数理解析研究所講究録 1252 巻 2002 年 197-203

Some Fuzzy Resource Constrained Scheduling Problems

Kanthen K Harikrishnan* and Hiroaki Ishii**

* INTI College, Malaysia
** Department of Applied Physics, Graduate School of Engineering, Osaka University, Yamada-oka, Suita, Osaka 565-0871, Japan

Abstract : We consider the following bi-criteria scheduling problem first.
(1) There are two identical machines (**or m identical machines**) and n jobs to be processed by either of these two machines. Each machine processes at most one job at a time and each job is processed on at most one machine at a time.
(2) There exists the set of resources and for each resource fuzzy bound is given which limits the total amount of the resource available at any given time. That is, available limit of each resource is flexible and is represented by the membership function which reflects the satisfaction degree of available limit for the resource.
(3) For the processing of each job, unit processing time of either machine and unit of the resource is required. That is, two jobs whose resource required sum is not over available limit are processed simultaneously.
(4) Under above setting, we consider two objectives, i.e., minimum satisfaction degree of the fuzzy bounds to be maximized and maximum completion time C_{max} to be minimized. That is, we optimize the limit vector and corresponding schedule to this vector. Usually we cannot optimize both objectives at a time and so we seek some non-dominated solutions.

Secondly we fuzzify the requirement of resource and it is assumed to be a fuzzy. Under this further fuzzified model, we seek non-dominated solutions based on the result of the first problem.

(5) We generalize the above problem to m identical machines.

1. Introduction

Any task, besides processors, may require for it's processing some additional scarce resources. In a scheduling model resources are notified as resource types, resource limits and resource requirements. There exists a tradition of works which have tried to apply fuzzy set theory in scheduling. Fuzzy constraints offer a very flexible method to devise composite and realistic objective functions.

In this paper, we introduce fuzzy constraints about available limits of resources to the problem considered by Garey Johnson [2]. Section 2 formulates a bi-criteria scheduling problem and Section 3 proposes solution procedure for the problem. Next in Section 4 we further assume resource requirements are fuzzy numbers and we investigate this problem and seek non-dominated solutions based on the result of the first problem. Section 5 generalizes to m identical machines (introducing a fuzzy constraint about available limits of resources to the problem considered by Blazewicz et al [1]). Continued by definitions and solution procedure for the generalized model in Section 6 and 7. Finally drawing a conclusion in Section 8.

2. Problem formulation

We consider the following bi-criteria scheduling problem first.

1. There are two identical machines M_1, M_2 and n jobs $\{J_1, J_2, \cdots, J_n\}$ to be processed by either of these two machines. Each machine processes at most one job is processed on at most one machine at a time.
2. There exists the set of resources $\{R_1, R_2, \cdots, R_s\}$ and for each resource R_j, fuzzy bound $\hat{B}_j$ is represented by the following membership function which reflects the satisfaction degree of available limit B_j for each resource R_j.

$$\mu_j(B_j) = \begin{cases} 1 & (B_j \le L_j) \\ 1 - \dfrac{B_j - L_j}{U_j - L_j} & (L_j < B_j < U_j) \\ 0 & (B_j \ge U_j) \end{cases}$$

3. For the processing of each job J_i, unit processing time of either machine and r_{ij} unit of the resource R_j is required. That is, two jobs J_i and J_k satisfying $r_{ij} + r_{kj} \le B_j, j = 1, \cdots, s$ are processed simultaneously.
4. Under above setting, we consider two objectives, i.e., minimum satisfaction degree μ_{min} of the fuzzy bounds for the simultaneously processed job pairs to be maximized and maximum completion time C_{max} to be minimized. That is, we optimize the limit vector $\mathbf{B} = (B_1, B_2, \cdots, B_s)$ and corresponding schedule π to this vector. Usually we cannot optimize both objectives at a time and so we seek some non-dominated solutions defined as follows. Solution $(\mathbf{B}^1, \pi_1)$ dominates solution $(\mathbf{B}^2, \pi_2)$ means:

$$min_j \mu_j(B_j^1) \ge min_j \mu_j(B_j^2),\ C_{max}(\pi_1) \le C_{max}(\pi_2)$$

and at least one inequality hold without equality. Solution $(\mathbf{B}, \pi)$ is called non-dominated if there exists no solution that dominates $(\mathbf{B}, \pi)$.

