A Heuristic Policy for Scheduling Of Concurrent Design Activities[*]

Qian Zhong and Yingping Zheng

The Institute of Automation
Chinese Academy of Sciences
Beijing.100080
THE PEOPLE'S REPUBLIC OF CHINA

Abstract: In large-scale manufacturing systems involving a large amount of information, resources and design activities, scheduling of these activities simultaneously in design process is an important step toward achieving concurrency. In this paper, a new heuristic policy for scheduling of concurrent design activities is developed. It decides which combination of activities to be performed simultaneously without violating any of the constraints by a trade-off between time order, resources, and information in a global view. At last an example is given to show the algorithm.

Keywords: Heuristic Scheduling, Information Weighted Factor (IWF), Composite Allocation Factors (CAF), Minimal Delaying Alternatives, Maximal Scheduling Alternatives

Qian Zhong is a Ph.D student at the Institute of Automation, Chinese Academy of Sciences. Her topics of interest include optimization, scheduling, decision and control of product system.

Yingping Zheng graduated from Beijing University in 1963 and obtained the MSc. degree from the Institute of Automation, Chinese Academy of Sciences in 1967. He is now Professor of Automatic Control. His research interests include discrete event dynamic systems, modeling of complex control systems and optimization, decision and control of product system.

1. Introduction

For large-scale and complex product, the design process involves a number of activities that are performed according to a number of constraints, such as time order, information interdependent relationships and resources, etc., which are the main factors to restrain the simultaneous execution of design activities. The scheduling of these activities is to simultaneously dispatch as many activities as possible without violating any of these constraints, to minimize the total completion time of design process. Therefore, it is a very important and efficient method to achieve the concurrency of design, and can lead to savings and increased quality through simplification of the entire design and manufacturing process [1] .

There are some difference and analogies between the problem of design activities scheduling and the project scheduling problem with multiple resource constraints for the new features of design activities in CE. Attempts to solve the latter problem concentrate in three areas [2]: the formulation and solution of the problem as a mathematical programming problem, the enumerative procedures for obtaining optimal solution and the development of heuristic solution procedures for satisfactory solutions. Numerous heuristic policies were developed, such as "Minimum Slack First", "Shortest Operation First" [3,4]. All these scheduling rules are based on the critical path. However, in concurrent design process, whether or not one activity can be implemented depends on the information that the preceding activities provide. Dispatching different activities will change the direction of design and the critical path. So, these policies have limitations in dealing with the problem of scheduling concurrent activities, and it is necessary to find some new policies.

There have been some work about scheduling problem in CE. Badiru [5] once used the *precedence diagramming method (PDM)* to establish the network of activities, and presented the *composite allocation factor (CAF)* heuristic to schedule concurrent manufacturing projects in the allocation of resources and time slots. Kusiak [6] has presented a pull system for the management of activities which ensure a smooth flow of information, and has developed a dynamic heuristic policy. However, at least to the authors' knowledge, there has been no previous work to concern the information factor in scheduling policies. But, in concurrent product development, the smooth flow of information from the beginning to the end of design process is a key factor to a successful and timely completion of the design process, thus a

---

[*] Supported by National 863 High-tech Project Foundation and National Natural Science Fund (China)

good and feasible scheduling policy should consider the information factor. In this paper we use a global view to propose a heuristic scheduling policy that simultaneously trades off the information, resources and time in order to make a satisfied decision on the dispatching of concurrent design activities.

To consider the information factor, *information weighted factor (IWF)* is proposed. It will measure the degree of importance of the information provided by each activity. The heuristic policy consists of the rule for allocation of the limited resources and the rule for optimization of information. In the next Section, some new features of the design activities are analyzed and the formulation of scheduling problem discussed in this paper is given. Section 3 focuses on heuristic scheduling policy and procedure, including how to allocate resources and how to decide the information weighted factor. In Section 4, a numerical example is given. The last Section is that of conclusions.

2. Scheduling Problem

2.1 New Features of Design Activities

CE differs from the traditional sequential product development in that it considers all the concerns in product life cycle at an early design stage, and dispatches the activities that are performed according to precedence constraints to be simultaneously executed as many as possible[7], especially the concurrency of information. Thus there are some new features of design activities in CE, such as:

1) Highly information interdependent relationships among activities, which is the main logic relationships among activities. The execution of succeeding design activities is based on the information provided by the preceding activities. Providing the information timely to those activities whose information is needed by downstream activities mostly is one key factor for successful realization of CE.

