

On the representation of the One Machine Sequencing Problem in the Shifting Bottleneck Heuristic

Saral Mukherjee* and A. K. Chatterjee¹

Indian Institute of Management Ahmedabad, India

Indian Institute of Management Calcutta, India

The Shifting Bottleneck heuristic decomposes the Job Shop problem into a series of One Machine Sequencing Problems (OMSPs) with release and due dates, precedence constraints and the minimization of maximum lateness objective. It is well known that delayed precedence constraints may exist between two operations to be performed on the same machine. We identify a new type of precedence constraint that may exist in an OMSP between the predecessor of an operation and the successor of another. The premise that an OMSP captures the sequencing relationships on other machines in the release and due date information is not valid when such precedence constraints exist. A modification of the OMSP representation is proposed based on a generalized lateness objective defined on a due window. The implications of such a representation for the OMSP solution procedure have been explored.

Keywords: Single machine scheduling; Shifting Bottleneck heuristic; Machine-based decomposition; Due window; Generalized lateness objective

* Corresponding author e-mail: saralm@iimahd.ernet.in, saral_m@hotmail.com

¹ e-mail: ac@iimcal.ac.in

1. Introduction

The Job Shop scheduling problem is concerned with scheduling jobs J_1, J_2, \dots, J_n on machines M_1, M_2, \dots, M_m . Each job J_i consists of operations $O_{i1}, O_{i2}, \dots, O_{im}$ having processing time requirements $p_{i1}, p_{i2}, \dots, p_{im}$ respectively. These operations must be performed according to a predetermined sequence, called the machine sequence of each job. A machine cannot process more than one operation at a time. In addition, preemption of operations is not allowed. We need to decide the job sequence of each machine such that the maximum completion time of all jobs, also known as the makespan, is minimized.

The Shifting Bottleneck (SB) heuristic, developed by Adams et al. (1988), has emerged as one of the prominent approximation methods for solving the Job Shop problem. It sets up and solves a series of One Machine Sequencing Problems (OMSPs). Each OMSP focuses on a particular machine M_k and abstracts the disjunctive selections on other machines in the form of release and due dates for operations to be processed on machine M_k . Let $OP_k = \{O_1, \dots, O_n\}$ be the set of operations requiring processing on machine M_k . Each operation O_i has a release date r_i , a processing time p_i and a due date f_i . We need to determine a processing sequence for the operations such that the maximum lateness L_{\max} for the set OP_k is minimized. If C_i denotes the completion time of O_i , then its lateness L_i equals $C_i - f_i$. The OMSP is solved and the optimal sequence S_k is implemented in the original Job Shop. In this way the SB heuristic either generates the sequence S_k for machine M_k when no such sequence existed before (Schedule Generation Step) or finds a possibly new sequence for the machine M_k if it already has a sequence implemented (Rescheduling Step).

The OMSP has been shown to be equivalent to a 3 machine (M_1, M_2, M_3) flow shop problem with makespan minimization objective where machines M_1 and M_3 do not have any capacity restrictions while M_2 can process only one operation at a time. This problem is known in literature as the Head and Tails problem. Each OMSP operation O_i is equivalent to a job in the Head and Tails problem which requires r_i, p_i and q_i amount of

processing on the three machines. The values r_i and q_i denote the “head” and “tail” of O_i . If C_{\max} denotes the makespan for the Job Shop from which this OMSP data is being abstracted, then $q_i = C_{\max} - f_i$.

Carlier (1982) provided a very efficient branch and bound solution procedure for solving the OMSP when all operations are independent. Adams et al. (1988) used this procedure in the original implementation of the SB heuristic. Dauzere-Peres and Lasserre (1993) showed that a Delayed Precedence Constraint (DPC) may exist between a pair of operations. These DPCs are the result of paths in the disjunctive graph connecting operations to be scheduled on the specified machine. Subsequently, Balas et al. (1995) proved that the OMSP with DPC is NP-hard and proposed a branch and bound procedure (BLV algorithm) for solving it. This OMSP solution procedure is an integral part of one of the best solution procedures for the Job Shop problem (Balas and Vazacopulos 1998).

In this paper we identify a new type of precedence constraint that may exist in an OMSP between the predecessor of an operation and the successor of another. We show that the equivalence of the OMSP and the Head and Tails problem breaks down in the presence of such precedence constraints. A modification of the OMSP representation is proposed incorporating a generalization of the lateness function based on a due window. We further show that the OMSP solution procedure need not be altered for solving the modified OMSP.