3. Solution procedure for the first problem

We calculate all pair-wise sum
$r_j^{ik} = r_{ij} + r_{kj}, i, k = 1, \cdots, n, i < k, j = 1, \cdots, s$ and corresponding minimum satisfaction degrees $\mu^{ik} = min_j \mu_j(r_j^{ik})$. Sorting all $0 < \mu^{ik} < 1$ in a non-increasing order, let the result be as follows:

$$\mu^0 \equiv 1 > \mu^1 > \cdots > \mu^m > \mu^{m+1} \equiv 0$$

where m is the number of different μ_j^{ik} between 0 and 1.

[Solution Algorithm 1]

Step 1: Set $\mathbf{B}^0 = (L_1, L_2, \cdots, L_s)$ and find π_0 by solving corresponding ordinary problem P_0 using the solution procedure of Garey and Johnson [2]. Then, set $DS = \{(\mathbf{B}^0, \pi_0)\}$ and $l = 1$. Go to step 2.

Step 2: Calculate $B_j^l = \mu_j^{-1}(\mu^l), j = 1, \cdots, s,$ solve corresponding ordinary problem P_l

under the resource limit $\mathbf{B}^l = (B_1^l, \cdots, B_s^l)$ and obtain corresponding schedule π_l.
If $(\mathbf{B}^l, \pi_l)$ is not dominated by any other solution in DS, let $DS = DS \cup \{(\mathbf{B}^l, \pi_l)\}$ and go to step 3. Otherwise go to step 3 directly.

Step 3: Set $l = l + 1$. If $l = m + 1$, terminate (DS is the set of some non-dominated solutions). Otherwise return to Step 2.

Now we briefly survey the algorithm by Garey and Johnson Algorithm [2].

Garey and Johnson Algorithm

Begin

Construct an n node (undirected) graph G with each node labeled as a distinct job and each edge joining J_i to J_k if and only if $r_{ij} + r_{kj} \le B_j^l, j = 1,2,\cdots,s$;
Find a maximum matching M of graph G;
Put the minimum value of schedule length $C_{max}^* = n - |M|$;
Process in parallel the pairs of jobs joined by the edges comprising set M and process other jobs individually;

End;

Note that a maximum matching M of graph G is the subset of edges satisfying the following conditions;

(1) at most one edge in M is connected to each node,
(2) the cardinality $|M|$ is maximum

and both jobs corresponding to those nodes connecting to each matching edge can be processed at a time.

Now we can show the following complexity of the algorithm 1.

Theorem 1

Using the algorithm in Even and Kariv [1] for the maximum matching, Algorithm 1 finds some non-dominated solutions in at most $O(n^{4.5})$ computational time if $s \le n^{2.5}$.

Proof: (validity) Validity is clear from the above discussion.

(Complexity) The number m is $O(n^2)$ and for each fixed bound $\mathbf{B}$, corresponding ordinary problem can be solved in $O(n^{2.5})$ computational time by using the algorithm in [2]. Determination of all μ^{ik} takes at most $O(n^{4.5})$ since the number of r_j^{ik} is $O(n^{4.5})$, that of $\mu_j(r_j^{ik})$ also $O(n^{4.5})$ and each μ^{ik} is minimum among $O(s) \le O(n^{2.5})$ number r_j^{ik} if $s \le n^{2.5}$. Sorting these μ^{ik} takes at most $O(m \log m) = O(n^2 \log n)$. So the complexity of algorithm 1 is at most $O(n^{4.5})$. Q. E. D.

4. Further fuzzified problem

Now we further fuzzified the problem and consider the case that each resource requirement is a fuzzy number $\tilde{r}_{ij}$. Then from the extension principle [3], each pair-wise sum $\tilde{r}_{ij} + \tilde{r}_{kj}$ is also fuzzy number and denoted by $\tilde{r}_j^{ik}$. Here fuzzy number $\tilde{A}$ is defined to be a fuzzy set whose membership function $\mu_{\tilde{A}}(x)$ satisfies the following conditions (for details, see [3]):

1. (normality) There exists a unique c such that $\mu_{\tilde{A}}(c) = 1$.
2. (convexity) Each α between 0 and1, α level set $A_\alpha = \{x \mid \mu_{\tilde{A}}(x) \geq \alpha\}$ is convex, that is, one continuous interval.
3. (upper semi-continuity) Membership function $\mu_{\tilde{A}}(x)$ is upper semi-continuous.

