An Improved Algorithm for the Exchange Heuristic for Solving Multi-Project Multi-Resource Constrained Scheduling with Variable-Intensity Activities

Jai-Keon Yu (ETRI)
and Won-Kyung Kim (Kyung Nam University)

Abstract

In this study, a modified algorithm for the exchange heuristic is developed and applied to a resource-constrained scheduling problem. The problem involves multiple projects and multiple resource categories and allows flexible resource allocation to each activity. The objective is to minimize the maximum completion time. The exchange heuristic is a multiple pass algorithm which makes improvements upon a given initial feasible schedule. Four different modified algorithms are proposed. The original algorithm and the new algorithms were compared through an experimental investigation. All the proposed algorithms reduce the maximum completion time much more effectively than the original algorithm. Especially, one of four proposed algorithms obviously outperforms the other three algorithms. The algorithm of the best performance produces significantly shorter schedules than the original algorithm, though it requires up to three times more computation time. However, in most situations, a reduction in schedule length means a significant reduction in the total cost.

1. Introduction

1.1. Problem Definitions

This section will be devoted to define, in detail, a Resource Constrained Scheduling(RCS) model that will be dealt with in this research. This particular model will be called the "MPMRCS" (Multiple Project Multiple Resource Constrained Scheduling) model, since it involves multiple projects and multiple resource categories.

1.1.1. Resource Constraints

The scheduling process most often arises in situations where resource availabilities are essentially fixed by the long-term commitments of a prior planning decision [1]. In job-shop scheduling, resources are assumed to be unlimited except machines. That is, it is usually assumed in job-shop environment that all the machines are always available to perform operations. However, it may be an unjustified assumption because, in practice, there may not be enough operators or enough power for all the machines to be working simultaneously.

Resources can be classified into three types, renewable, nonrenewable, and doubly constrained, according to the types of constraints. Renewable resources are used and constrained on a period-by-period basis. That is, constraints on the availability of the resources only concern total usage in every single time unit. The model assumes that all resources are renewable and discrete. Thus, the resource availability for each category is limited to a fixed integer value for each period.

1.1.2. Flexible Resource Requirements

Most models proposed for the RCS assume that an activity must be performed in a prespecified way. That is, each activity consumes certain types of resources at fixed rates and is completed in a fixed duration. However, in reality, many activities are capable of consuming resources at a certain range of degrees. These activities that have flexible resource requirements and variable durations are termed "variable-intensity" activities [4]. The processing time is a function of the amount of resources allocated to it.

1.1.3. Objective Function

The objective of the MPMRCS model is to minimize the maximum completion time. The maximum completion time, or schedule length (SL), is defined as the time, or interval, required to complete all the activities. Minimizing the maximum completion time implies that the cost of a schedule depends on how long the processing system is devoted to the entire set of activities.

1.1.4. Assumptions

The assumptions of the MPMRCS model are as follows:

(1) a set of projects is to be scheduled,
(2) each project must perform a set of predetermined activities,
(3) all activity durations are taken to be integers,
(4) an activity may have flexible resource requirements,
(5) activities cannot be started until specific, preceding activities are completed,
(6) each activity may have multiple predecessors and multiple successors, but looping and dangling of activities are not allowed,
(7) there is no mutual exclusiveness among activities,
(8) activities once started cannot be interrupted,
(9) the amounts of resources are integer-valued,
(10) the minimum amounts of different resources that an activity requires during a period, remain constant throughout the processing of the activity, and
(11) the amounts of resources available during each time period are constant.

1.2. Research Objectives

The research objective of this research is to improve the performance of the EH in terms of the maximum completion time and expand its applicability. Four modified algorithms of the EH are proposed to improve the performance of the EH. To expand the application area of the EH, the assumption of fixed resource requirements is relaxed in the MPMRCS model.

2. The Exchange Heuristic

Since the main research objective is to develop an improved algorithm for the (EH), this heuristic procedure is explained in this chapter. The EH was originally developed by Yang and Ignizio to solve a scheduling problem of army battalion training exercises [8].

