42114: Integer Programming

Solutions

Week 4

Updated: September 19, 2022

Week 4: Dynamic Programming

Wolsey Exercise 5.1

We will present a solution using the recursion presented in Wolsey and a solution using a similar recursion to that in the lecture.

The Uncapacitated Lot-Sizing problem is solved as example 5.1 on page 70 (1st edition)/82 (2nd edition) in Wolsey. *Note however, that we use x_t as the production in period t and y_t as the binary variable for whether or not we produce in time period t. This is opposite of Wolsey 2nd edition.* We go through Observation 8.4 at the end of the document for those who need it.

To use the recursion we need to compute the relevant input data $c_t = p_t + \sum_{i=t}^{n} h_i$ and the constant $\sum_{t=1}^{n} h_t d_{1t}$, where $d_{it} = \sum_{j=i}^{t} d_j$. We get

$$c_1 = p_1 + \sum_{i=1}^{4} h_i = 5$$

$$c_2 = p_2 + \sum_{i=2}^{4} h_i = 4$$

$$c_3 = p_3 + \sum_{i=3}^{4} h_i = 3$$

$$c_4 = p_4 + \sum_{i=4}^{4} h_i = 3$$

So $c = (5, 4, 3, 3)$, and

$$[d_{ij}] = \begin{bmatrix} 8 & 13 & 26 & 30 \\ & 5 & 18 & 22 \\ & & 13 & 17 \\ & & & 4 \end{bmatrix}$$

such that

$$\sum_{t=1}^{4} h_t d_{1t} = 77$$

The recursion is as defined in Wolsey:

$$H(k) = \min_{1 \le t \le k} \{H(t-1) + f_t + c_t d_{tk}\}$$

The dynamic programming is based on forward recursion. So we start with the optimal production plan for the case of 0 time periods:

$$H(0) = 0$$

Then we find the optimal production plan for the case of 1 time period:

$$\begin{aligned} H(1) &= \min\{H(0) + f_1 + c_1 d_{11}\} \\ &= \min\{0 + 20 + 5 \cdot 8\} \\ &= \min\{60\} \\ &= 60 \end{aligned}$$

We proceed to the optimal production plan for the case of 2 time periods:

$$\begin{aligned} H(2) &= \min\{H(0) + f_1 + c_1 d_{12}\,,\, H(1) + f_2 + c_2 d_{22}\} \\ &= \min\{0 + 20 + 5 \cdot 13\,,\, 60 + 10 + 4 \cdot 5\} \\ &= \min\{85, 90\} \\ &= 85 \end{aligned}$$

Then we make the optimal production plan for the case of 3 time periods:

$$\begin{aligned} H(3) &= \min\{H(0) + f_1 + c_1 d_{13}\,,\, H(1) + f_2 + c_2 d_{23}\,,\, H(2) + f_3 + c_3 d_{33}\} \\ &= \min\{0 + 20 + 5 \cdot 26\,,\, 60 + 10 + 4 \cdot 18\,,\, 85 + 45 + 3 \cdot 13\} \\ &= \min\{150, 142, 169\} \\ &= 142 \end{aligned}$$

And finally, we find the optimal production plan for the case of all four time periods:

$$\begin{aligned} H(4) &= \min\{H(0) + f_1 + c_1 d_{14}\,,\, H(1) + f_2 + c_2 d_{24}\,,\, H(2) + f_3 + c_3 d_{34}\,,\, H(3) + f_4 + c_4 d_{44}\} \\ &= \min\{0 + 20 + 5 \cdot 30\,,\, 60 + 10 + 4 \cdot 22\,,\, 85 + 45 + 3 \cdot 17\,,\, 142 + 15 + 3 \cdot 4\} \\ &= \min\{170, 158, 181, 169\} \\ &= 158 \end{aligned}$$

In the case of all four time periods, we see that in the optimal solution the last period in which we produce is $t = 2$. Hence $y_3 = y_4 = 0$ and $y_2 = 1$ and in time period $t = 2$ we have to produce enough for the rest of the time periods, hence $x_2 = 5 + 13 + 4 = 22$.
For the remaining 1 time period, we also produce. Hence $y_1 = 1$ and $x_1 = 8$.
The objective value of the optimal solution is then found by subtracting the constant $\sum_{t=1}^{n} h_t d_{1t}$ which has been ignored during the optimization process: $158 - 77 = 81$.

