Homework II
Advanced Topics in Optimization

Basil R. Yap

2018 February 05

1 Question 1

system description: $1|chains|\sum w_j C_j$

jobs	1	2	3	4	5	6	7
w_j	0	18	12	8	8	17	16
p_j	3	6	6	5	4	8	9

$$1 \rightarrow 2$$
$$3 \rightarrow 4 \rightarrow 5$$
$$6 \rightarrow 7$$

$\rho(1,\cdots,k) = \max_{l=1,\cdots,k} \frac{\sum_{j=1}^{l} w_j}{\sum_{j=1}^{l} p_j}$

$$\begin{aligned}
C_1 : \quad \rho(1,2) &= \max\left(\frac{0}{3}, \frac{0+18}{3+6}\right)\\
&= 2\\
C_2 : \quad \rho(3,4,5) &= \max\left(\frac{12}{6}, \frac{12+8}{6+5}, \frac{12+8+8}{6+5+4}\right)\\
&= \max\left(2, 1\frac{9}{11}, 1\frac{13}{15}\right)\\
&= 2\\
C_3 : \quad \rho(6,7) &= \max\left(\frac{17}{8}, \frac{17+16}{8+9}\right)\\
&= \max\left(2\frac{1}{8}, 1\frac{16}{17}\right)\\
&= 2\frac{1}{8}
\end{aligned}$$

$\rho(C_1, C_2, C_3) = 2\frac{1}{8}$

Considering that C_3's ρ value is maximized at job 6, job 7 has to be reevaluated while job 6 will be the first job in the optimal sequence.

$C_{31} : \ \rho(7) = 1\frac{7}{9}$

$\rho(C_1, C_2, C_{31}) = 2$

job 4 and 5 need to be reevaluated.

$$\begin{aligned} C_{321}: \quad \rho(4,5) &= \max\left(1\frac{3}{5}, 1\frac{7}{9}\right) \\ &= 1\frac{7}{9} \end{aligned}$$

$6 \to 1 \to 2 \to 3$ and $6 \to 3 \to 1 \to 2$ are part of an Optimal Sequence.

$\rho(C_{31}, C_{321}) = 1\frac{7}{9}$

both subchains share the same ρ value and, therefore, are interchangeable.

The list of possible optimal sequences are: $6 \to 1 \to 2 \to 3 \to 7 \to 4 \to 5$
$6 \to 1 \to 2 \to 3 \to 4 \to 5 \to 7$
$6 \to 3 \to 1 \to 2 \to 7 \to 4 \to 5$
$6 \to 3 \to 1 \to 2 \to 4 \to 5 \to 7$

2 Question 2

Consider the problem $1|r_j, prmp|\sum C_j$, show that the preemptive Shortest Remaining Processing Time(SRPT) first rule is Optimal.

SRPT: At any point in time, schedule the job with the shortest remaining time preempting when jobs with shorter processing times are released.

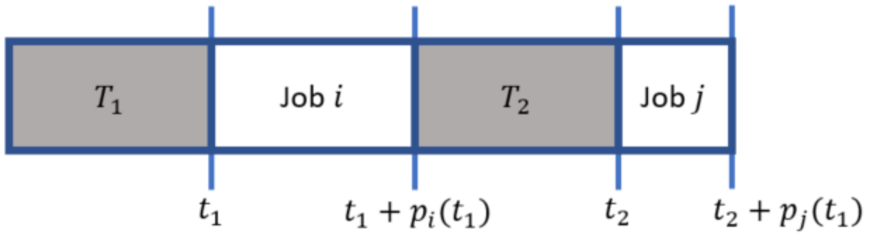
Inverse statement: All optimal schedules do not follow the SRPT rule. Assume inverse statement is true.

Given any schedule from the set of optimal schedules, consider any job i and job j, such that:

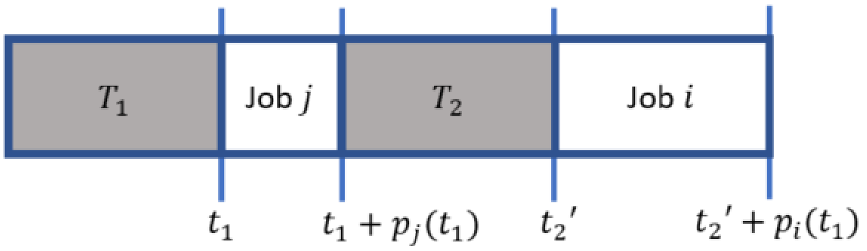
$$p_i(t) > p_j(t)$$

where $p_i(t)$ is the remaining processing time of job i at time t.