The heuristic is essentially the second phase in a two-phase approach. Phase I is concerned with obtaining an initial feasible schedule. Any solution procedure that finds such an initial schedule can be used. Phase II uses the EH in an iterative fashion to make improvements in terms of schedule length.

2.1. Phase I -- Preparing an Initial Schedule

An initial feasible schedule is obtained during Phase I. It is up to the users of the heuristic to find an initial feasible solution for their problems. The heuristic can be applied to any initial schedule as long as it is feasible, no matter which method is used to find it. Initial schedules may be generated preferably by a single pass heuristic which is computationally more efficient than a multiple pass heuristic. To the best of the author's knowledge, all single-pass heuristics are based on priority rules. However, it is conjectured that, when an initial schedule is obtained using a particular priority rule, such as the LPT (longest processing time) or SPT (shortest processing time), the rule prevents some particular blocks in the schedule from having flexibilities that are necessary in the search for obtaining improvements. Flexibility indicates the level of possibilities in exchanging activities within the schedule length. To obtain initial schedules having more flexibility, a random schedule generator (RSGEN) was developed in Yang et al. [9] and employed in this study. The procedure that the generator uses is classified as serial scheduling based on a random rule

2.2. Phase II -- The Exchange Heuristic Algorithm

A survey of heuristic approaches to RCS problems indicates that most researchers have hitherto adopted the forward-loading technique in which activities are scheduled as early as possible [5]. A different approach is the backward loading technique, which schedules activities as late as possible within the current schedule length. For the forward-loading technique, there is a strong tendency that resource consumption levels will become lower along the time horizon, as shown in Figure 1-(a). The EH takes advantage of this tendency. That is, the EH reduces schedule length through the leveling of such uneven resource consumption rates.

Figure 1-(b) and (c) show simplified representations of how the EH achieves the reduction of schedule length. First, the activities in a specific time region, which is called "search region," are rescheduled backward, therefore consuming the unused resources at the later part of the schedule. The resources in the search region are freed by this backward-scheduling and resource leveling is achieved after the search region, as shown in Figure 1-(b). Then, by rescheduling the activities forward that are previously scheduled after the search region, the schedule length may be reduced, as shown in Figure 1-(c).

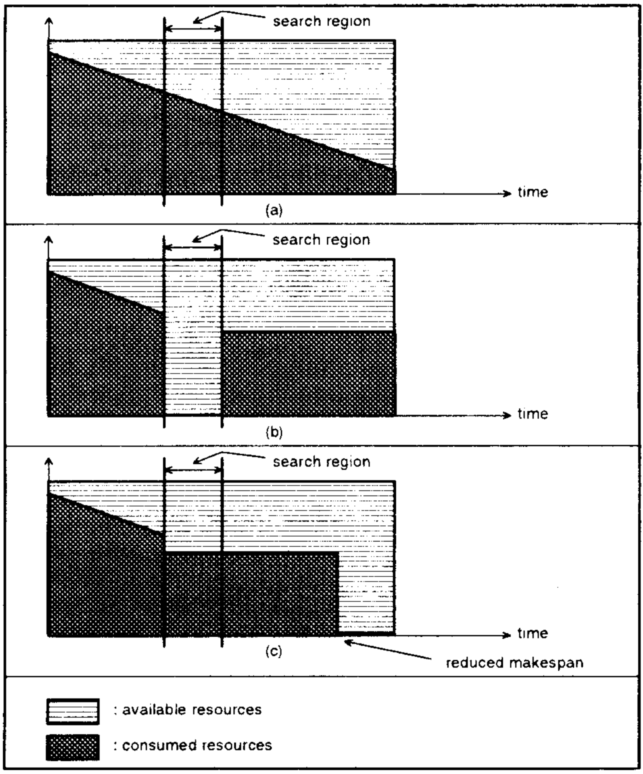

Figure 1. A simplified illustration of how the EH reduces schedule length.

