Development of Production Scheduling Model with Constraint Resources and Parallel Machines

M.B. Fakhrzad, Yazd University, Yazd, Iran
H. Khademi Zare, Yazd University, Yazd, Iran

Abstract

In this paper, a production scheduling model with constraint resources and parallel machines has been investigated. This problem is proposed as a multi-product production problem. Shortage is not allowed and the production horizon is indefinite. The objective is to maximize the level of resource usage and support the management's standpoint (delays reduction). In this paper, this problem is modeled as the popular Knapsack problem in 0 and 1 programming. Then due to being NP-hard type for this kind of problems to obtain an optimal solution, A heuristic approach has been used to obtain the acceptable solution. By using the branch-and bound method, a near optimal solution is provided. Finally, resultant solutions by the proposed approach have been compared with the optimal solutions of some real-world problems and it has been observed that deviation from the optimal solution is negligible that indicates the accuracy of the proposed approach.

Keywords
Production scheduling, Limited resources, Parallel machines, Branch-and-bound,

Introduction

So many organizations are trying to produce their under production goods as fast as possible. Each product is consisted of several activities and in demonstrated as a graph which contains a number of nodes and activities. Each activity uses various resources in addition to run-time. Furthermore, activities have precedence and delay over each other. Resources are limited and contain a finite and specific level. Allocating these limited resources to the products may cause tardiness in production finish time. These activities should be planned so firstly each product's activities finish at the deadline (latest finish time) and secondly the resources are used optimally and efficiently. In order to allocate the resources and planning the activities it is necessary to study a number of activities of a project simultaneously. Since the resources are limited, planning all of activities in a time is not possible.

Accordingly, a decision making problem is proposed which shows that according to the available resources which activities can be planned and which activities could be delayed. After making the possible decision and planning the activities, this decision-making problem is presented again but this time the level of the available resources is different from the previous case (more or less). Thus it is necessary to make proper decisions at different times.

Gupta [1] has divided the production scheduling problems to three category of long range, mid range and short range problems. They have compared different types of the scheduling problems and solutions under various modeling conditions. They also have assimilated different solution methods for several large-size, medium-size and small-size problems. They have categorized problems like products type determination, process type determination and equipment type determination as long-range problems; problems such as lot-size determination, master production scheduling and material requirement planning as mid-range problems and problems such as production scheduling and operations sequencing as short-range problems. Accordingly, the concerned subject belongs to short-range scheduling group.

In a basic model of parallel machine, there is a set $M = \{M_1, M_2, \ldots\ldots, M_m\}$ of identical machines, which are used to process n jobs $J_1, J_2, \ldots\ldots, J_n$. Each job J_j has one operation which can be processed on any machine with processing time P_j. The objective is to minimize the makespan $C_{\max}$. By the well-known three-field representation [1], this problem is denoted by $P\|C_{\max}$ .Usage of 0 and 1 mathematical models and goal programming when there are a few activities already have been practiced [2,3]. But by increasing the number of activities when production environment becomes more realistic, complexity of these models significantly increases. This complexity even gets more complicated when level of the available resources during the production process varies.

Making the right decision at a time with regard to available limited resources can be described as a mathematical model like 0 and 1 and multi-purpose Knapsack problem. It has been showed that given problem is a NP-hard type problem [4]. Martello and Toth [5] have studied various solution methods for this kind of problems and have provided several recommendations. Toyoda [6] has investigated the multi-purpose Knapsack problem using Gradian method and has provided a recommendation. Heng Yang et al. [7] have developed an approximate algorithm for scheduling two parallel machines with limited resources. Jan Remy [8] has investigated the limited resources scheduling in parallel machines and he has solved the problem using the approximate algorithm with regard to NP-hard type of the problem. His aim was to minimize the makespan.Kellerer and Strusevich [9] have investigated the parallel machines scheduling problem with limited resources using the approximate algorithm. He has shown that solution of this kind of problems is polynomial. Balas and Martin [10] have discussed the problem by disengaging the problem from integer variables and using the linear programming. Freville and Plateau [11] proposed to reduce the number of constraints. Magazine and Oguz [12] solved the problem using the Lagrange multipliers. Volgenant and Zoon [13] developed a heuristic approach. Approaches provided by these people are applicable to 0 and 1 problem and typically these solution methods are complicated and time consuming. As mentioned before, resource allocation and planning problem for products is reduced to several multi-purpose Knapsack problems at different times. Thus it's essential to obtain a near optimal solution through a simpler, faster and less complicated approach.

