Journal of the Korean
OR/ MS Society
Vol. 19, No. 3, December 1994

A New Approach for Resource Allocation in Project Scheduling with Variable-Duration Activities

Soo-Young Kim* · Robert C. Leachman**

Abstract

In many project-oriented production systems, e. g., shipyards or large-scale steel products manufacturing, resource loading by an activity is flexible, and the activity duration is a function of resource allocation. For example, if one doubles the size of the crew assigned to perform an activity, it may be feasible to complete the activity in half the duration. Such flexibility has been modeled by Weglarz [13] and by Leachman, Dincerler, and Kim [7] in extended formulations of the resource-constrained project scheduling problem. This paper presents a new algorithmic approach to the problem that combines the ideas proposed by the aforementioned authors. The method we propose involves a two-step approach: (1) solve the resource-constrained scheduling problem using a heuristic, and (2) using this schedule as an initial feasible solution, find improved resource allocations by solving a linear programming model. We provide computational results indicating the superiority of this approach to previous methodology for the resource-constrained scheduling problem. Extensions to the model to admit overlap relationships of the activities also are presented.

1. Introduction

We have observed that in many project-oriented production systems, there is considerable flexibility available in resource allocation, whereby one may upgrade/downgrade the resource levels allocated to an activity to lengthen/shorten the total duration of the activity. It is common in industrial practice to vary the resource allocation levels while an activity is in progress. For example, a shop manager may expedite a late job by supplementing the crew originally

* Dept. of Industrial Engineering, Pohang University of Science and Technology, Pohang, Korea
** Dept. of Industrial Engineering and Operations Research, University of California at Berkeley, Berkeley, Ca 94720, USA

assigned with additional workers, while slowing down an in-progress job which has sufficient slack time.

Flexibility of resource allocation in project scheduling has been modeled by researchers in two basic ways: (1) Assume that each activity has discrete alternative "operation modes" (e. g., fast, medium, slow). Once an operation mode is selected, it cannot be changed throughout the job progress (Wiest [14], Talbot [12]). (2) The range of resource allocation that is feasible for each activity is continuously variable between some upper and lower limits. The resource levels may be changed while the activity is in progress. The activity duration is not prespecified, but rather is a function of the resource allocation decisions (Weglarz [13], Leachman, Dincerler, and Kim [7]).

This paper provides an improved algorithmic approach for solving the second type of model of the resource-constrained scheduling problem. The resource-constrained scheduling problem we consider is the same as the one solved in Leachman et al [7], i. e., a project is constrained by limited availabilities of resources assumed to be non-storable and infinitely divisible, e. g., labor or machine services. The objective is to find a schedule with the minimum overall project duration.

Weglarz [13] proposes single-resource preemptive scheduling models in which he introduces a performance speed-resource usage function relating activity progress to resource allocation. His approach to find the minimum project completion time is based upon the prespecification of a complete ordering of the partially ordered nodes in the activity-on-arc network. When the performance speed-resource usage function is linear, a simple linear programming model is formulated and solved to find the minimum project duration for the given node ordering. For a large project network, there can be a very large number of possible node orderings; thus obtaining optimal solutions is impractical for real-world situations, and some means of identifying efficient node orderings is required to make use of Weglarz's approach.

On the other hand, Leachman et al [7] propose a multi-resource non-preemptive scheduling model in which they introduce the concept of activity intensity, linear speed-resource usage function in which the consumptions of multiple resources by an activity are assumed to be proportional. That is, resources are allocated to an activity in proportion to its total requirements, whereby the progress of an activity can be indexed with a single continuous curve termed activity intensity. Resource allocation decisions are made by assigning activity intensity levels at each decision epoch. Their results from solving the well-known set of test problems assembled by Patterson [11] as well as from solving very large-scale ship overhaul scheduling problems demonstrate that their heuristic approach provides substantial savings over the results obtained using traditional, fixed-intensity scheduling heuristics and yet is practical for large-scale use.

In this paper, we extend the heuristic approach proposed by Leachman et al [7] to find an improved solution by applying a linear program extended from the Weglarz's model. The modeling assumptions are explained in the following section. In section 3, we present the overall heuristic procedure. Section 4 provides computational results of comparing the proposed approach to the earlier methodology. We suggest extensions to the model and the proposed heuristic in section 5. Finally, we summarize and discuss some further research possibilities in section 6.

