Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 2, Issue. 5, May 2013, pg.430 – 434

SURVEY ARTICLE

A Study on Multi Project Resource Constrained Project Scheduling using Metaheuristic Approach

S. Siva Vidya[1], M. Suguna[2], Dr. D. Sharmila[3]
[1]PG Scholar, SNS College of Technology, Coimbatore, India
[2]Associate Professor, SNS College of Technology, Coimbatore, India
[3]Professor and Head EIE Department, Bannari Amman Institute of Technology, Erode, India

Abstract— ***Project management in the business field is defined as managing and directing time, material, personnel, and costs to complete a particular project. One important phase in the project planning is Project Resource Scheduling. This process identifies resources amount and type according to the activity that scheduled. Planning the efficient use of resources is a complex task. The purpose is to create a smoother distribution of resource usage i.e. to minimize the fluctuation of the resource usage. The resource leveling is used to produce a solution to the problem. In meta-heuristic algorithms, generating individuals in the initial step has an important effect on the convergence behavior of the algorithm and final solutions. Using some heuristics for generating one or more near-optimal individuals in the initial step can improve the final solutions obtained by meta-heuristic algorithms. Different criteria can be used for evaluating the efficiency of scheduling algorithms, the most important of which are makespan and flowtime. This paper provides a survey on various multi project resource constrained scheduling algorithms used for evolutionary computing.***

Key Terms: - Project Scheduling; Metaheuristic Approach; Ant Colony Optimization; Particle Swarm Optimization

I. INTRODUCTION

The multi project resource constrained project scheduling problem is a well-known NP-Hard problem in scheduling. It is a classical problem in operations research with broad applicability in project management and production scheduling. It involves minimizing the makespan of a project by scheduling its activities that are subject to preference and resource constraints. The amounts of available resources are fixed and known in advance. Resource requirements and processing times for each activity are deterministic and also known in advance and preemption of activities is not permitted. This problem has received the notice of many researchers for well over four decades. One of the recent research focuses in this area has been towards developing new metaheuristics using artificial intelligence and biologically- inspired techniques. For solving this problem, two schedule generation schemes are commonly used serial and parallel. Project scheduling has attracted an ever growing attention in recent years both from science and practice. It is concerned with single-item or small group production where limited 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 slant 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.

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.

The evolutionary algorithmic flow with randomly generating population of individuals that are possible solutions. Then in a unchanging number of iterations the algorithm tries to get optimal or near-optimal solutions using predefined operators and a fitness function that evaluates the optimality of solutions. Generating potential solutions at the beginning of the algorithm has an important effect in obtaining final solutions and if in this step of the algorithm bad solutions are generated randomly, then the algorithm provides bad solutions or local optimal solutions. To overcome the posed problem, we usually generate one or more individuals using well-known heuristics and others randomly in the initial step of the algorithm. These heuristics generate near-optimal solutions and the meta-heuristic algorithm combines random solutions with them for obtaining better solutions. Using this method we can obtain better solutions using meta-heuristic algorithms.

Project scheduling heuristics can be divided into two classes: on-line mode and batch-mode heuristics. In the on-line mode, a task is mapped onto a host as soon as it arrives at the scheduler. In the batch mode, tasks are not mapped onto hosts immediately and they are collected into a set of tasks that is examined for mapping at prescheduled times called mapping events. The online mode heuristic is suitable for the low arrival rate, while batch-mode heuristics can achieve higher performance when the arrival rate of tasks is high because there will be a sufficient number of tasks to keep hosts busy between the mapping events, and scheduling is according to the resource requirement information of all tasks in the set. Different criteria can be used for evaluating the efficiency of scheduling algorithms, the most important of which are makespan and flowtime. Makespan is the time when an HC system finishes the latest job and flowtime is the sum of finalization times of all the jobs. An optimal schedule will be the one that optimizes the flowtime and makespan.

II. TYPES OF SCHEDULES

Schedules can be classified into one of following three types of schedules:

• **Semi-active Schedule:** These are feasible schedules obtained by sequencing activities as early as possible. In a semi-active schedule, no activity can be started earlier without altering the processing sequences.

