International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 5, Issue 05, May-2019

SMART PROJECT SCHEDULING FOR CONSTRUCTION PROJECTS USING AI

[1]Joel Macwan, [2]Prof. Amitkumar N. Bhavsar, [3]Dr. Mohammedshakil Malek

[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
[3] Principal, F. D. (MUBIN) Degree College of Engineering, Gandhinagar

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 a big project is no easy feat. 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 make-span and two constraints (1)resource constrains (2)precedence constrains. For solution of such a problem an algorithm has been generated in this thesis using heuristics. This algorithm solves the RCPSP and tries to obtain the near optimal solution. A few sample examples have been performed to opt for better priority rule and evaluate the framework. Also, A case study also has been provided at the end for evaluation of the algorithm.

Key words: ***Project scheduling, Project management, software, scheduling algorithms, allocation, make-span.***

Introduction

Project management competence is of the crux in many industries. Indeed, be it a software development project, a shut-down and maintenance project, an R&D project or any other industrial project: the careful planning, scheduling, and management of the project is as vital as the project content itself. In this article project scheduling software is described that is generated as part of M.Tech thesis at B. V. M engineering college, V.V. Nagar. The software framework mainly focuses on the scheduling part of the project management field. In other words, it is assumed that the project content has been defined, the different project activities have been identified and precedence relations have been presented between the activities. Furthermore, it is also assumed that the required quantity of resources for the execution of every individual activity has been calculated and the activity durations have been estimated. This information, along with the limited availability of resources, can then be seen as an instance of the resource-constrained project scheduling problem (RCPSP), and can be solved with one of the many types of scheduling algorithms.

The rest of this article is ordered as follows. In the following section an introduction is given about the basic RCPSP. The third section then explains on the software features with respect to project scheduling. Finally, the discussion is ended with some concluding remarks and future scope.

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 $K = \{1, \ldots , m\}$. 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 make-span. 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 make-span is reduced.

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 make-span.

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

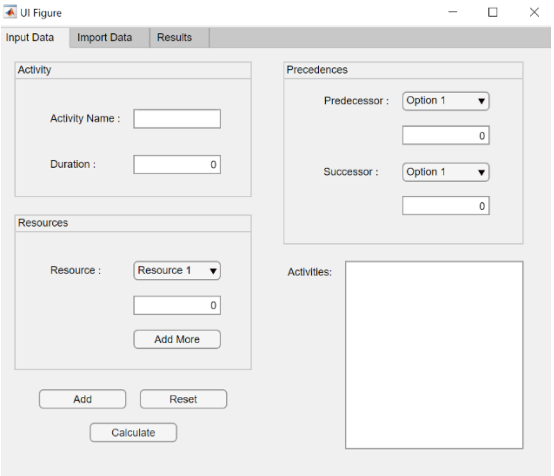
- List scheduling algorithms: Provided with a priority list of the project activities, the serial scheduling generation scheme or the parallel schedule generation scheme can be used to generate a schedule. These constructive heuristics are very quick when it comes to computation time but can potentially yield schedules with a make-span in excess of the optimal value.

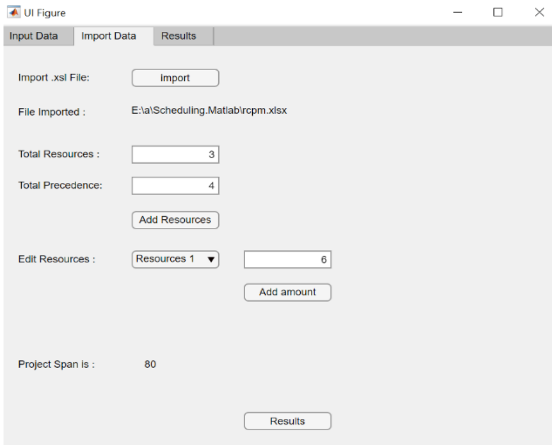
- Exact procedures: The dedicated branch-and-bound procedures are considered as the best of the exact procedures for solving the RCPSP. Yet as the complexity of the problem increases say with the rise in number of activities, the computational running time of these procedures intensifies hastily. Instances 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.