2. Modeling Assumptions

A project is described by an activity-on-node network consisting of N activities A_1, A_2,$\cdots$,A_N. The project requires K non-storable resources, which have capacities C_1, C_2,$\cdots$, C_N, respectively. The production model we assume is adapted from Leachman et al[7], in turn based on the general framework of Hackman and Leachman[4]. Multiple types of resources are required by each activity, which are applied proportionally throughout the activity duration. A constant mix of resources is utilized exactly proportional to the mix of total resource requirements for the activity. The rate of activity progress(the activity "speed") can be described by a single index for each activity, termed activity intensity. For example, if an activity is executed with a constant resource level, its intensity is the inverse of the resulting duration. This intensity concept was first introduced in Leachman[6].

Let $z_i(t)$ denote the intensity of activity of A_i at time t. Then, we assume

$$\underline{z}_i \leqslant z_i(t) \leqslant \bar{z}_i. \tag{1}$$

To calculate the rate resource K is applied to A_i at time t, one can simply multiply the total amount of resource K required by A_i, a_{ki}, times the intensity. We let

$$Z_i(t)=\int_0^t z_i(\tau)d\tau \tag{2}$$

i.e., $Z_i(t)$ denotes the cumulative intensity of A_i at time t, or, in other words, the fraction of A_i completed by time t. We assume $z_i(t)$ is normalized so that A_i is completed when $Z_i(t)$ becomes 1.

Each activity is assumed to have a normal intensity level pre-defined which is the inverse of the normal duration. It has been observed that in many parctical situations the shop managers utilize and maintain such "normal" durations to estimate and plan the event completion times. The normal intensity is assumed to represent a regular operation mode of the activity.

Leachman et al [7] proposed two different types of intensity assignment models, upgrading-only and upgrading-and-downgrading. The former model assumes that intensity cannot be lower than the previous level at any decision time, while the latter allows any feasible level of intensity at any time. In this paper, we illustrate the upgrading-and-downgrading case only. We present a conceptual formulation of the scheduling problem in the following.

Let

S_i=start time of A_i,

F_i=finish time of A_i,

FOL_i=set of activities immediately following A_i,

ACV_i=set of activities in progress at time t,

Then, the scheduling problem can be describes as:

Minimze F_N

Subject to

$$\int_{Si}^{Fi} z_i(t)dt=1 \qquad i=1,\cdots,N$$

$$S_j \geq F_i \qquad j \in FOL_i,\ i=1,\cdots,N$$

$$\sum_{i \in ACV_t} a_{ki}\ z_i(t) \leq C_k \qquad \text{for all } t$$

$$\underline{z_i} \leq z_i(t) \leq \overline{z}_i \qquad \text{for all } t$$

Obviously, the problem is quite complicated due to not only its combinatorial nature but also due to the continuity of the decision variables S_i, F_i and $z_i(t)$. However, if somehow the activities could be pre-ordered such that we knew which activities compete for resources in the same interval of time, we could find an optimal allocation of resources in each such interval. This specification of which activities operate in parallel is called an activity partitioning, and is tantamount to the node ordering proposed in Weglarz [13]. But instead of enumerating all possible orderings, we will try to find a "good" activity partitioning using an existing heuristic scheduling algorithm. That is the basis of our proposed approach to be explained in the following section.

3. Proposed Heuristic Algorithm

The algorithm proposed in the following consists of two steps:

Step 1. Solve the scheduling problem by utilizing the heuristic intensity-assignment algorithm proposed by Leachman et al [7].

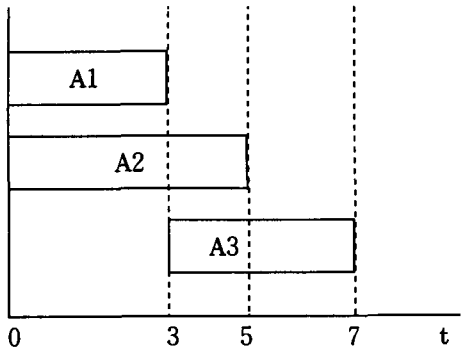
Step 2. Identify the activity partitioning, i. e., identify all activities active in the same time slice, and then find the optimal resource allocations for the observed activity partitioning by solving a linear programming model. The differences in our formulation from the Weglarz's are: (1) the intensities are limited by lower bounds, i. e., non-preemptive activity progress is assumed, and (2) multiple resource types are allowed.