2) The complex time order relationships among activities. CE allows the serial, overlapping or simultaneous execution of activities according to the information availability.

3) Uncertainty of processing time of activities. Some design activities are new and unique in nature, and so, it is not possible to predict their duration. Some are needed to evaluate and repeat, which is a stochastic behaviour. Thus the processing time of activities is uncertain, and it is necessary to find a feasible assessment of time distribution for design activities in CE.

4) Limitation of available resources. As there are only limited amounts of resources in design process, a simultaneous execution of activities will cause resource conflicts, thus the allocation of resources to activities is the other key factor in the scheduling problem.

Given the four features mentioned above, the scheduling problem in CE is different from conventional projects scheduling. It is necessary to find some new methods for satisfying the specific and unusual project scenarios of CE.

2.2 Formulation

The basis for a scheduling problem is the formulation. The details of formulations of two kinds of scheduling problems which CE is concerned with, have been provided in our paper [8].

The first step for formulation is to express all kinds of relationships among activities clearly. An extended activity-on-node (A-O-N) method is proposed to establish the *network of activities*, which is represented as a directed graph G = (N,A,C). N is a set of nodes, where one node indicates a design activity. A is a set of arcs, which indicates the information-dependent relationships between two activities. C is a set of weights of arcs, indicating the time order relationships among activities. $C_{ij} \in C$, $(i,j) \in N$, $C_{ij} \geq 0$ means that activity "j" must be performed after activity "i" finishes with the lag of C_{ij} , $C_{ij} < 0$ means that activity "i" and activity "j" have an overlap of $|C_{ij}|$. Thus, the extended A-O-N method can express the complex time order and the information-dependent relationships among activities clearly.

Another preparation for formulation is the assessment of time distribution of design activities. According to our experience, it is more suitable to use k-slot discrete approximation than to use k-point discrete approximation for estimating the processing time of activities. For activity "i", the time distribution of it could be approximated by the discrete random variables T_i . It is assumed that there are K possible time slots, and the time duration of activity "i" could be in one time slot (A_{ir1}, A_{ir2}) with probability p_{ir} (r =1,2,...,K) based on previous knowledge. Thus the time distribution of activity "i" can be expressed as:

$$P(A_{ir1} \le T_i \le A_{ir2}) = p_{ir}\ ,\quad r = 1,2,\ldots,K.$$

Each activity may have its own value of K. If K is chosen sufficiently large, it is obvious that the approximation will approach the true distribution to any desired degree of accuracy. Normally, K=6 is all right. And the estimated duration of activity "i" is: $\bar{t}_i = \sum_{r=1}^{K} (\frac{A_{ir1} + A_{ir2}}{2}) p_{ir}$, the estimation deviation σ_i is: $\sigma_i = \sum_{r=1}^{K} (\frac{A_{ir1} + A_{ir2}}{2} - \bar{t}_i)^2 p_{ir}$, and the standard deviation S_i is: $S_i = \sqrt{\sigma_i}$.

For this kind of time distribution, we can use a simple and easy-to-implement procedure proposed by Ord [9] to compute the earliest/latest start time and earliest/latest finish time of activities by collapsing the distribution.

To simplify the formulation, it is assumed that every design activity, once begun, cannot be interrupted, and resources are available per period in constant amounts and are also demanded by an activity in constant amounts throughout the duration of activity. A is a *set of activities*, $i \in A$, $i = 1,2,\ldots,n$. For convenience, let us consider two dummy nodes, namely 0 and n+1, such that their processing time and amount of resource requirements are 0. The scheduling problem discussed in this paper can thus be formulated as follows:

$$\min f_{n+1} \tag{1}$$

subject to:

$$f_{n+1} \ge f_i, \forall i \in A \tag{2}$$

$$f_j - f_i \ge C_{ij} + d_j, (i,j) \in H \tag{3}$$

$$\sum_{S^t} r_{ik}(t) \le R_k(t), t = 1,2,\ldots,f_n \tag{4}$$

$k = 1,2,\ldots,m$ where:

d_i : processing time of activity i,

f_i : finishing time of activity i,

H: set of pairs of activities indicating the precedence constraints,

$r_{ik}(t)$: amount of resource type k required by activity i at time t.

S^t : set of activities executed simultaneously at time t,

$R_k(t)$: total availability resource type k at time t.