In this paper, this problem is modeled as the popular Knapsack problem in 0 and 1 and multi-purpose state. Then due to being NP-hard type for this kind of problems to obtain the optimal solution, a heuristic approach has been provided to obtain the acceptable and practical solution. In this method, the problem is expanded as a tree at different levels. Development of each level of this tree is dependent on the available resources for allocation and the best acceptable resultant solution at that level. If creating a new level is not possible for any reason, the near optimal solution is obtained. Resultant solutions from several real-world problems have been compared with these solutions and efficiency of this method has been approved. This paper has the following structure. In section 2 we describe the given problem. Mathematical model of this problem is provided in section 3.Improvement of the proposed approach to obtain a near optimal solution is studied in section 5. Finally, section 6 is devoted to conclusion.

The Mathematical Model

In this section we try to develop and solve a mathematical model by planning and allocating the limited resources to various products which jointly use these resources and are implemented along each other. It is assumed that resource allocation to different production activities is possible and resource level varies according to the nature of the problem. So increase or decrease in resource level may lead to a production delay. Therefore, each planning with a makespan of *t* is not usable till the end of the planning horizon. It seems that this rule holds in practice since the program may change after t_1 time unit. Consequently, we want to plan the activities and allocate the resources in t time and for a relatively short horizon. In this case, the problem is converted to a 0 and 1 planning model like the popular multi-purpose Knapsack problem. As the time is running, by identifying the available resource level, the same model is repeated for another short horizon. In this case, consider a set of the production activities for different products as a set of activities that can be planned and their preceding activities has been planned before. With regard to resource constraint, planning of all activities may not be possible. Therefore, at the time of t we are dealing with a decision making problem with characteristics as below:

n = Number of activities that can be planned at time of t

m = Number of the available resources

q_i = Optimality resulted from planning the i th activity

$$X_i = \begin{cases} 1 & \textit{if } i\,th \textit{ activity is planned} \\ 0 & \textit{otherwise} \end{cases}$$

r_{ij} = Amount of the j th resource used in i th activity

R_j = Amount of the j th resource available at the time of t

t_i = i th activity's time

L_i = Latest finish time of the i th activity.

Activities should be selected according to a criterion. This criterion is a function of activity allowance time (floating) and importance of that activity from the management's point of view (delays reduction) which is identified as W_i. Allowance or floating time of an activity at the time of t is calculated as below:

$$S_i = L_i - t_i - t$$

Suppose that importance and criterion function of activity i is identified as $U_i(t)$. $U_i(t)$ is also a function of $S_i(t)$ .Therefore, $U_i(t) = f(S_i(t))$. $f(S_i(t))$ is defined as below:

$$U_i(t) = f(S_i(t)) = \begin{cases} |HS_i(t)| & \text{if} \quad S_i(t) \le 0 \\ \\ \dfrac{M}{S_i(t)} & \text{otherwise} \end{cases}$$

M and H are large and constant values that $H \ge M$. If an activity is finished earlier than its latest finish time then value of function f will be large. Otherwise the function has a small value. As mentioned before, selection criterion of the activities for planning is a function of $S_i(t)$ and W_i. Consequently q_i coefficient related to X_i in objective function is calculated from $W_i U_i(t)$ relation. According to above issues, activities selection problem for planning leads to a mathematical model described as below:

Max $\quad Z = \sum_{i=1}^{n} q_i X_i$

S.T.

$$\sum_{i=1}^{n} r_{ij} X_i \le R_j$$

$$X_i = 0,1$$

$$i = 1,2,.....,n \qquad j = 1,2,.....,m$$

In above model, a set of activities are selected which their resource usage is not greater than the available resource level. In addition, the objective function must be maximized. Before proposing an approach for activities selection, first we should make some changes in the above model. Each activity i is defined as below:

$$a_{ij} = \frac{r_{ij}}{R_j}$$

$a_{ij} =$ Percent of resource *j* required to process activity *i*

Thus, above model is changed to:

Max $\quad Z = \sum_{i=1}^{n} q_i X_i$

S.T.

