International Journal of Soft Computing 1(4): 255-260, 2006
© Medwell Online, 2006

Using Genetic Algorithms to Find Weights for Multiple Heuristic for The Stochastic Resource Constrained Project Scheduling Problem

Mahdi Nasereddin
Pennsylvania State University Berks Campus P.O. Box 7009 Reading, PA 19610

Abstract: The focus of this study is on resource constrained project scheduling with stochastic task durations. In the extensive research performed in project scheduling, little research has been done with projects that have stochastic activity durations. In this study, we explore combining two priority rule based heuristics (Longest Activity First (LAF) and Greatest Resource Demand (GRD) using weights assigned to each heuristic. The heuristics are then used to schedule the project activities. Genetic Algorithms (GA) are used to find the optimal weights on the heuristics. The GA search was compared to both random and interval searches. Two performance measures were used: average percent deviation from the best mean project duration found by the enumerative search and average percent deviation from the best variance found by the enumerative search. An experimental analysis was conducted to evaluate the performance of the three approaches. A full factorial design with 10 replications was used in this evaluation. It was found that the interval search performs better than the random search, which in turn performs better than the GA.

Key words: Genetic algorithms, project scheduling, priority rule based heuristics

INTRODUCTION

The PERT/CPM approach to solving the project scheduling problem does not take into account resource constraints and the priority rule heuristics are designed for deterministic problems. There exists a need for an efficient technique that solves the stochastic resource constrained project scheduling problems (SRCPSP). The technique should take into account that there are limited resources and that activity durations are stochastic (there exists some uncertainty in how long it will take to finish an activity). For real life large scale projects, it takes a long time to solve the deterministic problem optimally; consequently combinations of heuristics are often used. An exhaustive search over weights on heuristics is currently used to find the best weighted combination of a set of heuristics.

The usefulness of using genetic algorithms to search over the weights on heuristics to solve the SRCPSP has not been investigated. In this study the usefulness of this approach is investigated and compared to other techniques; namely, random search and interval search. The measure of usefulness is the percent deviation from the best solution and is evaluated as a function of the number of iterations.

There exists a large body of literature addressing the Resource Constrained Project Scheduling Problem (RCPSP). Most of this research does not take into account the stochastic nature of project scheduling. Ozdamar and Ulusoy[1] listed 83 articles that contained different approaches to the resource constrained project scheduling problem with deterministic activity durations. Using Meta-heuristics for the deterministic SRCPSP has been used by many researchers[2-4] and most notably the work done by Hartmann and Kolisch[5]. The authors evaluated the use of state of the art meta-heuristics; namely Genetic Algorithms, Simulated Annealing and Tabu Search for the deterministic SRCPSP. They found that Simulated Annealing and Genetic Algorithms to give the best results.

When introducing variance to the deterministic problem, we end up with a stochastic problem which is more complex to solve. The complexity is introduced by adding another factor (probability). Less research has been done in this area than has been done in the deterministic area. Brucker *et al.*[6] reviewed some of the latest development (up to 1999) for both the stochastic and deterministic RCPSP.

Pet-Edwards and Mollaghasemi[7] introduced the notion of using GA to search over weights on scheduling heuristics solutions. Although an experiment was not performed to determine the efficiency of the GA, the study discussed the methodology of how such an experiment is to be performed.

Priority rule based heuristics used in the literature: Efficient mathematical programming models have been developed to solve the small deterministic RCPCP. These models proved to be inefficient to solve practical sized problems. The computational complexity of using optimal mathematical methods to solve the project scheduling becomes very prohibitive for large projects. This need has motivated researchers to develop effective heuristics. Heuristics are classified into two main categories: Non-Priority Rule Heuristics and Priority Rule Based Heuristics. Some of the optimal methods can be used as Non-Priority Rule Based Heuristics. Other Non-Priority Based Heuristics focus on sophisticated search techniques (Brown,[8]).

In terms of how a heuristic calculates the priority associated with each activity, Priority Rule Based Heuristics are classified into two types:

- Serial (static) heuristics where the priorities for all the activities are calculated once and remain constant throughout the scheduling process. Most heuristics found in the literature are serial heuristics.
- Parallel (dynamic) heuristics where the priorities of all the activities to be scheduled are recalculated each time an activity is scheduled. Of the heuristics found in literature, only the Hybrid Solution Procedure (HSP) heuristic is dynamic.

