OR Spektrum (2001) 23: 383–394

OR Spektrum

Resource allocation with project management software*

Christoph Mellentien and Norbert Trautmann

Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 76128 Karlsruhe, Germany (e-mail: {mellentien,trautmann}@wior.uni-karlsruhe.de)

Abstract. We present results of a benchmark test evaluating the resource allocation capabilities of the project management software packages Acos Plus.1 8.2, CA SuperProject 5.0a, CS Project Professional 3.0, MS Project 2000, and Scitor Project Scheduler 8.0.1. The tests are based on 1560 instances of precedence– and resource–constrained project scheduling problems. For different complexity scenarios, we analyze the deviation of the makespan obtained by the software packages from the best feasible makespan known. Among the tested software packages, Acos Plus.1 and Project Scheduler show the best resource allocation performance. Moreover, our numerical analysis reveals a considerable performance gap between the implemented methods and state–of–the–art project scheduling algorithms, especially for large–sized problems. Thus, there is still a significant potential for improving solutions to resource allocation problems in practice.

Key words: Project management software – Resource allocation – Performance evaluation

1 Introduction

A project is a unique endeavour that can be divided into individual activities linked by precedence relationships. During their execution, activities

* This research is supported by the Deutsche Forschungsgemeinschaft under Grant Ne 137/4. We also would like to thank the software companies that supported this research by providing the respective software packages.

Correspondence to: N. Trautmann

require resources and thus incur costs. Project planning consists of temporal scheduling, resource allocation, and cost management. Project planning problems arise e.g. in software development, civil engineering, production planning, or audit–staff scheduling.

Different commercial software packages are available for computer–aided project management. In this paper we present an experimental performance analysis of the resource allocation modules implemented in the software packages Acos Plus.1 8.2 (Acos), CA SuperProject 5.0a (Computer Associates), CS Project Professional 3.0 (CREST Software), MS Project 2000 (Microsoft), and Project Scheduler 8.0.1 (Scitor). Previous versions of some of these packages have been assessed in earlier surveys (cf. Johnson, 1992; Farid and Manoharan, 1996; Kolisch and Hempel, 1996a; Kolisch, 1999). Our test set (cf. Kolisch et al., 1995) consists of 480 projects with 30 activities, 480 projects with 60 activities, and 600 projects with 120 activities, and four resources each. As no software package is designed to consider general temporal constraints (minimum and maximum time lags) between activities, the comparison is restricted to projects with completion–to–start precedence relationships.

The remainder of this paper is organized as follows. In Section 2 we introduce the resource–constrained project scheduling problem under consideration. Section 3 sketches the main resource allocation features of the individual software packages. In Section 4 the experimental parameters used for generating the problem instances are defined. In Section 5 we discuss the results of our benchmark tests obtained from different complexity scenarios. In particular, we study the impact of parameters like problem size, resource scarcity, and number of precedence constraints.

2 Problem statement

The resource–constrained project scheduling problem $PS|prec|C_{max}$ under consideration can be stated as follows (cf. Brucker et al., 1999). A project consists of a set $V := \{1, \ldots, n\}$ of n activities. Let p_i denote the duration of activity $i \in V$. Preemption of activities is not allowed. For each activity $i \in V$, a set $Pred_i$ of predecessor activities is given. An activity cannot be started before all of its predecessor activities have been completed. For executing the activities, a set $\mathcal{R}$ of renewable resources k with limited capacity $R_k > 0$ is available. An activity $i \in V$ requires $r_{ik} \in \mathbb{Z}_{\geq 0}$ units of resource $k \in \mathcal{R}$ during its execution time interval $[S_i, S_i + p_i[$. The resource allocation problem consists of determining start times S_i for each activity $i \in V$ such that the precedence and resource constraints are fulfilled and the corresponding makespan $\max_{i \in V}(S_i + p_i)$ is minimized. A formal description of the scheduling problem is given as follows:

$$\left.\begin{array}{lll}\text{Min.} & \max\limits_{i\in V}(S_i+p_i) & \\ \text{s.t.} & S_j \geq S_i + p_i & (j\in V, i \in Pred_j) \\ & \sum\limits_{i\in V: S_i \leq t < S_i+p_i} r_{ik} \leq R_k & (k \in \mathcal{R}, t \geq 0) \\ & S_i \geq 0 & (i \in V)\end{array}\right\} \quad (1)$$

3 Software packages

3.1 Technical data

Table 1 summarizes information on the hardware requirements and interfaces of the tested software packages. All of them can be used on a personal computer operating under Microsoft Windows.