$$\sum_{i=1}^{n} a_{ij} X_i \leq 1$$

$$X_i = 0,1$$

$$i = 1, 2, ..., n \qquad j = 1, 2, ..., m$$

A proposed approach to obtain an initial acceptable solution

With regard to the above model, a set of activities which maximize the objective function must be planned. Therefore, activities with delay should be identified according to a criterion. This criterion is named d_i that will be explained later. First, all of the activities that could be planned are placed in set *E.* With regard to the available resource level, planning of all activities is possible; in that case, developed solution is an initial acceptable solution. If planning of all activities is not possible then we make below definitions:

E = A set of activities that can be planned at the time of *t*

D = A set of delayed activities

V = A set of activities to be planned

Thus, it is necessary that all required resources be available to plan the activities of set E. Therefore, only the activities which have the least impact on the objective function Z are delayed. After allocating the resources, state of each resource is demonstrated in a vector as below:

$$B = [b_1, b_2, ..., b_j, ..., b_m]$$

$$b_j = 1 - \sum_{i \in v} a_{ij}$$

b_j = Remained resource *j* (percent)

B = Set of remained resources (percent)

Acceptable solution for above model is obtained when all elements of vector *B* are positive. If some elements are negative it means that some activities are not completed at deadline. Hence, some activities must be delayed. As described earlier, delaying the activities must be performed based on a criterion. d_i parameter is calculated as below:

$$d_i = \frac{q_i}{\sum_j \left| a_{ij} b_j \right|}$$

d_i = Delaying criterion of *i*th activity

Denominator of above fraction indicates the total used resource percent for *i* th activity. Therefore, an activity is delayed which has the minimum d_i. Thus, activities that cause negative elements in set B are added to set D. In the proposed algorithm, all E set activities are entered to set V. Then feasibility of set E is investigated. If solution is not feasible then an activity with minimum d_i is entered from set V to set D. This process is continued till set V is acceptable. Algorithm steps are as below:

Step 1- Set V set equal to set E and consider set D to be empty. Then calculate vector B and d_i for each activity of set V. Sort the activities in ascending order with respect to d_i. If all elements of vector B are positive then go to step 4. Otherwise go to step 2.

Step 2- Select an activity from set V with minimum d_i then remove that activity and add it to set D.

Step 3- Recalculate vector B. if B is positive then go to step 4. Otherwise go to step 2.

Step 4- Stop and start planning the V set activities. It may be required to solve many similar problems to determine all activities plan. In this method, increasing the number of activities does not affect the complexity of the proposed approach.

Numerical Example

Suppose that we want to determine the production scheduling program for products A and B as 3 parts of product A and 2 parts of product B can be scheduled at the time of *t*. Therefore, set E is consisted of 5 elements. Also suppose that there are 3 similar (parallel) resources with availability of $R_1 = 8$, $R_2 = 9$ and $R_3 = 7$ respectively at the time of *t*. Concerning the calculations of the critical path method, floating for each part, each type and amount of the resources required for production and weight of each part (relative importance) is given in Table 1.

Let H = 5 and M = 10 then $U_i(t)$ and q_i is calculated and summarized in Table 2.

Step 1. First let $V = E = \{1,2,3,4,5\}$ and $D \neq \varnothing$ then calculate vector B as below:

$B = \{-0.38, -0.56, -0.43\}$

Whereas elements of vector B are negative thus some activities of set V must be delayed. Relevant d_i measures are calculated as below:

$d_i = \{64.78, 73.49, 39.26, 58.84, 86.5\}$

Step 2. Sort the activities ascending by d_i and select the minimum d_i. Thus we have:

$D = \{3\}$, $V = \{1,2,4,5\}$

Step 3. Recalculate vector B disregarding part 3:

$B = \{0.0, -0.22, 0.0\}$

Whereas there is still a negative element in vector B then step 3 must be repeated.

Step 2. Select the minimum d_i. In this step activity 4 is selected. Therefore:

$D = \{3,4\}$, $V = \{1,2,5\}$

Step 3. Recalculate vector B disregarding parts 3 and 4. We have:

$B = \{0.5, 0.33, 0.43\}$

Whereas vector B is positive, the initial acceptable solution is obtained as below:

$D = \{3,4\}$, $V = \{1,2,5\}$

Now the sequence of the process can be determined according to q_i measures given in Table 2. Value of the objective function is equal to 60. Using LINDO software the optimal value of 85 is achieved.

