KTH Mathematics

Suggested solutions for the exam in SF2863 Systems Engineering. October 24, 2016 14.00–19.00

Examiner: Per Enqvist, phone: 790 62 98

---

1. We consider the interest rate accumulated at the time $T = t_1 + t_2 + t_3$ of completion of the last project.

The interest rate at time T coming from project j is $c_j((1+\epsilon/100)^{T-C_j} - 1) \approx c_j((1+\epsilon/100(T-C_j) - 1) = c_j\epsilon/100(T-C_j)$ where C_j is the completion time of job j.

Maximizing the sum of all accumulated interest rate is then equivalent to minimizing the sum $\epsilon/100 \sum_{j=1}^{3} c_j C_j$, which is the sum of weighted completion times.

The jobs should be sorted according to WSPT, i.e. with decreasing c_j/t_j quotient.

Here $6/8 = 3/4$, $4/2 = 2$, $5/5 = 1$.

So the schudule should be $2, 3, 1$.

2. (a) Model this as two independent multi-commodity problems, one for incoming trucks and one for outgoing trucks, where the commodities are the reception and delivery docks respectively.

Consider first the incoming side. Let the nodes represent the timeslots for each incoming truck.

Let $X_{ijk} = 1$ if gate k is allocated to truck i and then truck j, and 0 otherwise.

Then we have a constraint that every truck should be allocated to one dock

$$\sum_{k=1}^{m}\sum_{j=1}^{M} X_{ijk} = 1, \qquad i = 1, \cdots, M$$

We define truck 0 as the source node and truck $M+1$ as the sink node, and for these nodes $\sum_{j=1}^{M+1} X_{0jk} = 1$ and $\sum_{i=0}^{M} X_{i,M+1,k} = 1$ for $k = 1, \cdots, m$.

A dock can handle truck j after truck i if $t_i + 30 \le t_j$, so let d_{ij} be 1 if this inequality is satisfied and zero otherwise.

The conservation of flow at each node amounts to the equation

$$X_{0,i,k} + \sum_{j=1}^{M} d_{ji}X_{jik} = X_{i,M+1,k} + \sum_{j=1}^{M} d_{ij}X_{ijk}, \qquad k = 1, \cdots, m, \quad i = 1, \cdots, M.$$

The minimization of number of used docks is equivalent to maximize the number of docks that goes directly from the source to the sink:

$$\sum_{k=1}^{m} X_{0,M+1,k}.$$

The outgoing side can be formulated similarly with variables Y_{ijk}.

(b) The objective function should be

$$\sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{s=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{m}\sum_{\ell=1}^{n} X_{ijk}Y_{s,t,\ell}W_{k\ell}$$

A more efficient solution would be to only include i and j in the sum where $d_{ij} = 1$, and similarly for k and ℓ.

(c) Robustness considerations, such as introducing slack between different loading periods to facilitate operation if there are delays.

Compatibility considerations, such as certain trucks only allowed to use some of the docks, or certain goods need import restrictions or pay toll only handled at certain docks.

There can also be limitations depending on what is loaded on nearby docks.

3. Assumptions about the pizza deliveries: we assume that all pizzas are baked and ready at the pizzeria we choose to deliver from, and that the delivery car can hold all pizzas to be delivered the next two hours and keep them warm.

Alternatively, only a fixed number of cutomers may be delivered to at each run, or only for a certain time, and then new pizzas need to be retrieved at the pizzeria. But this is a bit more complicated to model.

Let $x_{ijk} = 1$ if car k delivers to customer i and then j, and zero otherwise.

Let $\sum_j x_{ijk} = 1$ for all i corresponding to a pizzeria, j corresponding to customers and $k = i$ denote the car belonging to pizzeria i. (if we want to associate a car with a pizzeria) This ensures that every car leaves to one customer.

Let $\sum_i x_{ijk} = 1$ for all i corresponding to a customer, j corresponding to a pizzeriaand $k = j$ denote the car belonging to pizzeria j. This ensures that every car returns to the same pizzeria.

Conservation of flow: For every customer j and car k $\sum_i x_{ijk} = \sum_\ell x_{j\ell k}$ incoming cars should also exit.

Ensure that each customer gets their pizzas: For every customer j $\sum_{i,k} x_{ijk} = 1$ one car should arrive to deliver.

We could also have constraints on the flow of the links $x_{ijk} \leq c_{ijk}$ that restricts cars to go to other pizzerias and satisfy other assumptions on allowed deliveries.

It is also necessary to impose a subtour condition to make sure that cars cannot satisfy conservation of flow by going around in loops that are not connected to the start and finish. For each car k let $Nx_{ijk} + u_{ik} - u_{jk} \leq N - 1$ where $i \neq j$ and N is the total number of customers. The $u_{\ell k}$ will increase by one for each visited customer and if the car returns to an already visited customer the condition will not hold.

Finally the objective function should describe the total driven distance: $\sum_{ijk} x_{ijk}D_{ij} = 1$, where D_{ij} denotes the distance between customers i and j. For example, we may model this as $D_{ij} = \sqrt{(z_i - z_j)^2 + (w_i - w_j)^2}$ between customers, and similarly for distances between customers and pizzerias.

If we obtain new orders after one hour, we may disregard all delivered orders, and only consider remaining orders and new orders. This generate a new optimization problem in which the starting positions for the cars are ' given by the actual location after one hour. Before new pizza orders can be fulfilled a pizzeria has to be visited. Old orders should be prioritized somehow to avoid that some customer gets too long delivery time.

Main differences to project 2 are the lack of time windows for deliveries/visits. No condition on having the same persons visiting. No key problems. Pizza cars may have to return to pizzeria to get new pizzas. Service time for pizza is almost zero. Alarms in project 2 needs to be handled immediately. Home care routes are expected to be more similar day to day.

