A SELECTING METHOD OF REASONABLE NON-DOMINATED SCHEDULES ON OPEN SHOP

Sangsu Han

Received December 21, 2008

Abstract. In this research, a selection method of non-dominated schedules is investigated on m machines open shop scheduling problem with maximum completion times for each of machines as multiobjective. In formulation of the problem, we introduce a concept of schedule vector and lexicographic ordering based on burdened machine. For this problem, the aim is to find a preemptive schedule that lexicographically minimizes the maximum completion times $Cmax_i,\ i = 1, 2, \cdots, m$ on m machines M_i, respectively. We propose a flexible algorithm and discuss its validity and computational complexity.

1 Introduction In this paper, we consider multiobjective scheduling problem (**GLSP**: Generalized Lexicographical Scheduling Problem) on m machine open shop with maximum completion times for each of machines as multiobjective. For this problem, the aim is to find a preemptive schedule that lexicographically minimizes the maximum completion times $Cmax_i,\ i = 1, 2, \cdots, m$ on m machines M_i, respectively. in [1], S.S.Han and H.Ishii analyzed problems on two and three machines. As a special case of vector formulation, we deal with the lexicographic formulation by introducing schedule vector. We utilize lexicographical ordering[2] to compare one schedule vector with the other. In order to solve our problem **GLSP**, we propose a solution procedure (**GLSA**:Generalized Lexicographical Scheduling Algorithm) which utilizes the max-min matching on level[2] weighted bipartite graph for constructing schedule and base on the similar idea to T.Gonzalez and S.Sahni [3].

Section 2 is devoted to the preliminary involving the notations and terminology, and formulate our problem. In section 3 and 4, we propose a solution procedure **GLSA** for solving our problem, and discuss its validity and computational complexity.

2 Formulation of problem The **GLSP** dealt with is specified as follows :

(1) There are m machines $M_1, M_2, \cdots, M_m$ and n jobs $J_1, J_2, \cdots, J_n$ each of which consists of m operations.

(2) Each job J_j has m processing times p_{ji} corresponding to operation on M_i, respectively.

(3) All jobs are open shop type, i.e., the processing order of m operations of each job is not specified. So processing of each job can be started on any machine.

(5) The objective is to find a schedule which minimizes the maximum completion times on m machines lexicographically.

We consider a following bipartite graph $G = (J \bigcup M, E)$. $J = \{J_1, J_2, \cdots, J_n\}$ is a set of jobs to be processed on a set of machines $M = \{M_1, M_2, \cdots, M_m\}$. $E = \{e_{j1} \mid t_{j1} \neq 0\} \bigcup \{e_{j2} \mid t_{j2} \neq 0\} \bigcup \cdots \{e_{jm} \mid t_{jm} \neq 0\}$ $(j = 1, 2, \cdots, n)$ is a set of edges, where p_{ji} denotes

2000 *Mathematics Subject Classification.* Primary 90B35; Secondary 90C27.
Key words and phrases. open shop, scheduling, multi-objective, DEA.

the processing time of operation of job J_j on machine M_i. We associate p_{ji} with edge e_{ji} and let $P(J_j)$ denote sum of processing time on edges incident to node J_j. So sorting different $P(J_j)$, let $0 \doteq L_0 < L_1 < L_2 < \cdots < L_t$ and define $L_h = \{J_j \mid L_h = P(J_j), j = 1, 2, \cdots, n, h = 1, 2, \cdots, t\}$. If $L_p > L_q (p, q = 1, 2, \cdots, t, p \neq q)$, then a job belongs to L_p has a higher level than a job belongs to L_q. When levels are associated with all jobs on above bipartite graph, we call this level bipartite graph[1]. Further we associate the weight w_{ji} with edges e_{ji} on the level bipartite graph and call this **level edge-weighted bipartite graph**. That is, (p_{ji}, w_{ji}) is associated with edge e_{ji} on the level edge-weighted bipartite graph. Let X be a **Modified Max-Min matching** from M to J defined as below : Given an level edge-weighted bipartite graph, X is the maximum-cardinality matching for which the minimum of weights of the edges with respect to w_{ji} in the matching is maximum. And we define μ as follows :

$$\mu = \min\{p_{ji} \mid e_{ji} \in X, j = 1, \cdots, n, i = 1, \cdots, m\}. \tag{1}$$