We now solve it using a similar method to that in the lecture.
The recursion given in the lecture was:

$$H(k) = \min_{1 \le t \le k} \{H(t-1) + f_t + h v_{tk}\}$$

where $H(k)$ is the cheapest production cost for periods $1 - k$ and we define

$$v_{tk} = \sum_{i=t}^{k} d_i (i - t)$$

In the lecture, we had assumed that the production costs were identical for each day. This means that no matter what we have to produce a certain amount of units (the sum of the demands) and it will cost the same regardless of when it is produced. Hence we could ignore this aspect during the optimization.
In our case, we do not have identical production costs and can thus not do this. Hence we have to modify the recursion slightly to include this. Let us define:

$$d_{it} = \sum_{j=i}^{t} d_j$$

Then the recursion should be:

$$H(k) = \min_{1\leq t\leq k}\{H(t-1) + f_t + hv_{tk} + p_t d_{tk}$$

Using the initial condition when considering 0 periods: $H(0) = 0$, we perform the forward recursion. We start by precomputing the v-matrix for v_{tk}:

$$v = \begin{bmatrix} 0 & 5 & 31 & 43 \\ & 0 & 13 & 21 \\ & & 0 & 4 \\ & & & 0 \end{bmatrix}$$

We find the optimal production plan for the case of one time period:

$$\begin{aligned} H(1) &= \min\{H(0) + f_1 + hv_{11} + p_1 d_{11}\} \\ &= \min\{0 + 20 + 1\cdot 0 + 1\cdot 8\} \\ &= 28 \end{aligned}$$

We proceed to the optimal production plan for the case of two time periods:

$$\begin{aligned} H(2) &= \min\{H(0) + f_1 + hv_{12} + p_1 d_{12}\,,\ H(1) + f_2 + hv_{22} + p_2 d_{22}\} \\ &= \min\{0 + 20 + 1\cdot 5 + 1\cdot(8+5)\,,\ 28 + 10 + 1\cdot 0 + 1\cdot 5\} \\ &= \min\{38, 43\} \\ &= 38 \end{aligned}$$

Then we make the optimal production plan for the case of three time periods:

$$\begin{aligned} H(3) &= \min\{H(0) + f_1 + hv_{13} + p_1 d_{13}\,,\ H(1) + f_2 + hv_{23} + p_2 d_{23}\,,\ H(2) + f_3 + hv_{33} + p_3 d_{33}\} \\ &= \min\{0 + 20 + 1\cdot 31 + 1\cdot(8+5+13)\,,\ 28 + 10 + 1\cdot 13 + 1\cdot(5+13)\,,\ 38 + 45 + 1\cdot 0 + 1\cdot 13\} \\ &= \min\{77, 69, 96\} \\ &= 69 \end{aligned}$$

And finally, we find the optimal production plan for the case of all four time periods:

$$\begin{aligned} H(4) &= \min\{H(0) + f_1 + hv_{14} + p_1 d_{14}\,,\ H(1) + f_2 + hv_{24} + p_2 d_{24}\,, \\ &\qquad H(2) + f_3 + hv_{34} + p_3 d_{34}\,,\ H(3) + f_4 + hv_{44} + p_4 d_{44}\} \\ &= \min\{0 + 20 + 1\cdot 43 + 1\cdot(8+5+13+4)\,,\ 28 + 10 + 1\cdot 21 + 1\cdot(5+13+4)\,, \\ &\qquad 38 + 45 + 1\cdot 4 + 1\cdot(13+4)\,,\ 69 + 15 + 1\cdot 0 + 2\cdot 4\} \\ &= \min\{93, 81, 104, 92\} \\ &= 81 \end{aligned}$$

In the case of all four time periods, we see that the optimal solution value is 81 and in the optimal solution, the last period in which we produce is $t = 2$. Hence $y_3 = y_4 = 0$ and $y_2 = 1$ and in time period $t = 2$ we have to produce enough for the rest of the time periods, hence $x_2 = 5 + 13 + 4 = 22$. For the remaining single time period we also produce. Hence $y_1 = 1$ and $x_1 = 8$.

Wolsey Exercise 5.3

The maximum weight rooted subtree problem/*Optimal Subtree of a Tree problem* is described in Wolsey on page 71-72 (1st edition)/83-84 (2nd edition). We solve the problem as in example 5.2 on page 72 (1st edition)/84 (2nd edition). We use the recursion:

$$H(v) = \max\{0, c_v + \sum_{w\in S(v)} H(w)\}$$

where $H(v)$ is the optimal solution value of the maximum weight rooted subtree problem defined on the subtree with node v as root, and $S(v)$ is the set of immediate successor nodes for node v.
Starting with the leaves (4, 9, 10, 6, 7, 11, 12) we get:

$$H(4) = 4$$

$$H(9) = 2$$
$$H(10) = 3$$
$$H(6) = 0$$
$$H(7) = 0$$
$$H(11) = 0$$
$$H(12) = 3$$

I.e. the optimal subtree of nodes with positive label is the node itself, while the optimal subtree of nodes with negative label is the empty subtree with cost 0.
We can now calculate $H(5)$ and $H(8)$, since the optimal solution for all immediate successor nodes are known:

$$H(5) = \max\{0, c_5 + H(9) + H(10)\} = \max\{0, -6 + 2 + 3\} = 0$$
$$H(8) = \max\{0, c_8 + H(11) + H(12)\} = \max\{0, 1 + 0 + 3\} = 4$$

