maximum 20\% security buffer)}$$

$$\text{work}_{pt} \leq \text{AVAIL}_p, \forall (p,t) \in \text{TYPES} \times \text{TIME} \qquad \text{(limit available generators)}$$

$$\text{start}_{pt} \geq 0, \text{work}_{pt} \geq 0, \text{padd}_{pt} \geq 0, \forall (p,t) \in \text{TYPES} \times \text{TIME} \qquad \text{(all variables are} > 0)$$

2.4 Results

The optimal solution was found with a total operating cost of $1,456,810.

Table 2: 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 $\text{start}_{pt}, \text{work}_{pt}, \text{padd}_{pt}$ with `a-6_report.py`.

3 C-2 Production of drinking glasses

This problem is taken from[3].

3.1 Parameters

1. $n = 12$: number of planning periods (weeks)
2. $\text{WEEKS} = \{0, \ldots, n-1\}$: enumerates periods
3. $m = 6$: number of product variants
4. $\text{PROD} = \{0, \ldots, m-1\}$: enumerates product variants
5. DEM_{pt} : demand for product p in week t,
6. PCOST_p : production cost for product p,
7. SCOST_p : storage cost for product p,
8. ISTOCK_p : initial stock for product p at start of planning
9. FSTOCK_p : final required stock for product p at end
10. TIMEW_p : worker time required to produce product p
11. TIMEM_p : machine time required to produce product p
12. SPACE_p : production area required for product p
13. $\text{CAPW} = 390$: available worker time capacity per period
14. $\text{CAPM} = 850$: available machine time capacity per period
15. $\text{CAPS} = 1000$: available production area per period

3.2 Decision variables

1. $0 \leq \text{prod}_{pt} \in \mathbb{Z}$: production volume of product p in week t, $p \in \text{PROD}$ and $t \in \text{WEEKS}$
2. $0 \leq \text{store}_{pt} \in \mathbb{Z}$: amount product p stored in week t, $p \in \text{PROD}$ and $t \in \text{WEEKS}$

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

3.3 Model

We seek to minimize total production cost

$$\text{minimize} \sum_{p \in \text{PROD}} \sum_{t \in \text{WEEKS}} \text{PCOST}_p \cdot \text{prod}_{pw} + \text{SCOST}_{pw} \cdot \text{store}_{pw}$$

Subject to the following constraints:

$$\text{store}_{p0} = \text{ISTOCK}_p + \text{prod}_{p0} - \text{DEM}_{p0}, \forall p \in \text{PROD} \qquad \text{(initial inventory)}$$

$$\text{store}_{pt} = \text{store}_{p(t-1)} + \text{prod}_{pt} - \text{DEM}_{pt}, \forall p \in \text{PROD}, 0 < t \in \text{WEEKS} \qquad \text{(storage-production-sale relation)}$$

$$\text{store}_{p(n-1)} \geq \text{FSTOCK}_p, \forall p \in \text{PROD} \qquad \text{(final stock requirement)}$$

$$\sum_{p \in \text{PROD}} \text{TIMEW}_p \cdot \text{prod}_{pt} \leq \text{CAPW}, \forall t \in \text{WEEKS} \qquad \text{(worker time capacity)}$$

$$\sum_{p \in \text{PROD}} \text{TIMEM}_p \cdot \text{prod}_{pt} \leq \text{CAPM}, \forall t \in \text{WEEKS} \qquad \text{(machine time capacity)}$$

$$\sum_{p \in \text{PROD}} \text{SPACE}_p \cdot \text{prod}_{pt} \leq \text{CAPS}, \forall t \in \text{WEEKS} \qquad \text{(production area space capacity)}$$

$$\text{prod}_{pt} \geq 0, \text{store}_{pt} \geq 0, \forall (p,t) \in \text{PROD} \times \text{WEEKS} \qquad \text{(non-negativity of all variables)}$$

3.4 Results

A optimal solution is found with a total production cost of $186,076.

Table 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 prod_{pt} and store_{pt}.

4 D-5 Cutting sheet metal

This problem is taken from[4].

4.1 Parameters

1. $n = 4$: number of sizes
2. $\text{SIZES} = \{0, \ldots, n-1\}$: enumerates sizes in $\{\texttt{36x50}, \texttt{24x36}, \texttt{20x60}, \texttt{18x30}\}$
3. $m = 16$: number of cutting patterns
4. $\text{PATTERNS} = \{0, \ldots, m-1\}$: enumerate cutting patterns
5. DEM_s : given demand for size $s \in \text{SIZES}$
6. $\text{COST} = 1$: cost is equal for all patterns; simply number of raw sheet materials
7. CUT_{sp} : number of pieces of size $s \in \text{SIZES}$ yielded by pattern $p \in \text{PATTERNS}$

4.2 Decision variable

$0 \leq \text{use}_p \in \mathbb{Z}$: number of times pattern $p \in \text{PATTERNS}$ is used

4.3 Model

We seek to minimize the raw material cost; i.e. the number of sheets used to meet demand.

$$\text{minimize} \sum_{p \in \text{PATTERNS}} \text{COST} \cdot \text{use}_p$$

Subject to the following constraints:

$$\sum_{p \in \text{PATTERNS}} \text{CUT}_{sp} \cdot \text{use}_p \geq \text{DM}_s, \forall s \in \text{SIZES} \qquad \text{(demand requirement)}$$

$$\text{use}_p \geq 0, \forall p \in \text{PATTERNS} \qquad \text{(non-negativity of decision variable)}$$

4.4 Results

An optimal solution is found which uses 11 sheets of raw material to satisfy demand, with a objective function value of 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 $\text{use}_p, \forall p \in \text{PATTERNS}$.

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

5 F-1 Flight connections at a hub

This problem is taken from[5].

5.1 Parameters

1. $n = 6$: number of planes; incoming flights is equal to outgoing flights at hub
2. $\text{PLANES} = \{0, \ldots, n-1\}$: enumerates aircraft
3. $0 \leq PASS_{ij} \in \mathbb{Z}$: number of passengers from origin $i \in \text{PLANES}$ traveling through to destination $j \in \text{PLANES}$

5.2 Decision variable

$\text{cont}_{ij} \in \{0, 1\}$: binary decision variable; indicates aircraft from origin $i \in \text{PLANES}$ travels to destination $j \in \text{PLANES}$ for next flight

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.

$$\text{maximize} \sum_{i \in \text{PLANES}} \sum_{j \in \text{PLANES}} \text{PASS}_{ij} \cdot \text{cont}_{ij}$$

Subject to the following constraints:

$$\sum_{j \in \text{PLANES}} \text{cont}_{ij} = 1, \forall i \in \text{PLANES} \qquad \text{(each aircraft has exactly one destination)}$$

$$\sum_{i \in \text{PLANES}} \text{cont}_{ij} = 1, \forall j \in \text{PLANES} \qquad \text{(each aircraft has exactly one origin)}$$

$$\text{cont}_{ij} \in \{0, 1\}, \forall i, j \in \text{PLANES} \qquad \text{(binary constraint)}$$

5.4 Results

The optimal solution has a total of 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: This mapping is simply constructed by the permutation matrix given by

$$\texttt{perm}(i, j) = \begin{pmatrix} \text{cont}_{0,0} & \cdots & \text{cont}_{0,n-1} \\ \vdots & \ddots & \vdots \\ \text{cont}_{n-1,0} & \cdots & \text{cont}_{n-1,n-1} \end{pmatrix}$$

where cont_{ij} is the optimal solution for the above model. Refer to `f-1_report.py` for implementation details.

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