The agreement index $\iota(\tilde{r}_j^{ik}, \hat{B}_j)$ of a fuzzy number $\tilde{r}_j^{ik}$ with regard to the fuzzy set $\hat{B}_j$ is defined to be a ratio, that is, (area surrounded by x axis, the membership function $\mu_j(x)$ of $\hat{B}_j$ and that of $\tilde{r}_j^{ik}$) / (area surrounded by x axis and membership function of $\tilde{r}_j^{ik}$) $\in [0, 1]$. Further the agreement index $\iota(\tilde{r}_{ij}, \hat{B}_j)$ of a fuzzy number $\tilde{r}_{ij}$ with regard to the fuzzy set B_j is defined to be a ratio, (area surrounded by x axis and $\mu_{\tilde{r}_{ij}}(x)$) / (area surrounded by x axis and $\mu_{\tilde{r}_{ij}}(x)$) very similarly and assumed to be 1. We calculate $h^{ik} \equiv min_j \iota(\tilde{r}_j^{ik}, \hat{B}_j), i, k = 1, \cdots, n, i < k$. Now we construct h graph $G^h = (V, E^h)$ as follows: $V = (J_1, J_2, \cdots, J_n)$ is the job set and $E^h = \{(J_i, J_k) \mid h^{ik} \geq h, i, k = 1, 2, \cdots, i < k\}$.

For the G^h, we calculate the maximum cardinality matching by using the algorithm in [1]. Then we find the schedule that minimizes C_{max} under the condition "minimal agreement index h_{min} among the simultaneously processed job pairs is not less than h by using the algorithm in [2]. Under above setting we seek the non-dominated schedules with respect to h to be maximized and C_{max} to be minimized since again, there may not the unique solution optimizing both objectives at a time. Then as Algorithm 1, we obtain the following algorithm.

[Algorithm 2 for the second problem]

Step 1: Sorting all h^{ik} such that $0 < h^{ik} < 1$, let the result be

$$h_0 \equiv 1 > h_1 > \cdots > h_t > h_{t+1} \equiv 0$$

where t is the number of different h^{ik} such that $0 < h^{ik} < 1$. Set l =0 and $DS = \phi$. Go to step 2.

Step 2: Construct h_l graph and calculate the maximum cardinality matching for h_l graph. From this maximum cardinality matching, construct the schedule π^l with minimum $C_{\max}\ value = C_{\max}^l$. If there exist a schedule $\pi^k \in DS$ such that $C_{\max}^k \leq C_{\max}^l$,then go to Step 3 directly. Otherwise, set $DS = DS \cup \{\pi^l\}$ and go to Step 3.

Step 3: Set l=l+1. If l=t+1, terminate. Otherwise return to Step 2.

Theorem 2

Algorithm 2 finds some non-dominated schedules in at most $O(n^{4.5})$ computational time if each agreement index can be calculated in a constant time and $s \leq n^{2.5}$.

Proof: Validity is clear from the above discussion and the proof of complexity is similarly done t Algorithm 1. Q. E. D.

5. Generalized Problem

We consider the following bi-criteria scheduling problem.

1. There are m identical machines $M_1, M_2, \cdots, M_m$ and n jobs $\{J_1, J_2, \cdots, J_n\}$ to be processed by any of these machines. Each machine processes at most one job at a time and each job is processed on at most one machine at a time.
2. There exists the set of resources $\{R_1, R_2, \cdots, R_s\}$ and for each resource R_j, fuzzy bound $\hat{B}_j$ which gives the total amount of the resources available at any given time. That is, available limit of each resource is flexible and $\hat{B}_j$ is represented by the following membership function which reflects the satisfaction degree of available limit B_j for each resource R_j.

$$\mu_j(B_j) = \begin{cases} 1 & (B_j \le L_j) \\ 1 - \dfrac{B_j - L_j}{U_j - L_j} & (L_j < B_j < U_j) \\ 0 & (B_j \ge U_j) \end{cases}$$

3. For the processing of each job J_i, unit processing time of either machine and r_{ij} unit of the resource R_j is required where r_{ij} is non-negative integer. That is, subset of jobs $\{J_{i1}, \cdots, J_{ip}\}$ satisfying $\Sigma_{\ell=1}^{p} r_{ij} \le B_j$, $p \le m$, $j = 1, \cdots, s$ can be processed simultaneously.
4. Under the above settings, we consider two objectives, i.e., minimum satisfaction degree μ_{min} of the fuzzy bounds for the simultaneously processed job subset to be maximized and maximum completion time C_{max} to be minimized. That is, we optimize the limit vector $\mathbf{B} = (B_1, B_2, \cdots, B_s)$ and corresponding schedule π to this vector.