The objective of scheduling problem is given in (1). Constraints (2) means the activity n+1 to be the last finished activity in design process. Constraints (3) indicates the time relationship and information interdependent relationship between activity i and activity j. The resource constraints given in constraints (4) indicate that at each time and for each resource type k, the resource amounts required by simultaneous execution of activities cannot exceed the resource availability.

3. Scheduling Policy and Procedure

The above-mentioned scheduling problem is usually NP-hard, so one can hardly find any efficient mathematical algorithm to solve it. We use a heuristic approach to obtain a sub-optimal solution. Firstly, we give the definition of *set of schedulable activities---S_t*.

A set S_t is the set of activities which can be implemented at time t according to the information they need and to the preceding design activities having been completed.

The two key factors for the successful and quick completion of the design process are:

1) Use of the limited resources effectively, and simultaneous scheduling of as many activities as possible without violating the constraints;

2) Optimization of the flow of information, in order to prioritize those activities whose information is mostly needed by downstream activities.

Therefore, the heuristic policy for the scheduling of concurrent design activities should consist of the rule for allocation of resources and the rule for optimization of information.

3.1 Rule for Allocation of Resources

If it is impossible to schedule all schedulable activities in S_t at time t, a so-called resource conflict occurs [10]. In order to resolve the resource conflicts, it is necessary to decide which combinations of activities can be given priority in being executed simultaneously, and which combinations of activities need be delayed. Demeulemeester and Herroelen [11] proposed *minimal delaying alternatives*, and proved that, *in order to resolve the resource conflicts, it is sufficient to consider only the minimal delaying alternatives*. At each time instant t, subsets D^k, k=1,2,...,L (different time instants may have a different value of L) consist of such activities, the delay of which would resolve the current resource conflicts. All subsets of activities D^k form *a delay set D(t)*. A delay alternative D^k is *minimal* if it does not contain any other delaying alternatives $D^v \in D(t)$ as a subset.

In this paper, supplementary sets of minimal delaying alternatives --- S^k, k=1,2,...,L are presented, named *maximal scheduling alternatives*. $S^k = S_t - D^k$. A set S^k is a subset of S_t, which is defined as the combination of design activities from the set S_t that can be performed simultaneously without violating the resource constraints, but adding any one more activity to S^k from set S_t would result in resource conflicts. All subsets S^k form a *scheduling set* --- S(t). In order to resolve the resources conflicts, the rule for allocation of resources is to assign the limited resources only to these activities in the maximal scheduling alternatives. This is also a sufficient condition.

The sets of S^k and D^k (K=1,2,...,L) can be obtained by enumerative procedures.

3.2 Rule for Optimization of Information

Proper dispatching at the upstream design activities can ensure an uninterrupted flow of information, and working on the wrong activity could delay the delivery of some important outcome (design information)[6]. Thus critical design activities that influence the total design time should not be allowed to be idle[6]. However, there has been no previous work to consider how to dispatch activities based on the information. In this paper, the concept of *information weighted factor (IWF)* is proposed to deal with this problem.

Information weighted factor is a measure of information provided by one activity to influence its succeeding activities. IWF of activity "i" is denoted by λ_i , $0 \leq \lambda_i \leq 1$, which could be decided by the number of activities that depends on the information activity i provides. λ_i is given under unified rules by analyzing the information dependence relationships among the activities in the *network of activities*. The more the activities that depend on the information delivered by activity i, the larger the value of λ_i will be. So the critical activities in design process are those activities that have a large value of IWF .

When a number of activities competes for the resources, the activities with larger values of IWF will be given higher priority and executed earlier, thus to avoid the resource conflicts and ensure a smooth flow of information to the critical activities. This is the rule for optimization of information aimed to ensure a smooth flow of information all through the design process.

There are several approaches to determine the value of IWF, such *as analytic hierarchy process (AHP), expert scoring method and fuzzy synthetic evaluation,* etc.

3.3 The Heuristic Scheduling Policy

Dispatching provides the direction for selecting a right activity of the design plan to perform. Given a number of known constraints and requirements, the scheduling problem of design activities is to choose a combination of activities to perform simultaneously without violating any of the constraints [6].

Badiru [5] has proposed the *composite allocation factor (CAF)* scheduling heuristic to prioritize activities in the assignment of resources and time slots in the project schedule. For each task i, CAF is computed as a weighted and scaled sum of two priority measures: *resource allocation factor (RAF) and stochastic activity duration factor (SAF)*, which takes into account both the resource requirement and the variability in activity times. But this scheduling policy does not consider the information factor, and needs be improved to satisfy the requirements of CE.

