SMART PROJECT SCHEDULING FOR CONSTRUCTION PROJECTS USING AI : A REVIEW

[1]Joel Macwan, [2]Prof. Amitkumar N. Bhavsar
[1]Student of construction engineering and management M. Tech, Birla Vishwakarma mahavidhyalaya, Anand, Gujarat, India
[2] Associate Professor, Civil Engineering Department, Birla Vishwakarma Mahavidhyalaya, vallabh vidyanagar, Anand, Gujarat, India

Abstract : Planning and scheduling are two stepping stones for any project as without careful planning and scheduling of any project, the project is destined to face a lot trouble down the road in completion. And so, planning and scheduling for any project should be considered vital and given required care. However, scheduling of big projects is not an easy task. Problems of scheduling of such projects are said as resource constrained scheduling problem(RCPSP).in RCPSP scheduling of the project is carried out considering the one objective function that is to reduce the makespan and two constraints (1)resource constrains (2)precedence constrains. For solution of such kind of problem there exist some conventional algorithms but these does not always provide optimal solutions. Although with the rise of AI there have been noteworthy improvement in the field of construction scheduling.in this review paper the problem of RCPSP is explained and the methods proposed for achieving optimal solutions are discussed.

***IndexTerms* -** Project scheduling, Project management, RCPSP, software, scheduling algorithms, allocation

INTRODUCTION

Project is considered as a complex one-time activity that performed to achieve specific objectives. Such projects are to be completed with the available resources and in available time period. Failing in doing so can lead to catastrophic outcomes for the project stakeholders. Therefore, the planning and scheduling of the project should be given utmost importance, also due care should be taken to plan the project and to control the project. Problems of scheduling such projects with an objective and provided constrains are known as resource constrain project scheduling problem. The solution of such problems is to achieve a near optimum schedule with possible minimum makespan.

Numerous researchers gave considerable to solve this problem using various methods. The rest of the paper discusses the gist of the basic RCPSP and various solution techniques at length.

THE BASIC RCPSP

The basic RCPSP includes a project network G(N, A) and a set N of nodes designating the activities of the project. The activities in the network are subject to self-styled zero-lag finish-start precedence constraints $(i,j) \in A$, designated by the arcs of the network. The existence of such an arc indicates that a predecessor activity i has to be completed before activity j can be started. The total of m renewable resource types are assumed, that has a per period availability of a_k, $k \in K$ and $K = \{1, \dots, m\}$. These resource types can be, e.g., men or equipment, and are assumed to be available continuously in specified limit per time period throughout the whole project makespan. The resources are renewable in the sense that even if we ‘‘use’’ the resources during a certain time period t, these resource still be available with full capacity for each and every successive time period t +1, t+2,.... The project activities $i \in N$ needs per period amount r_{ik} (where r_{ik} is an integer value) of resource type k, $k \in K$. A solution to the RCPSP then comprises of a vector of start times s_i, $i \in N$, in a manner that the resource and precedence constraints are fulfilled, and the project makespan is reduced.

The mathematical formulation of this RCPSP is given below.

Let F_j represent the finish time of task j. Let a Schedule S be given by a vector of finish times $(F_1, F_2 \dots, F_m)$. Let $A(t) = \{ j \in J \mid F_j - d_j \leqslant t < F_j \}$ be the set of activities which are being handled at time t. The goal is to minimize the finish time of the last task and thus the makespan of the project. The first constraint is for the precedence relationship while the second is for the resource usage. Set P_j represents the immediate predecessor activities for each activity j.

Minimize F_m

Subject to

$$F_h \leq F_j - d_j, \quad j = 1,\dots,m; \; h \in P_j$$
$$\sum_{j \in A(t)} r_{j,k} \quad , k \leq R_k \;\in K; \; t \geq 0$$
$$F_j \geq 0, \quad j = 1,\dots,m;$$

Mostly in projects, resource constraints will be obligatory and as a result, the optimum schedule will not be viable. Consequently, project management has to opt for scheduling techniques which yield resource feasible schedules with (preferably) a minimized project makespan.

