Dogo Rangsang Research Journal
ISSN : 2347-7180
UGC Care Group I Journal
Vol-08 Issue-14 No. 04, April 2021

The Usefulness of Technologies for Resource Levelling In Projects with Resource Constraints

[1*] Ms S.Madhumita Behera, [2]Ms. Rachita Padhi
[1*]Asst. Professor, Dept. OF Civil Engineering, NIT BBSR,
Asst. Professor DEPT. of Civil Engineering, SYNERGY, DKL
[1*]sushreemadhumita@thenalanda.com , padhiracchu11@gmail.com

Abstract

The premise underlying the PERT/CPM network approaches is that all required resources will be available. The lack of resources is frequently the cause of project delays. Several approaches have been used to solve the Resource Constrained Project Scheduling Problem (RCPSP). By taking into account resource restrictions and prioritisation issues, these strategies aim to reduce the project's duration. Software for project management uses resource levelling to resolve resource conflicts. The research compares the outcomes while levelling two actual construction projects as case studies in order to assess the efficacy of the resource levelling tools of three well-liked packages.

Keywords: RCPSPs; Levelling; Software evaluation

1. Introduction

The CPM has been widely utilised for project scheduling, assisting managers in ensuring that the project will be completed on schedule and within budget. The critical path(s), free and total float, and other pertinent information are all provided by CPM and are crucial for the effective planning of a project. The capacity to manage by exceptions (critical and near-critical operations), particularly in large-scale projects, is a benefit of CPM [1]. CPM, on the other hand, is predicated on the premise that there are endless resources available to carry out the operations. Nonetheless, resources are restricted in actual projects. Scheduling without taking resource limits into account results in unstable schedules. According to a survey by Liberatore et al. [2], resource levelling is utilised by 58% of planners and 44% of controllers in the construction sector, and 83% of professional project managers use project management software for planning and control.

The most popular software programmes used for building projects are MS Project and Primavera Project Planner, according to the same poll. This research evaluates the resource levelling capabilities of three PM software packages in two building projects, adding one open source package for comparison.

Five paragraphs make up the paper's structure. The problem's theoretical foundation is formulated in the first. The most recent methods for locating more reliable schedules while taking resource restrictions into account are discussed in the next paragraph as they appear in the international literature. The report then presents its goals and methodology before concluding with its findings and succinct conclusions.

2. Problem formulation

An Activity-On-Node network is used to represent a project (AoN). It comprises of n actions, where 1 and n are dummies that represent the beginning and end of the project, respectively.

The fixed integer duration of activity j is indicated by d_j ($1 \le j \le n$), its integer start by s_j ($1 \le j \le n$) and its integer finish by f_j ($1 \le j \le n$). There are K renewable resources with r_{jk} ($1 \le j \le n$, $1 \le k \le K$), the constant resource requirement of activity j for resource type k, and a_k, the constant availability of resource k. Consequently, the resource constrained problem can be formulated as follows:

$$\text{Min } f_n \quad (1)$$

Subject to

$$f_1 = 0 \quad (2)$$

$$f_j - d_j \ge f_i \quad \forall (i,j) \in H \quad (3)$$

$$\sum_{j \in S_t} r_{jk} \le a_k; \quad t = 1, 2, \ldots, K \quad (4)$$

where H denotes the set of pairs indicating precedence relationships and S_t, the set of activities in progress in time interval $]t-1,t]$: $S_t = \{i \mid f_i - d_i < t \le f_i\}$. Eq. (2) assigns a completion time of 0 to the dummy start activity 1. The precedence constraints given by Eq. (3) indicates that activity j can only be started only if all its predecessors i are completed. The resource constraint given in Eq. (4) indicate that for each time period $]t-1,t]$ and for each renewable resource k, the resource amounts required by the activities in progress cannot exceed the resource availability. The objective function is given as Eq. (1). The project duration is minimized by minimizing the finishing time of the unique dummy ending activity n [3].