By T. Gonzalez and S. Sahni., optimal value $Cmax^*$ of the maximum completion time is

$$Cmax^* = \max\{T_i = \sum_{j=1}^{n} p_{ji}, \max_j(\sum_{i=1}^{m} p_{ji})\} \tag{2}$$

Under the above setting, our problem is formulated as follows :

$$\textbf{GLSP}: \left| \begin{array}{ll} Lex \min & Lex \max(Cmax_1^\pi, \cdots, Cmax_m^\pi) \\ \text{subject to} & Cmax_1^\pi, \cdots, Cmax_m^\pi \leq Cmax^*, \ \pi \in \Pi. \end{array} \right.$$

Where, π denotes set of schedules.

¿From above $Cmax^*$, our problem can be divided into two cases. One case is $Cmax^* = T_i (i = 1, 2, \cdots, m)$. We call this **Case-1**. The other case is $Cmax^* = \max_j(\sum_{i=1}^{m} p_{ji})$ as **Case-2**.

3 Solution procedure Case-1: By giving $L_h (h = 1, 2, \cdots, t)$ as the level to all jobs, construct a level bipartite graph G. For considering lexicographical scheduling problem with respect to $Cmax_1$,$Cmax_2$,$\cdots$,$Cmax_m$, introduce vector T with $T_i (i = 1, 2, \cdots, m)$ as its components. By $Lex \max$ transform T_i, lexicographically the greatest vector ($T_{\pi(1)}$,$T_{\pi(2)}$,$\cdots$, $T_{\pi(m)}$) can be found, where $T_{\pi(1)} \geq T_{\pi(2)} \geq \cdots \geq T_{\pi(m)}$ and $\pi(i)$ is the corresponding permutation. Based upon this order, give weight $w_{\bar{j}\bar{i}} = mn - n(i-1) - (j-1)$ on edge e_{ji} on graph G, where $\bar{j}(\bar{i})$ denotes the j-th(i-th) job index of remaining job(machine) of $J(M)$. For this w_{ji}, find the max-min matching X from M_i to J_j using the **modified max-min matching algorithm**. Let this matching be $X = \{e_1, e_2, \cdots, e_k\}$ with maximum cardinality $|X| = k (\leq \min\{m, n\})$. Next, compute μ as described in expression (1) of section 2. The jobs incident to the edges $e_1, e_2, \cdots, e_k$ are scheduled on their respective machines for a time period of μ, and the processing time of at least one edge is deleted(i.e. the processing time of at least one edge becomes zero). By scheduling a job on its respective machine we mean that if (J_j, M_i) is one of the edges in the matching, then job J_j is processed on machine M_i for μ units of time. This process is repeated until all edges are deleted. Now we are ready to give an efficient algorithm **GLSA** minimizing maximum completion time on each machine and discuss the validity and computational complexity of **GLSA**.

Algorithm GLSA

Step 0 : Construct level bipartite graph $G = (J \bigcup M, E)$ with $n + m$ nodes.

Step 1 : Let $T = (T_1, T_2, \cdots, T_m)$ and $Lex\max$ transform of T, $T = (T_{\bar{1}}, T_{\bar{2}}, \cdots, T_{\bar{m}})$ where $T_{\bar{1}} \geq T_{\bar{2}} \geq \cdots \geq T_{\bar{m}}$.

Step 2 : Based on $\bar{T}$, determine processing sequence of machines.

Step 3 : Based on the sequence of machines, construct level edge-weighted bipartite graph $\bar{G}$ so that $w_{\bar{j}\bar{i}} = mn - n(i-1) - (j-1)$ associated with e_{ji}.

Step 4 : Execute subalgorithm **modified max-min matching algorithm**.