The scheduling schemes along with various solution methodologies for the RCPSP described is described below :

- List scheduling algorithms: Provided with a priority list of the project activities, the serial scheduling generation scheme or the parallel schedule generation scheme can be used to generate a schedule. These constructive heuristics are very quick when it comes to computation time but can potentially yield schedules with a makespan in excess of the optimal value.
- Exact procedures: The dedicated branch-and-bound procedures are considered as the best of the exact procedures for solving the RCPSP. Yet as the complexity of the problem increases say with the rise in number of activities, the computational running time of these procedures intensifies hastily. Instances with about 30 activities can be solved in reasonable time bounds using dedicated branch and bound procedures. Larger instances may take an excessive amount of time to solve, so that one has to opt to heuristic methods.
- Metaheuristics: These algorithms perform a broad search in the solution space in order to find a good solution for the RCPSP. They frequently use priority lists as a subroutine to produce intermediate schedules.

SCHEDULE GENERATION SCHEME

S. Hartmann et. al. suggested vital most important component related with cracking the RCPSP is the schedule generation scheme (SGS). SGSs, which work on the schedule space, are used to shape a feasible schedule by stepwise extension of a fractional schedule. ASGS is typically used with heuristic and meta-heuristic algorithms which work on an in-direct depiction such as the activity list or random key depiction. This in-direct depiction determines the order in which activities are to be handled and the SGS converts this into a schedule. Priority heuristics function in a like way, where the assigned priorities govern the order in which the SGS will process activities. Integer programming approaches do not utilize SGSs since they usually already operate on the schedule space. [1]

A Sprecher et. al. also gave two commonly used schedule generation schemes.

1. Serial SGS produces a schedule in n phases where n is the number of activities. At any phase the set of suitable activities comprises of unscheduled activities with all predecessor activities scheduled. In each of the n phases, one activity is designated from the appropriate set and scheduled at the earliest precedence and resource possible completion time .
2. The parallel SGS works based on time increase. At each time instant it identifies a set of appropriate activities (Dn) and from this set one activity is chosen to be scheduled. The appropriate set is the set which contain activities which can be started at present time considering both precedence and resource constrains. A time increase is done after the eligible set at the present time is entirely empty. **[1]**

R. Kolisch proved that the serial SGS produces active schedules. if none of the activities can be started earlier sans delaying some other activity the schedule is called an active schedule. Also , it has been proved that for makespan minimization the optimum solution will at all times lie in the set of active schedules. Additionally, it was also verified in that the parallel SGS produces non-delay schedules. None of the activities can be started earlier sans postponing some other activity even if activity pre-emption is permitted it is termed non-delay schedule. The set of non-delay schedules is merely a subset of the set of active schedules it might not hold optimum schedules for a regular performance measure. Nevertheless, regardless of producing non-delay schedules, it also has been shown that parallel outdid serial SGS when functioning on priority rules. **[2]**

SOLUTION METHODOLOGIES

There exist a number of approaches for solving the classical RCPSP defined previously. These range all the way from simple priority rules all the way to more complex exact and meta-heuristic methods.

PRIORITY RULES/HEURISTICS

Priority rules are simple heuristics which, based on certain instance or activity characteristics, determine the order in which activities should be scheduled. Based on this order/priority an SGS is used to construct a schedule. Priority heuristics are different from optimization algorithms in the sense that, in their traditional form, they do not perform systematic optimization. They usually generate a single "good enough" solution as opposed to optimization algorithms which explore thousands of solutions in search of the optimum.

Kolisch et. al. said Priority rules are simple to comprehend and easy to implement yet their performance for the classical static instances leaves a lot to be desired. Few of the best priority rules for RCPSP are Latest Finish Time (LFT), Latest Start Time (LST), Worst Case Slack (WCS) and Average Case Slack (ACS). LST and LFT determine priorities based on the start and finish times in the precedence feasible schedule constructed by ease up the resource constraints in which all activities are scheduled as late as possible

(Critical Path Method). ACS and WCS prioritise activities which would impose the greatest slack if not scheduled at the current time. They utilize the latest start time to compute the slack. These four rules could well be ranked as the top rules for RCPSP when utilizing forward scheduling. Two other notable rules are Earliest Finish Time (EFT) and Earliest Start Time (EST) which are again based on the critical path method in which priorities are determined based on a precedence feasible schedule constructed by scheduling as each activity as early as possible. **[3]**

A more detailed study of the different priority rules for RCPSP is presented by **Klein**. He performed an initial evaluation of as many as 73 different priority rules for his study which focused on benchmarking priority rules.

