Journal of Applied Science and Engineering Innovation, Vol.2 No.8, 2015, pp.302-304
ISSN (Print): 2331-9062
ISSN (Online): 2331-9070

A Cost Allocation Rule in Sequencing Situation

Shanshan Liu

Department of Mathematics, School of Science, East China University of Science and Technology, Shanghai, 200237, China

Abstract: In a sequencing situation, each agent owns exactly some jobs which need to be processed on some machines. Each job has a completion time, and the agent need to wait until his jobs are all finished. The agent occurs a cost which is the weighted completion time of his last job. The agents are free to discuss the final processing order. We propose a cost allocation rule being dependent on the processing order, and show how it works for two sequencing situations.

Keywords: scheduling; sequencing situation; cooperative game; cost allocation

INTRODUCTION

The sequencing situation raised from production environment is introduced in 1989 [Curiel et al. 1989]. In a sequencing situation there are some agents before some machines, each agent owns jobs that need to be processed with given processing times, and its staying in the system occurs a cost per unit time, which is called the weight. The agents negotiate to decide the final processing order, to make every agent satisfied transfer payment is allowed, which means one can pay those processed later some money to be processed earlier. A final processing order and the related transfer payment together solved the sequencing situation. The single machine sequencing situation has been fully studied, and different kinds of cost allocation are proposed [Curiel et al. 1989, Curiel et al. 1993, Curiel et al. 1994]. Latter single machine sequencing situation with ready time and due date are studied [Hamers et al. 1995, Borm et al. 2002]. Hamers also discussed the sequencing situation with chains precedence [Hamers et al. 2005], which means jobs are partitioned in several chains, a job can start to process only if jobs in the same chain before it have all finished.

We focus on a cost allocation rule called EWCS rule, which is generalized in the classic single machine sequencing situation, and is developed to solve two sequencing situations, one is the single machine sequencing situation with precedence constrains, the other is the two-machine sequencing situation. For single machine with precedence constraint, there is precedence constraint P between the jobs, and for any job i and j if $(i, j) \in P$ then job i can only be processed after job j is finished. Chains precedence mentioned before is a special case for precedence constraints. In a two-machine sequencing situation each agent has exactly two jobs need to be processed on machine 1 and machine 2, the agent need to wait until his last job is finished, thus his cost is weighted completion time of his last job.

COST ALLOCATION RULE

EWCS rule

In a sequencing situation, the grand coalition containing all agents is denoted by N , the jobs' processing time is p , while the weight is w . Let σ be a processing order, $\sigma(\mathrm{i})$ be the job on position i . Let the jobs preceding agent i be $P(\sigma,i)$, and jobs after agent i be $F(\sigma,i)$. The completion time of agent i under processing order σ is $C_i(\sigma)$, thus we have $C_i(\sigma)=\sum_{j\in P(\sigma,\mathrm{i})} p_j + p_i$. Then the total cost is $\sum_{i\in N} w_i C_i(\sigma)$, and a cost allocation rule must allocates exactly this cost among agents. Thus the rule satisfies efficiency. There may be many feasible processing orders for a sequencing situation, the optimal one makes the total cost minimized. Shapley value is a popular cost allocation solution in many problems, it is first introduced by Shapley [Shapley 1953]. The single machine sequencing situation without initial order is studied [Mishra and Rangarajan 2007], and a polynomial time formulation of the Shapley value for agent i is given:

$$SV_i = w_i p_i + \sum_{j\in P(\sigma^*,i)} \frac{1}{2} w_i p_j + \sum_{l\in F(\sigma^*,i)} \frac{1}{2} w_l p_i$$

where σ^* is an optimal processing order.

Corresponding Author: Shanshan Liu, School of Science, East China University of Science and Technology, Shanghai 200237, China.

The EWCS rule is inspired by this formulation. For a given processing order σ we decide a cost allocation according to it. The rule allocates to each agent his own processing cost, and half the waiting cost caused by his job for jobs after him, and also half the cost he suffered from those preceding him. With this meaning we call it EWCS (Equal waiting cost split) rule. And agent i 's cost under processing order σ is:

$$f_i(\sigma) = w_i p_i + \sum_{j \in P(\sigma,i)} \frac{1}{2} w_i p_j + \sum_{l \in F(\sigma,i)} \frac{1}{2} w_l p_i$$

It is easy to verify that the EWCS rule satisfies efficiency.