Several exact and heuristics methods have been proposed for the solution of the RSPCP [4]. Exact methods such as dynamic programming, zero–one programming and implicit enumeration with branch and bound, aim at finding the optimal solution. Thus they need a lot of computational time, so they are inappropriate for large and complex projects. Heuristic solutions, such as priority rule-based scheduling, truncated branch and bound, disjunctive arc-based heuristics, are very fast in finding a solution, which makes them very practical, but this solution may not be optimal but may be a near optimal. Lately, some meta-heuristic techniques, such as genetic algorithms, simulating annealing, and tabu search, have been applied.

Most project management software employ priority rule-based heuristics for resource levelling. Priority rule-based scheduling consists of two components: a scheduling scheme and a priority rule. There are two scheduling schemes: serial and parallel. In serial scheduling, a decision set is formed at each scheduling time and contains all the unscheduled activities that are precedence feasible (all their predecessors have been scheduled), while in parallel scheme the decision set is formulated from all the unscheduled activities that are precedence and resource feasible at the scheduling time. A priority rule is then applied in order to select one activity from the set which will be scheduled [5]. Some examples of priority rules are: minimum slack, minimum latest finish/start time, and shortest processing time, but there are many different priority rules.

When considering resource constraints in scheduling, some critical activities may be delayed because of resource unavailability. In that case, the sequence of the critical path is lost [6]. *Critical sequence* was proposed by Weist [6] to find critical activities by considering both precedence relations and resource constraints. Activities in the critical sequence affect the projects completion, like the critical path for the resource unconstrained case. Weist [7] proposed a heuristic approach for the solution of resource constrained problem with constant resource limits, and Woodworth and Shanahan [8] developed a method for the calculation of the resource constrained float. Li and Willis [9] used the critical sequence in their time/cost trade-off algorithm for the resource constrained problem, but they did not calculate the float. Bowers [1] proposed a set of heuristics based on the critical sequence for the calculation of the resource constrained float. These algorithms assumed a fixed critical sequence with an invariant resource allocation.

3. Literature review

However, a resource constrained project network may have a lot of equivalent schedules [1]. An example of equivalent schedules is given in Fig 2, where two equally good solutions of network A (Fig. 1) are depicted. Activities 3 and 4 have the same duration, resource requirement, predecessors and successors, but in schedule (a), Activity 3 is critical and Activity 4 has a significant float, while in schedule (b), Activity 4 is critical and Activity 3 has a significant float. Raz and Marshall [10] introduced the *early and late scheduled dates*, which are the early and late dates of the activities, by considering both precedence relations and resource constraints. A heuristic algorithm was applied for the calculation of the *early scheduled dates*. Inversing this algorithm and starting from the *early scheduled finish* of the project, they calculated the *late scheduled dates*. Then, they calculated the *scheduled total and free float* using the classic way. Goldratt [11] proposed a methodology for the identification of the *Critical Chain*, which is defined as ''the longest chain of dependant steps", where ''dependencies between steps can be a result of a path or a result of a common resource" [11]. Critical Sequence and Critical Chain are based on the same concept. The difference is that critical chain uses reduced activity durations. Goldratt reduced the activity duration by assuming that it contains a protection against uncertainty and used this protection in some strategic positions in the schedule by adding time buffers.

Bowers [12] used some perturbations of the network inorder to generate the alternative schedules. Tormos and Lova [13] complement the concept of *Backward and Forward Free Slack* from Lova et al. [14] with the *Backward and Forward Total Slack* for the calculation of *Resource Constrained Activity Criticality Index* (RC ACI), and they integrated their method into Microsoft Project '98. Forward Free Slack of an activity is the amount of time in

