Flex Serv Manuf J
DOI 10.1007/s10696-016-9236-8

A note on the selection of priority rules in software packages for project management

Philipp Baumann[1] · Norbert Trautmann[1]

Abstract Various software packages for project management include a procedure for resource-constrained scheduling. In several packages, the user can influence this procedure by selecting a priority rule. However, the resource-allocation methods that are implemented in the procedures are proprietary information; therefore, the question of how the priority-rule selection impacts the performance of the procedures arises. We experimentally evaluate the resource-allocation methods of eight recent software packages using the 600 instances of the PSPLIB J120 test set. The results of our analysis indicate that applying the default rule tends to outperform a randomly selected rule, whereas applying two randomly selected rules tends to outperform the default rule. Applying a small set of more than two rules further improves the project durations considerably. However, a large number of rules must be applied to obtain the best possible project durations.

Keywords Project scheduling · Software · Priority rules · Experimental analysis

1 Introduction

A project is a one-time endeavor dedicated to some objective and consists of a series of activities that are interrelated by precedence constraints and that require time and resources for execution (cf., e.g., Brucker et al. 1999). The lifecycle of a project

✉ Norbert Trautmann
norbert.trautmann@pqm.unibe.ch

Philipp Baumann
philipp.baumann@pqm.unibe.ch

[1] University of Bern, Schuetzenmattstrasse 14, Bern, Switzerland

Published online: 16 January 2016

consists of the conception, definition, planning, execution, and termination phases (cf. Klein 2000). We focus on the planning phase, which includes temporal scheduling, i.e., computing the earliest and the latest start times and the slack times of the activities, and resource allocation, i.e., allocating the scarce resources over time to the execution of the activities. In practice, project managers often use commercial project management software during the planning phase (see, e.g., Herroelen 2005; Liberatore and Pollack-Johnson 2003; White and Fortune 2002).

The focus of this paper are the resource-allocation capabilities of eight widely-used commercial project management software packages. We perform an experimental analysis based on 600 instances with 120 activities each of the widely-studied resource-allocation problem setting RCPSP. This resource-constrained project scheduling problem (RCPSP) is defined as follows. Given are a set of n project activities with prescribed durations, a set of precedence relationships among these activities, a set of renewable resources with finite capacities, and the requirements of each activity for these resources. Sought is a schedule, i.e. a start time for each activity, such that all precedence relationships are fulfilled, at no point in time the total requirement for any resource exceeds its capacity, and the project duration is minimized. In the deterministic RCPSP, all input data is assumed to be known in advance; for the stochastic variant SRCPSP, in which the activity durations are assumed to follow some probability distribution, we refer to Fang et al. (2015). The RCPSP represents an NP-hard optimization problem; therefore, the resource-allocation procedures of the software packages apply heuristic methods. In some of these packages, the user can influence the resource-allocation procedure by selecting a rule for computing the priority values for the activities. It is proprietary information which specific heuristic is used and how the priority values are considered internally. Therefore, the questions of how the performance of the implemented heuristics varies across the priority rules, and whether a good performance can be obtained by applying a single or a few priority rules arise.

In the studies of Assad and Wasil (1986), Wit and Herroelen (1990), Johnson (1992), Maroto and Tormos (1994), Kolisch and Hempel (1996), Farid and Manoharan (1996), Khattab and Søyland (1996) and Kolisch (1999), the resource-allocation capabilities of various project management software packages under the default settings were analyzed, using a relatively small set of RCPSP instances. Kastor and Sirakoulis (2009) used two real-world construction projects to evaluate the performance of three packages under the default settings and when specific priority rules were selected. The studies of Mellentien and Trautmann (2001), Trautmann and Baumann (2009a, b) and Baumann and Trautmann (2015) refer to a sizeable set of instances and various priority rules for those packages in which the user can select a rule; in these studies, the best-possible performance for each package is reported, i.e., it is assumed that the user enumerates all priority rules for each instance individually. The results indicate that under this best-case assumption, the heuristic methods perform considerably better than under the default rule. The best-case assumption, however, does not reflect the behavior of project managers who apply a single priority rule or only a few priority rules.

In this paper, we analyze the performance distribution of the priority rules for each software package in more detail using violin plots as a statistical method. The results of our analysis show that the default rule is superior to many priority rules, and thus, selecting an arbitrary priority rule is risky. Some priority rules, however, perform surprisingly well when applied individually. Subsequently, we analyze the trade-off between the number of applied rules and the obtained performance. We compare four strategies for selecting a set of priority rules; it turned out that for all strategies, the shortest project duration rapidly decreases in the set size, but an unexpected high number of rules needs to be applied to obtain the performance under the best-case assumption.

