Resource Constrained Multi-Project Scheduling using GA - Slack Mode

Siva Vidya.S[#1], Suguna.M[#2], Sharmila.D[#3]

[#1]PG Scholar
Department of Information Technology
SNS College of Technology, Coimbatore -641035
vidya.nce@gmail.com

[#2]Associate Professor
Department of Information Technology
SNS College of Technology, Coimbatore -641035
suguna.marappan@gmail.com

[#3]Professor & Head EIE Department,
Bannari Amman Institute of Technology,

Abstract

The resource-constrained project scheduling problem (RCPSP) is a well-known NP-Hard problem in scheduling. The aim of this project is to present a new genetic algorithm using slack mode for large scale multiple resource-constrained project-scheduling problems. The literature reveals that previous researches have developed numerous scheduling methods and techniques to overcome the complex nature of this problem. However, there are still no promising methods that guarantee optimal solutions as well as computational feasibility. Genetic Algorithms (GA) are very promising Artificial Intelligence approaches to this problem in terms of the computational feasibility and the quality of solutions. However, the most common models of GA are very difficult to implement in scheduling problems. On the other hand, using specific and proper design of GA can make scheduling problems tractable.

Keywords: ***Project Management, Resource Constrained Project Scheduling, Genetic Algorithm ,Slack mode***

I. INTRODUCTION

In project scheduling problems, a single project consists of a set of tasks, or activities. The tasks have precedence relationships, i.e. some tasks cannot be started until their predecessors have been completed. The tasks also have estimated durations and may include various other measures such as cost. Perhaps the most common objective in the project scheduling problem is to minimize the time to complete the entire project. Many specializations of the project scheduling problem have been defined. In resource constrained project scheduling problems, the tasks have resource requirements and the resources are limited. In multi-modal resource-constrained project scheduling problems, each task may be executed in more than one mode, and each mode may have different resource requirements. In multi-project scheduling problems, more than one project must be scheduled. Resource Constrained Project Scheduling Problem (RCPSP) consists on executing a group of activities limited by constraints. Precedence relationships force to some activities to begin after the finalization of others. In addition, processing every activity requires a predefined amount of resources, which are available in limited quantities in every time unit. The RCPSP purpose is to find the starting times for the activities in a way that the difference between the finalization time of the latest activity and the start of the earliest sequenced activity is minimized. RCPSP can be described as:

A project consists of a set of activities V= (1, 2, …, n). The activities 1 and n are dummy activities that represent the project start and end, respectively. Activities are related by two types of constraints. First, precedence relations are “finish-start” (FS) type that is today that j activity may just be started when i activity is concluded. For each activity j there is a Pred j set of activities that must be finalized before j can be started. The performance criteria are numerous: minimize schedule length (makespan); minimize mean (weighted) flow time; minimize mean or maximum lateness or tardiness (lateness is the difference between a job's completion time and its due date - the lateness for an early job

© 2013 IJAIR. ALL RIGHTS RESERVED

being negative; when a job is completed after its due date, it is tardy - tardiness being the maximum of zero and the lateness); minimize the number of tardy jobs; maximize throughput (number of jobs completed per time unit), etc.. Sometimes combined scheduling criteria are used: minimize mean Row time subject to no jobs late, search for the shortest mean flow time schedule, search for a schedule in which no job is early nor tardy (just-in-time), etc.. Problems can be studied in a static environment (all jobs simultaneously available) or in a dynamic environment (jobs have unequal ready times). Problems may be considered to be deterministic or stochastic.

Over the years, several (irrealistic) assumptions of the basic machine scheduling problems have been relaxed. A natural extension involves the presence of additional resources, where each resource has a limited size and each job requires the use of a part of each resource during its execution. This leads us to the area of resource-constrained project scheduling which covers again a tremendous variety of problem types. Certain types of resources are depleted by use (e.g. nonrenewable resources such as money and energy). Resources may be available in an amount that varies over time in a predictable manner (e.g. seasonal labor) or in an unpredictable manner (e.g. equipment vulnerable to failure).

Resources may be shared among several jobs, and a job may need several resources. The resource amounts required by a job may vary during its processing, and the processing time itself could depend on the amount or type of resource allocated, as in the case of the above mentioned uniform or unrelated parallel machines. Over the past few years, extensive research efforts have resulted in new results on the level of problem classification and complexity, and new optimal and suboptimal solution approaches. The elements of Pred j are called j immediate predecessors.