4. This is the EOQ problem for a serial supply chain

(a) The demand is $d = 2000$ per week. The holding cost is $h_0 = 100$ SEK per week at the warehouse and $h_1 = 200$ SEK at the retailer. The ordering cost is $a_0 = 2000$ and $a_1 = 1000$.

Then the quantity to be ordered and time between orders for the retailer is

$$\hat{Q}_1 = \sqrt{\frac{2a_1 d}{h_1}} = \sqrt{\frac{2a_1 d}{h_1}} = 100\sqrt{2}, \quad \hat{T}_1 = \sqrt{\frac{2a_1}{dh_1}} = \sqrt{2}/100 = 1/(10\sqrt{2}).$$

If the warehouse orders for m periods of the retailer every time he orders, then $T_0 = m\hat{T}_1$, the cost per period T_0 is $a_0 + h_0\frac{(m-1)Q_1}{2}mT_1$ and the cost per time unit is then

$$Z_0(m) = \frac{a_0}{m\hat{T}_1} + \frac{h_0 d\hat{T}_1(m-1)}{2} = 20000\left(\frac{\sqrt{2}}{m} + \frac{m-1}{2\sqrt{2}}\right).$$

Then, since Z_0 is integer-convex, $\hat{m}$ should be the smallest m such that $\Delta Z_0(m) \geq 0$, *i.e.*, satisfying

$$\frac{2a_0}{h_0 d\hat{T}_1^2} = 4 \leq m(m+1).$$

Hence, $m = 2$.

Therefore, $T_0 = 2T_1 = \sqrt{2}/10$, and $Q_0 = 2\hat{Q}_1 = 200\sqrt{2}$.

(b) Add up all the costs for the reorder interval $T_0 = m\hat{T}_1$,

$$a_0 + h_0\frac{(m-1)Q_1}{2}mT_1 + m(a_1 + h_1\frac{Q_1}{2}T_1)$$

Cost per time unit is then

$$Z = a_0/T_0 + ma_1/T_0 + h_0\frac{(m-1)Q_1}{2} + \frac{h_1 Q_1}{2} = (a_1 + a_0/m)/T_1 + T_1(h_0 m + (h_1 - h_0))d/2$$

which is minimized for

$$T_1(m) = \sqrt{\frac{a_1 + a_0/m}{h_0 m + h_1 - h_0}}\sqrt{\frac{2}{d}} = \sqrt{\frac{1 + 2/m}{m+1}}/10$$

Inserting this expression for $T_1 = T_1(m)$ the cost per time unit is then

$$Z(m) = \sqrt{2d}\sqrt{a_1 + a_0/m}\sqrt{h_0 m + h_1 - h_0}$$

$$= \sqrt{4000}\sqrt{1000 + 2000/m}\sqrt{100m + 200 - 100} = 20000\sqrt{1 + 2/m}\sqrt{m+1}.$$

Again using integer convexity. $\hat{m}$ is determined as the smallest m such that

$$m(m+1) \geq \frac{a_0(h_1 - h_0)}{a_1 h_0} = \frac{2000(200-100)}{1000 \cdot 100} = 2$$

This is satisified for $m = 1$.
Therefore, $T_0 = T_1 = T_1(1) = \sqrt{\frac{1000+2000}{100+200-100)}}\sqrt{\frac{2}{2000}} = \frac{1}{10}\sqrt{3/2}$,
and $Q_0 = Q_1 = T_0 d = 200\sqrt{3/2}$.

(c) If one more level is added, then the orders at the next level will be combined for a multiple of the warehouse order period. The system will be more and more sensitive to variations in the demand and expecially if there is a delay in the ordering and delivery process and lack of information between the levels. This is called the Bullwhip effect.

5. The pairings are sequences of flights and layovers that are legal, i.e., such that they start and end in A, each duty period is less than 12 hours (then a layover have to be scheduled or the pairing ends in A), and there should be at most one layover. The departure of a flight also has to be after the arrival of the previous flight, and start where the previous one ends.

The legal pairings are

α :	$1 : A \to B$ layover $2 : B \to A$.
β :	$1 : A \to B$ layover $3 : B \to C \quad 5 : C \to A$.
γ :	$6 : A \to C$ layover $5 : C \to A$.
δ :	$6 : A \to C \quad 4 : C \to B$ layover $2 : B \to A$.
ϵ :	$6 : A \to C \quad 4 : C \to B$ layover $3 : B \to C \quad 5 : C \to A$.

$x_i = 1$ if pairing i is used and 0 otherwise.

Since two aircrafts are needed to fly every pairing, one aircraft starts in A and makes a layover to the next day at another destination and the other start the next day from A and fly the flights of the pairing that are before the layover on the second day, in order that every flight in the pairing are flewn every day, the objective function is given by;

$$2\sum_{i=\alpha}^{\epsilon} x_i.$$

Furthermore, we need constraints that every flight are flewn each day. Therefore, we require (set cover formulation)

$$x_\alpha + x_\beta \geq 1$$

$$x_\alpha + x_\delta \geq 1$$

$$x_\beta + x_\epsilon \geq 1$$

$$x_\delta + x_\epsilon \geq 1$$

$$x_\gamma + x_\epsilon \geq 1$$

$$x_\gamma + x_\delta + x_\epsilon \geq 1$$

In a set partitioning problem we require equality in all constraints.

The minimum of this problem is 4. Optimal solutions are given by $x_\alpha = 1$, $x_\epsilon = 1$ and all other zero, and $x_\beta = 1$, $x_\delta = 1$ and all other zero, hence they are not unique.