Table 1 Weight, floating time and required resources for each product

P roduct	art	eight V	F loating Time	a_{ij}		
A		0 3 5 1	1 - 2 5			
B		5 2	1.6 - 1			

Table 2 Calculation of $U_i(t)$ and q_i for each part

P art					
U					

					0
q	5	5	0	0	0

Improvement of the proposed approach to obtain a near optimal solution

In this paper, to obtain a near optimal solution for a small-scale multi dimension and 0 and 1 Knapsack problem, is used the branch-and-bound method. In this method, it takes too much time to solve this kind of problems by computer. Required solution or computational time to obtain an optimal solution in the branch-and-bound algorithm is shown by $O(2^n)$ notation (n is the number of activities) [4]. Therefore, to obtain an optimal solution it is necessary to make some changes in the branch-and-bound algorithm. Simplicity of this method and reduction in computational time are some characteristics of these changes. In this method, problem is expanded like a tree at different levels. Creating a new level depends on obtaining the most acceptable solution at the previous level and availability of enough resource to be allocated to these activities. There are several nodes at each level. Number of each node represents the number of that activity. Selection of a node represents the planning of that activity. No activity is planned at level 0.

Selecting a node at a level means the planning of that activity. With regard to use of an activity from different resources, available resource level must be updated. Remaining unplanned activities of node 1 are planned using a proposed method to obtain an acceptable solution. The resultant objective function value from the proposed method is considered as evaluation result of node 1. This practice is repeated for other nodes. A node with maximum objective function value could be branched out. We avoid branch the other nodes. In branch-and-bound method, nodes are removed carefully but in this method, nodes removal is performed less precisely.

Generally, if it is planned to allocate the resources to n activities at the time of t , they will have n nodes at level 1. Therefore, we have $\left\{\frac{n!}{(n-k)!}\right\}$ nodes at k th level. At each level we select a node using the proposed approach to obtain an acceptable solution. This process continues until activities' planning is not possible due to resource shortage (lack of resources). The best objective function value at ultimate level is adopted as the near optimal solution. Number of the nodes investigated from level 1 to n are equal to $1, ..., n-2, n-1, n$ respectively. Thus, $O(2^n)$ represents the time complexity of the calculations [4]. Evaluating each node at each level is possible using a VB computer program. The results of study show that if two or more nodes at each level are adopted for splitting, the final solution will be more accurate and closer to the optimal solution. Results of these investigations are summarized in Table 3.

Conclusion

In this paper, development of a production scheduling with limited resources and parallel machine has been discussed. Considering that our purpose is to maximize the In this paper, development of a production scheduling with constraint resources and resource usage and fulfill the management's standpoint, we have modeled the problem as the popular Knapsack problem in 0 and 1 programming. Then due to being NP-hard type for this problem, a near optimal solution is obtained using a heuristic and the related branch-and-bound approach. The proposed approach has been used for 15 real-world problem with 5 to 50 activities that each activity uses 3 to 10 type of resources respectively. For each problem, resultant objective function value is compared with the resultant initial function value from LINDO software and deviation percent is calculated. Accordingly, the mean deviation percent is determined to be 5 percent. Experiments show that if 3 nodes are selected at each level, deviation from the optimal solution will be less that 0.5 percent which indicates the accuracy of the proposed method. Simplicity, speed and computational time reduction in obtaining a near optimal solution are the other advantages of this method.

Table 3 : Deviation from the optimal solution for a number of instance problems with evaluation of different nodes at different levels (percent)

Problem	**Number of activities at the time of *t***	**Number of resources**	**Level 0 result**	**Evaluation of one node at each level**	**Evaluation of 2 nodes at each level**	**Evaluation of 3 nodes at each level**
1	5	3	0	0	0	0
2	7	5	1.2	0	0	0
3	20	8	3.8	0.89	0	0
4	27	8	4.4	1.8	0	0
5	32	7	5.6	1.45	0.42	0
6	35	8	6.76	1.58	0.31	0
7	36	8	6.2	2.23	0	0
8	38	6	7.1	2.4	0.18	0
9	40	8	8.2	3.1	0	0
10	40	8	7.6	4	0.25	0.02
11	42	9	8.6	3	0.5	0
12	45	8	8.7	3.5	0.32	0
13	48	10	10.2	3.2	1.4	0.1
14	48	9	9.9	3.1	1.2	0
15	50	10	10.8	3.4	1.9	0.06