The scheduling software framework

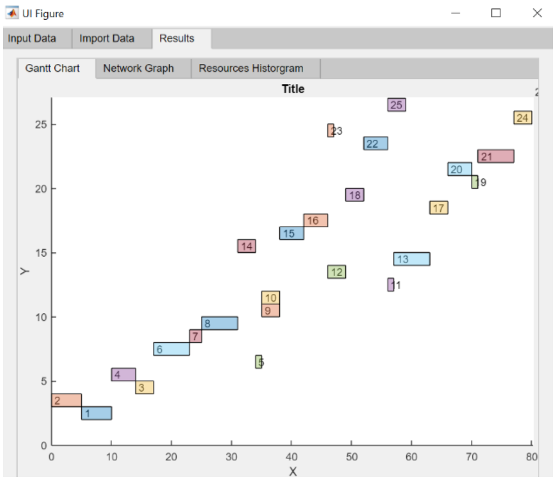
The software framework was written in MATLAB. A GUI for the software has also been provided for the data insertion.

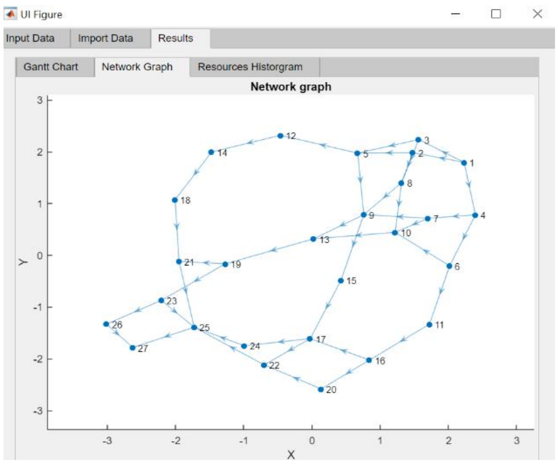
The framework is generated to solve RCPSPs with the use of heuristics. Serial scheduling scheme was adopted for scheduling and after comparing different priority rules MIS priority rule was kept as it gave the better performance.

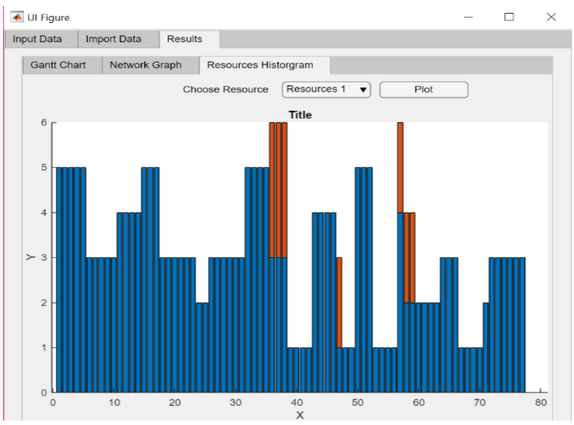




The figure below shows the GUI of the software generated.







Analytical results

Four heuristic priority rules were evaluated for different instances for implementation in the framework. These are:

I. Minimum job slack(MINSLCK)
II. Greatest Resource Demand(GRD)
III. Most Immediate Successor (MIS)
IV. Select Jobs Randomly(RAN)

Out of these four heuristic priority rules MIS rule performed better than the others and gave optimal schedules for the instances.

TABLE 1 PRIORITY RULES FOR DIFFERENT INSTANCES

Priority rule	*Make-span (In days)*			
	Instance 1	*Instance 2*	*Instance 3*	*Instance 4*
GRD	38	20	21	80
RAN	38	22	19	72
MIS	30	22	19	71
MINSLCK	38	21	21	71

From the above results it can be inferred that the MIS priority rule provides with near optimum solution. And based on that MIS rule is adopted for the software framework to achieve optimum project schedules.

