

Kolisch, Rainer; Hempel, Kai

Working Paper — Digitized Version

Finite scheduling capabilities of commercial project management systems

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 397

Provided in Cooperation with:

Christian-Albrechts-University of Kiel, Institute of Business Administration

Suggested Citation: Kolisch, Rainer; Hempel, Kai (1996) : Finite scheduling capabilities of commercial project management systems, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 397, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/149844>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

No. 397

***Finite Scheduling Capabilities of Commercial
Project Management Systems***

Rainer Kolisch and Kai Hempel



No. 397

***Finite Scheduling Capabilities of Commercial
Project Management Systems***

Rainer Kolisch and Kai Hempel

May 1996

Institut für Betriebswirtschaftslehre
Lehrstuhl für Produktion und Logistik
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40, 24098 Kiel, Germany

Abstract

This study assesses the quality of seven commercial project management systems to perform finite scheduling. We give a brief survey of the way commercial project management systems work and especially how they perform finite scheduling. Since the implemented methods are proprietary, we measure the quality of the software in terms of the deviation from a strong lower bound of the minimum makespan. As benchmark instances we use a set of problems generated under a factorial design. Statistical tests are performed in order to investigate the impact of the problem parameters and to detect significant differences between the quality of the software packages.

1. Introduction

Finite scheduling has matured to one of the mostly discussed topics in today's production management. The reason is twofold. First, companies have become aware that large capacities cause high fix costs which, if not efficiently utilised for manufacturing, lowers the profit. Second, in times of harsh global competition customers set a high value on short lead times and accurate delivery dates. Hence the ability to schedule the customer jobs efficiently subject to finite capacities plays an important role for all companies regardless of the size, the type of business, and the location. In manufacturing different planning tools are employed for make-to-stock and make-to-order production. Mass production / make to stock producers usually employ MRP II-systems at the aggregate production planning and lot sizing level, whereas job shop-based finite scheduling systems are utilised at the shop floor level. In contrast, small batch / make-to-order producers use project scheduling based systems for their specific needs. In order to support potential users of project management software, a research project has been performed in Germany with two main directions: A multi attribute value theory-based approach for selecting project management software and an appraisal of models and methods employed in project management systems. Results are documented in German in Kolisch/Hempel (1995a, b). In order to make interesting results available to an international audience, the paper at hand focuses at the finite scheduling capabilities of project management systems. The plan is as follows: In Section 2 we provide a brief outline of the way commercial project management systems work. Section 3 summarises recent research on the assessment of finite scheduling capabilities of commercial project management systems. The experimental design of our study is given in Section 4. Section 5 reports on the results of our research. The paper ends with a summary in Section 6.

2. Planning Approach of Commercial Project Management Systems

Today's commercial project management and scheduling systems are generally made up of the three components planning, realisation of the plan, and control/feedback of results versus

plan. The project planning component is the central component of the systems. It comprises three phases: structuring the project, time phasing the project, and scheduling the project subject to finite capacities [cf. Meredith/Mantel (1995)].

At an aggregate point of view, the structuring module defines, on the basis of the work breakdown structure (WBS), the entire project as a set of work elements. Each work element is associated with a duration, a resource demand, costs, and technological interrelations to each other. At a detailed point of view, every work element can be further divided into jobs, activities, or tasks. For the remainder of this paper we will refer to jobs. Again, each job can be defined by duration, resource requirements, costs, and by technological interrelations to other jobs. Whereas the aggregate structure of the project can be visualised by the WBS, the detailed structure is visualised by a network. Two different types are possible: Activity-on-node networks depict each job as a node and precedence relations are represented by arcs leading from the preceding to the succeeding node. Contrary, in activity-on-arc networks activities are represented by arcs and events are depicted by nodes.

Based on the network, the duration of the activities, the release date, and the due date, time phasing determines the earliest and latest start and finish times of the jobs by forward and backward recursion, respectively [cf. Elmaghraby (1977)]. In a production environment the process of time phasing is oftenly termed as (infinite) front and backward loading. With the calculated start times, a resource skyline can be generated which visualises the amount of resource capacity required over time. If the resource demand does not exceed the resource capacity in any single period of the planning horizon, the planning phase can be finished at this stage with a precedence and resource feasible schedule. Unfortunately, in the presence of scarce capacities this will usually not be the case. Hence the third planning phase, finite scheduling, has to be undertaken.