G. Koulinas, L. Kotsikas have tried to combine priority rules with sampling methods as a way to improve their performance. These approaches use sampling methods like random sampling, biased random sampling, and regret based biased random sampling, all of which try to bias the priority for each activity in the eligible set based on a certain probability, allowing the scheme to discover multiple solutions with a single priority rule. **[4]**

Exact Methods

J. A. Carruthers et. al. explored that Exact solution methods usually work on the principle of "guaranteed optimality". This of course comes at the cost of extended run-time.in their work they applied a dynamic programming approach. This was followed by a number of integer (zero-one) programming models/approaches**. [5]**

E. Demeulemeester visited Methods based on implicit enumeration with branch and bound or lower bounds rather lengthily, and said these techniques, though, are mostly useful for smaller instances as the run time becomes harder to manage with larger and/or extremely constrained instances. **[6]**

Heuristic/Meta-Heuristic Algorithms

A number of meta-heuristic and heuristic algorithms have been proposed over the years as a compromise between simple priority heuristics and exact solution approaches. These algorithms are more complex and time consuming in comparison to priority heuristics but offer a better approximate solution. They cannot guarantee optimality like exact approaches but offer a 'good enough' or 'near optimal 'solution in a 'reasonable' amount of time for static cases. These include approaches such as genetic algorithm, particle swarm optimization, ant colony optimization, tabu search and simulated annealing.

J. Mendes gave two commonly used forms of indirect representation. First is the activity list representation which is simply a list of activity IDs with the order indicating preference. The other one is the random key representation in which the solution is a vector of real numbers with the value at each index representing the priority for the corresponding activity. These indirect representations have come up as a way to counter the complexity of representing and manipulating a full schedule. These solutions have to paired with an SGS to generate a schedule and subsequently to determine their fitness. **[7]**

V. Valls et. al. proposed the most commonly used local search technique justification. Justification is a problem specific local search technique. It is an iterative forward and backward scheduling approach to improve the quality of the solution. Each successive iteration of this local search will result in a solution either better or equal in makespan in comparison to the solution in the previous iteration. **[8]**

There are also a number of hybrid methods given by **D. Debels et. al**. like hybrid scatter search/electromagnetism meta-heuristic, hybrid ant-colony/scatter search algorithm by **W. Chen et. al.** which harness the strength of multiple search paradigms. **R. Zamani** suggested Other interesting aspects including design of new recombination operators. **[9]**

Hoc et. al. said Some of these approaches are effective for smaller instances while others are better on larger instances. One of the main short comings of a number of such algorithms in literature is the excessive number of users defined parameters and the problem specific parameter tuning which can bring to question the quality of some of the results obtained planning is a intellectual activity accustomed to everybody. It plays a crucial part in decision making by allowing individuals to deal with altering and complex situations. Planning effects a extensive range of activities, from the most insignificant ones, such as how to get to work in the morning, to the most important, such as the allocation of resources in a country's economy. Plans are used, either formally or informally, for controlling any activity that has not been fully automatized **[10]**.

Business Roundtable, Laufer& Tucker said Planning is one of the vital elements of construction management. Even if a lot of research has been made throughout the last few decades, some dissatisfaction with the application and results of construction planning still remains. Nevertheless, it appears that people involved in construction management still consider that a more effective method to construction planning can fetch significant developments in the performance of the industry. **[5]**

Construction industry in India is facing an acute shortage of the construction management experts. With the use of AI in the field such shortage can be reduced till an extent. If used correctly AI can also provide services of a consultant in specific field.AI can also help manager take better decisions and r]train freshers in the field and thus, reducing training time. These are but a few good examples of AI in the construction industry.it is apparent from this that AI has a huge potential in construction industry and a field as dynamic as construction management can definitely take advantage of it. Use of AI in solving of RCPSP is one such use that is discussed here, efforts should be made to expand AI's reach in construction management field to improve the effectiveness of the field.

CONCLUSION

From above discussion it can be said that Resource Constrain Project Scheduling Problem(RCPSP) are the crux of scheduling problems. Since long many researchers have tried to solve it. However, after the introduction of artificial intelligence significant improvements have been reported in the field of RCPSPs, with various solution methodologies that have been developed. However, these methodologies not necessarily always give optimal solutions, they always try to achieve near optimal solutions. However, no single method can be said as the best method for solution of the RCPSP, these methods can be improved to obtain optimal solution for the RCPSPs.

FUTURE SCOPE

As said in conclusion RCPSP is the most sought-after topic in scheduling, and in such research various algorithms for solution of RCPSP has been evolved. These all achieve near optimum solutions; they are not giving the best optimum solution and every solution method has room for improvement. Every solution method should be explored further to improve upon and yield optimum solution to the problem.

