Solving a Permutation Flow Shop Problem with Blocking and Transportation Delays

Jacques Carlier, Mohamed Haouari, Mohamed Kharbeche, Aziz Moukrim

Solving a Permutation Flow Shop Problem with Blocking and Transportation Delays

Jacques Carlier[1], Mohamed Haouari[2], Mohamed Kharbeche[2], Aziz Moukrim[1]

(1) UMR CNRS 6599 Heudiasyc, Centre de Recherches de Royallieu,

Université de Technologie de Compiègne,

e-mail: jacques.carlier, aziz.moukrim@hds.utc.fr

(2) ROI - Combinatorial Optimization Research Group,

Ecole Polytechnique de Tunisie, 2078 La Marsa, Tunisie

e-mail: mohamed.haouari@ept.rnu.tn, medkharbeche@yahoo.fr

Keywords: Permutation flow shop, blocking, time delays, branch-and-bound.

1 Problem definition

Given a job set $J = \{1, 2, ..., n\}$ where each job has to be processed non preemptively on m machines M_1, M_2,..., M_m in that order. The processing time of job j on machine M_i is p_{ij}. At time $t = 0$, all jobs are available at an input device denoted by M_0. After completion, each job must be taken from M_m to an output device denoted M_{m+1} (for convenience, we set $p_{m+1,j} = 0, \forall j \in J$). The transfer between machine M_i and M_k $(i, k = 0, ..., m + 1)$ is performed by means of a single robot and takes τ_{ik} units of time. The machines have no input or output buffering facilities. Consequently, after processing a job j on machine M_i $(i = 1, ..., m)$, this latter remains blocked until the robot picks j and transfers it to M_{i+1}. Such a move could only be performed if machine M_{i+1} is free (that is, no job is being processed by or is waiting at M_{i+1}). At any time, each machine can process at most one job and each job can be processed on at most one machine. Moreover, the robot can transfer at most one job at any time. The problem is to find a processing order of the n jobs, the same for each machine (because of the blocking constraint, passing is not possible), such that the time $C_{\max}$ at which all the jobs are completed (makespan) is minimized.

In the sequel, we *partially* relax the constraint requiring that the robot can transfer at most one job at any time and we propose to investigate the flow shop problem with blocking and transportation delays. This problem might be viewed as a generalization of the much studied flow shop with blocking (Ronconi, 2005). An overview of the literature on flow shop scheduling blocking can be found in Hall and Sriskandarajah (1996).

2 Lower bounds

2.1 One-machine based lower bounds

As a consequence of the transportation delays, the minimum elapsed time on machine M_i between the completion of a job j and the starting of a job k is

$$\delta_i = \tau_{i,i+1} + \tau_{i+1,i-1} + \tau_{i-1,i} \quad \forall i = 1, ..., m \tag{1}$$

Also, by setting for each job $j \in J$ and each machine M_i $(i = 1, ..., m)$:

- a head $r_{ij} = \sum_{k=1}^{i-1} p_{kj} + \sum_{k=0}^{i-1} \tau_{k,k+1}$ if $i > 1$ and $r_{1j} = \tau_{01}$,
- a tail $q_{ij} = \sum_{k=i+1}^{m} p_{kj} + \sum_{k=i}^{m} \tau_{k,k+1}$ if $i < m$ and $q_{mj} = \tau_{m,m+1}$.

Hence, a simple lower bound is

$$LB_1 = \max_{1 \le i \le m} \{ \min_{1 \le j \le n} r_{ij} + \sum_{j=1}^{n} p_{ij} + (n-1)\delta_i + \min_{1 \le j \le n} q_{ij} \} \tag{2}$$

Actually, we can derive a better bound by observing that if job h is scheduled immediately after job j, then the minimum elapsed time on machine M_i $(i = 2, ..., m)$ between the completion of j and the starting of a job h is given by

$$s_{ijh} = \max(p_{ij} + \tau_{i,i+1} + \tau_{i+1,i-1}, p_{i-1,h} + \tau_{i,i-2} + \tau_{i-2,i-1}) - p_{ij} + \tau_{i-1,i} \quad \forall i = 2, ..., m \tag{3}$$