A case study

A case study was also considered for the evaluation of the algorithm. The case study was of a residential building in Vadodra, Gujarat, India. Schedule of which was generated by shakti construction group. The same schedule was again generated by the Heuristic algorithm and the results were compared.

The primary details of the residential project are given here.

Project duration	*510 days*
Project total cost	*□ 69,76,500.00*
Project start time	*19 / 08 / 2014*
Project end time	*07 / 04 / 2016*

Case study - Results

Different schedules generated with different heuristic priority rules for the above-mentioned projects are shown in the table below.

TABLE 2 PRIORITY RULE RESULTS

Priority rule	*Make-span (In days)*
GRD	497
RAN	487
MIS	487
MINSLCK	487

It can be said from the above results that the GRD rule gives the least feasible schedule, while the other rules give near optimum results. However, RAN rule seems to be giving different results each time depending upon the selection of the activities for scheduling and so it may not be considered as a reliable schedule and the MIS and MNSLCK rules are given due consideration.

This near optimum result is compared with the original MS. Project generated schedule.MS. Project generates schedule with a make-span of 482 days but this schedule over allocates resources, however after applying resource levelling the schedule gets lengthened and gives make-span of 510 days. The comparison results are as shown below.

TABLE 3 CASE STUDY RESULTS

	Before resource levelling		*After resource levelling*	
	Conventional schedule	*Heuristic schedule*	*Conventional schedule*	*Heuristic schedule*
Project duration	*482 days*	*487 days*	*510 days*	*487 days*
Total renewable resources	*14*	*14*	*14*	*14*
Overallocated resources	*5*	*0*	*0*	*0*

From evaluation of the case study following can be inferred:

(1) The evaluation of the different examples using different heuristic rules suggested that the MIS priority rule gives the least make-span and provide most optimum schedules. Same results are seen with the case study schedule evaluated with the same framework as MIS rule seems to give best schedule. Also, this schedule was compared with original MS project schedule and reduction in make-span by about 23 days or say 4.8% was observed. Not only that but also the resources were allocated while also maintaining the precedence and resource constrains.

(2) The over allocation of the resources found in the schedule generated with conventional algorithm does not occur while using heuristics.

(3) It can be said that using MIS priority rule as heuristic for scheduling better feasible schedules can be obtained. Also, it does not over allocate the resources and thus provide more realistic and feasible solution while keeping both the constraints.

Summery

Project controlling and monitoring through scheduling is a very daunting task, but it could be eased up by careful planning and scheduling. Project planning and scheduling are the basic phase of any project and if they are executed correctly and perfectly the rest of the project breezes through without any trouble such as time and /or cost overruns.

In this thesis a framework has been developed for Resource Constrained Project Scheduling Problems (RCPSP) by Heuristics. This framework generates schedules for the project and provides with the resource allocation for the same. The framework's efficiency was evaluated by considering a case study of a project at Baroda, the schedule of which was compared on both the MS project as well as on RCPSP with heuristics.

The outcomes are shown below.

Table 2 shows make-span of the project for different priority rules.it is apparent from the table that MIS rule gives consistent better schedules.

Table 3 shows a comparison between the schedules generated by conventional approach and heuristic approach.it can be seen from it that heuristic approach not only reduces the make-span but also better allocate the resources. While at first conventional approach makes shorter make-span of the schedule, it also over allocates resources, on levelling the resource allocation the schedule make-span increases, on the other hand heuristic approach provides a schedule with shorter make-span and no resource over allocation.

Conclusion

Based on the analytical results and comparative study in summary, it may be concluded that the heuristic algorithms generated far better results and reduced the project make-span by about 23 days or by 4.8%. Also, the algorithm allocated the resources of the project while maintaining the resource constrains, it does not over allocate a single resource and provided with near realistic or say feasible schedule.