The remainder of this paper is structured as follows. In Sect. 2, we describe how the user can influence the resource-allocation methods of the software packages. In Sect. 3, we present the design and the results of our analysis. In Sect. 4, we provide some concluding remarks and directions for future research.

2 Project management software packages

In this section, we illustrate how the user can influence the resource-allocation procedures of eight software packages; a detailed presentation is provided in Baumann and Trautmann (2015).

Our analysis was restricted to commercial software packages that contain a resource-allocation procedure and that can be controlled using Visual Basic for Applications (VBA); the latter is required for automatically performing the experimental analysis. Table 1 lists the name, the number of the release used, and the developer for each package.

All of the software packages address the scheduling of resource-constrained projects using the same types of heuristic methods. After each input or update of the project data, the package computes the precedence-feasible early-start schedule. The resource-allocation procedure then applies a schedule-generation scheme to devise a precedence- and resource-feasible schedule. The scheme that is implemented in the individual packages is proprietary information; however, all schemes select the activities to be scheduled or delayed based on the priority values of the activities.

Table 1 Analyzed software packages

	Name	Release (built)	Developer
ACO	Acos Plus.1	9.4b (755)	ACOS Projektmanagement
ADT	Adept Tracker Professional	3.13 (10953)	WangTuo Software
CSP	CS Project Professional	3.8 (.06)	CREST Software
M10	Microsoft Project 2010	14 (6129.5000)	Microsoft Corporation
M13	Microsoft Project 2013	15 (4420.1017)	Microsoft Corporation
PP6	Primavera P6	8.2 (1926)	Oracle Corporation
SCI	Sciforma 5	5.0c (2994)	Sciforma Corporation
TPP	Turbo Project Professional	4 (221.5)	OfficeWork Software

The same approach is used by many constructive heuristics that have been presented in the literature (e.g., Hartmann 2013). In most packages, the user can manually specify these priority values; however, we did not utilize this option. The packages ADT, M10, M13, SCI, and TPP contain a single rule (default rule) for computing the priority values. In the packages ACO, CSP, and PP6, the user may choose between different priority rules as follows:

ACO: The rules implemented in Acos Plus.1 include total float (A2), free float (A5), total number of predecessors (A6) or successors (A7) or both (A8), and shortest (A9) or longest duration (A4).

CSP: In CS Project, the user can define multi-level priority rules consisting of up to four levels. On each level, a rule such as total float (C1), free float (C2), duration (C4), early start (C5), or late finish (C8) can be applied in ascending (a) or descending order (d). In our analysis, we tested all combinations of rules C1, C2, C4, C5, and C8 (ascending and descending) for the first three levels plus the selection of no rule (Na) for the second and the third levels, resulting in a total of 570 of such multi-level rules. In addition, we tested the CARLO ("cost and resource levelling optimisation") algorithm, which automatically applies different priority rules and returns, according to the help file, the best schedule found. However, it is not mentioned which criterion is used to select the best schedule.

PP6: In Primavera P6, the user can define multi-level priority rules that consist of an unlimited number of levels. On each level, a rule such as duration (P8), free float (P5), total float (P13), early start (P4), early finish (P3), late start (P7), or late finish (P6) can be applied in ascending (a) or descending (d) order. In our analysis, we used two levels, resulting in a total of 196 multi-level rules.

Furthermore, we tested the default rules of the ACO, CSP, and PP6 packages. In CSP, the combination C3a-C1a-C5a-C2a is selected by default. In ACO and PP6, the default rule uses the user-defined priority values, even if the user has not specified any priority values.

3 Computational analysis

We installed the software packages on various standard PCs with a Windows operating system and applied the packages to the 600 instances with 120 activities (J120 set) of the well-known PSPLIB library (cf. Kolisch and Sprecher 1997), which we downloaded from the webpage http://www.om-db.wi.tum.de/psplib/. We measured performance in terms of the relative deviation between the obtained project duration and the critical path-based lower bound (CLB), which is the minimum project duration without considering the resource constraints. The reported results can thus be used as reference values for future studies that evaluate performance improvements of new or updated software packages. The

