for $i=1, 2, \ldots, m$. Let $\overline{W}_{jis}=t_0+(a_{ij}+b_i t_0)\sum_{k=0}^{s-1}(1+b_i)^k$. (Note that unlike W_{jir}, the quantity $\overline{W}_{jis}$ is independent of n_i.) Define $y_{jis}=1$ if J_j is the sth last job processed by M_i, and $y_{jis}=0$ otherwise. Then, for $i=1, 2, \ldots, m$,

$$\sum_{s=1}^{n_i} C_{\pi(i,s)} = \sum_{s=1}^{n_i}\sum_{j=1}^{n_i} \overline{W}_{jis} y_{jis} = \sum_{s=1}^{n}\sum_{j=1}^{n} \overline{W}_{jis} y_{jis}$$

where the second equality holds because $y_{jis}=0$ when $s > n_i$. Hence, the problem can be formulated as the following $n \times nm$ 'constrained asymmetric assignment problem', where we would like to assign the n jobs to nm positions, with n positions on each machine, leaving a total of $nm-n$ positions unassigned:

$$\begin{aligned}
\text{AP}': \quad \text{minimize} \quad & \sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{s=1}^{n} \overline{W}_{jis} y_{jis} \\
\text{subject to} \quad & \sum_{j=1}^{n} y_{jis} \leqslant 1 \quad (i=1, 2, \ldots, m; \\
& \quad s=1, 2, \ldots, n) \qquad (1) \\
& \sum_{i=1}^{m}\sum_{s=1}^{n} y_{jis} = 1 \quad (j=1, 2, \ldots, n) \qquad (2) \\
& \sum_{j=1}^{n} y_{ji1} \geqslant \sum_{j=1}^{n} y_{ji2} \geqslant \ldots \geqslant \sum_{i=1}^{n} y_{jin} \\
& \quad (i=1, 2, \ldots, m) \qquad (3) \\
& y_{jis}=0 \text{ or } 1 \\
& \quad (j=1, 2, \ldots, n; \\
& \quad i=1, 2, \ldots, m; \\
& \quad r=1, 2, \ldots, n). \qquad (4)
\end{aligned}$$

In this formulation, constraints (3) ensure that on every machine, the unassigned positions must precede all assigned positions, so that $y_{jis}=1$ if and only if J_j is truly the sth last job on machine M_i.

Note that $\overline{W}_{ji1} \leqslant \overline{W}_{ji2} \leqslant \ldots, \leqslant \overline{W}_{jin}$ for all $i=1, 2, \ldots, m$ and $j=1, 2, \ldots, n$. Thus, constraints (3) can be removed from the formulation without affecting the optimal solution value of the problem. In other words, solving the following problem will provide us with an optimal solution to AP′:

$$\begin{aligned}
\text{AP}'': \quad \text{minimize} \quad & \sum_{j=1}^{n}\sum_{i=1}^{m}\sum_{s=1}^{n} \overline{W}_{jis} y_{jis} \\
\text{subject to} \quad & \text{constraints (1), (2) and (4)}
\end{aligned}$$

Note: A rigorous proof of this statement can be developed easily by considering any optimal solution to AP″ and showing that an alternative feasible solution can be created through adjusting the y_{jis} values (say, by repeatedly swapping the values of y_{jis} and $y_{j,i,s+1}$ whenever $\sum_{j=1}^{n} y_{jis} < \sum_{j=1}^{n} y_{j,i,s+1}$) so that the new solution satisfies (3) and has an objective function value no worse than the original solution.

Problem AP″ is an $n \times nm$ (unconstrained) 'asymmetric assignment problem' with n jobs and nm positions. One simple way to solve this asymmetric assignment problem is to convert it to an $nm \times nm$ assignment problem by introducing $nm-n$ dummy jobs. Solving the converted assignment problem requires $O(m^3n^3)$ time. Thus, when m is fixed, the running time of this solution method becomes $O(n^3)$, regardless what the value of m is. Therefore, this method is significantly more efficient than Kuo *et al*'s method for large-sized problems.

When m is not fixed, this solution method has a polynomial running time of $O(m^3n^3)$. This is again a significant improvement over Kuo *et al*'s method. In fact, in this case, AP″ can be solved with a lower computational complexity using a more sophisticated method. Note that an $n_1 \times n_2$ asymmetric assignment problem is the same as a minimum weighted bipartite matching problem with an underlying bipartite graph $G=(N_1 \cup N_2, A)$, where $n_1=|N_1|$ and $n_2=|N_2| > n_1$, and all the nodes in N_1 are required to be matched. This minimum weighted bipartite matching problem can be solved in $O(n_1(|A|+n_2 \log n_2))$ time using the Successive Shortest Path Algorithm (see Cheng *et al*, 1996; Ahuja *et al*, 1993, Chapter 12). Therefore, AP″ can be solved in $O(n^3m+n^2m\log(nm))$ time. This implies that when m is not fixed, problem $P/p_{ij}=a_{ij}+b_i t_{ij}/\Sigma C_j$ can be solved in $O(n^3m+n^2m\log(nm))$ time.

Problem $P/p_{ij}=a_{ij}-b_i t_{ij}/\Sigma C_j$ can also be solved with the same efficiency using the above method, except that $\overline{W}_{jis}$ is redefined as $t_0+(a_{ij}-b_i t_0)\sum_{k=0}^{s-1}(1-b_i)^k$.

References

Ahuja RK, Magnanti TL and Orlin JB (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall: Englewood Cliffs, NJ.

Cheng TCE, Chen ZL and Li C-L (1996). Parallel-machine scheduling with controllable processing times. *IIE Trans* **28**: 177–180.

Kuo W-H, Hsu C-J and Yang D-L (2008). A note on unrelated parallel machine scheduling with time-dependent processing times. *J Opl Res Soc*, advance online publication, doi: 10.1057/palgrave.jors.2602576.

Mosheiov G (2001). Parallel machine scheduling with a learning effect. *J Opl Res Soc* **52**: 1165–1169.

The Hong Kong Polytechnic University Chung-Lun Li

Reply to Chung-Lun Li

Journal of the Operational Research Society (2008), **59**, 1697. doi:10.1057/palgrave.jors.2602649

The Viewpoint proposes an improved solution for Kuo *et al*'s method. I agree with all the comments.

National Formosa University D-L Yang