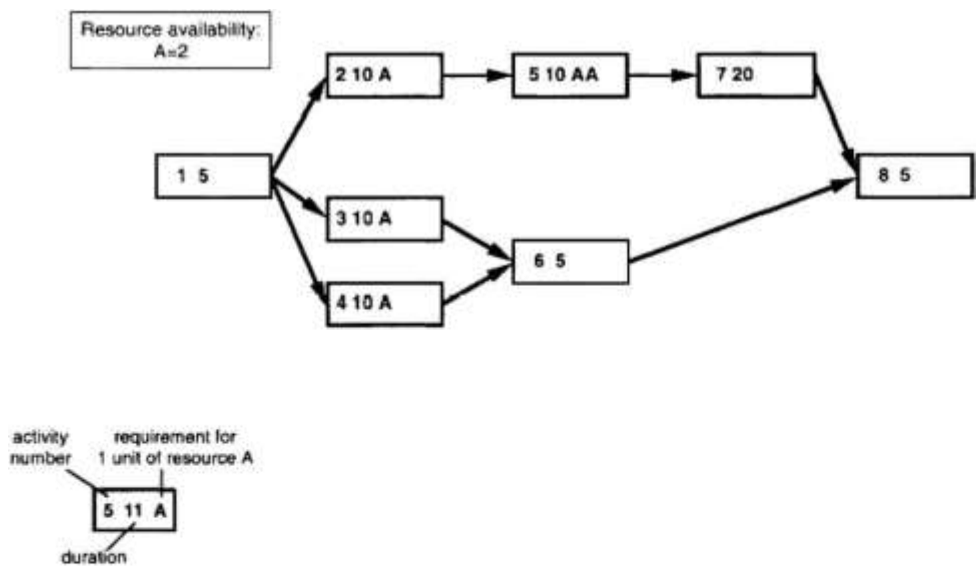


Fig. 1. Network A (source [1]).

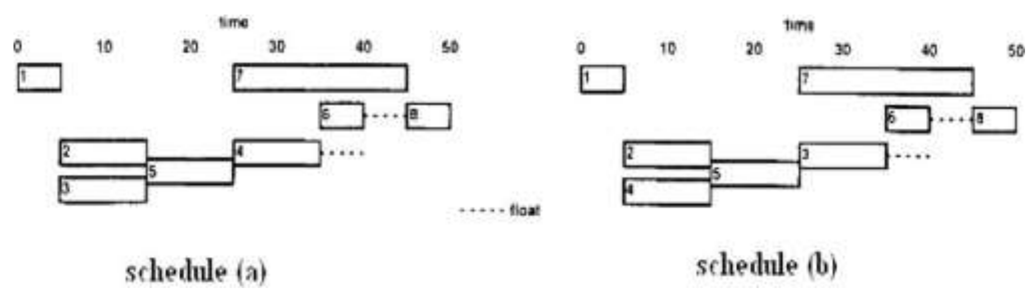


Fig. 2. Equivalent Schedules of Network A (source [1]).

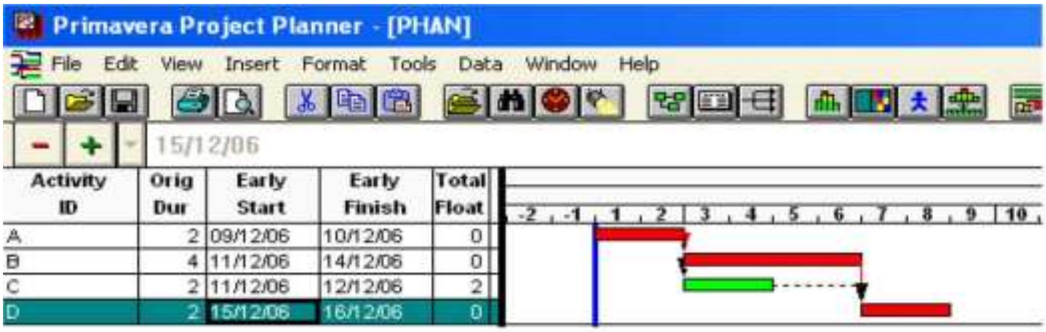
which the activity can be right-shifted allowing the successors to start on their scheduled dates. Backward Free Slack of an activity is the amount of time in which the activity can be left-shifted allowing the predecessors to start on their scheduled dates. Forward and Backward Free Slacks do not take into account resource constraints. Respectively, Forward and Backward Total Slack of an activity is amount of time in which the activity can be executed earlier or later, maintaining both the feasibility and the project completion time.