Step 5 : For X, compute μ by expression (1) in section 2 and transform an amount of μ to schedule with respect to machines and jobs corresponding.

Step 6 : Reduce an amount of μ from processing time of edges belong to X. As a result of reducing, at least one edge is deleted. Let E be set of remaining edges.

Step 7 : If $E = \emptyset$, that is ,all edges are deleted, then Stop. Otherwise, let remaining graph be G, return to step 3.

modified max-min matching algorithm

Step 4.0: For w_{ji} on $\bar{G}$, set $X = \emptyset$,the value of threshold $W = +\infty$ and $\sigma_j = -\infty$ for each node $j \in J$. No nodes are labeled.

Step 4.1: lexicographical labeling

S 4.1.1: Give the label $\emptyset$ to each exposed node in M.

S 4.1.2: If there are no unscanned labels, go to step 4.3. If there are unscanned labels, but each unscanned label is on node i in J for which $\sigma_i < W$, then set $W = \max\{\sigma_i \mid \sigma_i < W\}$.

Step 4.1.3: Find a node i with an unscanned label such that
case $i \in M$: $\max\{i \mid P(i)\}$ based on T,
case $i \in J$: $\max\{i \mid P(i)\}$ based on Job's level,
where either $i \in M$ or else $i \in J$ and $\sigma_i \geq W$. If $i \in M$, go to step 4.1.4; if $i \in J$, go to step 4.1.5.

S 4.1.4: Scan the label on node $i(i \in M)$ as follows. For each edge $e_{ji} \notin X$ incident to i, if $\sigma_j < w_{ji}$ and $\sigma_j < W$, then give node j the label i and set $\sigma_j = w_{ji}$. Return to step 4.1.2.

S 4.1.5: Scan the label on node $i(i \in J)$ as follows. If node i is exposed, go to step 4.2. Otherwise, identify the unique edge $e_{ji} \in X$ incident to node i and give node j the label i. Return to step 4.1.2.

Step 4.2: Augmentation
An augmenting path has been found, terminating at node i. The nodes preceding node i in the path are identified by backtracing from label to label. Augment X by adding to X all edges in the augmenting path that are not in X, and removing from X those which are. Remove all labels from nodes. Set $\sigma_j = -\infty$, for each node j in J. Return to step 4.1.1.

Step 4.3: Hungarian Labeling
No augmenting path exists, and the matching X is a max-min matching of maximum cardinality. Let $L \subseteq M \cup J$ denote the set of labeled nodes. Let $e_{j'i'} \in X$ be such that$w_{j'i'} = \min\{w_{ji} \mid e_{ji} \in X\}$.

In step 3 of **GLSA**, w_{ji} denotes a priority such that machine M_i may choose job J_j. Since w_{ji} is based on lexicographical order considering from the machine side, always $w_{jk} > w_{jl}(T_k > T_l, k \neq l)$. For w_{ji}, the modified max-min matching X in step 5 give the assignment such that the most burden machine with respected to remaining processing times choose job with the highest level as possible. Concretely, in step 4.1.2, we can control it under the T and job's level. Consequently we have to show that this assignment will lexicographically minimize the completion time on each machine. That is, in order to prove the validity of **GLSA**, we have to show the existence of schedule with completion time $Cmax^*$ and schedule such that corresponding schedule vector is lexicographically minimized.

Theorem 1

For our problem of finding lexicographically minimize completion time on each machine, **GLSA** finds an optimal preemptive schedule in the sense of $\bar{T}$.

proof : In order to prove the validity of **GLSA**, we have to show the existence of schedule with the completion time $Cmax^*$ and the lexicographical minimization. First of all, we consider bipartite graph $G' = (J' \bigcup M', E')$ to prove that the existence of schedule with the completion time $Cmax^*$. $J'(= J \bigcup J'')$ is a set of jobs to be processed on a set of machines $M'(= M \bigcup M'')$, where $J''(M'')$ represents $m(n)$ fictitious jobs(machines). $E(= E \bigcup E_1 \bigcup E_2 \bigcup E_3)$ is a set of edges between J' and M'. Now a set of edges E_1 connecting J to M'' is added in such a way that $P(J_j) = Cmax^*, 1 \leq j \leq n$.