Temporal constraints may be specified as start–to–start, start–to–completion, completion–to–start, or completion–to–completion precedences. CA SuperProject and Project Scheduler do not support start–to–completion precendences.

All packages provide tools for managing resource–related costs. With the exception of Acos Plus.1, which only consideres variable costs, all packages allow the user to define fixed costs per activity as well as overtime costs. In

Table 1. Technical data

Package	**Acos Plus.1**	**CA Super-Project**	**CS Project Professional**	**MS Project**	**Project Scheduler**
Release	8.2	5.0a 002	3.0	2000	8.0.1
CPU	486	486/25	386	P75	P120
OS	Win 9x Win NT4.0	Win 9x Win 2000 Win NT4.0 OS/2	Win 9x Win 2000 Win NT4.0 Win ME	Win 9x Win 2000 Win NT4.0	Win 9x Win 2000 Win NT4.0
RAM	16 MB	16 MB	4 MB	24 MB	8 MB
HD	20 MB	10 MB	11 MB	35 MB	40 MB
Max. act.	32.000	∞	n.a.	1 Mill.	99.999
Max. res.	32.000	∞	n.a.	1 Mill.	20.000
Data Exchange	ODBC, OLE DDE, SQL	ODBC, OLE, DDE, DAO	ODBC DDE	ODBC, OLE, DDE	ODBC, OLE, DDE, FTP
Main Import/ Export Formats	MPX, CSV, ASCII, Arriba, DBF, HTML	MPX, XLS, CSV, TXT, dBaseIII, WK1, Sylk	MPX, P3, CSV	MPX, XLS, CSV, MDB, TXT	MPX SAP R/3

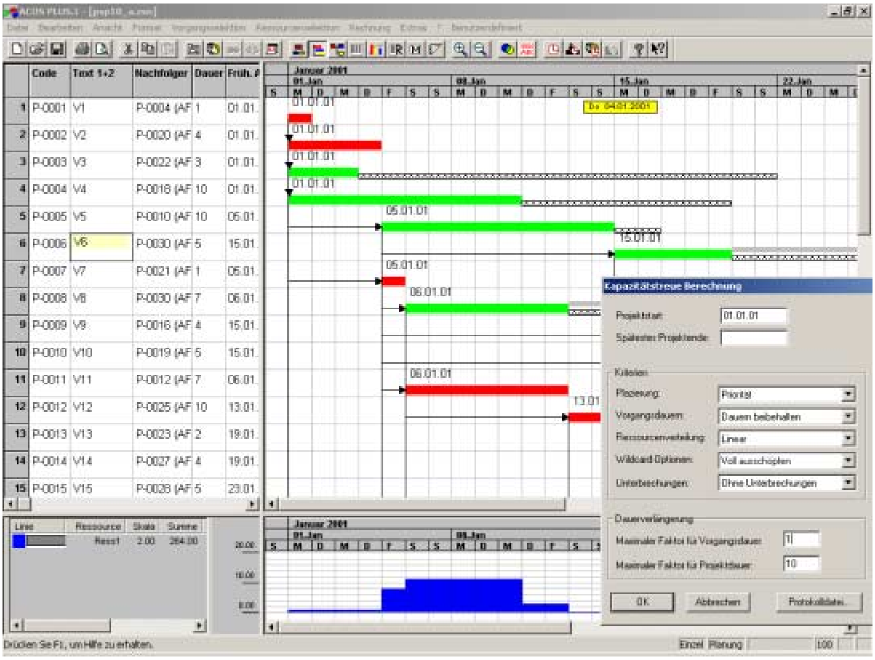

Fig. 1. Acos Plus.1

all packages different calendars may be defined for activities and resources, including working time and overtime per day, per week, or per month, as well as holidays. In our test, however, the focus is on resource allocation. Hence, we did not evaluate the impact of costs or calendars.