To provide a precise formulation of the problem which has to be tackled by finite scheduling we want to employ some formalism. Let denote the set of jobs with $j=1,...,J$. Each job j has a set of immediate preceding jobs P_j which may be empty. We assume that each predecessor job of j has to be finished before j can be started. Processing job j takes $d_j \geq 0$ periods of time; during each period j is in process $k_{jr} \geq 0$ capacity units of resource r are required. Once started, processing j cannot be preempted. Finally, we have $r=1,...,R$ different resources, e.g. manpower, machines etc., each with a finite period capacity of $K_r > 0$ capacity units. Defining the decision variable $t_j > 0$ as the finish time of processing job j and A_t to be the set of jobs which are processed in period t , a schedule, i.e. a vector t of finish times, which is feasible w.r.t. precedence and resource constraints can be depicted by the conceptual model given in Table 1 [cf. Alvarez/Tamarit (1989)].

(1) forces a job not to start before all its predecessor jobs are finished. (2) models constraints induced by finite capacities, and (3) defines the continuous variables. An important objective for (1) - (3) within a manufacturing context is the minimisation of the makespan, i.e. min

$\max \{t_j | j=1, \dots, J\}$. Several reasons are advocating this objective function. Minimizing the makespan reduces work in process and hence tied-up capital. Furthermore, it lowers the possibility of violating deadlines. Unfortunately, minimising the makespan renders the polynomially solvable finite scheduling problem (1) - (3) to be *NP*-hard where optimal solutions for instances of realistic size cannot be obtained [cf. Blazewicz et al. (1983)].

Table 1: Conceptual finite scheduling model.

| | | |
|------------------------------------|--------------------------------|-----|
| $t_i + d_i \leq t_j$ | $j=1, \dots, J, i \in P_j$ | (1) |
| $\sum_{j \in A_t} k_{jr} \leq K_r$ | $r=1, \dots, R, t=1, \dots, T$ | (2) |
| $t_j \geq 0$ | $j=1, \dots, J$ | (3) |

For solving the finite scheduling problem with the objective of minimising the makespan different heuristic and optimal procedures have been suggested in the literature. A detailed survey can be found in, e.g., Özdamar/Ulusoy (1995). To the best of our knowledge, all commercial project management systems are performing finite scheduling with a priority rule-based solution methodology. It is made of the following three components: loading direction, scheduling scheme, and priority rule [cf. Li/Willis (1992) and Johnson (1992)].

Three different ways of loading can be distinguished: Front loading begins with the release date and tries to schedule activities as early in time as possible. Backward loading begins with the due date and tries to schedule activities as late in time as possible. Finally, systems like OPT perform front loading behind the bottleneck and backward loading before the bottleneck resource [cf. Vollmann et al. (1992)].

The type of the scheduling scheme determines how the set of schedulable activities, that is all activities which can be selected for scheduling in the next planning step, is formed. Two types are available: Serial and parallel scheduling. Within serial scheduling the set of schedulable activities consists of all jobs which are feasible w.r.t. precedence constraints, i.e. constraint (1). One of the jobs from the set is selected with a priority rule and scheduled at the earliest precedence and resource feasible start time. Contrary, in parallel scheduling the set of schedulable activities consists of all activities which are precedence and resource feasible at a certain time period t . Again, one job is selected with a priority rule and then scheduled to start at t .

Finally, the priority rule determines which job from the set of schedulable jobs is selected. Well-known priority rules are, e.g., minimum slack, shortest processing time, and minimum latest finish time. Surveys in the literature document more than a 100 different priority rules [cf., e.g., Panwalkar (1977)].

Taking into account the three elements of a priority rule-based solution methodology, we have an enormous number of different methods which can be employed in commercial project management and scheduling systems. Although there are sound experimental results from scheduling theory about the performance of different strategies such as priority rules and scheduling schemes [cf.. e.g., Alvarez/Tamarit (1989), Kolisch (1996)] the actual realisation implemented in the commercial project management systems is proprietary. Hence, we cannot assess the finite scheduling capabilities of these systems on the basis of the methods employed. This is the reason for our experimental investigation which follows in Section 4. In the next section we provide a short review of the relevant literature.

3. Literature Review

There is only a limited body of literature on the assessment of commercial project management systems. We can roughly divide the research into general evaluations of project management programs and the evaluation of its finite scheduling capabilities. Work in the first area has been performed by Wasil/Assad (1983), Assad/Wasil (1986), De Wit/Herroelen (1990), and Maroto/Tormos (1994). Research in the second field has only emerged in recent years. This is due to the fact that commercial project management systems did not offer the capability of finite scheduling until the beginning of the nineties. Additionally, researchers were lacking powerful exact methods in order to derive optimal benchmark solutions for practically-sized problem instances.