Now, define $\delta_{ij} = \min_{h \neq j} \{s_{ijh}\}$, $\forall i = 2, ..., m, j = 1, ..., n$ ($\equiv$ minimum elapsed time after completion of j on M_i), $\delta^i_{[l]} = l^{th}$ smallest value of δ_{ij} $(j = 1, ..., n)$. Then we get the lower bound

$$LB_2 = \max_{2 \le i \le m} \{ \min_{1 \le j \le n} r_{ij} + \sum_{j=1}^{n} p_{ij} + \sum_{k=1}^{n-1} \delta^i_{[l]} + \min_{1 \le j \le n} q_{ij} \} \tag{4}$$

Clearly, a valid relaxation is a one-machine problem with heads, tails, and setup times $1 \mid r_j, q_j, s_{jk} \mid C_{\max}$. A relaxation of this problem is a $1 \mid r_j, q_j \mid C_{\max}$ obtained by setting $r'_j = r_{ij}$, $p'_j = p_j + \delta_{ij}$, and $q'_j = q_{ij} - \delta_{ij}$. Hence, a third lower bound is obtained by allowing preemption. For each machine M_i $(i = 1, ..., m)$, let $LB3_i$ denote the makespan of the corresponding optimal preemptive schedule. Then a valid lower bound is

$$LB_3 = \max_{2 \le i \le m} LB3_i \tag{5}$$

A further relaxation of $1 \mid r_j, q_j, s_{jk} \mid C_{\max}$ is obtained by setting all heads and tails to $\min_{j \in J} \{r_j\}$ and $\min_{j \in J} \{q_j\}$, respectively.

The resulting relaxation is equivalent to finding a shortest Hamiltonian path in a directed complete graph, where the nodes represent the jobs and the distance matrix is (s_{ijk}). We can transform this problem into an equivalent asymmetric traveling salesman problem (ATSP) by adding to the graph a dummy node $n+1$ and dummy zero-cost arcs $(j, n+1)$ and $(n+1, j)$, for $j = 1, ...n$. For a given machine M_i, let z^i_{ATSP} denote the value of the shortest cycle. Since solving this relaxed problem is $\mathcal{NP}$-hard, we compute a lower bound on z^i_{ATSP}. In our implementation, we have derived a tight lower bound LB^i_{ATSP} by solving an enhanced linear programming ATSP formulation which is based on assignment constraints as well as lifted Miller-Tucker-Zemlin subtour elimination constraints (See Desrochers and Laporte, 1991). The resulting lower bound is

$$LB_4 = \max_{2 \le i \le m} \{ \min_{1 \le j \le n} r_{ij} + LB^i_{ATSP} + \min_{1 \le j \le n} q_{ij} \} \tag{6}$$

2.2 A two-machine based lower bound

First, we consider the special case where $m = 2$. We shall prove that the problem is polynomially solvable. Given a permutation σ of the n jobs with a corresponding makespan $C_{\max}$, we denote by $S_{i,\sigma(j)}$ the start time on machine i $(i = 1, 2)$ of the job at position j $(j = 1, ..., n)$. The time interval $[0, C_{\max}]$ can be partitioned into $2n + 1$ sub-intervals I_1, J_1, I_2, J_2,..., I_n, J_n, I_{n+1} where

- $I_1 = [0, S_{2,\sigma(1)} - \tau_{12}]$,
- $I_j = [S_{2,\sigma(j-1)}, S_{2,\sigma(j)} - \tau_{12}]$ for $j = 2, ..., n$,
- $I_{n+1} = [S_{2,\sigma(n)}, C_{\max}]$
- $J_j = [S_{2,\sigma(j)} - \tau_{12}, S_{2,\sigma(j)}]$ for $j = 1, ..., n$

Note that :

- During each interval $I_j = [S_{2,\sigma(j-1)}, S_{2,\sigma(j)} - \tau_{12}]$ $(j = 2, ..., n)$ either machine M_1 or machine M_2 is blocked for

$$\lambda_{\sigma(j-1),\sigma(j)} = \left|(p_{2,\sigma(j-1)} + \tau_{23} + \tau_{31}) - (p_{1,\sigma(j)} + \tau_{20} + \tau_{01})\right| \tag{7}$$

  units of time

- During the interval $I_1 = [0, S_{2,\sigma(1)} - \tau_{12}]$, machine M_2 remains idle for $(\tau_{01} + p_{1,\sigma(1)})$ units of time
- During the interval $I_{n+1} = [S_{2,\sigma(n)}, C_{\max}]$, machine M_1 remains idle for $(p_{2,\sigma(n)} + \tau_{23})$ units of time