3.2 Acos Plus.1 8.2

The user interface of Acos Plus.1 (cf. Fig. 1) is the most clearly arranged one. A reason may be the limitation of the package to core functions and data management. In some cases, the user interface does not conform with Windows standards, e.g., the corresponding dialog box is not opened when double–clicking on a task bar in the Gantt–chart, and a context–sensitive menu is missing. Thus, program control is less comfortable than in the other packages. The handbook, online–help, and the examples are fairly compact.

Temporal scheduling has to be performed explicitly before resource allocation. Manual capacity planning is somewhat cumbersome as there is no easy way of displaying activities involved in resource overloads. They can only be identified using filter and sort functions.

Acos Plus.1 is the only package that offers the possibility to specify maximum time lags between activities. These maximum time lags are observed

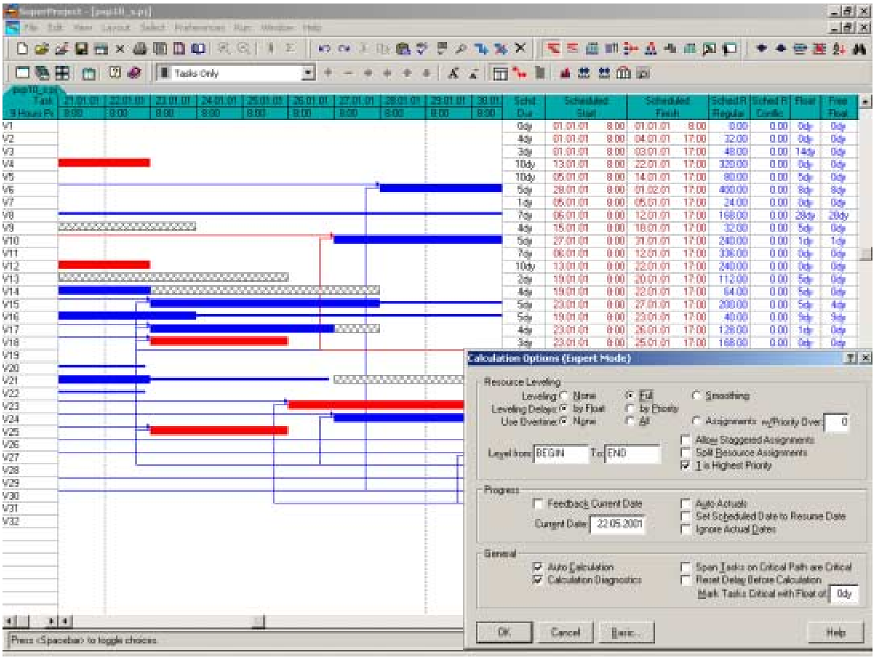

Fig. 2. CA SuperProject

during the temporal scheduling phase. Capacity planning, however, generally fails to find a feasible solution even for very simple project instances if maximum time lags have been prescribed.

Resource allocation can be performed for all or only for selected resources. Task priorities can be defined according to smallest total/free float time, affiliation to the critical path, longest/shortest processing time, number of predecessors/successors, total number of predecessors and successors, and user–defined. The result is displayed in a Gantt–chart, which also shows precedence relationships, free and total float times, and visualizes resource profiles on a common time–axis.

3.3 CA SuperProject 5.0a

The user interface of CA SuperProject offers extensive features. This results in a more demanding program control. Five alternative levels of information, menus, and dialogs are available. Data processing can be controlled via various filters and sort functions. Data can be entered into worksheets, Gantt–charts, or dialog boxes using mouse or keyboard. Detailed handbooks, a tutorial, and several assistants provide sufficient user support.

Resource allocation can be limited to a user–defined time interval. The priority of tasks has to be defined manually. For our test set we have chosen the same priority for all tasks. The resulting resource allocation is shown in a Gantt–chart (cf. Fig. 2) including precedence relationships as well as free and total float times.