The computations of CAF are performed as below:

$$CAF_i = (w)RAF_i + (1-w)SAF_i$$

where w is a weight between 0 and 1, and a simulation experiment may be conducted to find out the best value of w for a given project [12]. In order to consider the information factor, the computation of RAF could be improved as:

$$RAF_i = \lambda_i \frac{1}{\bar{t}_i} \sum_{j=1}^{m} \frac{r_{ij}}{R_j}$$

where λ_i = the value of information weighting factor of activity i, $\bar{t}_i$ = expected processing time of activity i, r_{ij} = number of units of the resource type j required by activity i, R_j = total units of resource type j available in design process, and m= number of resource types involved. RAF is a weighting measure of expected resource consumption per unit time. The resource-intensive and critical activities (with a larger value of IWF) have a larger value of RAF, and so, require greater attention in the scheduling process. For the same reason, to incorporate the uncertainty of activity processing time and information factor, the computation of SAF could be improved as:

$$SAF_i = \lambda_i \left(\bar{t}_i + \frac{S_i}{\bar{t}_i}\right)$$

where S_i = standard deviation of processing time for activity i, and $S_i / \bar{t}_i$ = coefficient of variation of the processing time of activity i. Thus the values of SAF are dependent on the processing time distribution and information weighted factor.

Therefore, the value of CAF is determined not only by the estimated design activity time ($\bar{t}_i$), the amount of resource requirements (r_i) and the amount of available resources (R_i), but also by the information weighted factor (λ_i).

Based on the value of CAF, it is decided whether an activity is given priority for resource allocation during the scheduling process or not. An activity that has larger values of IWF lasts longer, consumes more resources and varies more in processing time. It will also have a large value of CAF. Such an activity is given higher priority for resource allocation among the activities competing for resources.

If several activities can be executed simultaneously without violating any of the constraints, then those activities in maximal scheduling alternatives that have the largest sum of CAF should be given highest priority for resource allocation, and the activities in its supplementary set should be delayed. Thus the scheduling problem reduces to picking up a subset S^k from the scheduling set $S(t)$ such that the sum of CAF is maximized, as follows:

$$Z^* = \max_{S(t)} \left(\sum_{i \in S^k} CAF_i \right)$$

The main advantage of this approach is that in order to schedule a right combination of activities to execute with priority, it simultaneously trades off all key factors for a successful and quick completion of the design process, such as information, resource and time. Thus it can achieve a globally sub-optimal solution. The other advantage lies in a simple and easy -to-do computation of the heuristic policy.

3.4 The Scheduling Procedure

Since the *sets of schedulable activities* at any particular time are small (even for fairly large network of activities), the algorithm suggested below is computationally attractive.

The procedure of scheduling of design activities is as:

step 1 At time t, update the resource vector ($R_1, R_2, \ldots, R_m$) and the set S_t of all schedulable activities according to the information and data.

step 2 List resource requirements, IWF, estimated processing time, etc., for each activity in set S_t.

step 3 Compute the RAF, SAF and CAF of each activity in set S_t.

step 4 Find all subsets S^k and D^k from S_t , compute the values of Z of each S^k.

step 5 Find the subset S^k with maximum Z, schedule all the activities in it simultaneously at time t, delay the activities in its supplementary set , and draw it in the Gantt chart.

step 6 Set t = t+1, go to step 1.

4. A Numerical Example

In this Section, an example illustrating the application of the algorithm presented at Section 3.4. is presented. The example differs a little from the example in [6].

Suppose the available resource vector at time t =5 is (5 5 4 0.9). For example, assume the value of w is 0.5, and the set of schedulable activities consists of 5 activities {5,7,8,10,11} and the known condition of the activities is that in Table 1. To simplify the computation, assume that time distribution of each activity is as listed in Table 1, and the probability of each time duration is 1/3.

The computation values of t, σ, RAF, SAF, CAF are listed in Table 2.