3. Improving the Exchange Heuristic

3.1. Improving the Algorithm Performance

The objective of the Exchange Heuristic is to minimize the maximum completion time (schedule length). That is, the performance of the EH depends on how short a final schedule is. Since the EH is an improvement algorithm, its performance depends on how much it can reduce the length of an initial schedule. The procedures of the EH are examined below to find which steps of the algorithm can be improved. Each iteration of the algorithm tries to achieve some reduction, but it may or may not be successful. Its success strongly depends on the success of the following steps:

Step 1: The resources in the search region should be freed.

Step 2: The activities in and after the search region should be rescheduled earlier to reduce the schedule length.

It is at the second step that the actual reduction of the schedule length is achieved. However, if the first step is not successful, the second step is not even attempted. In other words, the failure of the first step certainly excludes any possibility of reducing the schedule length at the current iteration. The success of the second step totally depends on the first step. It is then believed that the performance of the original algorithm can be improved by modifying the first step.

Therefore, efforts to improve the performance of the heuristic are focused on improving the step of freeing the search region resources. Four different heuristic algorithms are obtained by applying different strategies to the step. All the strategies are fundamentally inspired by the fact that the activities in the search region are frequently blocked by subsequent activities when the former are rescheduled backward to free resources in the search region. Blocking occurs because of precedence constraints and resource conflicts. Note that the EH always maintains feasibility and the constraints are never violated. Thus, subsequent activities also need to be rescheduled backward. The question is then which subsequent activities should be rescheduled. The major difference between strategies lies in selecting subsequent activities to be rescheduled along with the activities in the search region. The four modified algorithms are labeled as EH1, EH2, EH3, and EH4, respectively.

3.2. Extending the Applicability

The EH was applied to a general form of RCS problem [10]. This RCS model assumes fixed resource requirements as do most of the RCS procedures found in the literature except for a few [4, 11]. In this research is being considered a more generalized RCS model which allows flexible resource requirements.

The following defines the relations between resource applications and processing time in the MPMRCS model:

1. Each variable-intensity activity has minimum resource requirements per period to make its processing possible. The amount of minimum resource requirements is fixed.
2. The mix of all the different types of resources which are minimally required by a variable-intensity activity is termed "basic-mix".
3. All the different types of resources which are required by variable-intensity activities are applied proportionally throughout the activity execution. That is, each variable-intensity activity utilizes a multiple of the basic-mix. Thus, the intensity of applications of resources to an activity can be indexed in terms of the number of basic-mixes.
4. The activity intensity is assumed to be an integer value from 1 to the upper limit. Each variable-intensity activity has a fixed upper limit.
5. The rate of progress of an activity is assumed to be proportional to its intensity. That is, by increasing the intensity of an activity, its processing time tends to be reduced. This concept of the relationship between the rate of progress and the intensity of the resources is used by Leachman et al. [4] and Weglarz [7].
6. The total amount of resources required to complete a variable-intensity activity is fixed and indexed in terms of the number of basic-mixes.

The function for the flexible duration is defined as follows:

$$\mathrm{CBM}_j(t) = \mathrm{Min}\left\{ \mathrm{MBM}_j, \left[\frac{\mathrm{AR}_k(t)}{\mathrm{mr}_{jk}} \right] \text{for } k \text{ such that } \mathrm{mr}_{jk} \neq 0 \right\}, \text{ for } j \in \mathrm{J},$$

$$= 1, \text{ for } j \notin \mathrm{J},$$

$$\mathrm{d}_j = \mathrm{Min}\left\{ \mathrm{T} \,\middle|\, \left(\sum_{t=t_0+1}^{t_0+\mathrm{T}} \mathrm{CBM}_j(t) \right) \geq \mathrm{TBM}_j,\ \mathrm{T} = 1, 2, \ldots \right\}, \text{ for } j \in \mathrm{J}, \text{ and}$$

d_j is fixed for $j \notin \mathrm{J}$,

where $\mathrm{AR}_k(t)$ = the amount of unused resources of type k at period t,
$\mathrm{CBM}_j(t)$ = number of basic-mixes consumed by activity j at period t,
d_j = duration of activity j,
MBM_j = number of basic-mixes which can be maximally applied to activity j in a time period,
mr_{jk} = the minimum amount of resources of type k required by activity j,
TBM_j = total number of basic-mixes required to complete activity j,
J = set of variable-intensity activities, and
$[x]$ = the greatest integer which is less than or equal to x.