Fondahl [15] remarks that, as soon as resources are considered in a project, the original calculated network data, which are used as priority rules, are activity attributes and may have no meaning in resource constrained scheduling. For that reason Lu and Li [16] proposed the *Resource-Activity Critical Path Method* (RACPM), in which ''the dimension of resources is considered in addition to activity and time" [16]. The RACPM is a serial path heuristic method based on the knowledge-based system of Waugh and Froese [17], which can handle both renewable and non-renewable resources and uses work content as a priority rule.

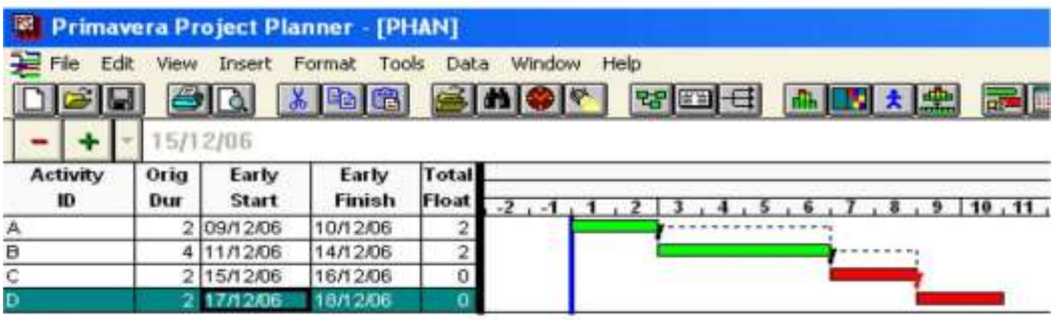
Kim et al. [18] developed the Resource Constrained Critical Path Method (RCPM) for the calculation of the resource constrained float without the *phantom float*. *Phantom Float* is called the difference between the theoretical remaining total float and the actual remaining total float [18]. This difference is the result of using the CPM for the backward scheduling after the resource constrained forward scheduling. This Phantom Float exists in project management software packages when executing resource levelling. An example of phantom float is shown in Fig. 3. Activities A–D share the same resource with constant availability of 1. Each activity requires 1 per day. When levelling the CPM schedule activities A and B seem to have a total float of 2 periods, but if activity B is delayed, the entire project will be delayed due to the use of a common resource. Hence, the float of the 2 periods does not exist. The advantages of RCPM are

- Gives more realistic schedules since it takes into account resource availability.
- Identifies the critical path and calculates floats correctly.
- Provides a stable schedule in a certain required level throughout the project duration.

At a specific time period of a project, a pair of activities may be critical (limit the advancement of the project). This



(a) CPM



(b) Resource Constrained Schedule

Fig. 3. Phantom float in P3.

means that if either of them is delayed, the project completion wouldnot be affected. But, any delay in both of them would delay the project. This pair of activities consists a *critical set* [19]. If there is more than one critical set within the same time period, then the project's completion is limited by a *critical cloud* [19]. Rivera and Duran [19] developed an algorithm for identifying critical sets and clouds.

Evaluations of resource levelling capabilities of project management software were performed by Johnson [20], Maroto and Tormos [21], Burley [22], Farid and Manoharan [23], Maroto et al. [24] and Kolisch [25]. Johnson [20] used the 110 problems of Patterson [26] to compare 13 versions of 7 project management software packages. Johnson compared the results given from the software packages with the optimal solutions of the problems as given by Demeulemeester and Herroelen [27]. Maroto and Tormos [21] used one problem with 51 activities and three resources to compare 7 versions of 6 software packages. Burley [22] compared Microsoft Project 3.0, Project Manager Workbench/w and Timeline 6.0, and Farid and Manoharan [23] compared Microsoft Project 3.0, Primavera Project Planner, Project Scheduler 5.0 and Time Line. Maroto et al. [24] evaluated six software packages: CA Superproject v.4.0, Microsoft Project v.4.0, Project Scheduler 6 v1.5, Time Line v.6, Primavera Project Planner v.2 and Artemis Schedule Publisher v.4.2. Ninety-six projects were generated for the evaluation of these software. The 96 projects were solved from the six software packages, and the results were compared against the optimal solutions obtained by the Branch and Bound procedure of Demeulemeester and Herroelen [27], 2 heuristics (MINLFT, FCFS) and the TRC Method. Kolisch [25] compared the solutions given by seven software packages: Artemis Schedule Publisher v.4.1, CA Superproject v.3.0 C, Microsoft Project v.4.0, Primavera Project Planner 1.0, Project Manager Workbench v.1.1.02w, Project Scheduler 6.0 v.1.02 and Time Line v.6.0 with the solution given by the Branch and Bound procedure of Demeulemeester and Herroelen [27] on 160 instances.