3.4 CS Project Professional 3.0

The user interface of CS Project Professional is well–organized and completely meets Windows standards. In most view modes, the main screen is split into a task and a resource window (cf. Fig. 3) using a common time axis. This seems to be the best way to visualize resource overloads and assignments of tasks to resource units. A vertical toolbar for quick changes between different views would be useful.

Data can be entered directly into worksheets or via dialog boxes, both with on–line update of the project data and charts. Standard filter and sort functions are available. Project Professional offers useful features like grouping of tasks or exchanging two tasks including their precedence constraints. Support is available via on–line help and an extensive handbook on CD.

For resource allocation, task priorities can be chosen as total float, free float, processing time, earliest start time, latest completion time, and as user–defined priority from a four–level hierarchical key. According to the

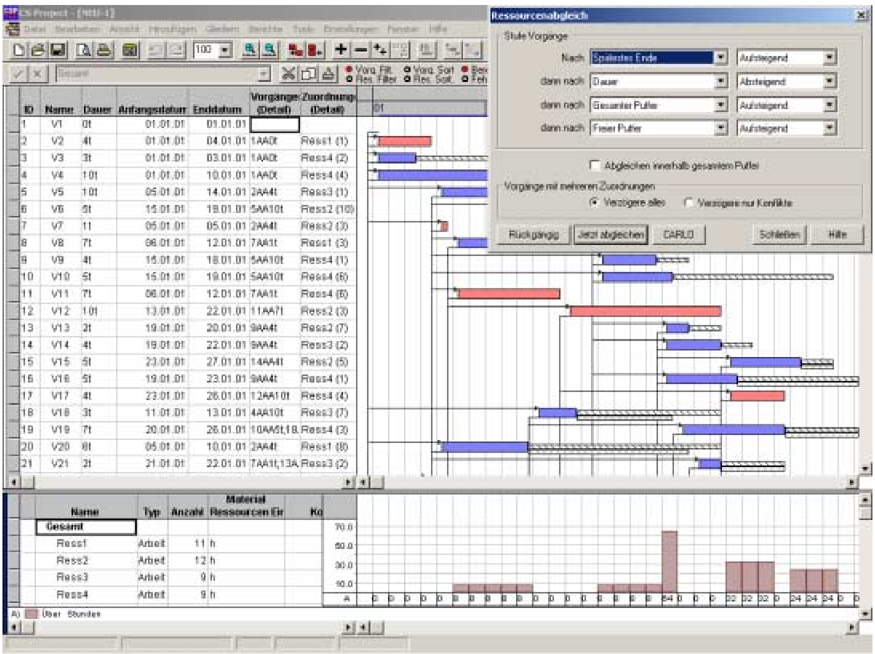

Fig. 3. CS Project Professional

handbook, the optional CARLO (cost and resource levelling optimization) algorithm automatically selects the best key. However, in several cases we obtained a shorter makespan using a manually selected key combination. The Gantt–chart of Project Professional also indicates precedence relationships as well as free and total float times of the activities.

3.5 Microsoft MS Project 2000

The user interface of Microsoft Project completely conforms with Windows and the MS–Office product family. Program control is very comfortable due to customizable toolbars, context sensitive menus, an undo function etc. Data can be specified in various ways via mouse and/or keyboard, e.g. by drag&drop in the Gantt–chart. All project data and charts are updated on–line. The GUI includes a vertical toolbar for quickly changing the project view. Individual views can be defined using filter and sort functions. An useful feature is the possibility to export user–defined worksheets in various file formats. The macro language Visual Basic for Applications VBA offers extensive possibilities for program automation. The support includes an user's manual, a tutorial, and a project assistant with general information on project management.

The resource allocation of Microsoft Project can be restricted to a certain time interval. The task priorities can be set either manually or predefined, where the latter is a combination of precedence relationships and float times. The Gantt–chart (cf. Fig. 4) is similar to the other packages. A resource utilization diagram can be displayed in a separate view.

3.6 Project Scheduler 8.0.1

Project Scheduler comes with a simple but functional user interface. Display filters can be defined for activities and resources. In addition to standard menu- and toolbars, a vertical view bar allows switching between different views very easily. In the lower part of the main window, additional information about activities can be displayed.