$$E_1 = \{e_{jm+j} \mid t_{j,m+j} = Cmax^* - P(J_j) \neq 0\}$$

A set of edges E_2 is included to connect M to J'' in such a way that $P(M_i) = Cmax^*, 1 \leq i \leq m$.

$$E_2 = \{e_{n+ii} \mid t_{n+i,i} = Cmax^* - P(M_i) \neq 0\}$$

Finally, a set of edges E_3 connecting J'' to M'' is added to make $P(J_j) = P(M_i) = Cmax^*, j = n+1, \cdots, n+m,\ i = m+1, \cdots, n+m$. One may easily verify that E_3 can be so constructed. Furthermore, if we suppose identical Lex max transform and weight w_{ji}, without of generality, we may construct a level edge-weighted bipartite graph $\bar{G}'$ connecting $\bar{G} = \{J \bigcup M, E\}$ to $\bar{G}'' = \{J'' \bigcup M'', E_3\}$ by E_1,E_2.

Now under above setting, in order to prove that the existence of schedule with the completion time $Cmax^*$, we have to show that there is a complete matching X on $\bar{G}(\subset \bar{G}')$ at each iteration by **GLSA**. From [4], since it is clear that there is always a complete matching X' with cardinality $n+m$ on $\bar{G}'$ at each iteration, it is sufficient to show that we may product X' involving X on constructing G'. Let the max-min matching X with cardinality $k(\leq \min\{n,m\})$ be X_{JM} for $\bar{G}$. By augmenting edges in $E_1(E_2)$ from $n - k(m-k)$ exposed nodes to $M''(J'')$, we may obtain matching $X_{JM''}(X_{MJ''})$ with cardinality $n-k(m-k)$. Accordingly, total cardinality $|X_{JM}| + |X_{JM''}| + |X_{MJ''}| = m+n-k$ is obtained. For remained $X_{J''M''}$ with cardinality k, we can easily find it from the property of connecting E_3 mentioned above.

Consequently, since a matching X' of size $n+m$ involving the max-min matching X can be found at each iteration, all $n+m$ machines are kept busy at all times(either processing real or fictitious jobs). The total processing time needed is $\sum_{1}^{n+m} P(M_i) = (n+m)Cmax^*$. Hence the maximal completion time of the schedule is $(n+m)Cmax^*/(n+m) = Cmax^*$.

Under the assumption that there is a schedule with completion time $Cmax^*$, it is sufficient to consider the precedence relation of two max-min matchings X_1 with cardinality

p and X_2 with cardinality q corresponding any schedules. If $p > q$,then it is clear that X_1 precedes X_2 from max-min matching. If $p = q$, then precedence relation depends on minimum weight w_1 and w_2 of X_1 and X_2, respectively. If $w_1 > w_2$, then X_1 precedes X_2. It is not difficult to prove this using the property, since the weight w_{ji} is based on lexicographical order considering from the machine side, always $w_{jk} > w_{jl}(T_k > T_l, k \neq l)$. If $w_1 = w_2$, then one with greater sum of weights precedes the other.

Theorem 2

The time complexity of **GLSA** is $O(rm^2n^2)$, where n is the number of jobs, m is the number of machines, and r is the number of nonzero operations. r is equal to $\max\{n, m\}$.

proof : Since each time a max-min matching is found one edge is deleted, all matchings have to be found at most $O(r)$ times. Hence the maximum number of preemptions for machines is $O(r)$. For a max-min matching, the number of W in step 4.1 does not exceed the number of distinct edge weights, i.e., mn. For each W, the augmentation computation is $O(mn)$. Thus, the procedure of finding a max-min matching takes $O(m^2n^2)$. Consequently, the computing time of **GLSA** is $O(rm^2n^2)$

4 Solution procedure Case-2:

For the case of $Cmax^* = \max_j(\sum_{i=1}^{m} t_{ki})$, if there is J_l with sufficiently great processing time less than J_k, it is not easy to find 2^m nondominated solutions with completion time $Cmax^*$.