4. Objectives and methodology

The objective of this paper is to analyze the quality of resource constrained scheduling of project management software, by evaluating two widely used software packages [2], Primavera P6.0 and Microsoft Project 2007, and one open source, Open Workbench 1.1.6. Furthermore, the paper compares the results with the previous versions of these software packages, e.g. Primavera Project Planner v.3.1, Microsoft Project 2003 and Open Workbench 1.1.3. Project management software packages usually use priority rule-based heuristic algorithms for the resource levelling, but they donot offer other information about the details of the algorithm, e.g. which scheduling scheme or which priority rule they use, if it is static or dynamic. Primavera Project Planner gives the user the opportunity to the user to choose the priority rule but it doesnot illustrate to the user the possible usefulness of changing the priority rule or using more than one. These criteria can cause many possible methods which can end up to a different schedule. Therefore, the paper examines the effectiveness of resource levelling of the software packages on two real construction projects to show that the duration of the project, when considering resource constraints, depends on the software and/or the method used.

The first benchmark instance is a real housing project which consisted of 96 houses. The focus was on the concreting of these 96 houses because there was problem. The project has 98 activities, and 1 renewable resource (concrete) was considered. According to the projects initial plan, the average daily concrete requirement, during the projects horizon, was 112.28m^3 with a maximum of 126.4m^3 but the constructor could procure only 80m^3. The following parameters were used inorder to determine the complexity of the problem [28]: The Network Complexity (*NC*), which measures the average number of precedence relations per activity. The Resource Factor (*RF)*, which defines the average number of constrained resources requested by an activity. *RF* is normalized to the interval [0, 1]. For *RF*=1, each activity requires every one of the R resources, and for *RF*= 0, none of the activities requires any of the resources. The resource strength (*RS)*, which measures the proportion of resource demand and availability *RS*, is also normalized to the interval [0,1], and for *RS*=0 the problem is highly resource constrained, while for *RS*=1, the project is resource unconstrained. For this instance, the values are *NC*=2.6, *RF*=1 and *RS*=0.62.

The second instance is the construction of a shopping mall consisting of 19 buildings. When the resource levelling was applied, the project was in progress and the contractor wanted to program the remaining work, especially electromechanological and activities requiring specialised work, by considering the availability of workgroups. The time-schedule consists of 668 activities and 7 renewable resources. The complexity factors for this instance are *NC* = 0.86, *RF* = 0.05, and *RS* equals 0 for 5 of the seven resources and 0.4 and 0.5 for the other two.

Microsoft Project and Open workbench use a standard built-in procedure so there is one result for each. For P6, besides the default resource levelling, six priority rules and all possible combinations between them were used, so P6 obtained 31 results. The priority rules used are Late Start Time (LST), Late Finish Time (LFT), Minimum Slack (MSLK), Ranked Positional Weighted Method (RPWM) and Enhanced Positional Weighted Method (EPWM). The last two rules do not exist in Primavera, but were computed and imported to Primavera as Custom Data Items. The comparison of these results made using the percentage deviation from the project duration without resource constraints, which is 464 days for instance 1 and 238 for instance 2.

5. Results

On comparing the default options given from the software (Table 1), Primavera P6 gives the best results and is followed by MS Project in both the instances. Open Workbench has a great deviation especially in the second instance.