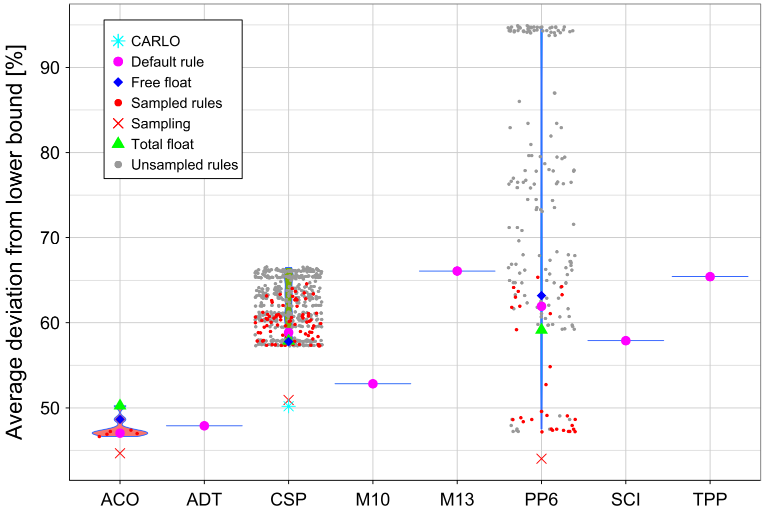

Fig. 1 Performances of individual priority rules

running time required for resource allocation for a single project never exceeded 30 s.

We first analyzed the performance of the packages when a single priority rule is applied to all 600 instances. Figure 1 presents the priority rules of the packages as points and reports the average relative deviation from the CLB for each rule. Certain rules, such as the default rule, are highlighted with specific markers. For the packages ACO, CSP, and PP6, which offer different priority rules, we used violin plots to visualize the performance distribution of the tested priority rules. The results show that the default rule is superior to many priority rules, and thus, selecting an arbitrary priority rule is risky. Even common priority rules such as the free float or the total float rule do not belong to the best performing rules. Some priority rules, however, considerably outperform the default rule. The selection of priority rules is most critical for the package PP6.

Next, we investigated for the packages ACO, CSP, and PP6 the best-possible performance that can be achieved by applying multiple priority rules to each instance rather than a single priority rule; in Fig. 1, the red marker "×" pinpoints this performance. The red dots represent the priority rules that needed to be applied to at least one instance of the test set to achieve this best-possible performance. The number of these rules is surprisingly high.

Assuming that project managers apply only a single or a few rules, it is of interest to investigate the trade-off between the number of rules applied and the performance obtained. We selected k rules according to different selection strategies and applied these rules to all instances. For each instance, we considered the smallest deviation from the CLB. The following four strategies were used to select k rules:

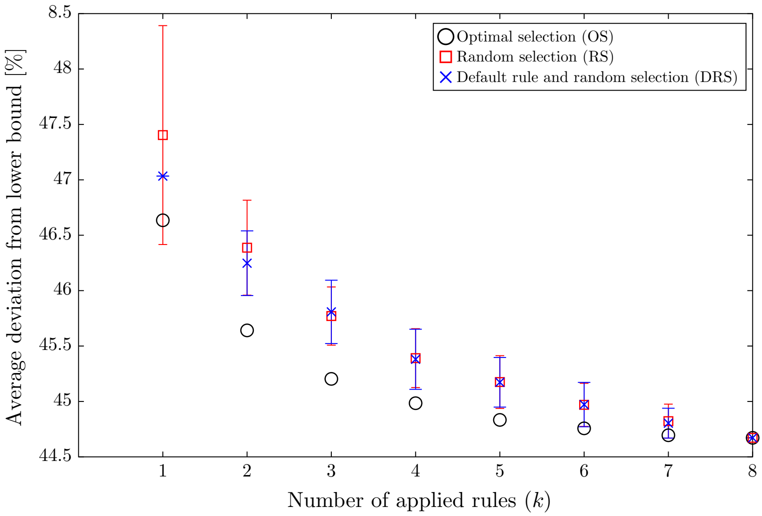

Fig. 2 Comparison of rule selection strategies for ACO

- Optimal selection (OS): Select the k priority rules that lead to the smallest average deviation from the CLB. We determined the optimal set of k rules by solving a binary linear program that takes as input the results of all priority rules. This selection strategy does not represent a practical approach, but serves as baseline.
- Random selection (RS): Select k rules randomly.
- Default rule and random selection (DRS): Select the default rule and select $k-1$ additional rules randomly.
- Default rule and random candidate selection (DRCS): Select the default rule and select $k-1$ additional rules randomly from a set of candidate rules. The set of candidate rules only contains single-level rules, resulting in 10 candidate rules for CSP and 14 candidate rules for PP6. This selection strategy is only used for the packages CSP and PP6, which have multi-level rules.