Now, for each machine M_i $(i = 1, 2)$, denote by P_i, T_i, and W_i the total processing time, transportation time from and to M_i, and waiting time, respectively. We have :

- $P_1 = \sum_{j=1}^{n} p_{1\sigma(j)}$ and $P_2 = \sum_{j=1}^{n} p_{2\sigma(j)}$
- $T_1 = \tau_{01} + (n-1)(\tau_{20} + \tau_{01}) + n\tau_{12}$ and $T_2 = n(\tau_{12} + \tau_{23}) + (n-1)\tau_{31}$
- $W_1 + W_2 = (\sum_{j=2}^{n} \lambda_{\sigma(j-1),\sigma(j)}) + (p_{2,\sigma(n)} + \tau_{23}) + (\tau_{01} + p_{1,\sigma(1)})$

Now, since $P_i + T_i + W_i = C_{\max}$ for $i = 1, 2$, we get

$$\sum_{j=1}^{n} p_{1\sigma(j)} + \sum_{j=1}^{n} p_{2\sigma(j)} + T_1 + T_2 + (\sum_{j=2}^{n} \lambda_{\sigma(j-1),\sigma(j)}) + (p_{2,\sigma(n)} + \tau_{23}) + (\tau_{01} + p_{1,\sigma(1)}) = 2C_{\max} \tag{8}$$

It follows that minimizing the makespan amounts to minimizing

$$(\sum_{j=2}^{n} \lambda_{\sigma(j-1),\sigma(j)}) + (p_{2,\sigma(n)} + \tau_{23}) + (\tau_{01} + p_{1,\sigma(1)}) \tag{9}$$

Setting, $a_j = p_{2,j} + \tau_{23} + \tau_{31}$, $b_j = p_{1,j} + \tau_{20} + \tau_{01}$ for $j = 1, ..., n$ and $a_0 = \tau_{20}$, $b_0 = \tau_{31}$. We get

$$\lambda_{jk} = |a_j - b_k|\,,\ \forall j,k = 0,...,n \tag{10}$$

The problem defined by (9) amounts to finding a permutation $\sigma = (\sigma(0), \sigma(1), ..., \sigma(n))$ with $\sigma(0) \equiv 0$ such that $\sum_{j=1}^{n} \lambda_{\sigma(j-1),\sigma(j)} + \lambda_{\sigma(n),\sigma(0)}$ is minimized. Clearly, this is a *Traveling Salesman Problem* with a Gilmore and Gomory distance matrix. This problem is solvable in polynomial time (Gilmore and Gomory, 1964). This result is a generalization of a previous similar result by Reddi and Ramamoorthy (1972) for a no-wait two-machine flow shop problem.

Clearly, if $m > 2$, then we consider a pair of consecutive machines (M_i, M_{i+1}) $(i = 1, ..., m-1)$ and we solve the corresponding two-machine problem. Let LB_5^i denote the optimal makespan. Then a valid lower bound is

$$LB_5 = \max_{1 \le i \le m-1} \{ \min_{1 \le j \le n} r_{ij} + LB_5^i + \min_{1 \le j \le n} q_{i+1,j} \} \tag{11}$$

3 Exact Branch-and-Bound

We have implemented a branch-and-bound algorithm that is based on the proposed lower bounds. We will present the results of extensive computational results on randomly generated instances that show that instances with up to 20 jobs and 4 machines can be solved to optimality.

References

[1] Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the Miller-Trucker-Zemlin subtour elimination constraints. Operations Research Letters 10, 27-36.

[2] Gilmore, P.C. and Gomory, RE. (1964). Sequencing a one state variable machine: a solvable case of the traveling salesman problem. Operations Research 12, 655-679.

[3] Hall, N.G. and Sriskandarajah, C. (1996). A survey of machine scheduling problems with blocking and no-wait in process. Operations Research 44, 510 -525.

[4] Reddi, S.S. and Ramamoorthy CV. (1972). On the flowshop sequencing problems with no-wait in process. Operational Research Quarterly 23, 323-330.

[5] Ronconi, D. P. (2005). A branch-and-bound algorithm to minimize the makespan in a flowshop with blocking. Annals of Operations Research 138, 53-65.