On comparing all the results (Table 2), Primavera P6 default and by LST rule give the best solution for both instances with an average deviation of 41.11%. MS Project is ranked 3 (46.14%) and is followed by P6 by RPWM and by LFT (47.19%). The worst result was given by Open Workbench with an average deviation of 167.79%. Especially in the second instance, Open Workbench gives a duration that is over the double from CPM duration. A point to be noted is that P6 by EPWM rule gives the best solution for the second instance, which is not the case for the first instance. Also MS Project in the first instance gives the same result with P6 by PWM and LFT, but in the second instance gives slightly better solution.

The evaluated software can be divided into three groups according to the quality of the schedules they produce. The first group consisted of those who have a better performance (average deviation less than 50%), which are P6 default, P6 by LST, MS Project, P6 by LFT and P6 by PWM. The second group contains the medium quality software (average deviation 50% to 60%), which are P6 by EPWM and P6 by MSLK, and the third group is formed by the worst quality software (average deviation over 60%), which is P6 by SPT, and the result of Open Workbench is not included as it is an outlier (Fig. 4).

The two instances that are also solved by Primavera, combining 2 priority rules and the average results are presented below (Table 3). The first rules are given in the rows and the second rules are given in the columns. Better results are given from LST, LFT and PWM when used as first rules. For tie breaking rule, better results are given by EPWM, PWM and LST. The combination of PWM and LST and the combination of LST and MSLK give the best solution for both the instances.

Table 1
Duration and percentage deviation obtained by the project management software for default options

Rule	1st Instance		2nd Instance		Average
	Duration	Percentage deviation from CPM (%)	Duration	Percentage deviation from CPM (%)	Average percentage deviation from CPM (%)
P6 Default	709	52.80	308	29.41	41.11
MS Project standard	744	60.34	314	31.93	46.14
Open workbench standard	863	85.99	832	249.58	167.79

Table 2
Duration and percentage deviation obtained by the project management software

Rule	1st Instance			2nd Instance			Average percentage deviation from CPM (%)
	Duration	Percentage deviation from CPM (%)	Rank	Duration	Percentage deviation from CPM (%)	Rank	
LST	709	52.80	1	308	29.41	1	41.11
P6 Default	709	52.80	1	308	29.41	1	41.11
MS Project standard	744	60.34	2	314	31.93	2	46.14
PWM	744	60.34	2	319	34.03	3	47.19
LFT	744	60.34	2	319	34.03	3	47.19
EPWM	823	77.37	3	308	29.41	1	53.39
MSLK	823	77.37	3	327	37.39	4	57.38
SPT	893	92.46	4	336	41.18	5	66.82
Open workbench standard	863	85.99	5	832	249.58	6	167.79

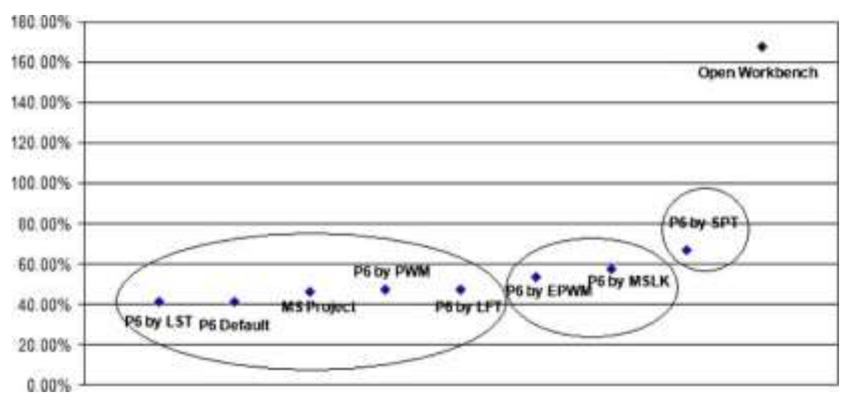


Fig. 4. Performance in resource constrained project scheduling.