Figures 2, 3, and 4 show the performances of the strategies for different sample sizes k. The strategies that involve a random selection (RS, DRS, and DRCS) were applied 100 times for each value of k. For these strategies, the markers report the mean average deviation, and the error bars span a range of σ_k above and below the marker, where σ_k denotes the standard deviation of the average deviations of the 100 repetitions. In Figs. 3 and 4, the OS and DRCS markers are only drawn up to the value of k for which they reach the lowest point. Table 2 lists for $k = 1, 2, 3$ the rules that are selected under the optimal selection strategy. The results indicate that if several rules are applied, then the shortest project duration rapidly decreases. For the package CSP, the selection strategy DRS considerably outperforms the strategy RS. For the package PP6, for small values of k, the strategy DRCS outperforms the strategies RS and DRS.

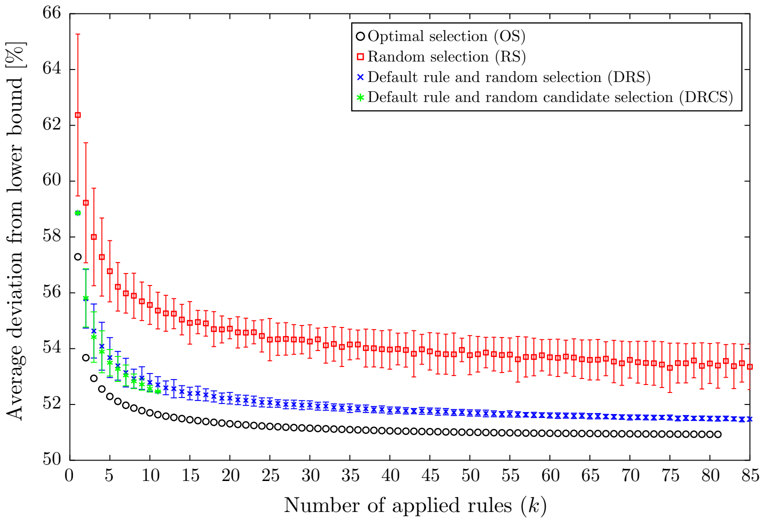

Fig. 3 Comparison of rule selection strategies for CSP

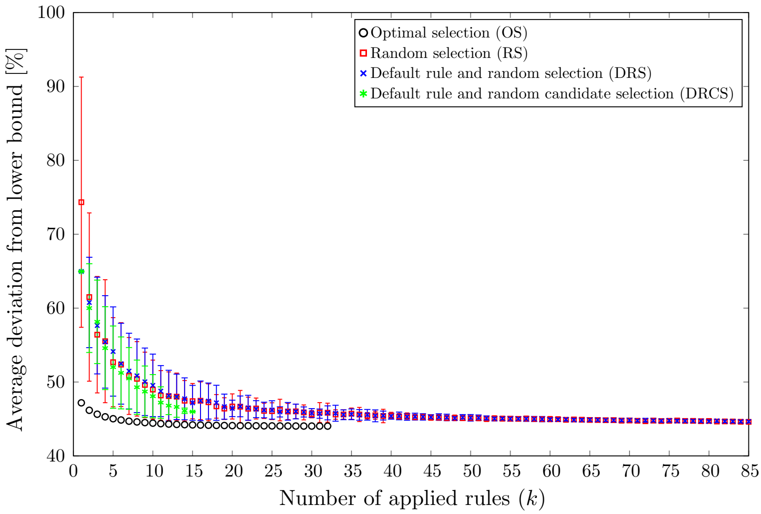

Fig. 4 Comparison of rule selection strategies for PP6

4 Conclusions

We analyzed the resource-allocation procedures of eight project-management software packages. Based on the 600 instances of the PSPLIB J120 test set, we investigated how the selection of priority rules influences the performance of the procedures. For some packages the performance distribution of the priority rules is rather wide. Hence, arbitrarily selecting a single priority rule bears a considerable

Table 2 Optimal priority rule subsets of size k for the packages ACO, CSP, and PP6

k	ACO		CSP		PP6	
	Rule (s)	AD [%]	Rule (s)	AD [%]	Rule (s)	AD [%]
1	A8	46.63	C8a-C1a-C4a	57.29	P5a-P3a	47.18
2	A3, A8	45.64	C8a-C1a-Na, D	53.68	P4a-P5a, P5a-P3a	46.17
3	A3, A6, A8	45.20	C2a-C8a-C4d, C8a-C2a-C5d, D	52.94	P4a-P6d, P5a-P1a, P5a-P1d	45.64

risk of generating poor schedules. We found that this risk rapidly decreases with the number of applied priority rules. However, to obtain the best-possible performance, a high number of rules must be applied. For individual packages, we identified effective priority rule selection strategies that allow a considerable reduction in the selection risk to be achieved with only few priority rules to be applied.