Brown[8] and Morse and McIntosh[9] surveyed priority rule based heuristics found in the literature. Some of the heuristics are resource based (Activity Resource, ACROS, Multiple Greatest Resource Demand, Greatest Resource Demand and Greatest Resource Utilization). Some heuristics are time based (Longest Activity First, Shortest Job First, Late Finish Time, Minimum Early Finish, Minimum Slack First, Maximum Slack First, Activity Time, Multiple Greatest Resource and Demand, Resource Scheduling Method). Some heuristics are a combination of resource and time based (Most Jobs Possible, TIMRES, GENRES and Priority Search Rule Technique). Others use other factors like activity number and random calculation to calculate the priorities (First Come First Served and Random Activity Selection). Only one heuristic (Stochastic Activity Factor) utilizes information about the variance of the task durations.

In the literature the combination of stochastic activity durations, multiple objectives, heuristics and GA has not been investigated. This study investigates all of the above to investigate the efficiency of using GA to find weights on heuristics for the SRCPCP.

Experimental design: The study of the experiment is to compare the performance of the Genetic Algorithms (GA) with an interval search and a random search for finding the best weights on heuristics for the SRCP. To accomplish this goal, 480 scheduling networks are used. The networks were created by R. Kolisch[10]. The Kolisch networks are scheduled using a combination of two heuristics (Longest Activity First (LAF) and Greatest Resource Demand (GRD)). The GA is used to assign weights to the two heuristics used.

The two heuristics were combined using the following formula:

LAF + GRD
WM(I)*Activity Duration + (W2*Total Resources Consumed by activity I) /4
where W2 = 1 - WM(I), WM(I) is the weight associated with LAF.

The above formula is normalized by dividing the GRD part by 4, because the maximum an activity duration can be is 10 and the maximum resources consumed by an activity is 40, thus a four to one ratio exists creating the need to normalize the formula.

The factors used to analyze the problem are Complexity Factor (CF), Resource Factor (RF), Resource Strength (RS), Method of evaluation (M) and the Evaluation Criteria (EC). CF, RF and RS were used by Kolisch to generate his benchmark network set. The other two factors were used to satisfy the requirements of the goals of this experiment (to evaluate the efficiency of genetic algorithms to find optimal weights on heuristics for stochastic resource constrained project scheduling problems).

Kolisch's full factorial model: Kolisch designed a full factorial experimental design which is used in this study to evaluate the GA's performance. The design consists of three factors which are:

- The network Complexity (CF) which is defined as the average number of non redundant arcs per node. The network complexity was first introduced by Pascoe for AOA networks and later adopted by Davis for the AON representation. The network complexity has three levels (1.5, 1.8, 2.1). The more complex the network is, the higher the value of CF by Kolisch, Sprecher and Drexel[11].
- The Resource Factor (RF) reflects the average proportion of resources requested per job. The resource factor was first introduced by Pascoe in 1966. An RF = 1 means that each job requests all resources and an RF=0 means that each job requires no resources. The resource factor has four levels (0.25, 0.5, 0.75, 1.0)[11].

- The Resource Strength (RS) reflects the relationship between the resource demand of the jobs and the resource availability. The resource strength was first introduced by Cooper in 1976. The resource strength has four levels (0.2, 0.5, 0.7, 1) .

The design contains 10 replications thus generating ten projects for each combination of CF, RF and RS. This results in a total of 3x4x4x10 = 480 instances.(Kolisch, Sprecher and Drexel,[11]).

In this study, to perform the analysis efficiently, five FORTRAN programs were developed. The main program is called PSA (Project Scheduling Algorithms). PSA schedules the activities based on the two combined heuristics using 100 different weights (0.01-1.0 using a step of 0.01), thus creating one hundred solutions for each network. One hundred replications are used to estimate the mean project duration and the project variance. A total of 10,000 runs (100 weights x 100 replications) are used for each network. The program ran for 40 hours and generated 480 files. Each file contains the 100 values of the mean project duration and variance for each of the weights.

How does PSA work?: The main steps taken by PSA are:

Kolisch network read: A file containing a Kolisch network is opened and the data within the file is stored

Initialization: A 0.01 is assigned to WM(1) (weight assigned toLAF), then W2 (weight assigned to GRD) is calculated as "W2 = 1-WM(1)". The time counter is reset to zero. All eligibility values "E(I)" are set to a zero value (E(I) = 0 means that the activity is not yet eligible to be scheduled due to precedence relationships), except for activity 1 which is given an eligibility value of 3 (Activity accomplished). Other variables are also initialized.

Priorities calculation: Priorities are calculated for all activities based on the two heuristics used (LAF and GRD) and the two weights assigned to them.