ACKNOWLEDGEMENT

I thankfully acknowledge to prof. Amitkumar N. Bhavsar, associate professor, civil engineering department, construction engineering and management, Birla Vishwakarma mahavidhyalaya,Vallabh Vidyanagar, Prof.(Dr.) I. N. Patel, Principal, Birla Vishwakarma mahavidhyalaya,Vallabh Vidyanagar,Prof. (Dr.) L. B. zala,H.O.D., Civil engineering department, Birla Vishwakarma mahavidhyalaya,Vallabh Vidyanagar for their motivation, encouragement and constructive support to carry out this research.

REFERENCES

[1] S. Hartmann, "Project scheduling with resource capacities and requests varying with time: A case study," Flexible Services and Manufacturing Journal, 2013.
[2] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation," European Journal of Operational Research, 1996.
[3] R. Klein, "Bidirectional planning: Improving priority rule-based heuristics for scheduling resource-constrained projects," European Journal of Operational Research, 2000.
[4] D. C. Paraskevopoulos, C. D. Tarantilis and G. Ioannou, "Solving project scheduling problems with resource constraints via an event list-based evolutionary algorithm," Expert Systems with Applications, 2012.
[5] A. A. B. Pritsker, L. J. Waiters and P. M. Wolfe, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach," Management Science, 2008.
[6] W. Herroelen, B. De Reyck and E. Demeulemeester, "Resource-constrained project scheduling: A survey of recent developments," Computers and Operations Research, 1998.
[7] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version," in European Journal of Operational Research, 2003.
[8] V. Valls, F. Ballestín and S. Quintanilla, "Justification and RCPSP: A technique that pays," in European Journal of Operational Research, 2005.
[9] S. Baradaran, S. M. Fatemi Ghomi, M. Mobini and S. S. Hashemin, "A hybrid scatter search approach for resource-constrained project scheduling problem in PERT-type networks," Advances in Engineering Software, vol. 41, no. 7-8, 2010.
[10]J. H. Patterson and W. D. Huber, "A Horizon-Varying, Zero-One Approach to Project Scheduling," Management Science, 2008.
[11]X. L. Zheng and L. Wang, "A multi-agent optimization algorithm for resource constrained project scheduling problem," Expert Systems with Applications, 2015.
[12]R. J. Vanderbei, "Linear Programming: Foundations and Extensions," Journal of the Operational Research Society, 2002.
[13]R. Slowinski, "Two approaches to problems of resource allocation among project activities- a comparative study," Journal of the Operational Research Society, 1980.
[14]J. H. Patterson and G. W. Roth, "Scheduling a project under multiple resource constraints: A zero-one programming approach," AIIE Transactions, 1976.
[15]A. Nguyen, T. Urli and M. Wagner, "Single- and Multi-Objective Genetic Programming: New Bounds for Weighted ORDER and MAJORITY," in Foundations of Genetic Algorithms, 2013.
[16]O. Lambrechts, E. Demeulemeester and W. Herroelen, "Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities," Journal of Scheduling, 2008.
[17] R. Kolisch and S. Hartmann, "Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis," 2011.
[18] T. Frankola, M. Golub and D. Jakobovic, "Evolutionary algorithms for the resource constrained scheduling problem," in Proceedings of the International Conference on Information Technology Interfaces, ITI, 2008.
[19] M. Đurasević, D. Jakobović and K. Knežević, "Adaptive scheduling on unrelated machines with genetic programming," Applied Soft Computing Journal, 2016.
[20] W. Chen, Y. j. Shi, H. f. Teng, X. p. Lan and L. c. Hu, "An efficient hybrid algorithm for resource-constrained project scheduling," Information Sciences, 2010.

[21] R. K. Chakrabortty, R. A. Sarker and D. L. Essam, "Multi-mode resource constrained project scheduling under resource disruptions," Computers and Chemical Engineering, 2016.

[22] J. Branke, S. Nguyen, C. W. Pickardt and M. Zhang, Automated Design of Production Scheduling Heuristics: A Review, 2016.

[23] M. Bartusch, R. H. Möhring and F. J. Radermacher, "Scheduling project networks with resource constraints and time windows," Annals of Operations Research, 1988.

[24] J. Alcaraz and C. Maroto, "A Robust Genetic Algorithm for Resource Allocation in Project Scheduling," Annals of Operations Research, 2001. li, A. 2001.Macroeconomic variables as common pervasive risk factors and the empirical content of the Arbitrage Pricing Theory. Journal of Empirical finance, 5(3): 221–240.