In future studies, the resource-allocation capabilities of the software packages in other problem settings, e.g., resource calendars, should be analyzed. Another interesting research direction is the simultaneous planning of multiple projects using such software packages (see Beşikci et al. 2013).

References

Assad A, Wasil E (1986) Project management using a microcomputer. Compute Oper Res 13:231–260

Baumann P, Trautmann N (2015) Resource-constrained project scheduling with project management information systems. In: Schwindt C, Zimmermann J (eds) Handbook on project management and scheduling, vol 2., International handbooks on information systems. Springer, New york, pp 1385–1400

Beşikci U, Bilge Ü, Gündüz U (2013) Resource dedication problem in a multi-project environment. Flex Serv Manuf J 25:206–229

Brucker P, Drexl A, Möhring R, Neumann K, Pesch E (1999) Resource-constrained project scheduling: notation, classification, models, and methods. Eur J Oper Res 112:3–41

De Wit J, Herroelen W (1990) An evaluation of microcomputer-based software packages for project management. Eur J Oper Res 49:102–139

Fang C, Kolisch R, Wang L, Mu C (2015) An estimation of distribution algorithm and new computational results for the stochastic resource-constrained project scheduling problem. Flex Serv Manuf J 27:585–605

Farid F, Manoharan S (1996) Comparative analysis of resource allocation capabilities of project management software packages. Proj Manag J 27:35–44

Hartmann S (2013) Project scheduling with resource capacities and requests varying with time: a case study. Flex Serv Manuf J 25:74–93

Herroelen W (2005) Project scheduling–theory and practice. Prod Oper Manag 14:413–432

Johnson R (1992) Resource constrained scheduling capabilities of commercial project management software. Proj Manag J 22:39–43

Kastor A, Sirakoulis K (2009) The effectiveness of resource levelling tools for resource constraint project scheduling problem. Int J Proj Manag 27:493–500

Khattab M, Søyland K (1996) Limited resource allocation in construction projects. Comput Ind Eng 31:229–232

Klein R (2000) Scheduling of resource-constrained projects. Kluwer, Boston

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

Kolisch R, Hempel K (1996) Experimentelle Evaluation der Kapazitätsplanung von Projektmanagementsoftware. Zeitschrift für betriebswirtschaftliche Forschung 48:999–1018

Kolisch R, Sprecher A (1997) PSPLIB—a project scheduling problem library. Eur J Oper Res 96:205–216

Liberatore M, Pollack-Johnson B (2003) Factors influencing the usage and selection of project management software. IEEE Trans Eng Manag 50:164–174

Maroto C, Tormos P (1994) Project management: an evaluation of software quality. Int Trans Oper Res 1:209–221

Mellentien C, Trautmann N (2001) Resource allocation with project management software. OR Spectr 23:383–394

Trautmann N, Baumann P (2009a) Project scheduling with precedence constraints and scarce resources: an experimental analysis of commercial project management software. In: Fleischmann B, Borgwardt KH, Klein R, Tuma A (eds) Operations research proceedings 2008, vol 2008. Springer, Berlin, pp 165–170

Trautmann N, Baumann P (2009b) Resource-allocation capabilities of commercial project management software: an experimental analysis. In: International conference on computers & industrial engineering, CIE 2009, IEEE, pp 1143–1148

White D, Fortune J (2002) Current practice in project management—an empirical study. Int J Proj Manag 20:1–11

Philipp Baumann received his BSc, MSc and PhD in Business Administration from the University of Bern (Switzerland) in 2007, 2009, and 2013, respectively. He did his postdoc at the Department of Industrial Engineering and Operations Research at the University of California, Berkeley. Currently, he is Assistant Professor at the Department of Business Administration at the University of Bern (Switzerland). His research interests include optimization in finance, project management and project scheduling, production planning and control, data mining, and network-based optimization. He has published in European Journal of Operational Research, Flexible Services and Manufacturing Journal, International Journal of Production Research, and Mathematical Methods of Operations Research.

Norbert Trautmann received his BSc and MSc in Business Engineering/Management Science and his PhD in Business and Economics from the University of Karlsruhe (Germany) in 1994, 1997, and 2000, respectively. Currently he holds the Chair in Quantitative Methods in Business Administration at the University of Bern (Switzerland). His research interests are in combinatorial optimization, project management and project scheduling, production planning and control, and portfolio optimization. He has published in European Journal of Operational Research, Flexible Services and Manufacturing Journal, International Journal of Production Research, Journal of Scheduling, Mathematical Methods of Operations Research, and OR Spectrum.