The rest of the paper is arranged as follows. In section 2 we identify two situations when representation failures can occur. The proposed representation is presented in section 3. The implications for the overall Job Shop problem are analysed in section 4 and the concluding remarks are presented in section 5.

2. Representation of the OMSP

Consider a particular scheduling instant when all machines have been scheduled, S_k being the sequence set for machine M_k . The steps involved in setting up and solving an OMSP for machine M_k in the Rescheduling Step of the Shifting Bottleneck heuristic are enumerated below. If the OMSP is created as part of the Schedule Generation Step then all steps except Step 1 would be performed.

Procedure OMSP

- Step 1: Unschedule machine M_k by deleting all disjunctive selections corresponding to sequence S_k .
- Step 2: Determine r values for operations to be performed on M_k .
- Step 3: Determine q values for operations to be performed on M_k .
- Step 4: Determine the values of delayed precedence (if any) between operations to be performed on M_k .
- Step 5: Set up the OMSP from steps 2, 3, &4.
- Step 6: Solve the OMSP by Carlier's algorithm or BLV algorithm.
- Step 7: Set disjunctive arcs corresponding to the solution returned by the OMSP algorithm in the original Job Shop

Let the makespan for the Job Shop before and after Step 1 be C_{\max}^1 and C_{\max}^A respectively. The makespan for the unscheduled OMSP set up in Step 5 is denoted by C_{\max}^B . Then $C_{\max}^B = \max_{i \in OP_k} \{r_i + p_i + q_i\}$. As the OMSP set up for machine M_k captures the scheduling considerations on the other machines at that scheduling instant through the r and q values, for proper representation, the makespan values C_{\max}^A and C_{\max}^B have to be identical. This follows from the equivalence of the Head and Tails problem and the OMSP as shown in Adams et al. (1988). However, the following example highlights that there can exist situations where this is violated.

Example 1

Consider the data for a 3 job 3 machine Job Shop presented in Table 1. A particular scheduling instant is shown in Figure 1 where the sequences set on M_1 and M_2 are O_{11} – O_{21} and O_{12} – O_{22} – O_{32} respectively. S and T are the dummy start and finish operations. The makespan for the Job Shop is now 106.

Table 1. Data for a 3 job 3 machine Job Shop

Operation	Immediate Job Predecessor	Duration	On Machine
O_{11}	-	2	M_1
O_{12}	O_{11}	3	M_2
O_{21}	-	4	M_1
O_{22}	O_{21}	5	M_2
O_{31}	-	100	M_3
O_{32}	O_{31}	6	M_2

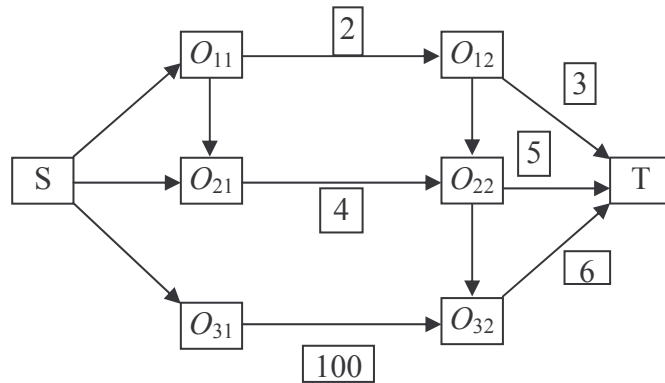


Figure 1. A particular scheduling instant

Suppose now that we choose M_1 for rescheduling. Step 1 of the rescheduling step is equivalent to deleting the directed arc O_{11} – O_{21} . The makespan for the Job Shop before and after the deletion of this arc are $C_{\max}^1 = 106$ and $C_{\max}^A = 106$ respectively. Since there is no delayed precedence relationship between O_{11} and O_{21} the OMSP data for M_1 is as shown in Table 2.

Table 2. OMSP data for M_1

Operation	r_i	p_i	q_i
O_{11}	0	2	14
O_{21}	0	4	11

Thus $C_{\max}^B = \max_i \{r_i + p_i + q_i\} = 16$. Hence $C_{\max}^B \neq C_{\max}^A$. The reason for the difference between the two makespan values stems from the fact that the processing time of operation O_{31} is not reflected in the release or delivery time data for any operation to be processed on M_1 . Hence the one machine abstraction does not remain faithful to the original Job Shop situation.