Sequencing Situation with Precedence Constriant

The single machine sequencing situation with precedence constraints with no initial order is denoted by (N, P, p, w) , where P is a set of 2-tuples representing the precedence constraints, and $(i, j) \in P$ implies that job j can be started only after job i has been finished. The precedence constraints can also be visually described by an acyclic directed graph. The specific precedence constraints of chains can be pictorially described as several parallel directed chains.

The processing orders satisfying the precedence constraints are called feasible orders, and we denote the set of feasible order as $A(N)$. Finding the optimal one among the feasible order may not be done in polynomial time, and in practical we can decide the final order by heuristic algorithm. For the case with chains precedence, the optimal processing order can be figured out: first cutting from each chain the head with maximum ratio of weight to processing time until all jobs are partitioned, and let jobs in a head be processed together, then order the partitioned job-sets in decreasing ratio of weight to processing time.

We generalize the EWCS rule to get the payment of each agent of sequencing situation (N, P, p, w) :

1) if $(j,i) \in P$ then i takes all the waiting cost caused by waiting for j since in any feasible order i is behind j ;
2) if $(i, j) \in P$ then j takes all the waiting cost caused by waiting for i ;
3) if $(j,i) \notin P$ and $(i, j) \notin P$ thus i and j share the waiting cost equally.

For any feasible processing order $\sigma \in A(N)$, we define value j as

$$q_i(\sigma) = \frac{1}{2}\Big(\sum_{j \in P(\sigma,i),(j,i) \notin P} w_i p_j + \sum_{j \in F(\sigma,i),(i,j) \notin P} w_j p_i \Big) + \sum_{(j,i) \in P} w_i p_j + \sum_{(i,j) \in P} w_j p_i + w_i p_i$$

where P is the precedence constraints.

It is easy to verify that $\sum_{i \in N} q_i(\sigma)$ is the total cost under processing order σ , in other words we have

$$\sum_{i \in N} q_i(\sigma) = \sum_{i \in N} C_i(\sigma) .$$

The modified EWCS rule applied to single machine sequencing situation satisfies efficiency.

Two-machine Sequencing Situation

The two-machine sequencing situation is proposed by Calleja et al. [Calleja et al. 2002], and they showed the balancedness of the related sequencing game when all jobs have uniform processing time and weight. A two-machine sequencing situation is denoted as $(M, N, w, (p, q))$, where M contains two machines, and N is the agent set, w is the weight of agent. And each agent owns exactly two jobs that need to be processed on the two machines respectively. We use i to denote agent i 's two jobs, i.e., agent i owns a job i with processing time p_i which need to be processed on machine 1, and also another job i with processing time q_i which need to be processed on machine 2. Let (π, φ) be a processing order, then π is the order on machine 1 and φ is the order on machine 2. If $\pi = \varphi$, (π, φ) is called a permutation order. We denote the completion time of agent i on machine 1 and machine 2 as $C_i^1(\pi)$ and $C_i^2(\varphi)$ respectively. The total cost is $\sum_{i \in N} w_i \cdot \max(C_i^1(\pi), C_i^2(\varphi))$.

For two-machine sequencing situation, the optimal processing orders have the following properties:

(1) There must be one optimal processing order being a permutation one;
(2) If $p = q$, then all optimal processing orders are permutation ones.

The general problem is difficult to find an optimal processing order, we can search from the permutation orders and choose a final order. There is a special case whose optimal processing order is easy to figure out: each agent's job with larger processing time is on machine 1, and the case on machine 2 is similar. In that case

$$\sum_{i\in N} w_i \cdot \max(C_i^1(\pi), C_i^2(\varphi))$$

$$=\sum_{i\in N} w_i \cdot \frac{1}{2}(C_i^1(\pi)+C_i^2(\varphi)-|C_i^1(\pi)-C_i^2(\varphi)|)$$

Thus if we let jobs on machine 1 in decreasing order of their ratio w_i / p_i (this order is optimal in minimizing the total weighted completion time on machine 1 [Smith 1956]), and jobs on machine 2 follow the order on machine 1, then we have

$$\sum_{i\in N} w_i \cdot (C_i^1(\pi) + C_i^2(\varphi))$$

$$= \min \sum_{i\in N} w_i \cdot C_i^1(\pi)$$

Which means the total cost is minimized, and the optimal processing order is found. For the general problem, there is a special order: the jobs on both machines are in their decreasing order of the ratio of weight to processing time. We denote the order as (π, φ), then it has a worst-case ratio less than 2, since the total cost won't be larger than