Data input is handled via keyboard and/or mouse. Changes in data are automatically updated in all project views. Standard help functions are available containing detailed information about external program control via VisualBasic and/or C++.

The resource allocation of Project Scheduler can be restricted to selected resources, activities, and/or a time interval. Priorities of tasks are predefined. The result of the resource allocation is visualized as Gantt–chart (cf. Fig. 5) that also contains precedence constraints and float times.

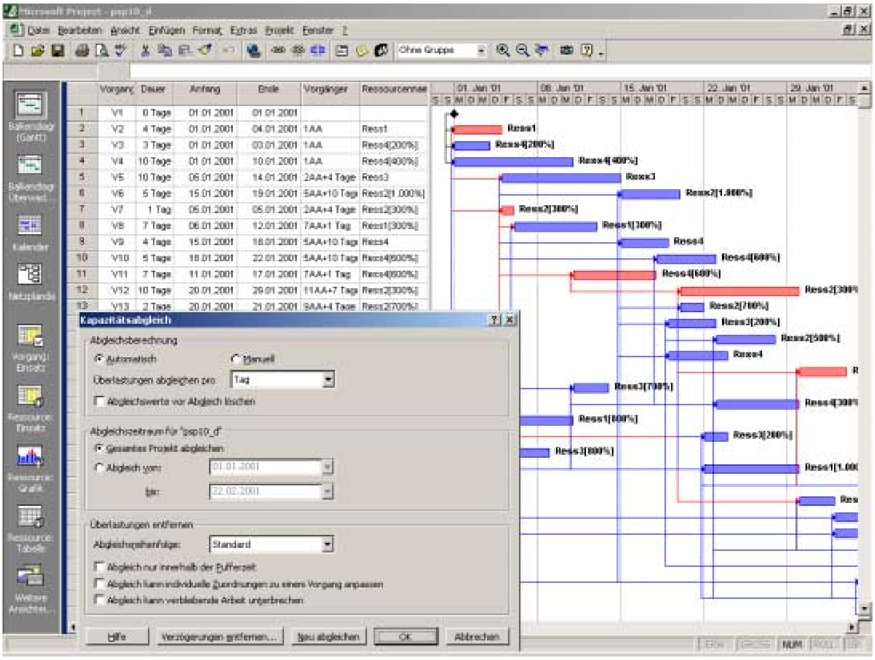

Fig. 4. Microsoft Project

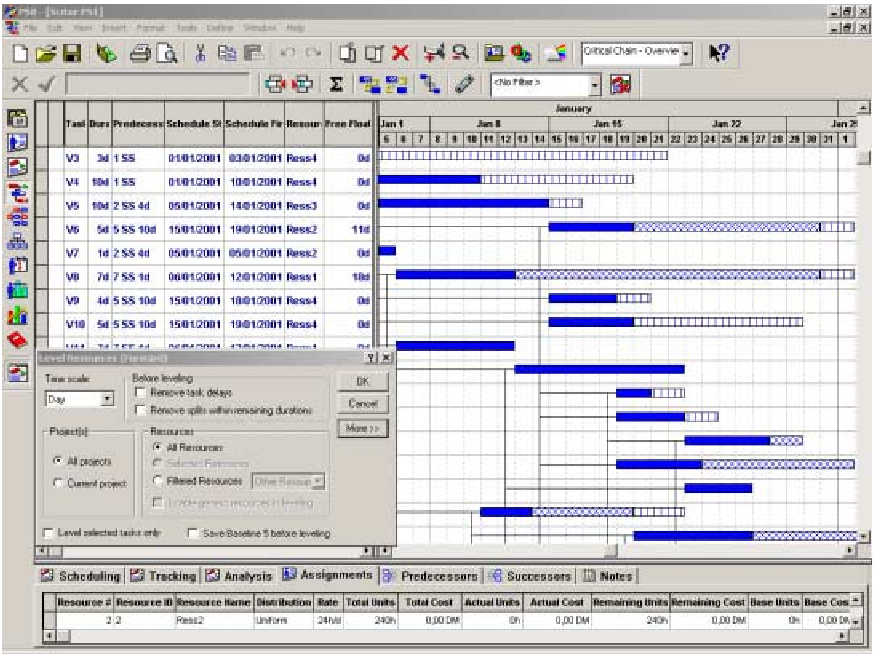

Fig. 5. Project Scheduler

4 Experimental design

The test set for our numerical investigation consists of 1560 instances of problem (1) characterized by the following parameters (cf. Kolisch et al., 1995):