In this optimal schedule, S, job i and job j are not processed in sequence T_2. Therefore, by definition, job i is placed before job j where t_1 is the released time of a job with a shorter processing time than the two.



If we were to swap job i and job j, we would get a new schedule, S'.



In this new schedule:

$$
\begin{aligned}
C_i &= t_1 + p_i(t_1) \\
C_j &= t_2 + p_j(t_1)
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k \in S} C_k &= \left(\sum_{k \in S/\{i,j\}} C_k \right) + C_i + C_j \\
&= (T_1 + T_2) + (t_1 + p_i(t_1)) + (t_2 + p_j(t_1))
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k \in S} C'_k &= \left(\sum_{k \in S/\{i,j\}} C'_k \right) + C'_i + C'_j \\
&= (T_1 + T'_2) + (t'_2 + p_i(t_1)) + (t_1 + p_j(t_1))
\end{aligned}
$$

$$
\sum_{k \in S} C'_k - \sum_{k \in S} C_k = T'_2 - T_2 + t'_2 - t_2
$$

$$
\begin{aligned}
T'_2 &< T_2 \\
t'_2 &< t_2
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k \in S} C'_k - \sum_{k \in S} C_k &< 0 \\
\therefore \sum_{k \in S} C'_k &< \sum_{k \in S} C_k
\end{aligned}
$$

Therefore, schedule S' is also optimal and also does not obey the SRFT rule.

As the new schedule does not follow the SRFT rule, there exists another pair of jobs i and j which the swap can be performed on.
This will result in an infinite set of schedules which are optimal.

As the set of optimal schedules is finite, the inverse statement must therefore be false, proving the original statement.

3 Question 3

Consider the problem $1|chains|\sum w_j(1-e^{-rC_j})$. Describe the algorithm that solves this problem and prove that it results in an optimal sequence.

$$\textbf{Let } \rho(1,\cdots,k) = \max_{l=1,\cdots,k} \frac{\sum_{j=1}^{l} w_j}{\sum_{j=1}^{l}(1-e^{-rC_j})}$$

Step 1: Evaluate $\rho(C_1, C_2, \cdots, C_n)$ where $\{C_1, C_2, \cdots, C_n\}$ is the set of all chains.

Step 2: For the chain that corresponds to the ρ value, examine the longest sub-chain that has the same ρ value.

Step 3: Append the sub-chain to the optimal sequence and return the remaining sub-chain to the set.

Step 4: Repeat **Step 1-3** until no more chains remain.

4 Question 4

Formulate the $1||\sum_j U_j$ scheduling problem where U_j is equal to 1 if job j is tardy. As an integer problem in the following three cases

(a) **Completion time decision variables and disjunctive constraints**

$$\begin{array}{lrll}
\textbf{minimize} & \sum_{j=1}^{n} Z_j & & \\
\textbf{subject to} & C_j + p_k & \leq C_k + MX_{jk} & \forall j,k \\
 & C_k + p_j & \leq C_j + M(1 - X_{jk}) & \forall j,k \\
 & C_j - d_j & \leq MZ_j & \forall j \\
 & C_j & \geq 0 & \forall j \\
 & X_{jk} & \in \{0,1\} & \forall j,k \\
 & Z_j & \in \{0,1\} & \forall j
\end{array}$$

$$\textbf{where} \quad X_{jk} = \begin{cases} 1 & \text{When } j \to k \\ 0 & \text{Otherwise} \end{cases}$$
$$Z_j = \begin{cases} 1 & \text{When job } j \text{ is tardy.} \\ 0 & \text{Otherwise} \end{cases}$$