This representation failure will occur whenever there exists an operation O^* such that (i) it is neither a predecessor nor a successor (immediate or otherwise) of any operation scheduled on machine M_k and (ii) O^* belongs to a critical path in the original digraph. This situation will not arise in the commonly used test problems in literature, such as those provided by Applegate and Cook (1991), Fisher and Thomson (1963) and Lawrence (1984); as every job in those test problems have operations to be performed on every machine. However, in real life applications, it is conceivable that a particular job requires processing only on a sub set of the available machines in the Job Shop.

A representation failure can also occur when the operations of the type O^* do not exist. We provide such a situation in the next example.

Example 2

Consider the data for a 2 job 2 machine Job Shop presented in Table 3. A particular scheduling instant is shown in Figure 2 where the sequences set on M_1 and M_2 are O_{11} – O_{22} and O_{12} – O_{21} respectively. The makespan for the Job Shop is now 211. Hence $C_{\max}^1 = 211$. If we now reschedule M_2 , the data for the OMSP is as presented in Table 4.

Table 3. Data for a 2 job 2 machine Job Shop

Operation	Immediate Job Predecessor	Duration	On Machine
O_{11}	-	100	M_1
O_{12}	O_{11}	6	M_2
O_{21}	-	5	M_2
O_{22}	O_{21}	100	M_1

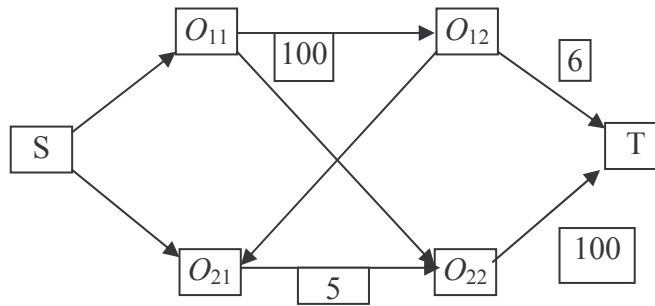


Figure 2. An example of a 2 job 2 machine Job Shop

Table 4. OMSP data for machine M_2

Operation	r_i	p_i	q_i
O_{12}	100	6	0
O_{21}	0	5	100

Clearly, $C_{\max}^A = 200$ while $C_{\max}^B = 106$, and hence $C_{\max}^B \neq C_{\max}^A$. Thus a representation failure can happen even if operations of the type discussed in the previous example are absent.

The two examples show the existence of a precedence relationship in the original Job Shop between the predecessor of one operation and the successor of another operation such that both the operations are to be performed on the same machine. For example, such a

relationship exists in Example 2 between O_{11} , the job predecessor of operation O_{12} and O_{22} , the job successor of operation O_{21} . Due to this precedence relationship, the operation O_{22} cannot start until time 100. This relationship is not captured by the current OMSP representation. Similarly, in Example 1, operation O_{32} cannot start before time 100. Hence a precedence relationship, not captured by the OMSP representation, exists between S, the job predecessor of operations O_{11} and O_{21} ; and O_{32} , the successor (not immediate) of O_{11} and O_{21} .

One possible way of overcoming the representation failure could be to introduce a dummy operation O^δ to be processed on selected machine M_k , having a release time of C_{\max}^A and zero processing time and q value. With this representation, the condition $C_{\max}^B = C_{\max}^A$ is always guaranteed. The OMSP data for machine M_2 in Example 2 would then be as shown in Table 5. The disjunctive graph representation of the equivalent 3 machine Head and Tails problem is shown in Figure 3.

Table 5. Modified OMSP data

Operation	r_i	p_i	q_i
O_{12}	100	6	0
O_{21}	0	5	100
O^δ	200	0	0

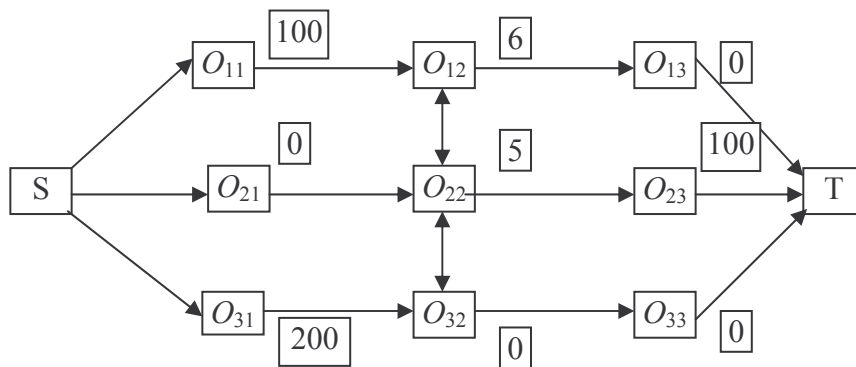


Figure 3. Disjunctive Graph of equivalent Head and Tails problem

It may be noted that the introduction of the dummy operation does not complicate the solution procedure for the OMSP as its processing time is zero. Once the OMSP is solved, the dummy operation may be ignored in subsequent steps. Thus the introduction of the dummy operation will let us avoid the creation of an OMSP representation which does not exactly mirror the prevailing Job Shop, without any detrimental effect on the OMSP solution procedure.

