数理解析研究所講究録
1015 巻 1997 年 200-204

Parallel Machines Scheduling with Resource Dependent Processing Times

Shintaro Mohri (毛利　進太郎) Osaka University
Teruo Masuda (益田　照雄) Tezukayama University
Hiroaki Ishii (石井　博昭) Osaka University

1 Introduction

Scheduling problems in real production are constrained some resources like man-power, room, fund and energy etc. And scheduling problems with resource dependent job parameters can be found in many practical settings [2, 3]. The study of scheduling problems with resource dependent job processing times was initiated by Vickson [4]. Most research has focused on single machine problem. This paper deals with parallel machines problem with a consumption of resources. Each job has a processing time which is a linear decreasing function of the amount of a common resource allocated to the job, and a due dates. The objective is to find resource allocation so as the deadlines are satisfied and total weighted resource consumption is minimized.

2 Sahni's algorithm

At first, we consider the following problem without consumption of resources. There are n independent jobs and 2 identical parallel processors. Each job J_i becomes available for processing at time zero, has a deadline d_i. Processing each job J_i is required the time p_i A job can be preempted at any time and can be resumed immediately or later on another machine. If there exits feasible schedule (i.e. Processing all jobs can be finished before its deadline d_i) then we can construct the schedule from using Sahni's algorithm [3]. We analyze the simpler version of it. But it is equivalent to original one without number of preemption and order of machines.

We assume $d_1 \leq d_2 \leq \cdots \leq d_n$ without loss of generality. Let $M_1(i), M_2(i)$ be the complete point of processing J_i on M_1, M_2 respectively.

Algorithm 1.

Step 0: Let $M_1(0) \leftarrow 0, M_2(0) \leftarrow 0, i \leftarrow 1,$ and all jobs J_i are not assigned.

Step 1: If $p_i \leq d_i - M_1(i-1)$ then Job $J_i$is processed on machine M_1 in time interval $(M_1(i-1), M_1(i-1)+p_i]$, and $M_1(i) \leftarrow M_1(i-1)+p_i$, go to Step 3.

If $p_i > d_i - M_1(i-1)$ then Job $J_i$is processed on machine M_1 in time interval $(M_1(i-1), d_i]$, and $M_1(i) \leftarrow d_i$, go to Step 2.

Step 2: Job $J_i$is processed on machine M_2 in time interval $(M_2(i-1), M_2(i-1)+p_i-(M_1(i)-M_1(i-1))]$, let $M_2(i) = M_2(i-1)+p_i-(M_1(i)-M_1(i-1))$, go to Step 3.

Step 3: Let $i = i+1$, go to Step 1.

This algorithm constructs the feasible schedule if and only if there exits feasible schedule. However if there exits no feasible schedule then it constructs infeasible schedule that a job is processed both M_1 and M_2 at a time. We obtain the following theorem from considering these situations.

Theorem 1 *There exits feasible schedule if and only if following inequality is hold.*

$$M_1(i) \leq M_2(i-1)(i = 1, \cdots, n) \tag{1}$$

Proof *Each job is assigned feasibility on machine M_1, If a due date d_i is too early to process the job J_i feasibly then the processing of the job J_i on machine M_2 overlaps that on machine M_1. Therefore if the equation (1) is hold for all jobs then the processing of the job J_i on machine M_2 does not overlap that on machine M_1 and there exits feasible schedule.* □

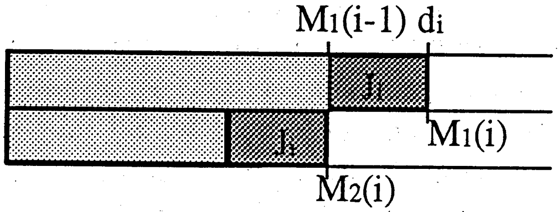

Fig 1. feasible case

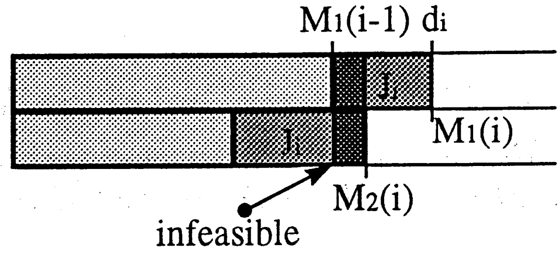

Fig 3. infeasible case

3 Resource minimize problem

At first we describe the detail of the problem. There are n independent jobs and 2 identical parallel processors. Each job J_i becomes available for processing at time zero, has a weight w_i, a deadline d_i and a resource dependent processing time

$$p_i = b_i - a_i x_i$$

where b_i is the normal processing time of job J_i which can be compressed by an amount of $a_i x_i$ if x_i units of a resource are allocated to this job and a_i is the unit processing time compression for job J_i. There is a limit on the amount x_i of the resource which can be allocated to job J_i

$$0 \leq x_i \leq \frac{b_i}{a_i}$$

A job can be preempted at any time and can be resumed immediately or later on another machine. The objective is to find resource allocation so as the deadlines are satisfied and total weighted resource consumption $\sum_{i=1}^{n} w_i x_i$ is minimized.