- The *number of activities* n. The test set contains 480 instances with $n = 30$, 480 instances with $n = 60$, and 600 instances with $n = 120$, and four resources each.
- The *resource strength* RS measures the degree of resource scarcity. $RS = 0$ implies $r_{ik} = R_k$ for at least one activity $i \in V$ and at least one resource $k \in \mathcal{R}$. In case of $RS = 1$ no explicit resource allocation is necessary because temporal scheduling already provides a resource–feasible schedule. For the test set we have $RS \in \{0.2, 0.5, 0.7, 1.0\}$ for $n = 30$ and $n = 60$, and $RS \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ for $n = 120$.
- The *resource factor* RF defines the mean fraction of resources used by an activity. $RF = 1$ indicates that each activity requires at least one unit of every resource. $RF = 0$ indicates that no activity requires any resource. For the test set we have $RF \in \{0.25, 0.5, 0.75, 1.0\}$.
- The *network complexity* NC defines the mean number of predecessor activities of an activity. A large value of NC indicates that many precedence constraints are prescribed. For the test set we have $NC \in \{1.5, 1.8, 2.1\}$.

All projects used for our performance analysis have been generated by Kolisch et al. (1995) using the problem generator ProGen. For each combination of the above control parameters RS, RF, and NC, the test set contains ten instances. The test set is available at http://www.bwl.uni-kiel.de/Prod/psplib/datasm.html. At the same location, optimal solutions to the instances with $n = 30$ and the best solutions currently known to the instances with $n = 60$ and $n = 120$ are available. We have used these reference values as a benchmark in our numerical investigation.

5 Numerical results

None of the packages was able to achieve a better solution than known so far for any of the 1560 projects. Individual makespan results are presented in Hartung et al. (2001). All tests have been performed on a Pentium III PC with 128MB RAM and 500MHz clock pulse. None of the packages required more than 60 seconds CPU time for resource allocation of a single project.

Table 2 lists the mean and the maximal relative makespan deviation from the reference solution for the projects with 30, 60, and 120 activities, respectively. For both criteria, Acos Plus.1 clearly outperforms the other packages. The results obtained by Scitor's Project Scheduler are satisfactory as well.

Table 2. Mean and maximal deviation of makespan [%] for instances with 30, 60, and 120 activities

	Mean deviation [%]			Maximal deviation [%]		
	30	60	120	30	60	120
Acos Plus.1	3.87	4.05	9.69	24.62	26.39	28.95
SuperProject	5.39	6.37	13.99	36.23	32.14	41.98
CS Project	3.50	5.28	13.70	25.42	23.16	30.00
MS Project	5.18	6.23	14.02	31.03	29.89	46.79
Scitor PS	4.85	4.98	11.15	37.93	36.89	31.11

Table 3. Variance of makespan deviation [$\%^2$] and No. of best solutions found (excluding projects with $RS = 1$) for instances with 30, 60, and 120 activities

	Variance [$\%^2$]			No. of best solutions found		
	30	60	120	30	60	120
Acos Plus.1	30.16	39.83	46.34	143	174	115
SuperProject	53.60	67.13	75.26	109	88	30
CS Project	23.04	46.06	52.79	135	125	57
MS Project	44.66	60.70	83.20	95	76	23
Scitor PS 8	48.23	53.44	54.35	129	145	98

Crest Software's Project Professional performs well on small projects. However, we note that the maximal deviation of makespan from the best known solution is almost 50% in the worst case. For this instance, poor resource allocation results in a project implementation that wastes one third of time and causes higher, unnecessary cost.