Resource management problems in project networks appear in a wide variety (Herroelen (1972), Herroelen and Demeulemeester (1992)). One of the best known problems is the problem of scheduling project networks subject to resource constraints. The classical resource-constrained project scheduling problem (RCPSP) involves the scheduling of a project to minimize its total duration subject to zero-lag finish start precedence constraints of the PERVCPM type and constant availability constraints on the required set of renewable resources.

II. SCOPE AND PURPOSE

Project scheduling has attracted an ever growing attention in recent years both from science and practice. It is concerned with single-item or small batch production where scarce resources have to be met when scheduling dependent activities over time. Project scheduling is important for make-to order companies where the capacities have been cut down in order to cope with lean management concepts. Project scheduling is very attractive for researchers also, because the models in this area are rich in the sense that many well-known optimization problems are special cases of the more general project scheduling models. For instance, the resource-constrained project scheduling problem contains the job shop scheduling problem as a special case. Without surprise, project scheduling problems in general are really challenging from a computational point of view.

Both practice and science of project scheduling have evolved fast recently, producing numerous acronyms to distinguish between different problem classes. Also, a variety of symbols are used by project scheduling researchers in order to denote one and the same subject. Hence, sometimes it is difficult to keep a clear view of what the subject is all about, because the models in this area are not standardized. Unfortunately, their scheme is not compatible with what is commonly accepted in machine scheduling. Hence, there is still a gap between machine scheduling on the one hand and project scheduling on the other with respect to both, viz. a common notation and a classification scheme. One purpose of our paper is to close this gap. We provide a classification scheme, i.e. a description of the resource environment, the activity characteristics, and the objective function, respectively, which is compatible with machine scheduling and which allows to classify the most important models dealt with so far. Also, we propose a unifying notation.

Project scheduling is of great practical importance and its general model can be used for applications in product development, as well as production planning and a wide variety of scheduling applications. Early efforts in project scheduling focused on minimizing the overall project duration (makespan) assuming unlimited resources. Well-known techniques include the Critical Path Method (CPM) and the Project Evaluation and Review Technique (PERT) . Scheduling problems have been studied extensively for many years by attempting to determine exact solutions using methods from the field of operations research .It was earlier shown that the scheduling problem subject to precedence and

© 2013 IJAIR. ALL RIGHTS RESERVED

resource constraints is NP-hard , which means that exact methods are too time-consuming and inefficient for solving the large problems found in real-world applications.

late 1950s the development of PERT and CPM techniques allowed projects to be described by network diagrams where jobs or activities are defined by the network structure. However, project scheduling with these techniques only deals with the time aspect without consideration of resource restrictions. Thus delays in the execution time of certain activities occur when resources required by these activities are not available in sufficient quantities.

A.Heuristic methods

Because of the inherent difficulty of the problem, exact methods can only be useful for small problems. In industry, people need to solve large-scale project scheduling problems quickly. This makes many researchers turn to heuristic approaches. Here, by heuristic we mean a method that creates “good enough” rather than optimal solutions. These heuristic methods are usually computationally efficient and use much less computational time than exact methods, but there is no guarantee that they will find optimal solutions. Basically, finding the near optimal results and using limited computation time are the major characteristics of any heuristic algorithms.

B.Simulated Annealing

Simulated Annealing (SA) originates from the physical annealing process, in which a melted solid is cooled down to a low-energy state. Starting with some initial solution, a so-called neighbor solution is generated by slightly perturbing the current one. If this new solution is better than the current one, it is accepted, and the search proceeds from this new solution. Otherwise, if it is worse, the new solution is only accepted with a probability that depends on the magnitude of the deterioration as well as on a parameter called temperature. As the algorithm proceeds, this temperature is reduced in order to lower the probability to accept worse neighbors.