6. Non-dominated Schedule and Elementary Instances

Usually we cannot optimize both objectives at a time and so we seek some non-dominated solutions defined as follows.

Solution ($\mathbf{B}^1$, π_1) dominates solution ($\mathbf{B}^2$, π_2) means:

$$min_j \mu_j(B^1{}_j) \ge min_j \mu_j(B^2{}_j) \;,\; C_{max}(\pi_1) \le C_{max}(\pi_2)$$

and at least one inequality hold without equality. Solution ($\mathbf{B}$, π) is called non-dominated if there exists no solution that dominates ($\mathbf{B}$, π).

Further we define elementary instances. For this purpose, first we divide the jobset into k classes $K_1, K_2, \cdots, K_k$ according to the resource requirement vector job J_i, $\mathbf{r} = (r_{i1}, r_{i2}, \cdots, r_{is})^T$, $i = 1, \cdots, n$ where T denotes the transpose and k the number of different resource requirement vectors. That is,

all jobs in the class K_ℓ have the same resource requirement vector $\bar{\mathbf{r}}^\ell = (\bar{r}_{\ell 1}, \bar{r}_{\ell 2}, \cdots, \bar{r}_{\ell s})^T$, $\ell = 1, 2, \cdots, k$. Let $n_\ell = |K_\ell|$ and $\mathbf{v} = (n_1, n_2, \cdots, n_k)^T$.

An elementary instance is defined to be a subset of jobs $\mathbf{J}_t = \{J_{t1}, J_{t2}, \cdots, J_{tp_t}\}$ satisfying the resource limit conditions,

$\Sigma_{\ell=1}^{p_t} r_{t\ell j} \le B_j$, $j = 1, \cdots, s$ and it is denoted by the corresponding elementary vector $\mathbf{b}_t = (b_{t1}, \cdots, b_{tk})^T$ where b_{tq}, $q = 1, 2, \cdots, k$ is the number of jobs in $\mathbf{J}_t$ belonging to the class K_q .

Vector $\mathbf{b}_\ell$ is enough to describe the elementary instance since all jobs in each class is considered to be the same identical job. Note that job set $\{J_1, J_2, \cdots, J_n\}$ is a union of elementary instances and each schedule is constructed from elementary instances since all jobs which are executed at the same time form an elementary instance.

For fixed resource limits, maximum elementary instances defined as follows are to be considered.

Elementary instance $\bar{\mathbf{b}} = (\bar{b}_1, \bar{b}_2, \cdots, \bar{b}_k)^T$ is called to be maximal if there exists no elementary instance $\mathbf{b}' = (b'_1, \cdots, b'_k)^T$ satisfying $b'_\ell \geq \bar{b}_\ell, \ell = 1,2,\cdots,k$ and at least one equality holds without equality.

For fixed available limit vector $\bar{\mathbf{B}} = (\bar{B}_1, \cdots, \bar{B}_\ell, \cdots, \bar{B}_s)$, let maximum elementary instances be $\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \cdots, \bar{\mathbf{b}}_{\bar{I}}$ where $\bar{I}$ is the number of maximum elementary instances. By using maximum elementary instances, we formulate the C_{max} minimization problem $\mathbf{P}_C$ when available limit of each resource is fixed.

$\mathbf{P}_C$: Minimize $\sum_{\ell=1}^{\bar{I}} e_\ell$

subject to $\sum_{\ell=1}^{\bar{I}} e_\ell \bar{\mathbf{b}}_\ell \geq \mathbf{v}$, $e_1, e_2, \cdots, e_{\bar{I}}$: nonnegative integers

$\mathbf{P}_C$ can be solved in a linear time with respect to input size of $\mathbf{P}_C$ if upper limit of $\bar{I}$ is fixed and applying the result due to Lenstra et al [5].

