Adv. Opt. HW1

Ji Yihong
100340

1. Solution

(a) $1||L_{max}$ is a special case of $1|prec|h_{max}$ where

$h_j(C_j) = C_j - d_j$, $\forall j \in \{1,2,3,4\}$

Let J be the set of job that do not proceed others, then $J = \{1, 2, 3, 4\}$ since there is no processing detail and constraints.

$$\begin{aligned}
\text{Last job} &= \operatorname*{argmin}_{l \in J} h_l\left(\sum_{j=1}^{4} p_j\right) \\
&= \operatorname*{argmin}_{l \in J} \sum_{j=1}^{4} p_j - d_l \\
&= \operatorname*{argmin}_{l \in J} 18 - d_l \\
&= \operatorname*{argmax}_{l \in J} d_l \\
&= 4
\end{aligned}$$

Apply this process iteratively, the optimal sequence is

$$1 \to 2 \to 3 \to 4$$

job	1	2	3	4
p_j	5	4	3	6
d_j	3	5	11	12
C_j	5	9	12	18
L_j	2	4	1	6

The value of the objective

$= \max\{2, 4, 1, 6\}$

$= 6$

```
In[88]:= jobset = {"Job1", "Job2", "Job3", "Job4"};
         processtime = Association[{"Job1" → 5, "Job2" → 4, "Job3" → 3, "Job4" → 6}];
         duedate = Association[{"Job1" → 3, "Job2" → 5, "Job3" → 11, "Job4" → 12}];
         lateness[schedule_] := Max[Accumulate[processtime /@ schedule] - duedate /@ schedule];
         MinimalBy[Permutations[jobset], lateness]

Out[90]= {{Job1, Job2, Job3, Job4}, {Job2, Job1, Job3, Job4}}
```

Verified by Mathematica, both

$1 \to 2 \to 3 \to 4$ and $2 \to 1 \to 3 \to 4$

yields the optimal value.

(b) Positioning job with earlier due date at the beginning would minimize the probability that due date is exceeded and avoids penalty

(c) Positioning job with smaller processing time can greedily complete as many jobs as possible within a given period and hopefully decrease penalty. But this may lead to suboptimal solution.

(d) (iii) per reasoning shown in (a).

2. Solution

The proof is done by showing there is

① A polynomial-time reduction from TSP to $1|s_{jk}|C_{max}$

② A polynomial-time reduction from $1|s_{jk}|C_{max}$ to TSP

① Given an algorithm to solve TSP, we can use it to solve $1|s_{jk}|C_{max}$ by interpretting the setting cost s_{jk} between job j and k as the distance between city j and k. We would like to minimize C_{max} which is to minimize the accumulated distance through the path. We then use the TSP to generate a sequence of city. City 0 is the start point of time and the path of city is then the sequence of jobs to be completed.

② Given an algorithm to solve $1|s_{jk}|C_{max}$, we can use it to solve TSP by removing city 0 from the set of cities. View each city as a job and the distance as clean up cost after each job. Feed this set of city into $1|s_{jk}|C_{max}$ and output a sequence of job. Add city 0 at the beginning

and the end of the sequence. We get a path to solve TSP.

Hence, since TSP and $1|S_{ik}|C_{max}$ are polynomial-time reducible to each other, they are equivalent.

3. Solution

The assignment problem is described mathematically as

$$\min \sum_i \sum_j C_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_i x_{ij} = 1 \quad , \forall \ j$$

$$\sum_j x_{ij} = 1 \quad , \forall \ i$$

$$x_{ij} \in \{0,1\} \quad \forall i, \forall j$$

Consider $1|p_j=1|\sum w_i T_j$, since $p_j=1$ is constant, we only need n time units to complete all n jobs. Equivalently, we need to assign each job to one of the time slots $\{1, 2, \ldots, n\}$.

Let $x_{ij} = \mathbb{1}\{\text{Job } i \text{ is assigned to time slot } j\}$

↑ indicator function

and

$$C_{ij} = w_j T_j$$
$$= w_i \cdot \max\{C_i - d_i, 0\}$$
$$= w_i \cdot \max\{i - d_i, 0\} \text{ since } C_i = i \text{ as } p_i = 1.$$

Hence the $1|p_j=1|\sum w_j T_j$ can be modelled as

$$\min \sum_i \sum_j w_i \cdot \max\{i - d_i, 0\} \cdot x_{ij}$$

s.t. $\sum_i x_{ij} = 1$ for $\forall j$ (one job is completed precisely at one time slot)

$\sum_j x_{ij} = 1$ for $\forall i$ (one time slot can process precisely one job at a time)

$x_{ij} = \{0, 1\}$ for $\forall i \; \forall j$

, which is an assignment problem.

4. Solution

To show that φ is a regular performance measure, it suffices to show that if $\exists S, S'$ such that $C_j^S \le C_j^{S'}$ for $\forall j \in \{1, \dots, n\}$, then $\varphi(S) \le \varphi(S')$.

If $\exists S, S'$ such that $C_j^S \le C_j^{S'}$,

$$\begin{aligned}
\varphi(C_j^S) &= \sum_{j=1}^{n} w_j U_j(C_j^S) \\
&= \sum_{j=1}^{n} w_j \cdot \mathbb{1}\{C_j^S > d_j\} \\
&\le \sum_{j=1}^{n} w_j \cdot \mathbb{1}\{C_j^{S'} > d_j\} \quad \text{since } C_j' > C_j > d_j \\
&= \sum_{j=1}^{n} w_j \cdot U_j(C_j^{S'}) \\
&= \varphi(C_j^{S'})
\end{aligned}$$

Hence $C_j^S \le C_j^{S'} \Rightarrow \varphi(C_j^S) \le \varphi(C_j^{S'})$.

If $w_k < 0$, then

$$\mathbb{1}\{C_k^{S'} > d_k\} \ge \mathbb{1}\{C_k^S > d_k\}$$

$$\not\Rightarrow \quad w_k \cdot \mathbb{1}\{C_k^{S'} > d_k\} \ge w_k \cdot \mathbb{1}\{C_k^S > d_k\},$$

and thus φ is not necessarily a regular performance measure.

5. Solution

(1) Regular.

Given φ_1 and φ_2 are regular performance measure.

If $C_j^S \le C_j^{S'}$ for $\forall\, j \in \{1, 2 \ldots, n\}$, then $\varphi_1(S) \le \varphi_1(S')$ and $\varphi_2(S) \le \varphi_2(S')$.

Let $\varphi = w_1\varphi_1 + w_2\varphi_2$

$$\begin{aligned}\varphi(S) &= w_1\varphi_1(S) + w_2\varphi_2(S) \\ &\le w_1\varphi_1(S') + w_2\varphi_2(S') \quad \text{Addition of two inequalities.} \\ &= \varphi(S')\end{aligned}$$

Hence, (1) is a regular performance measure.

(2) Not regular performance measure.

Consider $\varphi_1(S) = 1$ and $\varphi_2(S) = \sum_j C_j$, both of which are regular performance measure. If $\exists$ S and S' such that $C_j^S \le C_j^{S'}$ for $\forall j$, let $\varphi = \frac{\varphi_1}{\varphi_2}$

$$\varphi(S) = \frac{\varphi_1(S)}{\varphi_2(S)} = \frac{1}{\sum_j C_j^S} \ge \frac{1}{\sum_j C_j^{S'}} = \varphi(S'),$$ which is clearly

not a regular measure.