C. Ant Colony Algorithm

Ant Colony Optimization studies artificial systems that take inspiration from the behavior of real ant colonies and which are used to solve discrete optimization problems. It is well known

III. RELATED WORKS

It is well known that before computers were used in project scheduling, managers did the project scheduling manually. This was time consuming and could not guarantee optimal solutions. In the that the main means used by ants to form and maintain a path is a pheromone trail. Ants deposit a certain amount of pheromone while walking, and each ant probabilistically prefers to follow a direction rich in pheromone rather than a poorer one. This elementary behavior of real ants can be used to explain how they can find the shortest path that reconnects a broken line after the sudden appearance of an unexpected obstacle in the initial path. In fact, once the obstacle has appeared, those ants which are just in front of the obstacle cannot continue to follow the pheromone trail and therefore they have to choose between turning right or left.

IV.PROPOSED GENETIC ALGORITHM

Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. Genetic algorithms were formally introduced by Holland (1975). Continuing improvements have made Genetic algorithms attractive for many types of optimization problems. “In particular, GA work very well on mixed (continuous and discrete), combinatorial problems. They are less susceptible to becoming 'stuck' at local optimal compared with some other types of optimization techniques” .

Genetic algorithms are inspired by Darwin's theory of evolution of survival of the fittest. Simply stated, problems are solved by an evolutionary process resulting in a best (fittest) solution (survivor). To use a genetic algorithm, a solution of the problems must be represented as a genome (or chromosome). The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one(s). Crossover and Mutation operators play a central and dominant role in all aspects of GA

A. Genetic Representation

In 1866 Mendel recognized that in nature the complete information for each individual are in pair wise alleles. The genetic information that determines the properties, appearance, and shape of an individual is stored in Chromosomes (David 2002). A chromosome describes a string of a certain length where all the genetic information of an individual is stored. Although nature often uses

© 2013 IJAIR. ALL RIGHTS RESERVED

more than one chromosome, most GA only use one chromosome for encoding the genotypic information.

Each chromosome consists of many alleles. An allele is the smallest information unit in a chromosome. A gene is a region on a chromosome that must be interpreted together and which is responsible for a specific phenotypic property (David 2002).

B. Crossover Operator

Crossover is inspired by the role of sexual reproduction in the evolution of living things. GA attempts to combine elements of existing solutions in order to create a new solution, with some of the features of each "parent." The elements (e.g. decision variable values) of existing solutions are combined in a "crossover" operation, inspired by the crossover of DNA strands that occurs in reproduction of biological organisms. As with mutation, there are many possible ways to perform a crossover operation.After we have decided what encoding we will use, we can proceed to crossover operation. Crossover operators operate on selected genes from parent chromosomes and create new offspring.

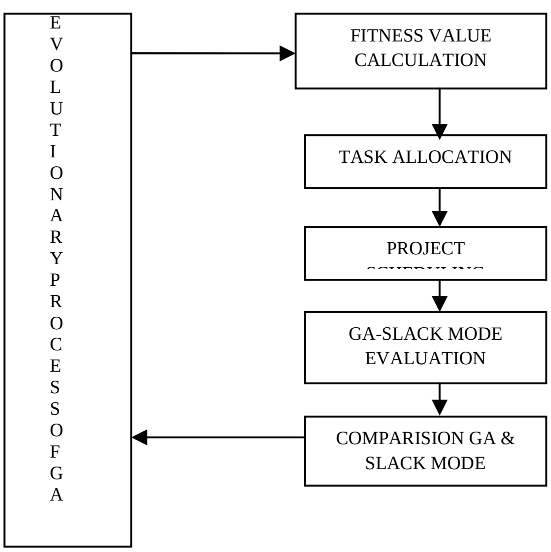

Figure 1: Block Diagram for GA-Slack Mode

The simplest way to do that is to randomly choose some crossover points and copy everything before these points from the first parent and then copy everything after the crossover point from the other parent. There are other ways to make crossovers. For example we can choose more crossover points. Crossover can be quite complicated and depends mainly on the encoding of chromosomes. Specific crossovers made for a specific problem can improve performance of the genetic algorithm. Generally they can fit into one or more of the following 4 principal types as their crossover points

C.Mutation Operator

Mutation is inspired by the role of mutation of an organism's DNA in natural evolution. An evolutionary algorithm operator periodically makes random changes or mutations in one or more members of the current population, yielding a new candidate solution (which may be better or worse than existing population members). There are many possible ways to perform a "mutation". The mutation Operator suddenly changes some parts of the offspring

D. Fitness Value Calculation

The objective function to minimize is the ending time of all activities which is makespan (*Cmax*). The fitness function is calculated as 1 / *Cmax*