Johnson (1992) was the first researcher who compared thirteen different versions of seven scheduling systems available in the USA. The packages included SuperProject Expert 1.0 and SuperProject 2.0, TimeLine 2.0 and 4.0, Primavera 4.0, 4.1 and 5.0, Microsoft Project for Windows 1.0 and 3.0, Harvard Total Project Manager II and Harvard Project Manager 3.0, Pertmaster Advanced and Hornet. As instance set Johnson employed the 110 instances of Patterson (1984). These instances were at that time the most widely used benchmark set. But as shown in Kolisch et al. (1995), it has not been generated subject to well defined problem parameters. In short the Patterson problem set can be characterised by 7 to 51 jobs and 1 to 3 finite capacities. Johnson solved each of the 110 instances with each of the altogether 13 scheduling systems and compared it to the optimal makespan documented in Demeulemeester/Herroelen (1992). He then computed the average percent increase above the optimal makespan and the number of optimal solutions for each system. On average, the tested programs achieved makespans which were 9.88% inferior than the optimal makespans. The range of the average deviation above the optimum was [5.03%, 25.6%]. The two programs with the best results turned out to be TimeLine 2.0 and Hornet with an average increase of 5.03% and 5.53%, respectively, above the minimum makespan. The worst results were derived by the systems Microsoft Project 1.0 and PertMaster Advanced with an average increase of 25.6% and 14.1%, respectively.

Maroto/Tormos (1994) compared CA SuperProject 2.0A, Instaplan 3.0B, Micro Planner for Windows 6.24A, Micro Planner Professional 7.3B, Microsoft Project for Windows 1.0 and 3.0 and Project Scheduler 1.0 on the basis of one instance with 51 activities and 3 resources. The increase above the precedence based lower bound, i.e., the optimal makespan for the resource-unconstrained schedule, ranged between 19.79 % and 37.43 %. The shortest makespan was obtained by CA SuperProject and by Microsoft Project 3.0 where the latter program was applied with the default setting. The worst schedule was generated by Microsoft Project 1.0 with the so-called levelling option "Date, Slack, and Priority".

In a subsequent investigation, Maroto et al. (1994) assessed Microsoft Project 3.0 and CA SuperProject in the three versions V.2.0 for DOS as well as V.2.0 and V.3.0 for Windows on the basis of a set of 48 benchmark problems. The instances were generated subject to a full factorial experimental design with the two problem parameters R , the number of constrained resources, and O , the resource overload. The latter parameter measures the average ration of the amount of resources demanded and resource capacity provided during the shortest possible length of the project. Maroto et al. set $R \in \{1,2,3\}$ and $O \in \{[0.5,0.99], [1,1.5]\}$ and realised for each combination of R and O eight instances. The number of jobs was arbitrary set between 30 and 100. As in the study of Maroto/Tormos (1994), Maroto et al. compared the makespan of the resource feasible solutions with the minimum makespan of the unconstrained problem. The impact of the project management software, the number of resources, and the resource overload on the quality of the solutions was investigated by an analysis of variance (ANOVA). With the exception of the number of resources, which was significant at the 5 % level of confidence only, all other parameters and all interactions were significant at the 1 % level of confidence.

Finally, current research has been reported by Burley (1995) and Maroto et al. (1996). Burley compared Microsoft Project 3.0, Project Manager Workbench/w and Timeline 6.0; Maroto et al. (1996) investigated on the performance of CA SuperProject 4.0 and Microsoft Project 4.0.

The paper at hand extends the work documented so far in two directions. First, we evaluate seven widespread programs which is, with the exception of Johnson (1992), more than in any other study. Second, we apply a larger and in terms of problem parameters better defined test set than in other studies. Details are given in the next Section.

4. Experimental Design

We have selected the seven most widespread PC-based project management programs in Germany. Table 2 gives an overview.