We see that the optimal subtree of node 5 is the empty subtree due to the large negative label, while for node 8 the optimal subtree consist of nodes 8 and 12.
Now we can continue with $H(2)$ and $H(3)$:

$$H(2) = \max\{0, c_2 + H(4) + H(5) + H(6)\} = \max\{0, -1 + 4 + 0 + 0\} = 3$$

$$H(3) = \max\{0, c_3 + H(7) + H(8)\} = \max\{0, -1 + 0 + 4\} = 3$$

We see that the optimal subtree of node 2 consists of nodes 2 and 4, while for node 3 it consists of nodes 3, 8 and 12.
We can then find the optimal value which is $H(1)$:

$$H(1) = \max\{0, c_1 + H(2) + H(3)\} = \max\{0, -2 + 3 + 3\} = 4$$

We can now use all the computations to see that the optimal subtree consists of the nodes: $1, 2, 3, 4, 8$ and 12.

Wolsey Exercise 5.5

(See page 67-68 (1st edition)/79-80 (2nd edition) in Wolsey for a description of the problem without passage times.)

(i) Let $f_k(j)$ be the shortest time required to go from node 1 to node j using at most k arcs. If node j cannot be reached $f_k(j) = \infty$.

For $k = 1$ let

$$f_1(j) = \begin{cases} \max\{c_{1j}, r_j\} & \text{, if } (1,j) \in A \\ \infty & \text{, otherwise} \end{cases}$$

The recursion for $2 \leq k \leq |A|$ is then:

$$f_k(j) = \max\{r_j, \min\{f_{k-1}(j), \min_{i \in V^-(j)} \{c_{ij} + f_{k-1}(i)\}\}\}$$

where $V^-(j)$ is the set of predecessors of node j.

(ii) We need two decision variables: $t_i \geq 0$ which is the time node i is visited (if node i is never visited t_i is unspecified) and $x_{ij}\{0,1\}$ indicating if the arc from node i to node j is traversed ($x_{ij} = 1$) or not ($x_{ij} = 0$).

We need to choose exactly one arc out of node 1 to traverse:

$$\sum_{i=1}^{n} x_{1i} = 1$$

Likewise, we need to choose precisely one arc into node n to traverse:

$$\sum_{i=1}^{n} x_{in} = 1$$

For all other nodes we must ensure that either no arc leaves or enters the node, or that exactly one arc enters and one arc leaves the node:

$$\sum_{i=1, i \neq j}^{n} x_{ij} = \sum_{i=1, i \neq j}^{n} x_{ji} \qquad \forall j = 2, ..., n-1$$

For each node j we must enforce the earliest passage times on the visit times:

$$r_j \leq t_j \qquad \forall j = 1, ..., n$$

We need a "balancing" constraint to ensure that the visit times are interconnected on the chosen route. Hence if an arc $(i,j) \in A$ is traversed, then the visit time t_j of node j cannot be earlier than the visit time t_i of node i plus the travel time from node i to node j:

$$t_i + c_{ij}x_{ij} - M(1 - x_{ij}) \leq t_j \qquad \forall (i,j) \in A$$

If $x_{ij} = 0$, hence the arc isn't traversed, then the term $-M(1-x_{ij})$ ensures that there is no limit on t_j (apart from the earliest passage time of course).
Finally, we want to minimize the visit time for node n and thereby the time required to go from node 1 to node n:

$$\min t_n$$

Hence the full Mixed Integer Program is:

$$\begin{array}{llcll}
\min & t_n & & & \\
\text{s.t.} & \sum_{i=1}^{n} x_{1i} & = & 1 & \\
 & \sum_{i=1}^{n} x_{in} & = & 1 & \\
 & \sum_{j=1, i \neq j}^{n} x_{ij} & = & \sum_{j=1, i \neq j}^{n} x_{ji} & \forall i = 2, ..., n-1 \\
 & r_j & \leq & t_j & \forall j = 1, ..., n \\
 & t_i + c_{ij}x_{ij} - M(1 - x_{ij}) & \leq & t_j & \forall (i,j) \in A \\
 & x_{ij} & \in & \{0,1\} & \forall i,j = 1, ..., n,\ i \neq j \\
 & t_j & \geq & 0 & \forall j = 1, ..., n
\end{array}$$

The number of binary variables is $O(m)$ where m is the number of arcs. Likewise the number of constraints is $O(m)$ (assuming $m \geq n$). This is to some extent fine - neither is exponential at least. So one way to go would be to solve the problem by solving the LP-relaxation and hope the optimal solution is integer or at least that the lower bound obtained is good. It is probably not - formulations involving Big-M are usually not nice and in fact the quality of the lower bound depends on what values of M are used.