It can be said conclusively that heuristics for the RCPSP gives better feasible/optimal schedules and reduce the make-span of the project while optimally allocating the available resources.

Limitations

The software framework was solved by heuristics. There is possibility of still improving and optimizing the schedule using better heuristic rules.

The software framework was tested for comparatively small and residential projects, the feasibility of the software framework for the large and complex commercial projects are remain to be tested.

Future scope

While the thesis presents substantial eff orts towards the heuristics for the resource constrained project scheduling problem, a number of other directions could be further pursued. These include (but are not limited to):

1) In this thesis the schedule was generated with the use of heuristics to solve the RCSPS, however there are many more heuristics approaches available than the one demonstrated here, which may give better results. These approaches could be explored as a future study in the field.
2) Thesis concluded that Heuristics give better results than conventional algorithms. Nevertheless, AI offers numerous algorithms and methodology for solution of RCPSP such as GA algorithm, neural etworks, tabu search, etc. these methodologies can be explored for the solution of RCPSP.

Acknowledgement

I thankfully acknowledge to Prof. Amitkumar N. Bhavsar, associate professor, civil engineering department, construction engineering and management section, Birla Vishwakarma mahavidhyalaya,Vallabh Vidyanagar, Dr. Mohammedshakil Malek, Principal, F. D. (MUBIN) Degree College of Engineering, Gandhinagar, 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 and support to carry out this research.

References

[1] R. Slowinski, "Two approaches to problems of resource allocation among project activities- a comparative study," *Journal of the Operational Research Society,* 1980.

[2] 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.

[3] 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.

[4] R. Kolisch, "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation," *European Journal of Operational Research,* 1996.

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

[6] J. H. Patterson and G. W. Roth, “Scheduling a project under multiple resource constraints: A zero-one programming approach,” *AIIE Transactions,* 1976.

[7] W. Herroelen, B. De Reyck and E. Demeulemeester, “Resource-constrained project scheduling: A survey of recent developments,” *Computers and Operations Research,* 1998.

[8] S. Hartmann, “Project scheduling with resource capacities and requests varying with time: A case study,” *Flexible Services and Manufacturing Journal,* 2013.

[9] O. Lambrechts, E. Demeulemeester and W. Herroelen, “Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities,” *Journal of Scheduling,* 2008.

[10] A. A. B. Pritsker, L. J. Waiters and P. M. Wolfe, “Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach,” *Management Science,* 2008.

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

[12] R. J. Vanderbei, “Linear Programming: Foundations and Extensions,” *Journal of the Operational Research Society,* 2002.

[13] V. Valls, F. Ballestín and S. Quintanilla, “Justification and RCPSP: A technique that pays,” in *European Journal of Operational Research,* 2005.

[14] R. Kolisch and S. Hartmann, “Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis,” 2011.

[15] 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.

[16] R. Klein, “Bidirectional planning: Improving priority rule-based heuristics for scheduling resource-constrained projects,” *European Journal of Operational Research,* 2000.

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

[18] 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.

[19] M. Đurasević, D. Jakobović and K. Knežević, “Adaptive scheduling on unrelated machines with genetic programming,” *Applied Soft Computing Journal,* 2016.

[20] J. Alcaraz and C. Maroto, “A Robust Genetic Algorithm for Resource Allocation in Project Scheduling,” *Annals of Operations Research,* 2001.

[21] 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.

[22] X. L. Zheng and L. Wang, “A multi-agent optimization algorithm for resource constrained project scheduling problem,” *Expert Systems with Applications,* 2015.

[23] J. H. Patterson and W. D. Huber, “A Horizon-Varying, Zero-One Approach to Project Scheduling,” *Management Science,* 2008.

IJTIMES-2019@All rights reserved