In step 1, the activity intensities are determined (i. e., start/finish times and resource allocations for each activity are determined) by the heuristic upgrading-and-downgrading algorithm. A quick overview of the intensity-assignment algorithm is as follows:

1. Set t=0.
2. At t, create two activity sets: ACTIV = set of currently in-progress activities, and READY= set of newly schedulable activities.
3. Prioritize activities in ACTIV∪READY in decreasing order of "lateness scores"(=expected finish time of A_i-latest finish time of A_i).
4. Assign intensities to the activities just large enough to make the latest finish times, following the priority order.
5. If resources are left over, upgrade intensities as much as possible in order of priority.
6. Find the next decision time t, i. e., the next time the ACTIV or READY sets will change. Go to 2. Stop if no such t exists.

In step 2, we make use of the schedule generated by the above heuristic, in particular, the activity partitioning over the time horizon. Let us explain the partitioning using the following example.

Suppose a schedule from step 1 for a network with 3 activities is: $S_1=0$, $F_1=3$, $S_2=0$, $F_2=5$, $S_3=3$, $F_3=7$(See Figure 1 for a gantt chart of the schedule). The first time slice between 0 and 3 has active activities 1 and 2. In the second slice between times 3 and 5, active activities are 2 and 3. Activity 3 is the only active one in the final slice between times 5 and 7.



[Figure 1] Gantt Chart of the Example

We now proceed to formulate the LP model which minimizes the project completion time. Let

l=index of the time slice, $l=1,\cdots,L$,

ACT_l=set of activities in time slice l,

SL_i=set of time slices which contain A_i,

T_l=length of time slice l,

Z_{il}=cumulative intensity of A_i in time slice l, i.e., the fraction of activity A_i completed in slice l.

LP : Minimize $\sum_{l=1}^{L} T_l$

Subject to

$$\sum_{l\in SL_i} Z_{il}=1.0 \qquad i=1,\cdots,N$$

$$\sum_{i\in ACT_l} a_{ki}\, Z_{il}\leqslant C_k T_l \qquad k=1,\cdots,K,\ l=1,\cdots,L$$

$$Z_{il}\geqslant \underline{z}_i\, T_l \qquad i=1,\cdots,N \text{ for all } l\in SL_i$$

$$Z_{il}\leqslant \bar{z}_i\, T_l \qquad i=1,\cdots,N \text{ for all } l\in SL_i$$

The first set of constraints requires that each activity must be completed. The second set of constraints expresses the resource capacity limitations, and the last two express the intensity limits. Note that there are no precedence constraints formulated, since the activity partitioning into time slices guarantees the precedence relationships. Feasibility of the LP is guaranteed by the fact that we start with a feasible activity partitioning specified by the solution from step 1.

One question which must be answered is: Is it optimal to assume a constant activity intensity level in an arbitrary time slice for a given activity partitioning? The answer is yes, because if there is an optimal solution with varying intensity level for an activity within a slice we can always find an quivalent solution with a constant intensity level for the same activity. Also, the

non-preemptiveness assumption is satisfied. We remark that if it is better to have a slice, say slice l, eliminated, the LP will do so by setting $T_l=0$ and still satisfies the precedence relationships.

4. Computational Results

The major question in the computational tests we strive to answer is: How good is the proposed approach compared to the heuristic intensity-assignment algorithm? This question is answered by re-solving the set of 110 test problems (also known as "Patterson data set") as solved previously. (See Patterson (1984) for details of the data set.) The proposed two-step algorithm was programmed in FORTRAN77 on an IBM-PC, and the LINDO LP package operating on a VAX minicomputer was used to solve the LP models. In Table 1, the results are summarized. Average, maximum, and minimum percentage decreases in project completion times over previous methods (Intensity-assignment heuristic, and Fixed-intensity heuristic (equivalent to the minimum slack rule), Fixed-intensity optimal solution) are reported. Obviously, our two-step approach is better than the intensity-assignment heuristic, since we start with the heuristic solution and try to improve it further. Depending upon the allowable intensity range, the reductions in project duration achieved by the LP are between 1.8 and 3.2 percent on average. Due to the limited accuracy of floating point arithmetic on a PC compared to the mainframe used in previous research, there were slight differences in the results obtained for the intensity-assignment heuristic (for example, there was an average score of 8.1% under fixed-intensity optimum for the 0.8-1.2 range in the previous tests). Substantial savings are also obtained over the fixed-intensity models (assuming each activity has fixed normal duration). Note that the reductions over the fixed-intensity optimal solutions average 9-16.2%, which strongly suggests that the flexibility in the resource allocations is a significant factor to be considered to shorten the project duration.