Dynamic Programming Exercises

Exercise 1

We will first solve the problem by constructing a recursion ourselves and then we will see how we can consider the problem in another way to reuse a recursion we already know.

Let $v_j(x)$ be the minimal number of crimes for districts $1, ..., j$ given x patrol cars.

We define the recursion as

$$v_j(x) = \min_{0 \leq w \leq x} \{v_{j-1}(x-w) + r_j(w)\}$$

which consists of the number of crimes $r_j(w)$ in district j if we assign w patrol cars to the district and the minimal number of crimes for the remaining districts $1, ..., j-1$ when using the remaining $x - w$ patrol cars.

We define the "base case" for when we consider no districts ($j = 0$):

$$v_0(x) = 0 \qquad \forall x$$

We then consider for only one district:

$$\begin{aligned} v_1(3) &= \min\{v_0(0) + r_1(3)\,,\, v_0(1) + r_1(2)\,,\, v_0(2) + r_1(1)\,,\, v_0(3) + r_1(0)\} \\ &= \min\{0 + 4\,,\, 0 + 7\,,\, 0 + 10\,,\, 0 + 14\} \\ &= 4 \end{aligned}$$

$$\begin{aligned} v_1(2) &= \min\{v_0(0) + r_1(2)\,,\, v_0(1) + r_1(1)\,,\, v_0(2) + r_1(0)\} \\ &= \min\{0 + 7\,,\, 0 + 10\,,\, 0 + 14\} \\ &= 7 \end{aligned}$$

$$\begin{aligned} v_1(1) &= \min\{v_0(0) + r_1(1)\,,\, v_0(1) + r_1(0)\} \\ &= \min\{0 + 10\,,\, 0 + 14\} \\ &= 10 \end{aligned}$$

$$\begin{aligned} v_1(0) &= \min\{v_0(0) + r_1(0)\} \\ &= \min\{0 + 14\} \\ &= 14 \end{aligned}$$

We then consider two districts (district 1 and 2):

$$\begin{aligned}
v_2(3) &= \min\{v_1(0)+r_2(3)\,,\ v_1(1)+r_2(2)\,,\ v_1(2)+r_2(1)\,,\ v_1(3)+r_2(0)\}\\
&= \min\{14+14\,,\ 10+16\,,\ 7+19\,,\ 4+25\}\\
&= \min\{28, 26, 26, 29\}\\
&= 26
\end{aligned}$$

$$\begin{aligned}
v_2(2) &= \min\{v_1(0)+r_2(2)\,,\ v_1(1)+r_2(1)\,,\ v_1(2)+r_2(0)\}\\
&= \min\{14+16\,,\ 10+19\,,\ 7+25\}\\
&= \min\{30, 29, 32\}\\
&= 29
\end{aligned}$$

$$\begin{aligned}
v_2(1) &= \min\{v_1(0)+r_2(1)\,,\ v_1(1)+r_2(0)\}\\
&= \min\{14+19\,,\ 10+25\}\\
&= \min\{33, 35\}\\
&= 33
\end{aligned}$$