• **Active Schedule:** These are feasible schedules in which no activity could be started earlier without delaying some other activity or breaking a precedence constraint. Active schedules are also semi-active schedules. An optimal schedule is always active, so the search space can be safely limited to the set of all active schedules.

• **Non-delay Schedule:** These are possible schedules in which no resource is kept idle when it could start processing some activity. Non-delay schedules are active and hence are also semi-active.

III. METAHEURISTIC APPROACHES

Several metaheuristic strategies have been developed to solve hard optimization problems. The following summary briefly describes those general approaches that have been used to solve the RCPSP. EAs can be applied as multi-objective optimization tools to obtain the most suitable solutions. In multi-objective problems, the decision maker is required to select a solution from a Pareto front solution by making compromise, which provide for suitable performance over all objectives classified multi-objective methods in three categories: generating techniques with a posteriori articulation of preferences; techniques which rely on prior articulation of preferences; and techniques which rely on progressive expression of preferences. Genetic Algorithm, Particle Swarm Optimization, and Ant Colony Optimization have been widely applied in optimization of construction problems. We briefly describe the algorithms in the following Sections.

A. Genetic Algorithm

As the first introduced evolutionary algorithm, GA has been used widely in various aspects of engineering problems such as constrained or unconstrained optimization, scheduling and reliability optimization. GA is a searching and optimization tool based on natural evolution. It directs the initial population toward the global optimum points according to the objective function. A solution to a given problem is represented in the form of a string, called chromosome, consisting of a set of elements called genes that hold a set of values for the optimization variables. In general, GA includes four important steps: (1) Generation of an initial population, (2) selection of the best chromosomes based on their fitness value; (3) crossover of the old chromosome to produce new chromosome in the next generation; (4) mutation of new chromosomes to extend the scope of searching. Usually some of the best chromosomes called an superiority genes go directly to the next generation. Based on the GA process, four important parameters including population size, Pcrossover ,Pmutation and size of crowding distance have significant impact on the convergence ratio and quality of Pareto front solution.

***B.** Ant Colony Optimization*

The Ant Colony Optimization is a metaheuristic developed by Marco Dorigo, which was inspired by colonies of real ants that leave a chemical substance on the ground called *pheromone*. This substance influences the behavior of the ants: they tend to take those paths where there is a larger amount of pheromone. Pheromone trails can thus be seen as an indirect communication mechanism among ants. From a computer science perception, the ACO is a multi-agent system where low level relations between artificial ants result in a complex behavior of the entire ant colony.

There are three main ideas from colonies of real ants that have been adopted in the ACO

- Indirect communication through pheromone trails.
- Shortest paths tend to have a higher pheromone growth rate.
- Ants have a higher preference (with a certain probability) for paths that have a higher amount of pheromone.

Additionally, an ACO algorithm has certain capabilities imaginary in colonies of real ants. For example:

- Each ant is able of estimating how far it is from a certain state.
- Ants have information about the environment and use it to make decisions. Therefore, their "behavior" is not only adaptive, but also extensive.
- Ants have memory, since this is necessary to ensure that only adequate solutions are generated at each step of the algorithm.

The ACO was originally planned for the traveling salesman problem, and most of the current applications of the algorithm require the problem to be reformulated as one in which the goal is to find the best path of a graph.

C. Artificial Bee Colony Algorithm

The ABC algorithm is a swarm based, meta- heuristic algorithm based on the model first proposed by on the foraging performance of honey bee colonies. The ABC consists of three groups of artificial bees: employed foragers, onlookers and scouts. The employed bees comprise the first half of the colony whereas the second half consists of the onlookers. The employed bees are connected to particular food sources. In other words, the number of working bees is equal to the number of food sources for the hive. The onlookers watch the dance of the employed bees within the hive, to select a food source, whereas scouts search randomly for new food sources.