4. Experimental Study

4.1. Aims

The aims of the experimental investigation were to answer the following questions:

1. Which version of the EH is the most effective at reducing SL ?
2. How much is the SL shortened by modeling the flexibility of resource requirements in scheduling procedures ?

4.2. Measures of Performance

Two measures of effectiveness can be used for each schedule: the schedule length, SL, as defined in section 1.1.4, and the "utilization factor", UF, as given by

$$\mathrm{UF} = \left(\sum_{k=1}^{\mathrm{K}} \frac{\sum_j \mathrm{d}_j \mathrm{mr}_{jk}}{\mathrm{R}_k \cdot \mathrm{SL}} \right) \times \frac{100}{\mathrm{K}},$$

where R_k is the amount of available resources of type k,
K is the total number of resource types.

UF is more useful as a measure of effectiveness across different problems, as it is normalized, dimensionless, independent of work content, and has better statistical properties [3].

4.3. Description of Problems

A scheduling problem is characterized by its size, network structure, and resource structure. The size parameters are the number of projects, activities, variable-intensity activities, and resource types. The structure of the network is characterized by the parameter "order strength" which is the ratio of the actual number of precedence relations to the possible number of precedence relations. The number of different kinds of resources used by activities in a problem is measured by the parameter "resource factor" which is the ratio of the average number of the different kinds of resources used per activity to the total number of the different kinds of resources. The relative quantities of a resource type required by the project, in relation to the amount available, is measured by the parameter "resource strength" which is the ratio of the resource availability to average requirements per activity [2].

Sixteen problems were prepared as shown in Table 1. Problems 4, 6, and 7 are modifications of well-known benchmark job shop scheduling problems from Muth and Thompson [6].

Table 1. Characteristics of the example problems.

Problem Serial No.	1, 10	2, 11	3, 12	4, 13	5, 14	6, 15	7, 16	8	9
No. of Projects	1	4	5	6	6	10	20	10	20
No. of Activities	11	12	12	36	36	100	100	200	400
No. of Var-Int Act.'s[1]	4	4	4	12	12	32	32	0	0
No. of Res. Types	3	3	5	6	9	10	5	10	20
Order Strength (os)	045	0.18	0.14	0.13	0.13	0.08	0.04	0.05	0.05
Resource Factor (rf)	0.94	0.33	0.35	0.26	0.23	0.14	0.39	0.50	0.45
Average Res. Stren.(rs)	3.13	4.89	4.00	7.23	6.82	26.77	10.21	13.39	88.25

4.4. Computational Results

Four modified algorithms were coded in Microsoft FORTRAN v.5.0. These codes were run on IBM 386/25 MHz without a math co-processor. The programs are designed to solve problems that have up to 400 activities and 25 resources types without any variable-intensity activity, or up to 200 activities and 50 resources types with some variable-intensity activities.

For each example problem, 30 random active schedules were generated by the random schedule generator and rescheduled by each of the four modified algorithms to compare their performance. The algorithm EH4 can be varied by using different values of a parameter. The parameter is the maximum number of an extra rescheduling. Three variations of the algorithm were tested by using the parameter values of 1, 2, and 3. Those are labeled as EH4.1, EH4.2, and EH4.3. The original algorithm is also labeled as EH0. If the modified algorithms are ordered by decreasing average utilization factor, they have the

---

[1] This data item is for problems 10~16. Note that problems 1~9 has no intensity-variable activity.

following ranking: EH4.3, EH4.2, EH4.1, EH2, EH3, and EH1. Figure 2 more clearly illustrates the ranking of modified algorithms and the strong tendency that as the parameter value increases, the performance is getting better.