V. EXPERIMENTAL RESULTS

It seems that the proposed algorithm is a good compromise which utilizes the potentials of GA-Slack Mode and genetic algorithms. A set of experiments are conducted in order to investigate the performance of the proposed algorithm. The experiments are divided in two categories. The first category evaluates the proposed method in comparison with the GA algorithms proposed by the authors in their previous works in terms of success rate. The second category investigate the performance of the proposed method against other heuristic algorithms in terms of average deviation. The MPRCPSP instances were solved by using a program built in Java language. The program was run in a Pentium IV processor of 3.0 Mhz speed.

VI. DISCUSSION

In general, the results demonstrate that the GA-Slack Mode method has good performance on the Multi Project Resource Constrained Project Scheduling Problem. The GA-Slack Mode method has the ability to be integrated with some other algorithms, such as genetic algorithm, PSO algorithms, in order to provide more efficiency in solving hard problems,

© 2013 IJAIR. ALL RIGHTS RESERVED

such as MPRCPSP. Hence, in this work, the slack mode method is integrated with the GA, and a new method called GA-slack mode is emerged. The results showed that the GA-Slack Mode method utilizes the efficiency of genetic algorithms along with the A more sophisticated decoding procedure in which problem specific information is included. The present research showed that the GA-Slack Mode has the ability to hybridize with the other heuristic, meta-heuristic, and local search methods in order to obtain better solutions.

VII.CONCLUSIONS

Resource-constrained project scheduling problems can be solved using several methods. Because of the great computational complexity of precise methods, they are used for planning only a small number of processes. In this paper presents a genetic algorithm for the resource constrained multi-project scheduling problem (RCMPSP). The chromosome representation of the problem is based on random keys. The schedules are constructed using a heuristic that generates parameterized active schedules based on priorities, delay times, and release dates defined by the genetic algorithm. Slack Mode fully implements the advantages of GA and chooses special designs for specific requirements of the problem such as the representation of the chromosome, the precedence and resources constraints. The computational results show that GA-Slack Mode is high quality algorithm.Compared with other methods, GA-Slack Mode can search the results fast and efficiently. Further work could be conducted to explore the possibility of using activities with multi-mode usage of resources.

REFERENCES

[1] Bottcher, J., Drexl, A., and Kolisch, R. (1996) A branch and bound procedure for project scheduling with partially renewable resource constraints, Proceedings of the fifth international workshop on project management and scheduling, Poznan, April 11-13, 48-51.

[2] Brucker, P., Drexl, A., Mohring, R., Neumann, K. and Pesch, E. (1999) Resource-constrained project scheduling: Notation, classification, models and methods, European Journal of Operational research 112, 3-41.

[3]. Patrick, Construction Project Planning and Scheduling, Pearson Prentice Hall, Ohio, 2004.

[4] H. Pan and C.H. Yeh, GA for Fuzzy Multi-Mode Resource-Constrained Project Sched-uling, WSEAS Transactions on Systems, 4 (2), (2003), 893-990.

[5] C. Darwin, The Origin of Species by Means of Natural Selection, London, John Murray, Albemarle Street, 1859.

[6] J. Magalhães-Mendes, Active, Parameterized Active, and Non-Delay Schedules for Pro-ject Scheduling with Multi-Modes, Proceedings of the 16th WSEAS International Con-ference on Applied Mathematics, Montreaux, Switzerland, December 29-31, (2011), 134-139.

[7] R. Kolisch, Project Scheduling under Resource Constraints, Physica-Verlag Heidelberg, 1995
.

[8]. L.G. Fendley. Towards the development of a complete multiproject scheduling system. Journal of Industrial Engineering, 19(10):505.515, 1968.

[9].M.R. Garey and D.S. Johnson. Computers and intractability. A Guide to the theory of NP-completeness. W.H. Freeman and Company, New York, 1979.

[10].D.E. Goldberg. Genetic algorithms in search optimization and machine learning. Addison-Wesley, 1989.

[11].J.F. Gonc¸alves and J.R. Almeida. A hybrid genetic algorithm for assembly line balancing.
Jornal of Heuristics, 8:629.642, 2002.

© 2013 IJAIR. ALL RIGHTS RESERVED