Variance calculation: The variance for the mean provided by the Kolisch network is calculated. The probability distribution function of each activity's duration is a step uniform distribution function (Fig. 1). The function is divided into two uniform functions having equal probabilities of occurring. The area to the left of the mean ($\overline{X}$) has a lower bound (a) equal to the mean minus 10 percent of the mean. The area to the right of the mean has an upper bound (b) located between 10 to 30% of the activity mean. The range value for the area located to the

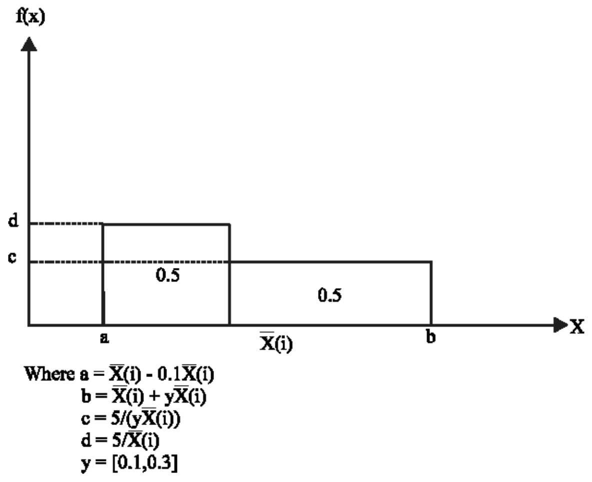

Fig. 1: The step uniform probability distribution used

right of the mean depends on how many resources the activity requires in order to be accomplished. The more resources the activity requires, the larger the range.

Eligibility calculation: In this part eligibility of all activities (precedence relationship check) is examined. E(I) is the activity eligibility index: a value of 0 is assigned for activities that are not eligible for service, a value of 1 is assigned for eligible activities, a value of 2 is assigned for activities in progress and a value of 3 is assigned for completed activities.

Highest priority activity is determined: In this part the eligible (both precedence, resource eligible) activity, X , with the highest priority is determined and scheduled.

Resource allocation: Resources are allocated to the activity X and its eligibility index is given a value of 2.

Resource release: The timer is advanced and resources of the finished activity are released. The activity's eligibility index (E(I)) of the activity accomplished is given a value of 3.

Project duration is determined and stored: After all activities have been scheduled, the project duration is determined and stored.

Mean project duration and project variance are calculated and stored: Mean project duration and variance are calculated using the results from the 100 runs using the same weights, but different random numbers to generate the value of activity durations.

- WM(I) is incremented by a 0.01.
- Process repeated for one hundred weights (until WM(I)=1)

The analysis programs: The other FORTRAN programs created are:

Eanalysis: This program reads the files generated by PSA and for each network finds the weights that generate the best mean and variance.

Sanalysis: This program reads the files generated by PSA and then compares the means and variances of specific weights. The weights are equidistant from each other. When the program is started, the user is asked to input the number of weights (4, 8, 12, 16 and 20) to be considered.

Ganalysis: The genetic algorithm program: In this program the following occurs:

- Four weights (chromosomes) are randomly generated and evaluated using the project duration or variance.
- The four chromosomes are transformed from decimal format to binary format.
- The mating pool: In this section the pair of chromosomes that will mate are established using a probability rule.
- Crossovers and Mutation: Chromosomes are crossed and mutated using a probability rule.
- Out of the eight chromosomes (four original chromosomes and four new crossed and mutated chromosomes). The best four are selected.

Ranalysis: This program reads the files generated by PSA and then compares the means and variances of specific weights. The weights are randomly generated with the help of the RAND(I) subroutine. When the program is started, the user is asked to input the number of weights (4, 8, 12, 16 and 20) to be considered.

Experimental design used: The Kolisch full factorial problem set with replications was used. It included the three factors described earlier where CF had three levels and both RF and RS had four levels. Two additional factors were added:

- The method used to solve the network problem (M). Three methods were used: random search, interval search and using genetic algorithms.
- The objective function (O). Two objectives were used: minimizing the project mean duration (PMD) and minimizing the project variation (PVar).

Adding the two factors increases the number of runs to 480x3x2 = 2880 and each of the 480 networks requires 100 simulation runs to compute the mean and variance.

The experiment: The main program PSA was run. It generated 480 files consisting of the results from the 100 simulations of each network. Each file contained the means and variances for all 100 weights used. The analysis programs (EANALYSIS, GANALYSIS, SANALYSIS and RANALSYSIS) were then run.