[Table 1] Summary of the Perfomance of the Proposed Approach

Allowed Intensity Range	% Reduction in Project Completion Times								
	vs. Intensity-Assignment Heuristic			vs. Fixed-Intensity Heuristic(Min Slack)			vs. Fixed-Intensity Optimum		
	Avg.	Max.	Min	Avg.	Max.	Min	Avg.	Max.	Min
0.8-1.2	1.8	8.2	0.0	13.3	25.9	0.0	8.9	20.5	−5.3
0.5-1.5	3.2	10.6	0.0	20.3	31.1	0.0	16.2	31.1	0.0

A notable result is that the two-step approach is never worse than the fixed-intensity heuristic in any test problem, which was not the case using the intensity-assignment heuristic. In the previous tests, Leachman et al reported that there were a few cases in which the intensity-assignment heuristic was worse than the fixed-intensity heuristic, especially, for small, significantly serial networks.

For a large-scale project, it may take considerable time to solve the LP in the second step. In our computational test, the maximum size problem was with 50 activities and 3 resource types, which required only a few seconds to solve the LP on a VAX machine. But, in the real-world project scheduling environment, one may easily face networks with thousands of activities and tens of resource types. From our computational experience, we estimate that a rough upper bound on the number of time slices L would be the number of the activities. The average number of activities appearing in a slice will be a function of the network complexity, especially, the number of parallel paths and arcs. A crude estimate of the upper bound on the size of the LP is $(M+1)N$ variables and $(1+K+2M)N$ constraints, where M represents the maximum number of activities appearing in a slice. For example, if one has a 1000-activity network with 20 resource types and $M=50$, a rough bound on the problem size will be 51,000 variables and 121,000 constraints. From our recent experience with solving large-scale LP's using IBM's OSL package on RISC-based workstations, we conjecture that it may take on the order of 6-8 hours to solve such an LP to optimality. Since the planning of a project is normally done in batch mode periodically, we believe the proposed approach may be viable for large industrial-type networks.

5. Extension To Accommodate Overlap Relationships

Leachman et al [7] also has proposed an extension to their heuristics such that the activity network may include overlap relationships. It is quite common in many project-oriented production systems that similar or identical activities requiring the same mixture of resources are aggregated into a single activity for scheduling purposes. Such phenomena leads many practitioners to utilize the so-called Precedence Diagramming Method (PDM). (For more details of PDM, see Crandall [1] and Moder et al [9].) While PDM is one of the most popular methods in practice, not much has been studied and reported in regard to its complications. Some time ago, Wiest [15] pointed out an anomaly of PDM, in which the project duration becomes shorter if certain activity durations are lengthened. Leachman and Kim [8] showed that the PDM is not a proper production model when activities with variable durations are overlapped. They also

proposed a modified CPM to correctly calculate the earliest/latest start and finish times of the activities.

In this section, we present how the LP in the second step of the proposed approach can be extended to accommodate overlap relationships. First, the model in section 2 must be extended to correctly describe activity progress in the case of an overlap. Suppose activity j follows activity i with a given progress lag f_{ij}. Such a lag represents the amount of lag (in terms of percentage) in the cumulative intensity of the follower so as to maintain the proper inventory balance with the predecessor. In the extended model we assume the following relationship between A_i and A_j :

$$Z_j(t) \leqslant \begin{cases} 0.0 & \text{if } 0 \leqslant Z_i(t) < f_{ij} \\ Z_i(t) - f_{ij} & \text{if } f_{ij} \leqslant Z_i(t) < 1.0 \\ 1.0 & \text{if } Z_i(t) = 1.0 \end{cases} \tag{3}$$

From the above constraint, A_j cannot start until A_i is $100\,f_{ij}$ % cpmplete, and the progress of A_j must lag $100\,f_{ij}$ % behind the progress of A_i until the completion of A_i. We now explain the necessary changes in the LP. Between the above activities A_i and A_j, let

l_j^- = time slice in which A_j appears for the time,

l_j^+ = time slice in which A_i appears for the last time,

SB_{ij} = set of time slices in which A_i and A_j both appear.

Then, by adding the following constraints for all activity pairs with overlap relationships to the LP in step 2, we can solve the same scheduling problem with the extended model.

$$\begin{aligned} &\sum_{l \in SL_i, l \leqslant l_j^- - 1} Z_{il} \geqslant f_{ij} \\ &\sum_{h=l_j^-}^{l} Z_{jh} \leqslant \sum_{h \in SL_i,\, h \leqslant l} Z_{ih} + f_{ij} \qquad \text{for all } l \in SB_{ij} \\ &\sum_{l \in SL_i, l \geqslant l_j^+ + 1} Z_{jl} \leqslant f_{ij} \end{aligned}$$

The first constraint ensures that the predecessor is progressed at least by $100\,f_{ij}$ % before the start of the follower. While the two activities are overlapped in the same slice, the cumulative intensity of the follower must not exceed the cumulative intensity of the predecessor plus the lag, which is represented by the second constraint. Finally, after the predecessor is completed, the follower must work on at least $100\,f_{ij}$ % of the total work, as is expressed in the third constraint.