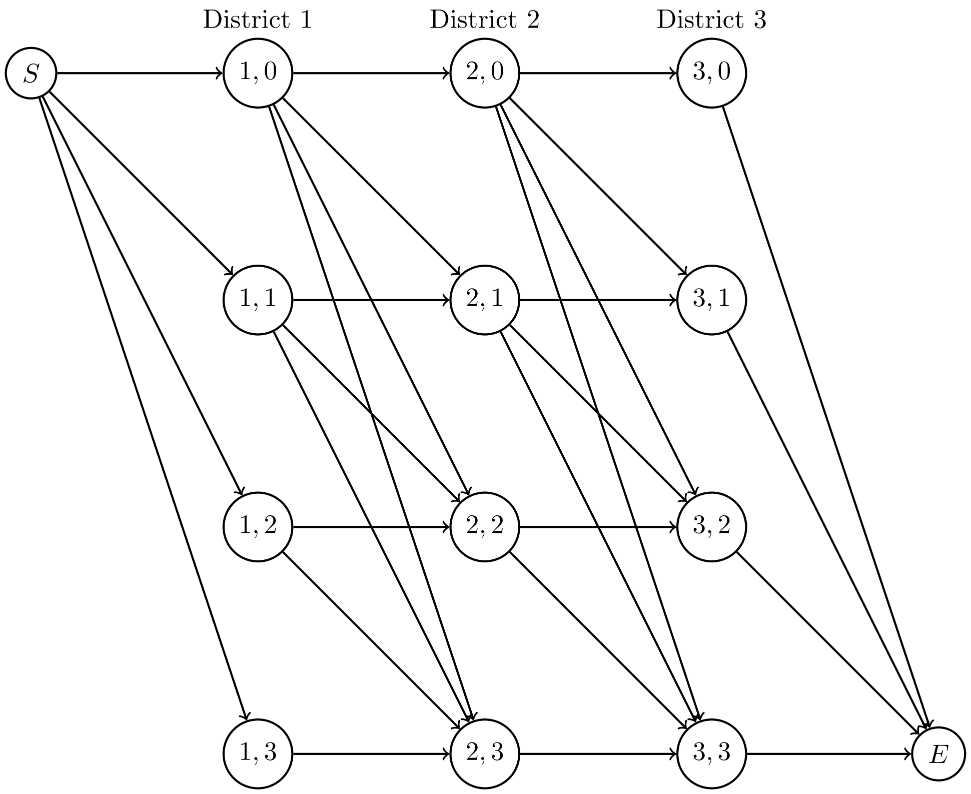
$$\begin{aligned}
v_2(0) &= \min\{v_1(0)+r_2(0)\}\\
&= \min\{14+25\}\\
&= 39
\end{aligned}$$

To get the optimal solution we consider all three districts:

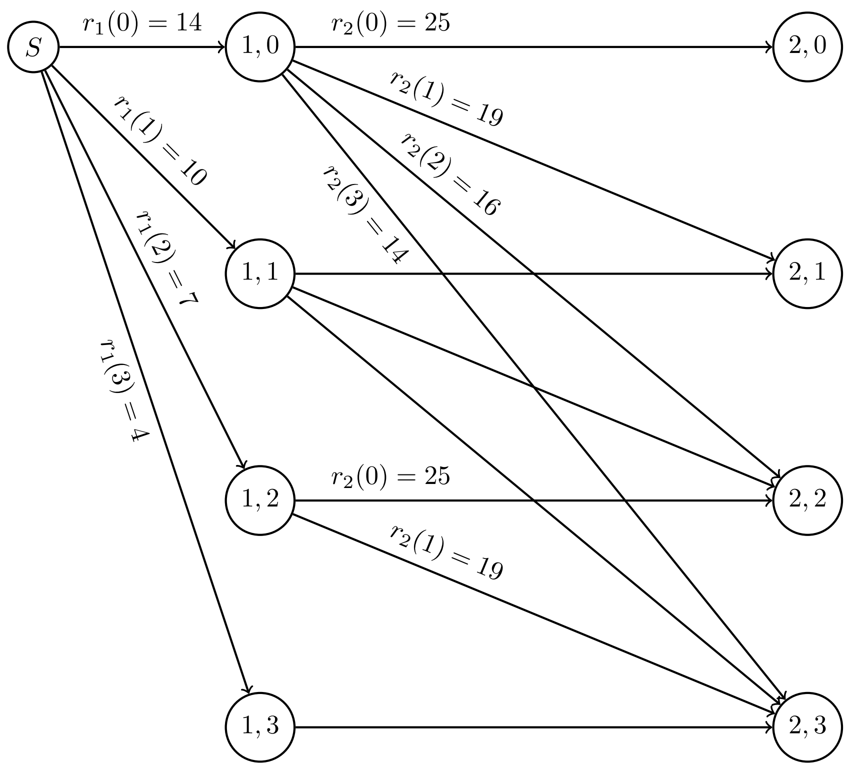
$$\begin{aligned}
v_3(3) &= \min\{v_2(0)+r_3(3)\,,\ v_2(1)+r_3(2)\,,\ v_2(2)+r_3(1)\,,\ v_2(3)+r_3(0)\}\\
&= \min\{39+8\,,\ 33+11\,,\ 29+14\,,\ 26+20\}\\
&= \min\{47, 44, 43, 46\}\\
&= 43
\end{aligned}$$

Going backwards through the calculations we see that district 3 should be assigned 1 patrol car, district 2 should be assigned 1 patrol car and district 1 should also be assigned 1 patrol car, leading to the minimal number of crimes of 43.

Another way to solve the problem is to write it up as a shortest path problem. Consider the directed acyclic graph below.