Table 2: Project Management Software

| Program |
|-------------------------------------|
| Artemis Schedule Publisher V.4.1 |
| CA SuperProject V.3.0 C |
| Microsoft Project V.4.0 |
| Primavera Project Planner V.1.0 |
| Project Manager Workbench V.1.1.02w |
| Project Scheduler 6.0 V.1.02 |
| Time Line V.6.0.0 |

In order to generate test instances we employed the following problem parameters [cf. Kolisch et al. (1995)]: J , the number of jobs, R , the number of constrained resources, NC , the network complexity, RF , the resource factor, and RS , the resource strength. While the first two problem parameters are clearly defined, we now introduce the latter three parameters. For details cf. Kolisch et al. (1995).

$$NC = \frac{1}{J} \sum_{j=1}^J |P_j| \quad (4)$$

NC measures the average number of technological precedence relations per job. A large (small) NC stands for a highly (lowly) technologically constrained network.

$$RF = \frac{1}{(J-2) \cdot R} \sum_{j=1}^J \sum_{r=1}^R \begin{cases} 1, & \text{if } k_{jr} > 0 \\ 0, & \text{else} \end{cases} \quad (5)$$

RF is normalised to the interval $[0,1]$ and it defines the average number of constrained resources which is requested by a job. For $RF=1$ each job requests every of the R resources, while for $RF=0$ none of the jobs requests any of the resources and hence we have the resource-unconstrained problem.

$$RS_r = \frac{K_r - K_r^{min}}{K_r^{max} - K_r^{min}} \quad r=1, \dots, R \quad (6)$$

RS measures the proportion of resource demand and availability. K_r^{min} is the minimum availability of resource r in order to process each job. K_r^{max} is the capacity of resource r which is required when the resource unconstrained CPM-time has to be achieved. Again, RS is normalised to the interval $[0,1]$. For $RS=0$ we obtain $K_r=K_r^{min}$ and the project is highly resource-constrained, for $RS=1$ we have $K_r=K_r^{max}$, that is, a resource-unconstrained scheduling problem. Table 3 reports the levels of the systematically varied problem parameters of our

experiment. All other parameters, i.e. d_j , k_{jr} , $|P_j|$, and $|S_j|$, the number of immediate successors of job j , were drawn randomly from the uniformly distributed intervals given in Table 4.

Table 3: Systematically generated problem parameters.

| Parameter | J | R | RS | RF | NC |
|-----------|-----|-----|------|------|------|
| Levels | 10 | 2 | 0.2 | 0.5 | 1.5 |
| | 20 | 3 | 0.5 | 1 | 2 |
| | 30 | 4 | 0.7 | | |

Table 4: Randomly generated parameters.

| Parameter | d_j | k_{jr} | $ S_j $ | $ P_j $ |
|---------------|-------|----------|---------|---------|
| Minimum level | 1 | 1 | 1 | 1 |
| Maximum level | 10 | 10 | 3 | 3 |

Realising a full factorial test design with 10 replications for each combination of the systematically varied parameter levels we would have $10 \cdot 3^3 \cdot 2^2 = 1.080$ instances where each instance would then have to be solved with each of the 8 project management programs. In order to reduce this prohibitive effort we chose a fractional design on the basis of orthogonal matrices [cf. Addelman (1962)]. We used the implementation given in the program Conjoint Designer V.2.0. This way we reduced the number of parameter level combinations to 16 and thus the overall number of instances to 160. In order to obtain a benchmark solution for each instance, we applied the optimal branch-and-bound based algorithm of Demeulemeester/Herroelen (1992). Giving a time limit of 3600 CPU-seconds for each problem on a personal computer with 40 MHz and mathematical coprocessor the method derived optimal solutions for 153 problems. For the remaining 7 instances it obtained an upper bound of the optimal makespan. We then employed each of the 7 project management programs to solve all test instances as follows. First, the ASCII-file of the instance under consideration was read. Afterwards we applied a so-called full levelling procedure with the default-settings of the program. It has to be noted that the term levelling, although used in all programs, is from the theoretical point of view confounding because levelling refers to the objective of obtaining a smooth resource demand profile and not a minimum makespan. Finally, the obtained schedule was checked for feasibility and the project length was adjusted by possible off-days in the calendar. Indexing the test instances with $j=1,...,160$ and the project management software with $i=1,...,7$, we denote with Z_{ij} the makespan for test instance j when solved with project management software i . Z_j^* denotes the benchmark objective function value of instance j . The response variable in our experiment is then Π_{ij} , the percentage deviation of the upper bound when instance j is solved by project management software i :