6. Summary and Discussion

We have proposed a two-step heuristic approach to finding a schedule with minimum project duration for a single-project resource-constrained scheduling problem. We first apply the intensity-assignment heuristic proposed by Leachman et al [7], and then improve the schedule by solving a linear programming model. The computational results show that significant improvements are obtained over the single-step intensity assignment heuristic, with substantial reductions in project completion times compared to the traditional fixed-duration solutions. We also introduced the extension of the proposed approach to admit overlap relationships of the activities.

Even though flexible resource loading and overlap relationships between activities are quite common in many project-oriented production systems, relatively little research effort has been directed toward modeling such phenomenon. Most of the commercially available software has implemented the PDM method, yet it is an inaccurate model to represent activity overlap relationships in the case of flexible resource loading (Leachman and Kim [8]).

Several extensions to the proposed model are possible. One attempt to apply it to the multi-project scheduling situation is proposed by Kim and Leachman [5]. It also is possible to consider different objectives, such as minimizing a cost-related function. For example, Padman and Smith-Daniels [10] have proposed discounted cash-flow (NPV) models. In lieu of the traditional time-cost trade-off model, it may be possible to consider a time-resource trade-off problem, in which the total resource allocations are limited.

References

[1] Crandall, K. C. "Project Planning with Precedence Lead/LagFactors," *Project Management Quarterly*, Vol.4, No. 3 (1973), pp.18-27.

[2] Davis, E. W., "Project Scheduling under Resource Constraints-Historical Review and Categorization of Procedures," *AIIE Transactions*, Vol. 5(1973), pp. 297-313.

[3] Davis, E. W., "*Project Management: Techniques, Applications, and Managerial Issues*, Industrial Engineering and Management Press, Institute of Industrial Engineers, 1983.

[4] Hackman, S. T. and R. C Leachman, "A Genearl Framework for Modelling Production," *Managenment Science*, Vol. 25, No. 4(1989), pp. 478-495

[5] Kim, S. and R. C. Leachman, "Multi-Project Scheduling with Explicit Lateness Costs, *IIE Transactions*, Vol. 25, No. 2 (1993), pp. 34-44.

[6] Leachman R. C., "Multiple Resource Leveling in Construction Systems Through Variation of Activity Intensities," *Naval Research Logistics Quarterly*, Vol. 30 No. 3(1983), pp. 187-198.

[7] Leachman R. C., A. Dincerler, and S. Kim, "Resource-Constrained Scheduling of Projects with Variable-Intensity Activities," *IIE Transactions*, Vol. 22, No. 1(1990), pp.31-40.

[8] Leachman R. C. and S. Kim, "A Revised CPM for Networks Including Both Overlap Relationships and Variable-Duration Activities," *European Journal of Operational Research*, Vol. 66(1993), pp. 229-248.

[9] Moder J. J., C. R. Phillips, and E. W. Davis, *Project Management with CPM, PERT and Precedence Diagramming*, 3rd Edition, Van Nostrand Reinhold, New York, 1983.

[10] Padman, R. and D. Smith-Daniels, "Optimization-Guided Heuristics for Maximizing the Net Present Value of Capital Constrained Projects," presented at the ORSA/TIMS national meeting, Anaheim, 1991.

[11] Patterson, J. H., "A Comparison of Exact Approaches for Solving the Multiple Constrained Resource, Project Scheduling Problem," *Management Science*, Vol. 28, No. 7(1984), pp. 854-867.

[12] Talbot, F. B., "Resource-Constrained Project Scheduling with Time-Resource tradeoffs: The Nonpreemptive Case," *Management Science*, Vol. 28, No. 10 (1982), pp.1197-1210.

[13] Weglarz, J., "Project Scheduling with Continuously-Divisible, Doubly Constrained Resources," *Management Science*, Vol. 27, No. 9(1981), pp.1040-1053.

[14] Wiest, J. D., "A Heuristic Model for Scheduling Large Projects with Limited Resources," *Management Science*, Vol. 13, No. 6(1967), pp.1120-1148.

[15] Wiest, J. D., "Precedence Diagramming Methods: Some Unusual Characteristics and Their Implications for Project Managers," *Journal of Operations Management*, Vol. 1, No. 3 (1981), pp.121-130.