However, a modification of the OMSP representation using a dummy operation fails to capture the additional information regarding start time of operation O_{22} in Example 2. We could have accommodated the precedence relationship between O_{11} and O_{22} in Example 2 as a directed arc between the corresponding operations O_{11} and O_{23} on M_1 and M_3 respectively in the Head and Tails problem. However, precedence relationships of the type presented in Example 1 would require the use of a new Head and Tails formulation consisting of 4 machines. In general such an approach would become clumsy, as precedence relationships dictated by disjunctive constraints can exist between any two operations in a Job Shop. The existence of such precedence relationship implies that the Head and Tails problem ceases to be an equivalent way of representing the sequencing decisions faced by the OMSP. Hence the need arises for a new representation of the OMSP which captures the additional information.

3. Proposed Representation

In the OMSP representation, the lateness of an operation O_i is calculated as $L_i = C_i - f_i$ where C_i and f_i denote the completion time and due date for O_i . The Lateness Function (LF) is plotted in Figure 4 to depict how the lateness of the operation varies with completion time.

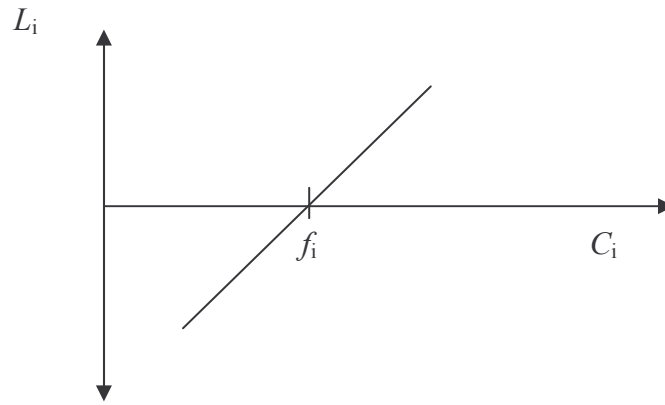


Fig. 4. The Lateness Function

As we have shown in the previous section, the LF may not adequately represent the lateness of all operations in the OMSP. For certain operations O_i it may be worthwhile to replace the due date with a due window. The left extremity (LE) of this due window is denoted as f_i^L . The right extremity of the due window is the original due date f_i . For such operations, $L_i = (f_i^L - f_i)$ for all $C_i \leq f_i^L$ and $L_i = C_i - f_i$ otherwise. The plot of the Modified Lateness Function (MLF) depicts the situation in Figure 5.

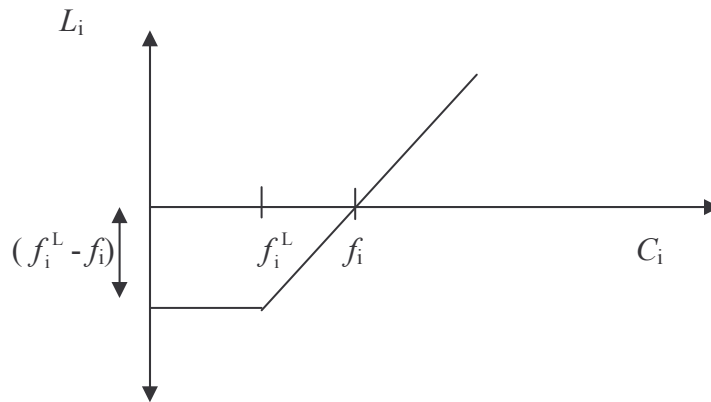


Fig. 5. The Modified Lateness Function (MLF)

We divide the set OP_k of operations to be performed on machine M_k into mutually exclusive and exhaustive sets O_1 and O_2 such that the LF (MLF) adequately represents

the lateness of all operations in set O_1 (O_2). The lateness of an operation O_i can then be computed as $L_i = C_i - f_i$ if $O_i \in O_1$; $\max \{(f_i^L - f_i), (C_i - f_i)\}$ otherwise.