$$\Pi_{ij} = \frac{Z_{ij} - Z_j^*}{Z_j^*} \cdot 100 \quad (7)$$

5. Results

Figure 1 gives the frequency distribution of the $7 \cdot 160 = 1.120$ response values. The median is 3.57 % and the mean is 5.79 %, respectively, while the standard deviation calculates to 7.51 % and the range comes up to [0%; 51.85%]. On the average 45 % of the instances were solved to optimality by the project management systems. The frequency distribution resembles to the one obtained by Davis/Patterson (1975) when performed front loading-based parallel scheduling with the minimum slack priority rule. This might be an indication of the way scheduling is performed by the majority of the systems. The mean and the standard deviation for each of the seven scheduling systems is provided in Table 5. The average percentage increase above the reference makespan $\Pi=5.79$ % is drastically better than the value of 9.88 % as documented in Johnson (1992). Generally this effect can be caused either by the fact that commercial management systems have drastically improved or that the test instances employed in this study are easier to solve than the Patterson-set which has been used by Johnson. From detailed computational tests with heuristic [cf. Kolisch (1996)] and optimal solution procedures [cf. Kolisch et al. (1995)] performed on the basis of similar test instances to the ones employed in this study, we know that our instances should be even harder to solve than the Patterson-set. Eventually it could be argued that we only got optimal solutions for 153 of the 160 problems which results in a smaller average percentage deviation from the benchmark solutions. But since only 4.3 % of the instances have not been solved to optimality and the bounds obtained are quite close to the optimum we can neglect this possibility. Hence, we can conclude, that the finite scheduling performance of commercial project management systems has drastically improved in recent years.

Figure 1: Frequency distribution of Π_{ij} .

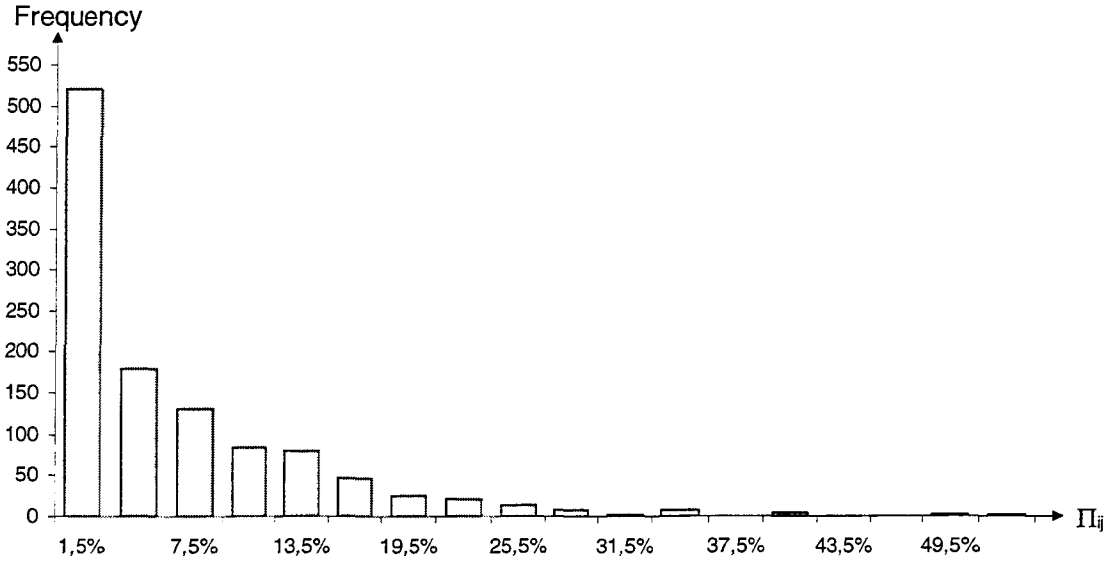


Table 5: Mean and standard deviation of Π_i

| Project Management System | $\mu(\Pi_i)$ | $\sigma(\Pi_i)$ |
|----------------------------|--------------|-----------------|
| Artemis Schedule Publisher | 9.76 % | 9.82 % |
| CA SuperProject | 4.41 % | 6.02 % |
| Microsoft Project | 5.35 % | 6.53 % |
| Primavera Project Planner | 4.39 % | 6.04 % |
| Project Manager Workbench | 6.69 % | 8.60 % |
| Project Scheduler 6 | 5.43 % | 7.98 % |
| Time Line | 4.49 % | 5.09 % |

5.1 Impact of the Problem Parameters

Tables A1 - A5 in the Appendix show the influence of the systematically varied problem parameters on the average increase of the makespan. The following observations can be made:

- The scheduling performance deteriorates with an increasing number of jobs. An exception holds for the system Artemis Schedule Publisher where the results are slightly better for the 30-job than for the 20-jobs problems. Nevertheless, this result has to be interpreted with care, because the performance of the Artemis Schedule Publisher software is far beyond the performance of the other systems.
- All seven project management systems show inferior results for $R=3$ constrained resources than for $R=1$ constrained resource. Three out of seven systems, i.e. Microsoft Project, Project Manager Workbench, and Time Line, show consistently deteriorating results for an increasing number of constrained resources. Three of the remaining four systems, i.e. CA SuperProject, Primavera Project Planner, and Project Scheduler 6, show a small improve-

ment when altering R from 2 to 3. Again, only Artemis Schedule Publisher performs much better when altering the number of constrained resources from 2 to 3.

- With an increasing scarcity of the resource capacity, i.e. with decreasing RS -levels, the planning results of all project management systems do deteriorate consistently.
- Enlarging the number of requested resources, i.e. enlarging the RF -level, causes a declining quality of the solutions.
- Finally, we observe better results for highly technologically interrelated projects as given by large NC -levels.

As a preliminary result we can state, that each of the systematically varied problem parameters has an impact on the performance of the project management systems. In order to test if this impact is statistically significant, we have performed a single factor analysis of variance (ANOVA) for each problem parameter and for each project management system. Testing the assumptions of ANOVA, we found that neither the assumption of a normally distributed response variable Π_{ij} nor the assumption of variance homogeneity was met [cf. Alvarez-Valdes / Tamarit (1989)]. Hence, we additionally applied more robust non-parametric tests. More precisely, we used the U-test of Mann and Whitney (MW) in order to test the influence of problem parameters with two levels and the H-test of Kruskal and Wallis (KW) to test the impact of problem parameters with three different levels. All calculations were performed with the program SPSS. Table 6 reports the levels of significance for the parametric and non-parametric test for all single effects of the independent parameters and all project management systems. Shaded cells indicate a significant influence of the corresponding problem parameter on the program at the 5 % level of confidence. With one exception regarding the influence of R on CA SuperProject all parametric and non-parametrical tests show (non-)significant results for the same cells. From Table 6 it can be concluded that the number of requested scarce resources measured by the parameter RF significantly effects the performance of all tested project management systems. With one exception for the system Time Line the same holds for the scarcity of the resources measured by the parameter RS . The level of the parameter R , the number of scarce resources, significantly effects only the results of the program packages Artemis Schedule Publisher, Project Scheduler 6, and Time Line; J , the number of jobs, has a significant impact on the planning results of CA SuperProject, Microsoft Project, Project Scheduler 6, and Time Line. Finally we can state that the problem parameter NC , the network complexity, is the only factor which does not have a significant influence on any of the systems. The observed influence of the problem parameters on the performance of the commercial systems has been analogously reported for an assessment of widely known priority rules [cf. Kolisch (1996)].

Table 6: Levels of significance for all parameter level-program combinations.

| Problem parameter | <i>R</i> | | <i>V</i> | | <i>NC</i> | | <i>RF</i> | | <i>RS</i> | |
|----------------------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|
| Test | <i>KW</i> | <i>ANOVA</i> | <i>KW</i> | <i>ANOVA</i> | <i>MW</i> | <i>ANOVA</i> | <i>MW</i> | <i>ANOVA</i> | <i>KW</i> | <i>ANOVA</i> |
| Artemis Schedule Publisher | 2.69 | 0.92 | 20.47 | 30.33 | 74.19 | 88.34 | 0.50 | 2.30 | 0.04 | 0.06 |
| CA SuperProject | 4.07 | 15.98 | 0.07 | 0.37 | 10.24 | 23.25 | 0.00 | 0.01 | 0.00 | 0.00 |
| Microsoft Project | 7.56 | 2.49 | 0.07 | 0.25 | 32.69 | 36.01 | 0.00 | 0.00 | 0.02 | 0.00 |
| Primavera Project Planner | 19.65 | 23.36 | 37.53 | 67.83 | 26.01 | 39.15 | 0.78 | 4.56 | 0.00 | 0.00 |
| Project Manager Workbench | 9.81 | 6.18 | 21.71 | 39.26 | 91.34 | 63.22 | 0.90 | 0.63 | 0.00 | 0.00 |
| Project Scheduler 6 | 3.46 | 2.78 | 0.67 | 0.77 | 54.99 | 83.29 | 0.00 | 0.00 | 0.00 | 0.00 |
| Time Line | 3.78 | 3.58 | 0.16 | 0.45 | 98.86 | 86.50 | 0.96 | 1.76 | 39.78 | 11.43 |