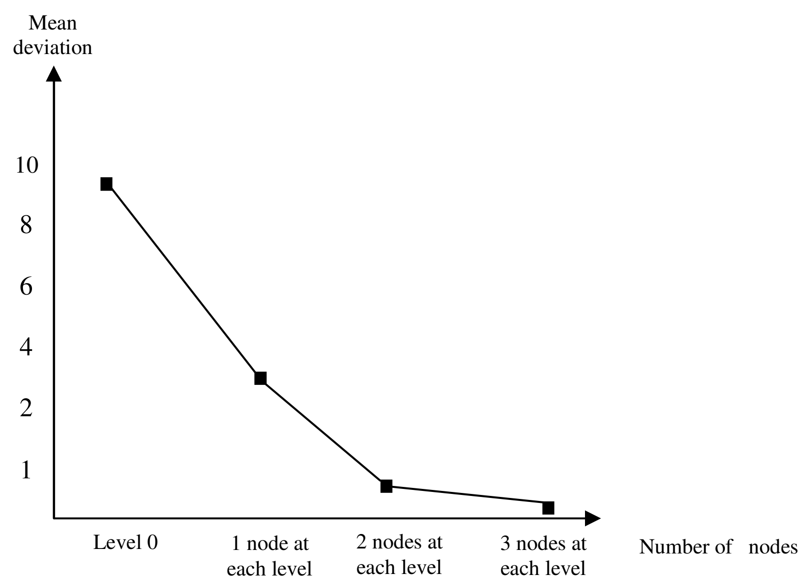

Fig 1 : Mean deviation percent from the optimal solution

References

[1] Gupta, J.N.D., (2002).,An excursion in scheduling theory: an overview of scheduling research in the century, *Production Planning & Control*, 13, 289-308

[2] Shabtay, D., Kaspi, M., (2006)., Parallel machine scheduling with a convex resource consumption function, *European Journal of Operational Research* 173, 92-107.

[3] Ventura, J. A., Kim, D., (2003)., Parallel machine scheduling with earliness–tardiness penalties and additional resource constraints, *Computers & Operations Research* 30, 1945-1958.

[4] Kellerer H., Strusevich, V. A., (2003)., Scheduling problems for parallel dedicated machines under multiple resource constraints, *Discrete Applied Mathematics* 133, 45-68.

[5] Martello, S. and Toth , P., Algorithms for Knapsack problems, In surveys in combinatorial optimization Martello, S. Amesterdam , The Netherlands: North-Holland (1987) pp.259-282.

[6] Toyoda, Y., (1985)., A simplified algorithm for obtaining approximate solutions to zero - one programming problems, *Management Science* 21,1417-1427.

[7] Heng Yang, Yinyu Ye, Jiawei Zhang, (2003)., An approximation algorithm for scheduling two parallel machines with capacity constraints, *Discrete Applied Mathematics* 130, 449 – 467.

[8] Jan Remy, (2004)., Resource constrained scheduling on multiple machines, *Information Processing Letters* 91, 177–182.

[9] H. Kellerer , V.A. (2004)., Strusevich, Scheduling problems for parallel dedicated machines under multiple resource constraints, *Discrete Applied Mathematics* 133, 45–68.

[10] Balas, E. and Martin, C., (1980)., Pivot and complement – A heuristic for 0-1 programming, *Management Science* 26, 86-96.

[11] Freville, A. and Plateau, (1989)., Heuristics and reduction methods for multiple constrains 0-1 liner programming problem, *European Journal of Operations Research* 24, 206-215.

[12] Magazine, M. and Oguz, O., (1986)., A heuristic algorithm for the multidimensional 0-1 knapsack problem, *European Journal of Operation Research* 16, 319-326.

[13] Volgenant, A. and Zoon, J.(1995)., An improved heuristic for multidimensional 0-1 knapsack problem, *Journal of Operational Research Society* 41, 963-970.

Index of Authors