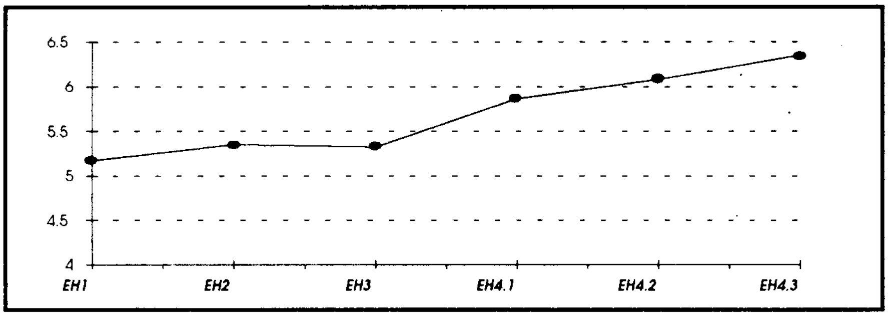

Figure 2. Percentage increases of average utilization factors over the original algorithm.

Table 3 and Table 4 show the percentage difference in schedule length and computing time, respectively, between the EH4.1 algorithm and the original algorithm. The average percentage decreases in the average, minimum, and maximum are up to 20%, 6%, and 30%, respectively. The computing time is up to three times longer for EH4.1.

Table 3. Summary of the performance of the original algorithm vs. EH4.1.

	Percentage decrease over the original algorithm								
Prob. serial No.	1	2	3	4	5	6	7	8	9
Average	0.00	0.00	1.84	6.77	5.68	19.57	2.95	6.41	5.87
Minimum	0.00	0.00	0.00	0.00	0.00	0.00	2.39	0.00	5.69
Maximum	0.00	0.00	5.88	11.29	16.13	29.87	3.56	14.29	6.66

Table 4. Summary of computing time of the original algorithm vs. EH4.1.

	Percentage increase over the original algorithm								
Problem serial No.	1	2	3	4	5	6	7	8	9
Avg. computing time	30.27	-17.50	-22.22	39.22	9.33	62.95	35.16	299.20	316.79

To address the second question which is stated in section 4.1, the maximum completion times for the fixed-intensity cases (problems 1~7) were compared with completion times for the variable-intensity cases (problems 10~16). The results are summarized in Table 5.

The decrease of 7~15 % in the average schedule length is obtained by allowing flexible resource allocation for a third of all activities in problem 6. However, this is not the case for other problems. The reason is that there are not enough resources available to get some benefit from flexible resource allotment. Note that the average resource strength of problem 6 is 26.77 and about 3~10 for other problems.

Table 5. Summary of the average schedule length for variable-intensity cases vs. fixed-intensity cases.

	Percentage decrease over fixed-intensity case						
Problem No.	1,10	2,11	3,12	4,13	5,14	6,15	7,16
Initial Schedules	0.00	5.70	0.00	3.54	0.00	15.29	2.73
Algorithm EH0	0.00	0.00	-0.68	3.38	0.86	14.80	-2.03
Algorithm EH1	0.00	0.00	-2.80	7.00	-0.20	15.04	-1.58
Algorithm EH2	0.00	0.00	-1.59	3.31	0.00	13.30	-2.06
Algorithm EH3	0.00	0.00	-1.61	4.85	-0.52	7.57	-2.67

5. Conclusions and Further Research

5.1. Conclusions

In this dissertation, four different modifications of the original algorithm for the Exchange Heuristic have been proposed for improving its performance in solving a RCS problem. Also, a RCS model, which accommodates multiple projects and multiple resource categories, has been defined and successfully solved by the EH. It is recognized that the algorithm step which free resources in the search region is critical for improving the performance of the EH. The modified algorithms were obtained by modifying the step of the original algorithm.