The MLF can be seen as a generalization of the LF where $f_i^L = 0, \forall O_i \in O_1$; $f_i^L > 0$ otherwise. Hence we call this modification of the classical lateness function as the generalized lateness function L_i^G . Thus $L_i^G = \max \{(f_i^L - f_i), (C_i - f_i)\}, \forall O_i$. Consequently, $L_{\max}^G = \max_i L_i^G$. The OMSP is thus reclassified as a $1|r_j|L_{\max}^G$ problem.

We are in effect proposing that instead of a due date based objective function, a due window based objective function is more suitable for the OMSP. Due window based objective functions have been used by researchers in Earliness-Tardiness scheduling problems (Liman et. al. 1998, Yeung et. al. 2001). In such problems, an activity is penalized if it does not finish inside the due window. No penalty accrues if completion time falls within the due window. However, for the generalized lateness function L_i^G , the lateness of an operation continues to vary inside the due window.

Let $w_i = f_i - f_i^L$ represent the length of the due window for O_i and let T_i^L represent the tardiness measure using the Left Extremity f_i^L as the due date. Then $T_i^L = L_i^G + w_i; \forall O_i$. In case $w_i = w, \forall O_i$; $T_i^L = L_i^G + w; \forall O_i$. Hence $T_{\max}^L = L_{\max}^G + w$. Thus the MLF can be visualized as a tardiness objective with a shifted origin if the length of the due window is same for all operations. A solution procedure for the OMSP can then be to use solution procedures for the $1|r_j|T_{\max}$ problem with due dates defined as f_i^L values. It is highly unlikely that $w_i = w, \forall O_i$. A counterexample is presented in Example 3 to highlight that the $1|r_j|T_{\max}$ problem with due dates defined as f_i^L values would not correctly represent the MLF in general.

Example 3

Consider the data for an OMSP presented in Table 6

Table 6. A 2-operation OMSP data

Operation	r_i	p_i	f_i	f_i^L
A	0	2	5	2
B	0	2	4	3

The optimal sequence for the $1|r_j|T_{\max}$ problem, with due dates defined as f_i^L values, is A-B. However, the optimal sequence under the $1|r_j|L_{\max}^G$ formulation is B-A. Hence the $1|r_j|T_{\max}$ problem, with due dates defined as f_i^L values is not equivalent to the $1|r_j|L_{\max}^G$ formulation if the length of the due window is not same for all operations.

The computation of f_i^L values for all activities in an OMSP can be effected by including the following steps in the Procedure OMSP described earlier.

```

Step 4a:   begin
            for each OMSP operation  $O_i$ ;
             $f_i^L \leftarrow 0$ ;
            for each successor operation  $O_j$  of  $O_i$ ;
            if  $L(S, j) - L(i, j) + p_i > f_i^L$ ;
             $f_i^L \leftarrow L(S, j) - L(i, j) + p_i$ ;
            end
    
```

Here $L(i, j)$ denotes the longest path in the Job Shop problem between two operations O_i and O_j . Since release time of operation O_j is $r_j = L(S, j)$, nothing is gained if operation O_i is completed before $L(S, j) - (L(i, j) - p_i)$. Note that the $L(S, j)$ values had already been

calculated during the forward pass calculations in Step 2 of the Procedure OMSP. Further, all successor activities of operation O_i need to be visited in Step 4 of Procedure OMSP to check whether delayed precedence constraints exist. The $L(i, j)$ values can be calculated during this search. Hence the determination of the left extremity values can be integrated with the work involved in Steps 2 and 4 of existing Procedure OMSP. Thus the additional work involved in the determination of left extremity values is minimal.

Example 4

For the 3 job 3 machine Job Shop presented earlier in Table 1, the left extremity (LE) values for operations O_{11} and O_{21} are calculated as follows:

$$\begin{aligned} LE(O_{11}) &= \max \{0, (L(S, O_{12}) - p_{11} + p_{11}), (L(S, O_{22}) - p_{11} - p_{12} + p_{11}), (L(S, O_{32}) - p_{11} - p_{12} - p_{22} + p_{11})\} \\ &= \max \{0, (2), (5 - 3), (100 - 3 - 5)\} \\ &= 92 \end{aligned}$$

$$\begin{aligned} LE(O_{21}) &= \max \{0, (L(S, O_{22}) - p_{21} + p_{21}), (L(S, O_{32}) - p_{21} - p_{22} + p_{21})\} \\ &= \max \{0, (5), (100 - 5)\} \\ &= 95 \end{aligned}$$