$$\sum_{i\in N} w_i \cdot (C_i^1(\pi) + C_i^2(\varphi)),$$

and meanwhile it won't be less than

$$\sum_{i\in N} w_i \cdot \frac{1}{2}(C_i^1(\pi) + C_i^2(\varphi)).$$

We focus on the cost allocation problem of the general problem in the following part. And we choose the final order a permutation one denoted by (σ, σ). Let agents be finished on machine 1 (the job with larger completion time is on machine 1) be S_1, and jobs finished on machine 2 be S_2, then $S_1 \cup S_2 = N$. We generate the EWCS cost allocation rule to the general case: let each agent take his own job's processing cost (his last finished job's processing cost), and let the waiting cost be allocated between those agents preceding him and himself, he also share the waiting cost of jobs delayed by him. For agent i, let

$$f_i(\sigma,\sigma) = \frac{1}{2}\Big(\sum_{l\in F(\sigma,i)\cap S_1} w_l p_i + \sum_{l\in F(\sigma,i)\cap S_2} w_l q_i + \sum_{j\in P(\sigma,i)} w_i p_j\Big) + w_i p_i, \quad i \in S_1$$

$$f_i(\sigma,\sigma) = \frac{1}{2}\Big(\sum_{l\in F(\sigma,i)\cap S_1} w_l p_i + \sum_{l\in F(\sigma,i)\cap S_2} w_l q_i + \sum_{j\in P(\sigma,i)} w_i q_j\Big) + w_i q_i, \quad i \in S_2$$

Under this rule, an agent's waiting cost is his last finished job's waiting cost, and he share the waiting cost with those who caused waiting equally. The rule also satisfies efficiency.

We give an instance of three agents to explain how the rule works. Suppose the optimal processing order which is also a permutation order, and $\sigma=\{1,2,3\}$ on both machines, the total cost is

$$w_1 C_1^1(\sigma) + w_2 C_2^2(\sigma) + w_3 C_3^1(\sigma)$$

$$= w_1 p_1 + w_2(q_1+q_2) + w_3(p_1+p_2+p_3)$$

For agent 1, 2, 3, their payments are respectively

$$f_1(\sigma,\sigma) = w_1 p_1 + \tfrac{1}{2} w_2 q_1 + \tfrac{1}{2} w_3 p_1,$$

$$f_2(\sigma,\sigma) = w_2 q_2 + \tfrac{1}{2} w_2 q_1 + \tfrac{1}{2} w_3 p_2,$$

$$f_3(\sigma,\sigma) = w_3 p_3 + \tfrac{1}{2} w_3 p_1 + \tfrac{1}{2} w_3 p_2.$$

CONCLUSION

We first introduced the EWCS rule for single machine sequencing situation without initial order. The rule assigns to each agent a cost which varies over different processing orders. When there are precedence constraints, the EWCS rule is modified according to the precedence constraints. For two-machine sequencing situation, the rule is generalized according to the final order. For both sequencing situations discussed, the generalized cost allocation rule is practical.

REFERENCES

Borm P, Fiestras-Janeiro G, Hamers H, Sanchez E, Voorneveld M, 2002 "On the convexity of games corresponding to sequencing situation with due date", European Journal of Operational Research, vol.136, pp 616-634.

Calleja P, Borm P, Hamers H, Klijn F, Slikker M, 2002, "On a new class of parallel sequencing situations and related games", Annals of Operations Research, vol.109, pp 265-277.

Curiel I, Pederzoli G, Tijs S, 1989, "Sequencing games", European Journal of Operational Research, vol.40, pp 344-351.

Curiel I, Potters J, Prasad R, Tijs S, Veltman B, 1994, "Sequencing and Cooperation", Operations Research, vol.42, pp 566-568.

Hamers H, Borm P, Tijs S, 1995, " On games corresponding to sequencing situations with ready times", Mathematical Programming, vol.69, pp 471-483.

Hamers H, Klijn F, Van Velzen B, 2005, "On the convexity of precedence sequencing games", Annals of Operations Research, vol. 137, pp 161-175.

Mishra D, Rangarajan B, 2007, "Cost sharing in a job scheduling problem", Social Choice and Welfare, vol.29, pp 369-382.

Shapley LS, 1953, "A value for n-Person Games", In: H.W. Kuhn and A.W. Tucker (eds.) Contributions to the Theory of Games Ⅱ, Princeton University Press, pp 307-317.

Smith WE, 1956, "Various optimizers for single stage production", Naval Research Logistics Quarterly, vol. 3, pp 59-66.