It was experimentally proved that all the modified algorithms are better than the original algorithm with respect to the schedule length, which is the objective function, but they require more computation time. The algorithms EH1, EH2, and EH3 are discarded because they produce longer schedule lengths and take longer computation time than EH4. The algorithm EH4 is absolutely the best at reducing schedule length as compared to the other algorithms. Furthermore, as more extra rescheduling is performed, the schedule length tends to decrease substantially. The computation times of EH4 are 2~5 times more than other algorithms. However, it is quite likely that the increased computing time is a relatively insignificant factor to the total project cost. On the other hand, the schedule length usually has a serious impact on the total cost.

The MPMRCS model defined in section 1.1 assumes that flexible resource allocation is allowed for variable-intensity activities. The computational results show that the extra reduction in schedule length can be achieved by modeling flexible resource requirements only if the resource strength and the order strength are high enough.

EH4 may be used in two ways. One way is to generate one random schedule and obtain one improved final schedule. The other way is to generate multiple random schedules, obtain a set of improved final schedules, and select the best. It is obvious that the latter way will find a better schedule. Thus, it is recommended that the latter method be used, when the computing time is not a significant cost factor.

5.2. Further Research

The performance of a heuristic can be affected by such problem characteristics as size, network structure, and resource requirements/availability. The relationship between the performance of the EH and some parameters, such as those defined in section 4.3, may be found for RCS problems. It can be done by performing extensive experiment and thus gathering computational results on more test problems of various characteristics.

Though the MPMRCS model has generous assumptions, it still has some nonrealistic restrictions such as renewable resource types, fixed resource availabilities, discrete time, discrete resource amounts, and no mutual exclusiveness. It is strongly felt that by further research these limitations can be removed to expand the applicability of the EH. There are a variety of real-world applications of the MPMRCS model which urge further extensive and rigorous study.

References

[1] Baker, K.R. (1975) *Introduction to Sequencing and Scheduling*. John Wiley & Sons, New York.

[2] Cooper, D.F. (1974) A Computer Analysis of Some Project Scheduling Heuristics. *Proceedings of the Sixth Australian Computer Conference*, 11-24.

[3] Cooper, D.F. (1976) Heuristics for Scheduling Resource-Constrained Projects: An Experimental Investigation. *Management Science*, **22** (11), 1186-1194.

[4] Leachman, R.C., Dincerler, A., and Kim, S. (1990) Resource-Constrained Scheduling of Projects with Variable-Intensity Activities. *IIE Transactions*, **22** (1), 31-40.

[5] Li, K.Y. and Willis, R.J. (1992) An Iterative Scheduling Technique for Resource-Constrained Scheduling. *European Journal of Operational Research*, **56**, 370-379.

[6] Muth, J.F. and Thompson, G.L. (1963) *Industrial Scheduling*, Prentice-Hall, Chapter 21 by J.E. Kelley, Jr.

[7] Weglarz, J. (1981) Project Scheduling with Continuously Divisible, Doubly Constrained Resources. *Management Science*, **27** (9), 1040-1053.

[8] Yang, T. and Ignizio, J.P. (1987) An Algorithm for the Scheduling of Army Battalion Training Exercises. Computers and Operations Research, **14** (6), 479-491.

[9] Yang, T. and Ignizio, J.P., and Song, J. (1989) An Exchange Heuristic Algorithm for Project Scheduling with Limited Resources. *Engineering Optimization*, **14**, 189-205.

[10] Yang, T., Ignizio, J.P., Yoo, J., and Santos, D. (1992) An Exchange Heuristic for Resource Constrained Scheduling. *Engineering Optimization* (accepted upon revision).

[11] Yoo, J., Yang, T., and Ignizio, J.P. (1991) An Exchange Heuristic for Resource Constrained Scheduling with Consideration given to Opportunities for Parallel Processing. *Production Planning and Control* (under review).