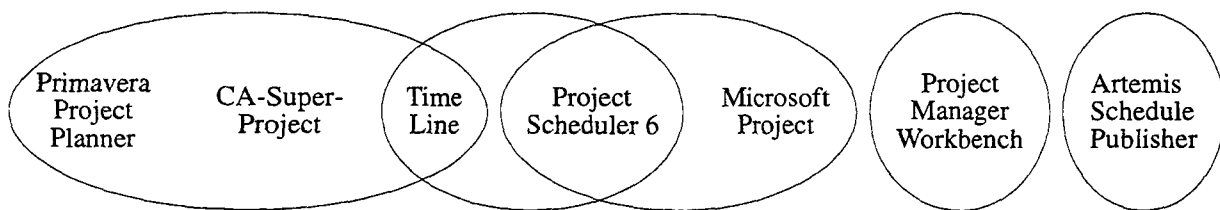
5.2 Comparison of the Project Management Systems

We have seen in the last subsection that the tested project management systems show a divergent performance when solving different types of problems as characterised by the systematically varied problem parameters of our test design. Hence the best choice of the scheduling systems would be parameter dependent. Unfortunately the cost of using different systems, i.e. the price of the program, the cost for instructions etc., is usually prohibitively high. We thus are interested if there is a program which significantly outperforms the other project management systems on all test instances. As already indicated by Table 6, the average performance is quite widespread with the best program being within reach of 4.39 % of the benchmark makespan and the worst program with a deviation of 9.76 %. In order to test if there is a significant distinction between the programs w.r.t. the solution quality we performed the non-parametric Friedmann-test. The test detected with a confidence level of 0,00 % a significant difference between at least two of the seven programs of our study. The next step was a pairwise comparison of the programs with the non-parametric Wilcoxon-test. Table 7 shows the confidence level obtained for each of the program pairs. Again, a grey shaded cell stands for a significant difference between two programs at the 5 % level of confidence. We can see that only for five pairs of the project management systems there is no significant difference in performance. Figure 2 summarises the results by ranking the programs on the basis of the average deviation from the benchmark makespan from left to right; systems which do not show a statistical significant difference from each other are placed in the same ellipse. From Figure 2 we can roughly divide three classes of project management systems: In the first class, with the best performance, we have Primavera Project Planner, CA SuperProject, and Time Line; the second class is constituted by Time Line, Project Scheduler 6, and Microsoft Project. The two programs with rather poor results are Project Manager Workbench and Artemis Schedule Publisher.

Table 7: Levels of significance for all program-pairs.

| | CA SuperProject | Microsoft Project | Primavera Project Planner | Project Manager Workbench | Project Scheduler6 | Time Line |
|----------------------------|--------------------|----------------------|---------------------------------|---------------------------------|-----------------------|-----------|
| Artemis Schedule Publisher | 0.00 % | 0.00 % | 0.00 % | 0.00 % | 0.00 % | 0.00 % |
| CA SuperProject | | 2.76 % | 98.25 % | 0.02 % | 3.13 % | 62.02 % |
| Microsoft Project | | | 1.66 % | 4.88 % | 93.35 % | 3.65 % |
| Primavera Project Planner | | | | 0.01 % | 4.35 % | 66.98 % |
| Project Manager Workbench | | | | | 3.53 % | 0.22 % |
| Project Scheduler 6 | | | | | | 8.58 % |

Figure 2: Performance classes of project management software.



6. Summary

We undertook a thorough experimental study to assess the performance of finite scheduling capability of commercial project management software subject to different problem parameters. The results revealed a significant different performance of the systems and a significant impact of the scarcity of resources. This confirmed the findings of prior studies. Furthermore, we showed that there is a significant influence of the number of jobs, of the number of scarce resources, and of the average number of resources requested by one job. Compared with the results obtained by Johnson (1992) we revealed that the average performance of project management systems improved by almost 100 %. Nevertheless, measured with the results derived by modern heuristics, e.g. Leon/Balakrishnan (1995) and Kolisch/Drexl (1996), there is still enough margin to improve the quality of commercial project management systems.