The variance of the makespan deviation (cf. Table 3) confirms these results. Again, the schedules obtained by Acos Plus.1 are markedly better on the average than those of the other packages. In addition, Table 3 provides the number of instances where the tested packages were able to achieve the reference solution. Again, Acos Plus.1 and Scitor's Project Scheduler perform best.

We note that for all of these criteria, the performance clearly decreases with increasing number of activities, which indicates that for real–life projects with hundreds of activities, there is still a significant performance gap between the algorithms implemented and modern state–of–the–art heuristics from literature. Closing this gap at moderate run times constitutes a challenge for future research.

In the following, we evaluate the performance of the resource allocation modules for different complexity scenarios varying the mean number of resources used, the scarcity of resources, and the number of precedence relationships. The analysis is based on the 600 instances with 120 activities

Table 4. Mean deviation of makespan [%] for various resource strengths ($n = 120$)

RS	0.5	0.4	0.3	0.2	0.1
Acos Plus.1	2.17	6.06	9.58	13.45	17.21
SuperProject	6.07	10.28	13.67	17.21	22.74
CS Project	6.13	11.46	14.47	17.07	19.36
MS Project	4.60	9.12	12.97	18.28	25.12
Scitor PS	3.35	7.56	10.76	15.42	18.67

Table 5. Mean deviation of makespan [%] for various resource factors and various network complexities ($n = 120$)

	Resource factor RF				Network complexity NC		
	0.25	0.5	0.75	1	1.5	1.8	2.1
Acos Plus.1	5.69	11.34	11.39	10.35	8.92	9.50	10.67
SuperProject	6.27	16.98	18.70	14.02	13.81	13.84	14.33
CS Project	8.17	15.80	16.63	14.20	13.15	13.67	14.27
MS Project	8.37	16.53	16.80	14.37	13.47	13.90	14.68
Scitor PS	6.30	13.06	13.34	11.90	10.63	10.85	11.98

only. The results for the smaller instances are similar (cf. Hartung et al., 2001). Table 4 shows the mean makespan deviation as a function of the resource strength. With increasing resource scarcity, the deviation noticeably increases for all five packages. Table 5 shows an interesting result concerning the number of resources used. If all activities use one resource only (not necessarily the same), the mean deviation of makespan is much smaller than in the case of two or more resources used. Again, all five packages show the same behavior. As can be seen from Table 5, the number of precedence relationships does not affect the resource allocation quality of any of the tested packages.

6 Conclusions

Our analysis shows that the project management software packages Acos Plus.1, CA SuperProject 5.0a, CS Project Professional 3.0, Microsoft MS Project 2000, and Scitor Project Scheduler 8.0.1 offer valuable support for the management of resource–constrained projects. The results of our benchmark test, however, indicate that none of the tested resource allocation methods is currently competitive with the best state–of–the–art algorithms from literature. It turns out that the heuristic methods of Acos Plus.1 and Scitor Project Scheduler 8 outperform their competitors. The quality of the schedules obtained significantly decreases when realistic scenarios are considered, i.e., when dealing with projects comprising a large number of activities and scarce resources.

All tested software packages do not offer an exact algorithm for resource allocation, but use fast heuristic methods. The exact solution to a project scheduling problem requires extensive computational time, which does not meet the user's requirement for interactive use of software. On the other hand, the makespan deviation from solutions that can be achieved with modern resource–constrained project scheduling methods (cf. Brucker et al., 1999) would justify the implementation of additional algorithms.

References

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

Farid F, Manoharan S (1996) Comparative analysis of resource–allocation capabilities of project management software packages. Project Mgmt Journal 24: 35–44

Hartung T, Mellentien C, Trautmann N (2001) Software zur ressourcenbeschränkten Projektplanung im Vergleich. Report WIOR 604, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, Germany

Johnson R (1992) Resource constrained scheduling capabilities of commercial project management software. Project Mgmt Journal 22: 39–43

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

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

Kolisch R, Sprecher A, Drexl A (1995) Characterization and generation of a general class of resource–constrained project scheduling problems. Mgmt Science 41: 1693–1704