Analogously in the optimization context, the number of food sources in ABC algorithm, is equivalent to the number of solutions in the population. Furthermore, the position of a food source signifies the position of a promising solution to the optimization problem, whereas the quality of nectar of a food source represents the fitness quality of the associated solution.

D. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a newer algorithm which is developed by Kennedy and Eberhart. It is based on an similarity with the choreography of flight of a group of birds. A large number of birds group synchronously, change direction suddenly, and scatter and regroup together. Each particle adjust its flying from the experience of its own and that of the other members of the swarm during the search for food. Like the evolutionary algorithm, PSO search operates through updating swarms of particle. This techniques use a population of solutions from the search space which are initially randomly generated. Solutions belonging to the same population interact with each other during the search process. Solutions are evolved using techniques inspired from the real world.

On the basis of classical PSO, the algorithm maintains a best set of non-dominated solutions and redefines the selections of guides during the optimization process. PSO has the advantage of keeping the continuity between individuals to converge faster although it may easily get into the local optimum,. In PSO, each particle corresponds to a candidate solution of the underlying problem. PSO that reproduces chromosomes of the next generation from unspecified survivals, PSO updates a population of particles with the internal velocity and attempts to profit from the discoveries of themselves and prior experiences of other companions.

E. Simulated Annealing

Simulated Annealing is 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. Clearly, SA can be viewed as an extension of a simple greedy procedure, sometimes called First Fit Strategy, which immediately accepts a better neighbor solution but rejects any deterioration.

F. TabuSeach Algorithm

TabuSearch is essentially a steepest descent/mildest ascent method. That is, it evaluates all solutions of the neighborhood and chooses the best one, from which it proceeds further. This concept, however, bears the possibility of cycling, that is, one may always move back to the same local optimum one has just left. In order to avoid this problem, a tabu list is set up as a form of memory for the search process. Usually, the tabu list is used to forbid those neighborhood moves that might cancel the recently performed moves and might thus lead back to a recently visited solution. Typically, such a tabu status is overrun if the corresponding neighborhood move would lead to a new overall aspiration criterion. It is obvious that TS extends the simple steepest descent search, often called Best Fit Strategy (BFS), which scans the neighborhood and then accepts the best neighbor solution, until none of the neighbors improves the current objective function value.

G. Cuckoo Search Algorithm

Cuckoo Search is an optimization algorithm which was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds. Some host birds can engage direct conflict with the intruding cuckoos. For example, if a host bird discovers the eggs are not their own, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. If the newly arrived job satisfies the constraints of the jobs in the resources approximately, then the job is chosen for execution. Else the job is discarded from that resource and some other optimal resource is chosen for that job.

Cuckoo optimization algorithm is based on three simple principles that emerge from the cuckoo's strategy:

1) Each cuckoo lays one egg (a design solution) at a time, and dumps it in a randomly chosen nest.

2) The best nests with a high quality egg carry over to the next generation.

3) The number of available host nests is fixed, and a host and there is a finite probability of the cuckoo in the nest being discovered.

H. Hybrid Genetic Algorithm- Particle Swarm Optimization (HGAPSO)

A new evolutionary learning algorithm based on a hybrid of GA and PSO called HGAPSO presented by Juang. In this hybrid algorithm, solutions in a new generation are created, not only by crossover and mutation operations as in GA, but also by PSO. The concept of elite strategy is adopted in HGAPSO, where the upper-half of the best-performing solutions in a population are regarded as elite. However, instead of being reproduced directly in the next generation, these elites are first enhanced.

The group constituted by the elites is regarded as a swarm, and each elite corresponds to a particle within it. In this regard, the elites are enhanced by PSO, an operation which mimics the maturing phenomenon in nature. These enhanced elites constitute half of the population in the new generation, whereas the other half are generated by performing crossover and mutation operation on these enhanced elites.

I. Shuffled Frog Leaping Algorithm

The SFLA combines the benefits of the genetic-based Meme -tic Algorithm and the social performance based PSO. Instead of using genes in GA, SFLA uses memes to improve spreading and convergence ratio. In the SFLA, the population consists of a set of frogs that is partitioned into subsets referred to as memeplexes. SFLA, in essence, combines the benefit of the local search tool of PSO and the idea of mixing information from parallel local searches, to move toward a global solution which is called a Shuffled Complex Solution.