Lemma 1: $f_i^L \leq f_i, \forall O_i$

Proof: Note that $f_i^L = 0, \forall O_i \in O_1$. Hence $f_i^L \leq f_i, \forall O_i \in O_1$ since $f_i > 0, \forall O_i$. For operation $O_i \in O_2$, let operation O_j be the immediate successor of O_i such that O_j is constrained to start only on or after f_i^L . Then $f_i^L = L(S, j)$. If C_{\max} be the makespan of the Job Shop,

$$\begin{aligned} L(S, j) + L(j, T) &\leq C_{\max} \\ \Rightarrow L(S, j) &\leq C_{\max} - L(j, T) \end{aligned}$$

$$\Rightarrow L(S, j) \leq C_{\max} - q_i$$

$$\Rightarrow L(S, j) \leq f_i$$

$$\Rightarrow f_i^L \leq f_i \quad \blacksquare$$

For any given sequence S , let $L_{\max}^G(S)$ represent the maximum lateness under the proposed $1|r_j|L_{\max}^G$ representation. The maximum lateness for any sequence S under the existing $1|r_j|L_{\max}$ representation of is denoted as $L_{\max}(S)$.

Lemma 2: For any sequence S ,

$$(i) L_i \leq L_i^G, \forall O_i$$

$$(ii) L_i = L_i^G \Leftrightarrow L_i^G = C_i - f_i$$

$$(iii) L_{\max}(S) \leq L_{\max}^G(S)$$

$$(iv) L_{\max}(S) = L_{\max}^G(S) \Leftrightarrow L_i^G = L_{\max}^G(S) = C_i - f_i$$

Proof: Follows from definitions of L_i and L_i^G . ■

Lemma 3: Let $L_i^G = L_{\max}^G(S)$. Then S is optimal for $1|r_j|L_{\max}^G$ if $L_i^G = f_i^L - f_i$.

Proof: Otherwise there exists sequence S' such that $L_{\max}^G(S') < f_i^L - f_i$. This is impossible since $L_i^G \geq (f_i^L - f_i)$ by definition. ■

Theorem 1: If S is optimal sequence for $1|r_j|L_{\max}$ then S is also optimal for $1|r_j|L_{\max}^G$.

Proof: Otherwise there exists sequence S' such that $L_{\max}^G(S') < L_{\max}^G(S)$. Let $L_i^G = L_{\max}^G(S)$. Then either $L_i^G = f_i^L - f_i$ or $L_i^G = C_i - f_i$.

Case I: $L_i^G = f_i^L - f_i$. Then using Lemma 3, S is also optimal for $1|r_j|L_{\max}^G$ which is a contradiction.

Case II: $L_i^G = C_i - f_i$. Then using Lemma 2, $L_i^G = L_{\max}(S)$ and $L_{\max}(S') \leq L_{\max}^G(S')$. Hence $L_{\max}(S') \leq L_{\max}^G(S') < L_{\max}^G(S) = L_{\max}(S)$
 $\Rightarrow L_{\max}(S') < L_{\max}(S)$

which is a contradiction since S is an optimal sequence for $1|r_j|L_{\max}$. ■

The converse of Theorem 1 is not true as is shown in Example 5.

Example 5

Consider the data for an OMSP presented in Table 7.

Table 7. A 2-operation OMSP data

Operation	r_i	p_i	f_i	f_i^L
A	0	2	5	4
B	0	2	4	4

Both A-B and B-A are optimal sequences for the $1|r_j|L_{\max}^G$ problem. However, only sequence B-A is optimal under the $1|r_j|L_{\max}$ formulation. Hence it is not necessary that an optimal sequence for the $1|r_j|L_{\max}^G$ problem is also an optimal sequence for the $1|r_j|L_{\max}$ problem.

Theorem 1 implies that we do not need to modify the current OMSP solution procedures in order to arrive at a solution for the new representation. The BLV algorithm, originally designed for the $1|r_j|L_{\max}$ problem, would be an exact solution procedure for the $1|r_j|L_{\max}^G$ problem. The extra information available in the $1|r_j|L_{\max}^G$ formulation then seems to be of no practical use for the purpose of optimally sequencing the OMSP.

However, as we show in the next section, the extra information may be of use for the solution of the overall Job Shop problem.