7. Solution Procedure

Since resource requirements are nonnegative integers, nonnegative integer available of each resource is enough to be considered. That is,

$$B_j = L_j, L_j + 1, \cdots, U_j - 1, \quad j = 1,2,\cdots,s$$

Let $\mu_{jq} = \mu_{\tilde{B}_j}(L_j + q)$, $q = 1,2,\cdots,U_j - L_j - 1$, $j = 1,2,\cdots,s$. First sorting all $0 < \mu_{jq} < 1$ in a non-increasing order, let the result be as follows:

$$\mu^0 \equiv 1 > \mu^1 > \cdots > \mu^a > \mu^{a+1} \equiv 0$$

where a is the number of different μ_{jq} between 0 and 1.

Now we are ready to describe our algorithm.

[Solution Algorithm]

Step 1: Set $\mathbf{B}^0 = (L_1, L_2, \cdots, L_S)$ and find π_0 by solving corresponding C_{max} minimization problem $\mathbf{P}_C^0$ using the solution procedure of Lenstra [5]. Then set $DS = \{(\mathbf{B}^0, \pi_0)\}$ and $\ell = 1$. Go to Step 2.

Step 2: Calculate $B_j^\ell = \lfloor \mu_j^{-1}(\mu^\ell) \rfloor$, $j = 1,2,\cdots,s$, solve corresponding C_{max} minimization problem P_C^ℓ under the resource limit $\mathbf{B}^\ell = (B_1^\ell, \cdots, B_s^\ell)$ and obtain corresponding schedule π_ℓ where $\lfloor \bullet \rfloor$ means the grateset integer not greater than $\bullet$. If $(\mathbf{B}^\ell, \pi_\ell)$ is not dominated by any other solution in DS, let $DS = DS \cup \{(\mathbf{B}^\ell, \pi_\ell)\}$ and go to Step 3. Otherwise go to Step 3 directly.

Step 3: Set $\ell = \ell + 1$. If $\ell = a + 1$, terminate (DS is the set of some non-dominated solutions). Otherwise return to Step 2.

Theorem 3

Using the algorithm in Lenstra [5], our algorithm finds some non-dominated solutions in at most

$$O(max\{n, log\sum_{j=1}^{s}(U_j - L_j)\} \times \sum_{j=1}^{s}(U_j - L_j))$$

competion time if upper limit of elementary instances of each C_{max} minimization problem P_C^ℓ is fixed constant.

Proof: Validity is clear from the above discussion and the proof of complexity is as follows:
Sorting $0 < \mu_{jq} < 1$ takes at most

$$O(max\{n, log\sum_{j=1}^{s}(U_j - L_j)\} \times \sum_{j=1}^{s}(U_j - L_j))$$

computational time. The number of P_C^ℓ is at most $\sum_{j=1}^{s}(U_j - L_j)$ and each problem P_C^ℓ can be solved in at most $O(n)$ using the algorithm in Lenstra et al [5] if upper limit of the number of elementary instances is fixed for . Q. E. D.

8. Conclusion

In this paper, we have investigated two machine scheduling (and generalized m identical machines) problem with fuzzy resource constraints and proposed solution algorithms for two problems. But these algorithms are straightforward and so we should make refinements of them. Our approaches to fuzzified scheduling models are relatively new. We should endeavor to peruse this direction to other classical scheduling models with resource constraints and construct more actual schedules applicable to real situations.

Acknowledgements

This research was supported by JSPS Ronpaku Program No. VCC-10144.

References

[1] Even, S. and O. Kariv, "An $O(n^{2.5})$ algorithm for maximum matching in general graphs", *System Symposium on Foundations of Computer Sciences, IEEE*(1975) 100-112.

[2] Garey, M. R. and D. S. Johnson. "Complexity results for multiprocessor scheduling under resource constraint", *SIAM J. Computing* 4 (1975) 397-411.

[3] Kaufmann, A. and M. M. Gupta, *Introduction to Fuzzy Arithmetic: Theory and Applications*(1985) Van Nostrand Reinhold, New-York.

[4] Blazewicz J., Lenstra J. K. and Rinnooy Kan A. H. G (1983) Scheduling subject to resource constraints: classification and complexity *Discrete Applied Mathematics* **5**, 11-24.

[5] Lenstra H. W., Jr. (1983) Integer Programming with a fixed number of variables *Mathematics of Operations Research* **8**, 538-548.