Each layer in the graph corresponds to the total number of patrol cars we have currently used. That is, in nodes (S), $(1,0)$, $(2,0)$ and $(3,0)$ we have assigned a total of 0 patrol cars, while in nodes $(1,1)$, $(2,1)$ and $(3,1)$ we have assigned a total of 1 patrol car etc.
Each of the three columns in the middle corresponds to a district. Hence moving through the graph from (S) to (E) will represent assigning patrol cars to District 1, then to District 2 and finally to District 3.
For instance the path from $(S) \to (1,1) \to (2,1) \to (3,2) \to (E)$ represents that we assign 1 patrol car to District 1, then assign 0 patrol cars to District 2 (hence staying on the same layer in the graph) and then assigning 1 patrol car to District 3.
The edge weights in the graph will be the number of crimes corresponding to the assignment of patrol cars represented by the edge. A smaller part of the graph is shown in the following figure with some of the edge weights shown.



Each of the edges out of (S) and into $(1, 0)$-$(1, 3)$ represents assigning 0-3 patrol cars to District 1. Hence the edge weights will be the number of crimes in District 1 when assigned with 0-3 patrol cars. That is $r_1(0) - r_1(3)$.
For each of the edges out of $(1, 2)$ and into $(2, 2)$ and $(2, 3)$, they represent assigning 0 or 1 patrol car to District 2. Hence their edge weights will be $r_2(0)$ and $r_2(1)$.
All edge weights are defined in the same way, except for the edges from $(3, 0) - (3, 3)$ and into (E) which have an edge weight of 0.

Now the problem of finding a shortest path from (S) to (E) in the graph corresponds to the problem of distributing the 3 patrol cars between the districts minimizing the number of crimes. For this we have a recursion in Wolsey in Observation 5.2 page 68 (1st edition)/79 (2nd edition):

$$d(v) = \min_{i \in V^-(v)} \{d(i) + c_{iv}\}$$

where $d(v)$ is the length of the shortest path from (S) to node v, $V^-(v)$ is the set of immediate predecessors to node v and c_{iv} is the edge weight of the edge from node i to node v.
The initial condition will then be $d(S) = 0$.

Using this forward recursion we start by considering the first column of nodes:

$$\begin{aligned}
d(1,0) &= \min\{d(S) + r_1(0)\} \\
&= \min\{0 + 14\} \\
&= 14
\end{aligned}$$

$$\begin{aligned}
d(1,1) &= \min\{d(S) + r_1(1)\} \\
&= \min\{0 + 10\} \\
&= 10
\end{aligned}$$

$$\begin{aligned}
d(1,2) &= \min\{d(S) + r_1(2)\} \\
&= \min\{0 + 7\} \\
&= 7
\end{aligned}$$

$$\begin{aligned}
d(1,3) &= \min\{d(S) + r_1(3)\} \\
&= \min\{0 + 4\} \\
&= 4
\end{aligned}$$

Then we consider the second column of nodes:

$$\begin{aligned}
d(2,0) &= \min\{d(1,0) + r_2(0)\} \\
&= \min\{14 + 25\} \\
&= 39
\end{aligned}$$

$$\begin{aligned}
d(2,1) &= \min\{d(1,0) + r_2(1)\,,\ d(1,1) + r_2(0)\} \\
&= \min\{14 + 19\,,\ 10 + 25\} \\
&= \min\{33, 35\} \\
&= 33
\end{aligned}$$

$$\begin{aligned}
d(2,2) &= \min\{d(1,0) + r_2(2)\,,\ d(1,1) + r_2(1)\,,\ d(1,2) + r_2(0)\} \\
&= \min\{14 + 16\,,\ 10 + 19\,,\ 7 + 25\} \\
&= \min\{30, 29, 32\} \\
&= 29
\end{aligned}$$

$$\begin{aligned}
d(2,3) &= \min\{d(1,0) + r_2(3)\,,\ d(1,1) + r_2(2)\,,\ d(1,2) + r_2(1)\,,\ d(1,3) + r_2(0)\} \\
&= \min\{14 + 14\,,\ 10 + 16\,,\ 7 + 19\,,\ 4 + 25\} \\
&= \min\{28, 26, 26, 29\} \\
&= 26
\end{aligned}$$