4. Implications of information loss for the overall Job Shop problem

Consider the data in Example 2 where the current OMSP representation for machine M_2 fails to take into account the information that operation O_{22} cannot start before 100. Can this information loss while scheduling the OMSP result in a degradation of solution quality for the overall Job Shop? The following example shows that such a situation can indeed arise.

Example 6

Consider the data for a 4 job 4 machine Job Shop presented in Table 8. A particular scheduling instant is shown in Figure 6 where the sequences set on M_1 , M_2 , M_3 and M_4 are $O_{11}-O_{22}$, $O_{21}-O_{31}-O_{12}$, $O_{41}-O_{23}$ and $O_{42}-O_{32}$ respectively. If we choose to reschedule machine M_2 , the OMSP data is presented in Table 9.

Table 8. Data for a 4 job 4 machine Job Shop

Operation	Immediate Job Predecessor	Duration	On Machine
O_{11}	-	100	M_1
O_{12}	O_{11}	6	M_2
O_{21}	-	60	M_2
O_{22}	O_{21}	99	M_1
O_{23}	O_{22}	1	M_3
O_{31}	-	40	M_2
O_{32}	O_{31}	99	M_4
O_{41}	-	60	M_3
O_{42}	O_{41}	50	M_4

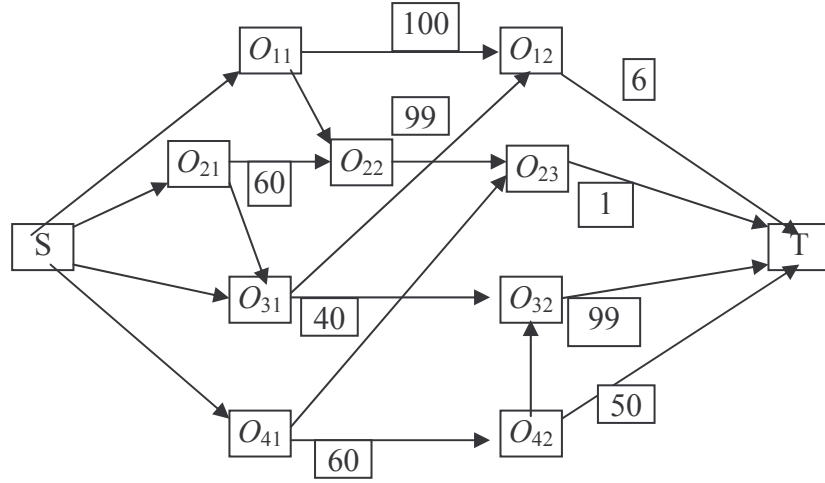


Figure 6. A particular scheduling instant for the 4 job 4 machine Job Shop

Table 9. OMSP data for machine M_2

Operation	r_i	p_i	q_i
O_{12}	100	6	0
O_{21}	0	60	100
O_{31}	0	40	99

Both Carlier's algorithm and BLV algorithm would return the sequence $O_{21}-O_{31}-O_{12}$ with $L_{\max} = 0$. Setting the sequence $O_{21}-O_{31}-O_{12}$ on M_2 would result in a makespan of 209 for the Job Shop. However, if we had supplied the additional information that operation O_{22} cannot start before 100, a possible sequence could have been $O_{31}-O_{21}-O_{12}$ resulting in a makespan of 209 for the Job Shop. While the two sequences return the same makespan, the differences crop up in the next iteration of the SB heuristic. If the $O_{21}-O_{31}-O_{12}$ sequence is implemented, the SB heuristic will settle down in a local optima as rescheduling any machine would not improve the makespan. On the other hand if sequence $O_{31}-O_{21}-O_{12}$ is implemented, we can reschedule machine M_4 in the next iteration leading to a makespan of 200 as shown in Figure 7. This is because sequence $O_{31}-O_{21}-O_{12}$ provides more opportunities for scheduling operation O_{32} earlier.