(b) **Time-indexed decision variables**

$$\begin{array}{lrll}
\textbf{minimize} & \sum_{j=1}^{n} \sum_{t=1}^{T} Z_{jt} & & \\
\textbf{subject to} & \sum_{t=0}^{T} X_{jt} & = 1 & \forall j \\
 & \sum_{j=1}^{n} \sum_{k=\max(0,t+1-p_j)}^{t} X_{jk} & \leq 1 & \forall j,t \\
 & \sum_{t=0}^{T} X_{jt}(t - 1 + p_j - d_j) & \leq MZ_{jt} & \forall j,t \\
 & X_{jt} & \in \{0,1\} & \forall j,t \\
 & Z_{jt} & \in \{0,1\} & \forall j,t
\end{array}$$

$$\textbf{where} \quad X_{jt} = \begin{cases} 1 & \text{When job } j \text{ starts at time } t \\ 0 & \text{Otherwise} \end{cases}$$
$$Z_j = \begin{cases} 1 & \text{When job } j \text{ is tardy.} \\ 0 & \text{Otherwise} \end{cases}$$

(c) **Sequence-position decision variables**

$$\begin{array}{lrll}
\textbf{minimize} & \sum_{k=1}^{n} Z_k & & \\
\textbf{subject to} & \sum_{j=1}^{n} X_{jk} & = 1 & \forall k \\
 & \sum_{k=1}^{n} X_{jk} & = 1 & \forall j \\
 & C_{k-1}^* + \sum_{j=1}^{n} p_j X_{jk} & \leq C_k^* & \forall k \\
 & C_k^* - \sum_{j=1}^{n} d_j X_{jk} & \leq MZ_k & \forall k \\
 & C_k^* & \geq 0 & \forall k \\
 & X_{jk} & \in \{0,1\} & \forall j,k \\
 & Z_k & \in \{0,1\} & \forall k
\end{array}$$

$$\textbf{where} \quad C_k^* : \quad \text{Completion time of job at sequence position } k$$
$$X_{jk} = \begin{cases} 1 & \text{When job } j \text{ is assigned to position } k \\ 0 & \text{Otherwise} \end{cases}$$
$$Z_j = \begin{cases} 1 & \text{When job } j \text{ is tardy.} \\ 0 & \text{Otherwise} \end{cases}$$

5 Question 5

Solve the $1|r_j|\sum_j w_j T_j$ scheduling problem with time-indexed variables in AMPL for the dataset provided.

```
CPLEX 12.8.0.0: optimal integer solution within mipgap or
    absmipgap; objective 139
4270685 MIP simplex iterations
1033392 branch-and-bound nodes
absmipgap = 0.000159996, relmipgap = 1.15105e-06
wT = 139 # Objective function value

C [*] := # Completion time values
 1   2
 2  18
 3   4
 4  19
 5  10
 6   9
 7   8
 8  15
 9  25
10  12
11  33
12  30
13  37
14  22
15   1
```

A Data file

```
param: J: r p w d:=
1 1 1 7 8
2 4 3 3 15
3 0 2 4 4
4 8 1 2 18
5 5 1 2 15
6 6 1 4 11
7 0 4 4 11
8 8 3 9 16
9 4 3 2 8
10 10 2 3 14
11 11 3 1 21
12 0 5 3 10
13 14 4 1 21
14 12 3 6 21
15 0 1 8 5
```

B Model file

```
### Sets ###
set J; # set of jobs released

### Variables ###
var C{J} >= 0; # completion times
var T{J} >= 0; # tardiness
var X{J, J} binary; # or variable

### Parameters ###
param r{J}; # release time
param p{J}; # processing time
param w{J}; # weight
param d{J}; # due date
param M := sum{j in J}p[j]+max{j in J}r[j]; # large M

### Objective ###
minimize wT: sum{j in J}w[j]*T[j]; # weighted tardiness

### Constraints ###
s.t. tardiness{j in J}: C[j]<=d[j]+T[j];
s.t. release_time{j in J}: C[j]>=r[j]+p[j];
s.t. overlap1{j in J, k in J: j<k}:
   M*X[j,k]+C[j]>=C[k]+p[j];
s.t. overlap2{j in J, k in J: j<k}:
   M*(1-X[j,k])+C[k]>=C[j]+p[k];
```

C Run file

```
model pset2.mod;
option solver cplex;
data pset2.dat;
solve;
display wT;
display C;
```