Table 3
Combinations of priority rules in Primavera P6

	EPWM(%)	PWM(%)	LFT(%)	LST(%)	MSLK(%)	SPT(%)	Average(%)
EPWM	–	53.39	51.87	53.39	57.38	60.28	55.26
PWM	43.42	–	45.30	41.11	45.10	55.72	46.13
LFT	43.84	45.85	–	41.95	47.61	48.03	45.45
LST	44.39	42.08	42.08	–	41.11	42.08	42.34
MSLK	56.12	53.18	57.38	57.38	–	57.38	56.29
SPT	48.58	48.65	54.67	54.67	66.07	–	54.53
Average	47.27	48.63	50.26	49.70	51.45	52.70	

6. Conclusions

This paper evaluated the resource levelling capabilities of three project management software packages on two real construction problems. The results show that the project duration depends on the software or the method used. It could be supported that, for problems of that size and complexity and objective function the makes pan minimization, there is an increase in makespan from 41.11% to 167.79% of the resource unconstrained schedule and that Primavera P6 outperforms MS Project and Open Workbench. Because of the big range of the results, project managers should not rely on the first result they get but try other rules or even software if possible.

Primavera P6 is the most effective as it allows the user to define theoretically unlimited number of criteria as priority rules. For the other software, the main problems are that in MS Project the user cannott have constraint on a material (user has to manage it as labor) and that Open Workbench was too slow when scheduling. Previous versions of the same software programs, e.g. Primavera Project Planner v.3.1, MS Project 2003 and Open Workbench 1.1.3, gave exactly the same results.

Regarding the different methodology used in the previous evaluations, it is not easy to make an exact comparison of the results. Johnson [20] compared seven software packages, with MS Project (1.0 and 3.0) and Primavera (4.0, 4.1 and 5.0) among them, against Talbot's Optimizer [29] and Patterson Heuristic [30] on the 110 Patterson's problems. The results showed that Timeline was the most successful with an average of 5.03% over the optimal solution. MS project was the least successful. Found the optimal solution for 10 projects, and had an average of 25.6% over the optimal solution. Primavera had an average of 7.45% over the optimal duration. Maroto and Tormos [21] compared the results from seven software packages against CPM on a project with 51 activities and three resources. They did not use the product of Primavera. MS Project 3.0 was within the first three that outperformed. Farid and Manoharan [23] used 10 of the 110 Patterson's problems to compare five software packages against Talbot's Optimizer [29] and Patterson Heuristic [30]. MS Project v.3.0 and Primavera Project Planner v.5.0 are ranked last (fourth and fifth, respectively). Maroto, Tormos and Lova [24] compared six software packages on 96 auto-generated problems against Branch and Bound procedure from Demeulemeester [31]. Primavera standard, Primavera by LFT and MS Project are placed in the group of medium quality. Primavera by MSLK was placed in the third group of worst performance. Kolisch [25] used seven software packages on 160 instances against Branch and Bound procedure from Demeulemeester and Kolisch [31]. Primavera Project Planner v.1.0 was placed in the first group of best performance, Microsoft project in the second and Project Manager Workbench v.1.1.02w in the third.

Our results seem to be consistent with the evaluation of Maroto et al. [24]. Primavera standard, Primavera by LFT and MS Project are placed in the same group (first for us second for Maroto et al., while Primavera by MSLK is placed in the next one.

CPM is inappropriate for scheduling resource constrained projects, and project managers shouldnot rely only on this method when dealing with limited resources [32]. Project management software packages should incorporate other more efficient methods for resource constrained scheduling and should try to illuminate some drawbacks such as Phantom Float. It would be also interesting if project management software could have other objectives, besides the makespan minimization, such as maximizing the net present value.

References

[1] Bowers JA. Criticality in resource-constrained networks. J Oper Res Soc 1995;46(1):80–91.

[2] Liberatore MJ, Pollack-Johnson B, Smith CA. Project management in construction: software use and research direction. J Constr Eng Manage 2001;127(2):101–7.

[3] Herroelen W, De Reyck B, Demeulemeester E. Resource-constrained project scheduling: A survey of recent developments. Comput Oper Res 1998;25(4):279–302.

[4] Kolisch R, Padman R. An integrated survey of deterministic project scheduling. Omega 2001;29(3):249–72.