Table 1. Activities With Resource Requirement, IWF and Processing Time Distribution

Activity	Resource requirement	IWF	Time distribution
5	(2 1 1 0.15)	0.2	(5, 8, 10)
7	(2 3 1 0.2)	0.8	(7,11,12)
8	(3 3 3 0.5)	0.4	(3, 4, 5)
10	(4 5 3 0.25)	0.7	(10,15,20)
11	(1 1 2 0.4)	0.5	(5,7,9)

Table 2. The Computation Values

活动	t	σ	RAF	SAF	CAF
5	8	1.63	0.025	1.64	0.83
7	10	2.16	0.118	8.17	4.14
8	4	0.82	0.251	1.68	0.97
10	15	4.08	0.132	10.69	5.41
11	7	1.63	0.096	3.62	1.86

step 1: t = 5, Resource status vector at time t = (5 5 4 0.9), S_t = { 5, 7, 8, 10, 11}

step 2: List the known values in Table 1.

step 3: Compute the values of t, σ, RAF, SAF, CAF , listed in Table 2.

step 4: $S^1 = \{5,7,11\}$, $D^1 = \{8,10\}$, $S^2 = \{10\}$, $D^2 = \{5,7,8,11\}$, $S^3 = \{5,8\}$, $D^3 = \{7,10,11\}$

step 5: 在 S^1 中, $Z_1 = 0.83+4.14+1.86=6.83$, $Z_2 = 5.41$, $Z_3 = 0.83+0.97=1.80$, $Z^* = 6.83$. The subset with maximum sum of CAF is found to be $S^1 = \{5,7,11\}$. Hence, the activities 5, 7, 11 are scheduled simultaneously at time t, and the activities 8 and 10 will be delayed.

step 6: t = t+1, go to step 1.

The problem of management and scheduling of concurrent design activities is as a sub-system of "the distributed concept design management system". The prototype of this system has been developed under Windows LAN with Lotus Notes and Visual C++ language.

5. Conclusion

In this paper, according to some new features of design activities, a heuristic policy is proposed in a global view to suit the scheduling problem in CE, which simultaneously trades off all key factors for a successful and timely completion of design process, such as time order, information allocation and information optimization, etc. This policy can satisfy the special requirements of scheduling problem in CE, and can guarantee the successful and quick completion of design process. At last, a numerical example of the scheduling procedures is presented.

REFERENCES

1. KUSIAK, A. and PARK, K., **Concurrent Engineering: Decomposition and Scheduling of Design Activities**, INT. J. PROD. RES., Vol.28, No.10, 1990, pp. 1883-1900.

2. DAVIS, E.W., **Project Scheduling Under Resource Constraints: Historical Review and Categorization of Procedures**, AIIE TRANSACTIONS, 5, 4 , 1973, pp. 297-313.

3. DAVIS, E.W. and PATTERSON, **A Comparison of Heuristic and Optimal Solutions in Resource-Constrained Project Scheduling**, MANAGEMENT SCIENCE, 21, 8 ,1975, pp. 944-955.

4. KURTULUS, L. and DAVIS, E.W., **Multi-Project Scheduling: Categorization of Heuristic Rule Performance**, MANAGEMENT SCIENCE, 28, 2, 1982, pp. 161-172.

5. BADIRU, A.B., **Scheduling of Concurrent Manufacturing Projects**, Concurrent Engineering: Contemporary Issues and Modern Design Tools, 1993, pp. 446-464.

6. KUSIAK, A., **Scheduling Design Activities**, in S.Y. Nof (Ed.) Information and Collaboration Models of Integration, KLUWER ACADEMIC PUBLISHERS, 1994, pp. 43-60.

7. PALMER, B., KORBLEY, L. et al, **Modeling the Concurrent Engineering Process**, CERC Technical Report, West Virginia University, 1992.

8. ZHONG, Q. and ZHENG, Y.P., **Management and Scheduling of Design Activities in Concurrent Engineering**, accepted at the 4th International Conference On Computer Integrated Manufacturing Systems, Singapore, October 1997.

9. ORD, J., **A Simple Approximation To the Completion Time Distribution for A PERT Network**, RES. SOC. Vol. 42, No.11, 1991, pp. 1011-1017.

10. CHRISTOFIDES, N., ALVAREZ-VALDES, R. and TAMARIT, J.M., **Project Scheduling To Resource Constraints: A Branch and Bound Approach**, EUROPEAN J. OPEN. RES., 29, 1987, pp. 262-273.

11. DEMEULEMEESTER, E. and HERROELEN, W., **A Branch-and-Bound Procedure for the Multiple Resource-Constrained Project Scheduling Problem**, MANAGEMENT SCIENCE, 38, 12,1992, pp. 1803-1818.

12. BADIRU, A.B., **Project Management Tools for Engineering and Management Professionals**, INDUSTRIAL ENGINEERING & MANAGEMENT PRESS, Norcross, GA,1991.