Then for the Case-2, we shall find not the nondominated solutions but an optimal solution to lexicographically minimize maximum completion time on each machine as Case-1. It is not trivial from results of Case-1. The Case-2 can be reduced to Case-1 whose $Cmax^*$ is $T_i' = T_i + P(J_d)$, simply, by supposing dummy job J_d and edge (J_d, M_i) with $t_{di} = \min\{Cmax^* - T_i, i = 1, 2, 3\}$ as processing time and $w_{di} = 0$.

Hence we can construct level edge-weighted bipartite graph with $n + m + 1$ nodes and may obtain the solution by solving **GLSA**. Under above setting, we can obtain following theorem.

Theorem 3

For our problem of finding lexicographically minimize completion time on each machine, by supposing J_d, **GLSA** finds an optimal preemptive schedule.

proof : It is clear from theorem 1.

Without Considering burdened machine, importance ranking of machines depends on decisin maker in general. So, we suppose that z non-dominated schedules $\pi(= 1, 2, \cdots, z)$ are obtained by the above assumption and m elements $Cmax_y(y = 1, 2, \cdots, m)$ of schedule vector V are ranked in decreasing order of importance of machines by decision maker. The problem(**SM**) to select reasonable non-dominated schedules is likely to obtain overall desirability index $\theta_{\pi_o\pi_o}$ for each non-dominated schedule π_o and is formulated as follows.

$$(3) \qquad \textbf{SM}: \left| \begin{array}{lll} \theta_{\pi_o\pi_o}(\epsilon) = & \text{Maximize} & \sum_{y=1}^{m} w_y Cmax_{\pi_o y} \\ & \text{subject to} & \sum_{y=1}^{m} w_y Cmax_{\pi y} \leq 1 \\ & & (\pi = 1, 2, \cdots, z) \\ & & w_y - w_{y+1} \geq d(y, \epsilon) \\ & & (y = 1, 2, \cdots, m-1) \\ & & w_s \geq d(m, \epsilon) \end{array} \right.$$

Where, w_y, $d(y, \epsilon)$ and $Cmax_{\pi y}$ denote weight reflecting importance objective, a nonnegative function to be nondecreasing in ϵ(discrimination intensity function) and objective $Cmax_y$'s value for schedule π, respectively.
Note that this problem is equivalent to the well known DEA-Assurance Region Model. See [3].

5 Conclusion We proposed a selection method to find reasonable non-dominated schedules based on lexicofraphical ordering and Cook and Kress's voting model on open shop. As remained subject, another multiobjective problem on shop and parallel environment can be considered.

Acknowledgement

The author would like to express his sincere appreciation to Professor Hiroaki Ishii of Osaka University for his enthusiastic discussion and persistent encouragement.

References

[1] S.S.Han: Scheduling Algorithm for Minimizing Number of Machines on Identical Parallel Environment with Restricted Processing Time, *International Journal of Information for Logistics and Management*, **Vol. 3**, No. 2, pp.89-94 (2008)

[2] S.S.Han, H,Ishii, S.Fujii and T.Nishida: Lexicographical Scheduling Problem on Three Machine Open Shop, *Sci. Math. Japonicae* , **Vol. 53**, pp.1001-1014 (1991)

[3] W.D.Cook and M.Kress: A Data Envelopment Model for Aggregating Preference Rankings, *Management Science*, **Vol. 3**, pp.1302-1310 (1990)

[4] T. Gonzalez and S.Sahni: Open shop Scheduling to Minimize Finish Time, *JACM*, **Vol. 23**, pp.665-679 (1976)

Department of Information Design, Faculty of Contemporary Social Studies
sugi 3-50-1, hirakata, Osaka 573-0192, JAPAN

E-mail : han@inf.oiu.ac.jp