[5] Kolisch R, Hartmann S. Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis. In: Weglarz J, editor. Project Scheduling: recent models, algorithms and applications. Boston: Kluwer Academic Publishers; 1998.

[6] Wiest JD. Some properties of schedules for large projects with limited resources. Oper Res 1964;12(3):395–428.

[7] Wiest JD. A heuristic model for scheduling large projects with limited resources. Manage Sci 1967;13(6):b359–77.

[8] Woodworth BM, Shanahan S. Identifying the critical sequence in a resource constrained project. Int J Project Manage 1988;6(2):89–96.

[9] Li RK-Y, Willis RJ. Resource constrained scheduling within fixed project durations. J Oper Res Soc 1993;44(1):71–80.

[10] Raz T, Marshall B. Effect of resource constraints on float calculations in project networks. Int J Project Manage 1996;14(4):241–8.

[11] Goldratt E. Critical chain. Great Barrington: The North River Press Publishing Corporation; 1997.

[12] Bowers J. Multiple schedules and measures of resource constrained float. J Oper Res Soc 2000;51(7):855–62.

[13] Tormos P, Lova A. Tools for resource-constrained project scheduling and control: forward and backward slack analysis. J Oper Res Soc 2001;52(7):779–88.

[14] Lova A, Maroto C, Tormos P. A multicriteria heuristic method to improve resource allocation in multiproject scheduling. Eur J Oper Res 2000;127(2):408–24.

[15] Fondahl JW. The development of the construction engineer: past progress and future problems. J Constr Eng Manage 1991;117(3): 380–92.

[16] Lu M, Li H. Resource-activity critical-path method for construction planning. J Constr Eng Manage 2003;129(4):412–20.

[17] Waugh LM, Froese TM. Constraint knowledge for construction scheduling. In: IEE Conf. Publ. Institute of Electrical Engineers, London, UK; 1990. p. 114–118.

[18] Kim K, De La Garza JM. Phantom float. J Constr Eng Manage 2003;129(5):507–17.

[19] Rivera FA, Duran A. Critical clouds and critical sets in resource-constrained projects. Int J Project Manage 2004;22(6):489–97.

[20] Johnson RV. Resource constrained scheduling capabilities of commercial project management software. Project Manage J 1992;22(4): 39–43.

[21] Maroto C, Tormos P. Project Management: An evaluation of software quality. Int Trans Oper Res 1994;1(2):209–21.

[22] Burley E. Resource levelling with project management software. In: Paper presented at the INFORMS International Meeting, Singapore; 1995.

[23] Farid F, Manoharan S. Comparative analysis of resource-allocation capabilities of project management software packages. Project Manage J 1996;24:35–44.

[24] Maroto C, Tormos P, Lova A. The evolution of software quality in project scheduling. In: Weglarz J, editor. Project scheduling–recent models, algorithms and applications. Boston: Kluwer Academic Publishers; 1998.

[25] Kolisch R. Resource allocation capabilities of commercial project management software packages. Interfaces 1999;29(4):19–31.

[26] Patterson JH. A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. IIE Trans 1984;30(7):854–67.

[27] Demeulemeester E, Herroelen W. A branch and bound procedure for the multiple resource-constrained project scheduling problem. Manage Sci 1992;38(12):1803–17.

[28] Kolisch R, Sprecher A, Drexl A. Characterization and generation of a general class of resource-constrained project scheduling problems. Manage Sci 1995;41(10):1693–703.

[29] Talbot FB, Patterson JH. An efficient integer programming algorithm with network cuts for solving resource constrained scheduling problems. Manage Sci 1978;24(11):1163–74.

[30] Patterson JH. Project scheduling: the effects of problem structure on heuristic performance. Naval Res Logist Quart 1976;23(1): 95–123.

[31] Demeulemeester E, Kolisch. A branch and bound procedure for the multiple resource constrained projects scheduling problem. Manage Sci 1992;38(12):1803–18.

[32] Herroelen W, Leus R. Identification and illumination of popular misconceptions about project scheduling and time buffering in a resource-constrained environment. J Oper Res Soc 2005;56:102–9.