The philosophy behind SCE is to treat the global search as a process of natural evolution. After a defined number of memetic evolutionary steps, frogs are shuffled among memeplexes, enabling frogs to exchange messages among different memplexes and ensuring that they move to an optimal position.

IV. CONCLUSION

In this paper, the survey is about the current trends and applications of multi project resource constrained scheduling algorithms in the area of evolutionary computing. There are lots of researches on project scheduling however, all the researches were carried out on single or multi-projects. Large scale project scheduling in evolutionary computing provides suitable solution for managing the resources. This problem involves optimization of several objectives including resource utilization, QOS metrics, cost, reliability factors, completion time etc.

REFERENCES

[1] Holland, J. H. (1976) Adaptation in Nature and Artificial Systems, University of Michigan Press, Ann Arbor, MI.

[2] Kolisch, R. and R. Padman (2001) An integrated survey of deterministic project scheduling. Omega, 29, 249-272.

[3] Kurtulus, I. S. and E. W. Davis (1982) Multi-project scheduling: Categorization of heuristic rules performance. Management Science, 28, 161-172.

[4] Kurtulus, I. S. and S. C. Narula (1985) Multi-project scheduling: Analysis of project performance. IIE Transactions, 17, 58-66.

[5] Law, A. M. and W. D. Kelton (1991) Simulation Modeling and Analysis, MacGraw-Hill, Singapore.

[6] S. Hartmann, "Project scheduling with multiple modes : A genetic algorithm," Annals of Operations Research, Vol.102, 2001, pp.111-135.

[7] J. Blazewicz, "Complexity of computer scheduling algorithms under resource constraints," Proc. 1st meeting AFCET-SMF on Applied Mathematics, 1978, pp. 169-178.

[8] J. Kelley, "The critical path method: Resource planning and scheduling," in Muth and Thompson editors, Industrial Scheduling, Prentice Hall, 1963, pp. 347-365.

[9] P. Brucker, A. Drexl, R. Mohring, K. Neumann, and E. Pesch, "Resource-constrained Project Scheduling: Notation, Classification, Model, and Methods," European Journal of Operational Research, vol. 112, 1999, pp. 3-41.

[10] K. Nonobe and T. Ibaraki (2002), Formulation and tabu search algorithm for the resource constrained project scheduling problem (RCPSP). In: Ribeiro, C.C., Hansen, P. (Eds.), Essays and Surveys in Metaheuristics, Kluwer Academic Publishers, 557 - 588.

[11] Yin, P.Y., Yu, S., and Wang, P. (2006). "A Hybrid Particle Swarm Optimization Algorithm for Optimal Task Assignment in Distributed Systems." Journal of Computer Standards and Interfaces, 28(4), 441-450.

[12] Yang, C.S., Chuang, L.Y., Ke, C.H., and Yang, C.H. (2008). "A Combination of Shuffled Frog-Leaping Algorithm and Genetic Algorithm for Gene Selection." Journal of Advanced Computational Intelligence and Intelligent Informatics, 12(3), 218-226.

[13] Xiong, Ying., and Kuang, Y. (2008). "Applying an Ant Colony Optimization Algorithm-base Multi-objective Approach for Time-Cost Trade-off." Journal of Construction Engineering and Management, 134(2), 153-156.

[14] Shahgholi Ghahfarokhi, B., abaeizadeh, M., and Movahedinia, N. (2007). "An Improved Hybrid Genetic Algorithm Using Particle Swarm Optimization." 15th ICEE Electrical Conference, Tehran, 102-108.

[15] Shi, X.H., Liang, Y.C, Lee, H.P., Lu, C., and Wang, L.M. (2004). "An Improved GA and a Novel PSO-GA based Hybrid Algorithm." Information Processing Letters, 93(5), 255-261.