We then consider the third column of nodes:

$$\begin{aligned}
d(3,0) \quad &= \min\{d(2,0) + r_3(0)\} \\
&= \min\{39 + 20\} \\
&= 59
\end{aligned}$$

$$\begin{aligned}
d(3,1) \quad &= \min\{d(2,0) + r_3(1)\,,\; d(2,1) + r_3(0)\} \\
&= \min\{39 + 14\,,\; 33 + 20\} \\
&= \min\{53, 53\} \\
&= 53
\end{aligned}$$

$$\begin{aligned}
d(3,2) \quad &= \min\{d(2,0) + r_3(2)\,,\; d(2,1) + r_3(1)\,,\; d(2,2) + r_3(0)\} \\
&= \min\{39 + 11\,,\; 33 + 14\,,\; 29 + 20\} \\
&= \min\{50, 47, 49\} \\
&= 47
\end{aligned}$$

$$\begin{aligned}
d(3,3) \quad &= \min\{d(2,0) + r_3(3)\,,\; d(2,1) + r_3(2)\,,\; d(2,2) + r_3(1)\,,\; d(2,3) + r_3(0)\} \\
&= \min\{39 + 8\,,\; 33 + 11\,,\; 29 + 14\,,\; 26 + 20\} \\
&= \min\{47, 44, 43, 46\} \\
&= 43
\end{aligned}$$

Then the objective value of the shortest path problem which is also the optimal value for minimizing crime is:

$$\begin{aligned}
d(E) \quad &= \min\{d(3,0) + 0\,,\; d(3,1) + 0\,,\; d(3,2) + 0\,,\; d(3,3) + 0\} \\
&= \min\{59, 53, 47, 43\} \\
&= 43
\end{aligned}$$

Going backwards through the calculations we see that it is best to assign 1 patrol car to District 3, 1 patrol car to District 2 and 1 patrol car to District 1 for the minimum of 43 crimes.

Exercise 2

(a) The principle of optimality states that the optimal solution to the problem can be composed of optimal solutions to the subproblems.
If the given recursion followed the principle of optimality, then we should gain the optimal solution (the sequence $3 \to 1 \to 2$ with the value $p_3 + s_{31} + p_1 + s_{12} + p_2 = 21$) when solving the subproblems and combining them.

The initial state values are:

$$c(\{1\}) = 4$$

$$c(\{2\}) = 9$$

$$c(\{3\}) = 5$$

We then look at the subproblems with subsets of two tasks:

$$\begin{aligned} c(\{1,2\}) &= \min\{c(\{2\}) + p_1 + s_{21}\,,\, c(\{1\}) + p_2 + s_{12}\} \\ &= \min\{9+4+4\,,\, 4+9+1\} \\ &= \min\{17, 14\} \\ &= 14 \qquad \text{with } f(\{1,2\}) = 2 \end{aligned}$$

$$\begin{aligned} c(\{1,3\}) &= \min\{c(\{3\}) + p_1 + s_{31}\,,\, c(\{1\}) + p_3 + s_{13}\} \\ &= \min\{5+4+2\,,\, 4+5+1\} \\ &= \min\{11, 10\} \\ &= 10 \qquad \text{with } f(\{1,3\}) = 3 \end{aligned}$$

$$\begin{aligned} c(\{2,3\}) &= \min\{c(\{3\}) + p_2 + s_{32}\,,\, c(\{2\}) + p_3 + s_{23}\} \\ &= \min\{5+9+3\,,\, 9+5+3\} \\ &= \min\{17, 17\} \\ &= 17 \qquad \text{with } f(\{1,2\}) = 3 \text{ (or 2)} \end{aligned}$$

Finally we can consider all three tasks, which should result in the optimal solution if the recursion follows the principle of optimality:

$$\begin{aligned} c(\{1,2,3\}) &= \min\{c(\{2,3\}) + p_1 + s_{31}\,,\, c(\{1,3\}) + p_2 + s_{32}\,,\, c(\{1,2\}) + p_3 + s_{23}\} \\ &= \min\{17+4+2\,,\, 10+9+3\,,\, 14+5+3\} \\ &= \min\{23, 22, 22\} \\ &= 22 \qquad \text{with } f(\{1,2,3\}) = 2 \text{ (or 3)} \end{aligned}$$

We see that we get the sequence $\{1,2,3\}$ with the value 22, which is not the optimal solution.

When solving the subproblem $c(\{1,3\})$ the choice of the "optimal" sequence is based only on the processing times of the tasks and the setup time between the two tasks in the subproblem. However, when we concider the full problem the choice of sequence also affects which setup cost should be added before the last task. Hence while $\{1,3\}$ is better than $\{3,1\}$ when only considering the two tasks, if also considering the setup time before task 2 then $\{3,1\}$ is actually better than $\{1,3\}$.