We modify Sahni's algorithm to solve this problem. We assume $d_1 \leq d_2 \leq \cdots \leq d_n$ without loss of generality. Let $M_1(i), M_2(i)$ be the completion time of processing J_i on M_1, M_2 respectively. And let $u_j = (M_2(i-1) - M_2(j-1))/a_j$, and C is the list of jobs nominated for assignment of resource with priority order.

The basic idea is follows. In previous section, We obtain the condition that there exits feasible schedule. If equation (1) is hold for all jobs then no consumption of resource is required. If equation (1) is not hold for Job J_i then it is necessary that we assigned resource to the jobs included the partial schedule from Job J_1 to J_i to hold the equation (1). For assignment of resource, following theorem hold.

Theorem 2 *For partial schedule from J_1 to J_i, Let $M_2'(i)$ be completion time of processing job J_i on machine M_2 with assignment of resource amount of $x_j = u_j$*

for a job $J_j (1 \leq j \leq i)$. And let $M_2''(i)$ be completion time of processing job J_i on machine M_2 with assignment of resource amount of $x_j > u_j$ for a job $J_j (1 \leq j \leq i)$.
Then following equation hold.

$$M_2'(i) = M_2''(i)$$

Proof *When $u_j = 0$, job J_j and all successor of job J_j are not processed on machine M_2. Therefore theorem obviously hold. When $u_j > 0$, The value of u_j decrease to 0 by assignment of resource $x_j = u_j$, This case is similar to above case. Therefore theorem hold.* □

From this theorem, when a job J_i in a partial schedule is infeasible, it is useless that the assignment of the resource x_j more than u_j to a job J_j that precedes job J_i. We assign resources such as $x_j \leq u_j$ and increasing order of w_i/a_i. Then we can decide the assignment of resource using following algorithm.

Algorithm 2.

Step 0: Let $M_1(0) \leftarrow 0, M_2(0) \leftarrow 0, i \leftarrow 1$, List $C \leftarrow \emptyset$ All job and resource has not been assigned.

Step 1: If $p_i \leq d_i - M_1(i-1)$ then Job $J_i$is processed on only machine M_1 in time interval $(M_1(i-1), M_1(i-1)+p_i]$. $M_1(i) \leftarrow M_1(i-1)+p_i$. Go to Step 4. If $p_i > d_i - M_1(i-1)$ then Job $J_i$is processed on machine M_1 in time interval $(M_1(i-1), d_i]$. $M_1(i) \leftarrow d_i$. Go to Step 2.

Step 2: If $M_2(i-1)+p_i-(M_1(i)-M_1(i-1)) \leq M_1(i-1)$ then the remainder of Job J_i can be processed without assigned resource. Therefore It is processed on machine M_2 in time interval $(M_2(i-1)+p_i-(M_1(i)-M_1(i-1))]$. $M_2(i) \leftarrow M_2(i-1)+p_i-(M_1(i)-M_1(i-1))$. Go to Step 4. If $M_2(i-1)+p_i-(M_1(i)-M_1(i-1)) \leq M_1(i-1)$ then go to Step 3.

Step 3. Let J_j is the top of the list C.
If $a_j u_j \geq M_2(i-1)+p_i-M_1(i)$ then we assign the resource $x_j \leftarrow x_j + (M_2(i-1)+p_i-M_1(i))/a_j$, and go to Step 4.
If $a_j u_j < M_2(i-1)+p_i-M_1(i)$ then we assign the resource $x_j \leftarrow x_j + u_j$, and delete Job J_j from list C, go to Step 3.
If list C is empty then There is not feasible schedule. This algorithm terminated.

Step 4: If $i < n$ then renew $i \leftarrow i+1$, $u_j \leftarrow (M_2(i-1) - M_2(j-1))/a_j (1 \leq j \leq i)$ and reconstruct the list C that include job $J_j (1 \leq j \leq i)$ such as $u_j > 0$ and is arranged such as increasing order of w_j/a_j. Go to Step 1. If $i = n$ then algorithm terminated successfully.

From correctness of Sahni's algorithm and Theorem 2, it os clear that total weighted resource $\sum_{i=1}^{n} w_i x_i$ of above algorithm is minimized that of all other feasible schedule. To obtain actual schedule, we applied original Sahni's algorithm to modified processing time $b_i = p_i - a_i x_i$ that assigned the resource x_i obtained from above algorithm.

4 Conclusion

We considered the parallel machines shop problem with a consumption of resources, and give the algorithm that minimize total weighted resource $\sum_{i=1}^{n} w_i x_i$. We conjecture that this result can be extended to the problem that constrained total weighted resource. And the problems with multi-criteria such as total resource and maximum completion time or total resource and maximum lateness are also interested.

References

[1] A. Janiak, M.Y.Kovalov "Single machine scheduling subject to deadlines and resource dependent processing times", European Journal of Operational Research 94(1996) 284-291.

[2] A. Janiak, "Single machine scheduling problem with a common deadline and resource dependent release dates", European Journal of Operational Research 53(1991) 317-325.

[3] S.Sahni "Preemptive Scheduling with Due Dates", Operations Research, 27 (1979) 925-934.

[4] R. G. Vickson "Choosing the job sequence and processing time to minimize total processing plus flow cost on a single machine", Operations Research, 28 (1980) 1155-1167.