EANALYSIS (Enumerative Analysis Program) took the output files of PSA and then found the smallest mean and variance for each network and stored it in an output file. The program considered all 100 weights when doing the analysis.

GANALYSIS (Genetic Algorithms Analysis Program) took the output files of PSA and then using GA with a particular number of weights (8, 12, 16 and 20) which is inputted when the program is first started, found the smallest mean and variance for each network and stored it in an output file. GANALYSIS was run for 1, 2, 3 and 4 iterations (8, 12, 16 and 20 weight values, respectively).

SANALYSIS (Step Function Analysis Program) took the output files of PSA and only the results from particular weights according to a step function were included. The program then found the smallest mean and variance generated by these weights. The number of weights was entered when the program was started. The step function determined which weights are to be included using the formulas:

$\text{Weight}_k = (1 + \text{INT}(100/(R-1))*(k-1))/100$ for $k = 1, R$

where INT() is a function that takes the integer part of what is between the parenstudy and R is the number of weights to be considered. For example for 4 weights, R=4 for k=1,4:

$\text{Weights}_1 = (1 + (\text{INT}(100/3)*(1-1))/100) = 0.01$
$\text{Weights}_2 = (1 + (\text{INT}(100/3)*(2-1))/100) = 0.34$
$\text{Weights}_3 = (1 + (\text{INT}(100/3)*(3-1))/100) = 0.67$
$\text{Weights}_4 = (1 + (\text{INT}(100/3)*(4-1))/100) = 1.00$

SANALYSIS was run for 4, 8, 12, 16 and 20 weights.

RANALYSIS (Random Analysis Program) took the output files of PSA and only analyzed randomly selected weights and then found the smallest mean and variance generated by these weights. The number of weights was an input to the program. RANALYSIS was run for 4, 8, 12, 16 and 20 weights.

RESULTS

After running the programs, data about enumerative, random, interval and GA searches were available. For the analysis, two objectives needed to be accomplished:

- Determine the Effect of the Weights on the Performance of the Heuristic (GRD and LAF): Is there a relationship between changing the weights and the mean and the variance?
- Compare the Three Methods Used (random, interval and GA search techniques): Which method(s) result in smaller deviation of mean and/or variance from the best solution?

Effect of the weights on the performance of the heuristic (GRD and LAF): The relationship between the weights and the performance of the heuristic was examined. Forty-eight graphs were generated representing all combinations of complexity, resource factor and resource strength. The graphs were generated using the data obtained after running the enumerative search analysis program (EANALYSIS).

Some of the functions were monotone and the weights did not affect the mean or the variance. In some networks, both the mean and variance were affected by the change in weights, but most networks changing the value of the weights affected the variance but not the mean.

It was expected that the GA would perform better than random and interval searches for non monotone responses. For monotone responses, random and interval search are expected to do as well as the GA search. The reason is that many weights will give the same optimal response, thus random and interval searches have a better probability of obtaining that optimum solution.

Comparison of the three methods used: The GA was compared to interval search and random search. Two evaluation criteria were used

- The percent deviation in the project mean duration from the best enumerative search result. The percent deviation in the project mean duration = ((best mean achieved by a search technique (GA, interval, or random))- (best mean achieved by an enumerative search)) /(best mean achieved by an enumerative search)
- The percent deviation in the variance from the best enumerative search result. The percent deviation in the variance = ((best variance achieved by a search technique (GA, interval, or random))- (best variance achieved by an enumerative search)) /(best variance achieved by an enumerative search).
- The data was obtained by running the analysis programs (GANALYSIS, EANALASIS, SANALYSIS and RANALYSIS) for 4, 8, 12, 16 and 20 different weights. Overall it was found that GA performed worse than random and interval searches. The best search technique was the interval search for both the mean (Fig. 2) and the variance (Fig. 3).

Graphs that contain only the data for a certain number of weights (4, 8, 12, 16, or 20) for the different search techniques were also generated.

Implications of this research: In order for GA to be effective, a bigger solution space is needed. For this project scheduling problem, the problem space is too small (100 points) for GA to be effective. Thus GA did not perform well because of the small solution space.

Currently, most researcher use a weight step size of 0.1 when using multiple heuristics to find the best set of weights. From the 48 generated graphs (Fig. 4 fro example chart), this step size value appears to be good for the mean, but appears to be small for the variance. It can be seen that a lot of variations occurs within the 0.1 step, thus a smaller step of 0.01 would be more practical.