(b) The initial state values are:

$$c(\{1\},1) = 4$$

$$c(\{2\},2) = 9$$

$$c(\{3\},3) = 5$$

We then consider the subproblems with subsets of two tasks:

$$\begin{aligned} c(\{1,2\},1) &= c(\{2\},2) + p_1 + s_{21} = 9+4+4 = 17 \\ c(\{1,2\},2) &= c(\{1\},1) + p_2 + s_{12} = 4+9+1 = 14 \end{aligned}$$

$$\begin{aligned} c(\{1,3\},1) &= c(\{3\},3) + p_1 + s_{31} = 5+4+2 = 11 \\ c(\{1,3\},3) &= c(\{1\},1) + p_3 + s_{13} = 4+5+1 = 10 \end{aligned}$$

$$\begin{aligned} c(\{2,3\},2) &= c(\{3\},3) + p_2 + s_{32} = 5+9+3 = 17 \\ c(\{2,3\},3) &= c(\{2\},2) + p_3 + s_{23} = 9+5+3 = 17 \end{aligned}$$

And then consider the subproblems with all three tasks to find the optimal solution:

$$\begin{aligned}
c(\{1,2,3\},1) &= \min\{c(\{2,3\},2)+p_1+s_{21}\,,\ c(\{2,3\},3)+p_1+s_{31}\}\\
&= \min\{17+4+4, 17+4+2\}\\
&= \min\{25,23\}\\
&= 23
\end{aligned}$$

$$\begin{aligned}
c(\{1,2,3\},2) &= \min\{c(\{1,3\},1)+p_2+s_{12}\,,\ c(\{1,3\},3)+p_2+s_{32}\}\\
&= \min\{11+9+1, 10+9+3\}\\
&= \min\{21,22\}\\
&= 21
\end{aligned}$$

$$\begin{aligned}
c(\{1,2,3\},3) &= \min\{c(\{1,2\},1)+p_3+s_{13}\,,\ c(\{1,2\},2)+p_3+s_{23}\}\\
&= \min\{17+5+1, 14+5+3\}\\
&= \min\{23,22\}\\
&= 22
\end{aligned}$$

So we see that we get the optimal solution of 21 for the sequence $3 \to 1 \to 2$.

Observation 8.4 - elaboration

Considering the Uncapacitated Lot-Sizing problem (ULS), the objective function of the MIP formulation is:

$$\min \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t$$

where x_t is the production in period t, s_t is the amount stored at the end of period t and $y \in \{0,1\}$ indicates whether or not we produce anything in period t.

First let $d_{ij} = \sum_{t=1}^{j} d_t$ be the sum of demands in periods from i to j. We consider the amount we store at the end of period t: s_t. This must be equal to the total amount of production thus far ($\sum_{i=1}^{t} x_i$) minus the total amount of demand thus far (d_{1t}):

$$s_t = \sum_{i=1}^{t} x_i - d_{1t}$$

By substituting this into the objective function we have:

$$\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t s_t + \sum_{t=1}^{n} f_t y_t$$

$$= \sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t \left(\sum_{i=1}^{t} x_i - d_{1t} \right) + \sum_{t=1}^{n} f_t y_t$$

We notice that h_t is multiplied onto all x_i for $i \leq t$. Likewise for a given x_t it is multiplied by all

h_i for $i \geq t$. Then:

$$\begin{aligned}
&\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} h_t \left(\sum_{i=1}^{t} x_i - d_{1t} \right) + \sum_{t=1}^{n} f_t y_t \\
= &\sum_{t=1}^{n} p_t x_t + \sum_{t=1}^{n} \left(\sum_{i=t}^{n} h_i \right) x_t - \sum_{t=1}^{n} h_t d_{1t} + \sum_{t=1}^{n} f_t y_t \\
= &\sum_{t=1}^{n} \left(p_t + \sum_{i=t}^{n} h_i \right) x_t - \sum_{t=1}^{n} h_t d_{1t} + \sum_{t=1}^{n} f_t y_t
\end{aligned}$$

Now let us define $c_t = p_t + \sum_{i=t}^{n} h_i$. Then we have:

$$\begin{aligned}
&\sum_{t=1}^{n} \left(p_t + \sum_{i=t}^{n} h_i \right) x_t - \sum_{t=1}^{n} h_t d_{1t} + \sum_{t=1}^{n} f_t y_t \\
= &\sum_{t=1}^{n} c_t x_t - \sum_{t=1}^{n} h_t d_{1t} + \sum_{t=1}^{n} f_t y_t
\end{aligned}$$

Since $\sum_{t=1}^{n} h_t d_{1t}$ is a constant, we can ignore it while finding the optimal solution and then subtract it from the final objective function value to get the real objective value. Hence the objective function becomes:

$$\min \sum_{t=1}^{n} c_t x_t + \sum_{t=1}^{n} f_t y_t$$

which is what is used in the recursion in example 5.1.

Julia Code

Contents