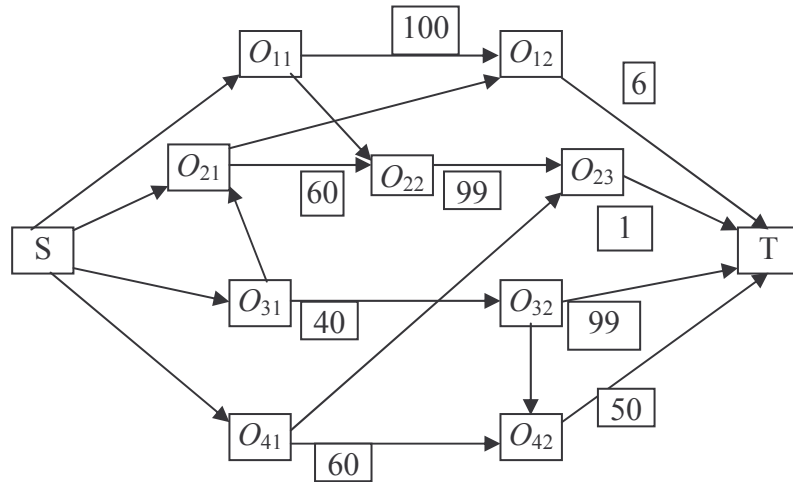


Figure 7. Status after rescheduling machine M_4

Several neighbourhood structures are known for the Job Shop problem (Blazewicz et. al. 1996). All such neighbourhood structures involve moves defined on operations on the critical path in the digraph. Note that the critical path $S-O_{41}-O_{42}-O_{32}-T$ in Figure 6 does not pass through any operation on M_2 . Thus the current neighbourhood structures do not allow a move to sequence $O_{31}-O_{21}-O_{12}$. Example 6 clearly demonstrates that a neighbourhood structure involving resequencing operations *not* on the critical path may be of value.

Thus the information contained in the $1|r_j|L_{\max}^G$ representation has the potential of being profitably used in solving the overall Job Shop problem even though it is little consequence for the optimal solution of the OMSP. However, a solution procedure designed keeping this information in mind may have to focus on more than one machine. Such a procedure can still be a machine based decomposition procedure, but instead of a One Machine Sequencing Problem we may have to deal with a Two Machine Sequencing Problem or a higher version.

5. Conclusion

The Shifting Bottleneck heuristic has emerged as one of the prominent approximation procedures for solving the Job Shop problem. The OMSP lies at the core of the SB heuristic. In this paper we have presented two examples to highlight a lacuna in the current representation of the OMSP and have proposed an improved representation. We have identified a new type of precedence constraint that may exist in an OMSP between the predecessor of an operation and the successor of another. In the presence of such constraints, the current OMSP representation does not capture all the sequencing relationships on other machines in the release and due date information. A modified OMSP representation has been proposed with a generalized lateness function based on the concept of a due window. We have proved that the current optimal solution procedures for the OMSP would also be optimal for the OMSP under the proposed representation. Further research can be aimed at proper utilization of the information contained in the $1|r_j|L_{\max}^G$ formulation for better schedules for the Job Shop problem.

References

- J. Adams, E. Balas and D. Zawack, "The Shifting Bottleneck Procedure for Job Shop Scheduling," *Management Science*, 34, 391-401, 1988.
- D. Applegate and W. Cook, "A Computational Study of Job shop Scheduling," *ORSA J. Computing*, 3 (2), 149-156, 1991.
- E. Balas, J.K. Lenstra and A. Vazacopoulos, "The One Machine Sequencing Problem with Delayed Precedence Constraints and its Use in Job Shop Scheduling," *Management Science*, 41, 94-109, 1995.

E. Balas and A. Vazacopoulos, "Guided Local Search with Shifting Bottleneck for Job Shop Scheduling," *Management Science*, 44, 262-275, 1998.

J. Blazewicz, W. Domscke, and E. Pesch, "The job shop scheduling problem: Conventional and new solution techniques", *European Journal of Operational Research*, 93, 1-33, 1996.

J. Carlier, "The One Machine Sequencing Problem," *European Journal of Operational Research*, 11, 42-47, 1982.

S. Dauzere-Peres and J.B. Lasserre, "A modified shifting bottleneck procedure for job-shop scheduling", *International Journal of Production Research*, 31, 923-932, 1993.

H. Fisher and G. L. Thomson, "Probabilistic Learning Combinations of Local Job Shop Scheduling Rules," in J.F. Muth, G. L. Thomson (eds), *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs, NJ, 1963.

S. Lawrence, Supplement to "Resource Constrained Project Scheduling: An experimental investigation of heuristic scheduling techniques", GSIA, Carnegie-Mellon University, 1984.

S.D. Liman, S.S. Panwalkar and S. Thongmee, "Common due window size and location determination in a single machine scheduling problem", *Journal of the Operational Research Society*, 49, 1007-1010, 1998.

W.K. Yeung, C. Oguz and T.C. Edwin Cheng, "Single-machine scheduling with a common due window", *Computers and Operations Research*, 28, 157-175, 2001.