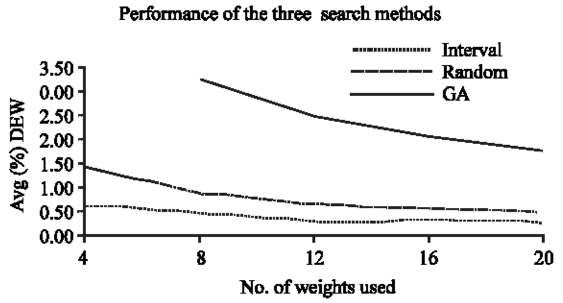

Fig. 2: Average percent deviation in the mean for all the runs

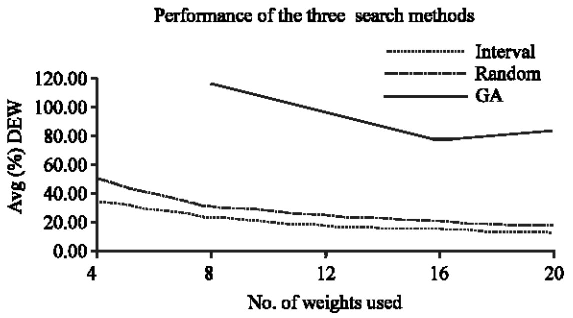

Fig. 3: Average percent deviation in the variance for all the runs

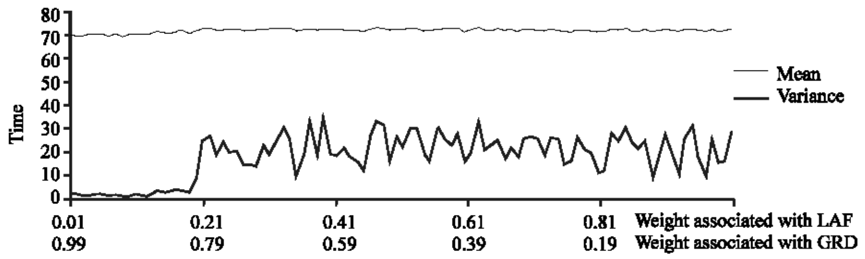

Fig. 4: An example of a non monotonous variance and a monotone mean

CONCLUSION

In this study, the efficiency of GA to find the optimal weights on heuristics for the stochastic resource constrained project scheduling problem was evaluated. The GA search was compared to both random and interval searches. Two performance measures were used: average percent deviation from the best mean project duration found by the enumerative search and average percent deviation from the best variance found by the enumerative search. An experimental analysis was conducted to evaluate the performance of the three approaches. A full factorial design with 10 replications was used in this evaluation.

It was found that the interval search performs better than the random search, which in turn performs better than the GA

REFERENCES

1. Ozdamar, L. and G. Ulusoy, 1995. A survey on the resource--constrained project scheduling problem. IIE Transactions, 27: 574--586.
2. Kim, K.W., M. Gen and G. Yamazaki, 2003. Hybrid genetic algorithm with fuzzy logic for resource-constrained project scheduling, Applied Soft Computing, pp:174-188 .
3. Viana, A. and J. de Sousa, 2000. Using metaheuristics in multiobjective resource constrained project scheduling, European J. Operational Research, pp: 359-374 .
4. Debels, D., B. De Reyck, R. Leus and M. Vanhoucke, 2006. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling, European J. Operational Research, pp: 638-653 .
5. Hartmann, S. and R. Kolisch, 2000. Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem, European J. Operational Research, pp: 394-407 .
.6. Brucker, P., A. Drexl, R Möhring, K. Neumann and E. Pesch, 1999. Resource-constrained project scheduling: Notation, classification, models and methods, European J. Operational Research, 112: 3-41 .
7. Pet-Edwards, J. and M. Mollaghasemi, 1995. Resource Constrained Project Scheduling using Genetic Algorithms, Proceedings of the IEEE Conference on Systems, Man and Cybernetics, Vancouver, B.C.
8. Brown, J.T., 1995. Priority Rule Search Techniques for Resource Constrained Project Scheduling, Unpublished Ph.D. Dissertation, University of Central Florida, Orlando, Fl.
9. Morse, L.C. and J.O. McIntosh, 1995. Using Heuristics to Schedule Activities of Constrained Multiple Resource Projects, Unpublished Paper, University of Central Florida, Orlando, Fl.
10. Kolisch, R., 1995. Project Scheduling Under Resource Constraints: Efficient Heuristics for Several Problem Cases, Physica-Verlag, Berlin Heidelberg.
11. Kolisch, R. and S. Hartmann, 2005. Experimental investigation of heuristics for resource-constrained project scheduling: An update, European Journal of Operational Research, In Press, Corrected Proof, Available online.