Literature

- Addelman, S. (1962): Orthogonal main-effect plans for asymmetrical factorial experiments, *Technometrics*, Vol. 4, S. 21-58.
- Alvarez-Valdes, R. and J.M. Tamarit (1989): Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis, in: Slowinski, R. und J. Weglarz (Hrsg.): *Advances in project scheduling*, Elsevier, Amsterdam, S. 113-134.

- Assad, A.A. and E.A. Wasil (1986): Project management using a microcomputer, *Computers & Operations Research*, Vol. 13, pp. 231-260.
- Blazewicz, J., J.K. Lenstra, and A.H.G. Rinnooy Kan (1983): Scheduling subject to resource constraints: Classification and complexity, *Discrete Applied Mathematics*, Vol. 5, S. 11-24.
- Burley, E. (1995): Resource levelling with project management software, *INFORMS International Meeting*, Singapore.
- Davis, E.W. and J.H. Patterson (1975): A comparison of heuristic and optimum solutions in resource-constrained project scheduling, *Management Science*, Vol. 21, S. 944-955.
- Demeulemeester, E. und W. Herroelen (1992): A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, *Management Science*, Vol. 38, S. 1803-1818.
- Elmaghraby, S.E. (1977): *Activity networks: Project planning and control by network models*, Wiley, New York.
- Johnson, R.V. (1992): Resource constrained scheduling capabilities of commercial project management software, *Project Management Journal*, Vol. 22, Nr. 4, S. 39-43.
- Kolisch, R. (1996): Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation, *European Journal of Operational Research*, to appear.
- Kolisch, R., A. Sprecher und A. Drexl (1995): Characterization and generation of a general class of resource-constrained project scheduling problems, *Management Science*, Vol. 41, pp. 1693-1703.
- Kolisch, R. and K. Hempel (1995a): Entscheidungstheoretisch fundierte Bewertung von Standardsoftware für das Projektmanagement, Working Paper 368, University Kiel.
- Kolisch, R. and K. Hempel (1995b): Experimentelle Evaluation der methodischen Fundierung von Projektmanagementsoftware, Working Paper 372, University Kiel.
- Kolisch, R. and A. Drexl (1996): Adaptive search for solving hard project scheduling problems, *Naval Research Logistics*, Vol. 43, S. 23-40.
- Leon, V.J. und R. Balakrishnan (1995): Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling, *OR Spektrum*, Band 17, pp. 173-182.
- Li, K.Y. and R.J. Willis (1992): An iterative scheduling technique for resource-constrained project scheduling, *European Journal of Operational Research*, Vol. 56, pp. 370-379.
- Maroto, C., S. Balasch, P. Tormos, C. Capilla und F. Crespo (1994): Project management software: Capabilities and efficiency in resource constrained scheduling, Working Paper, 4-th International Workshop on Project Management and Scheduling, Leuven.
- Maroto, C. and P. Tormos (1994): Project management: An evaluation of software quality, *International Transactions in Operational Research*, Vol. 1, pp. 209-221.
- Maroto, C., P. Tormos, A. Lova, and F. Crespo (1996): Professional software for resource constrained scheduling: Heuristic and exact methods, in: *Institute of Computing Science Poznan University of Technology (Ed.): 5-th International Workshop on Project Management and Scheduling*, Poznan, pp. 163-166.
- Meredith, J.R. and S.J. Mantel (1995): *Project Management - A managerial approach*, 3-rd edition, Wiley, New York.
- Özdamar, L. and G. Uluzoy (1995): A survey on the resource-constrained project scheduling problem, *IIE Transactions*, Vol. 27, pp. 574-586.
- Panwalkar, S.S. and W. Iskander (1977): A survey of scheduling rules, *Operations Research*, Vol. 25, pp. 45-61.
- Patterson, J.H. (1984): A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem, *Management Science*, Vol. 30, S. 854-867.
- Vollmann, T.E, W.L. Berry, and D.C. Whybark (1992): *Manufacturing planning and control systems*, 3-rd edition, Irwin, Homewood.
- Wasil, E.A. and A.A. Assad (1983): Project management on the PC: Software, applications, and trends, *Interfaces*, Vol. 18, pp. 75-84.
- de Wit, J. and W. Herroelen (1990): An evaluation of microcomputer-based software packages for project management, *European Journal of Operational Research*, Vol. 49, pp. 102-139.

Appendix: Estimated influence of the systematically varied parameters on the percent increase of the benchmark makespan.

Table A1: Estimated percent increase of benchmark makespan as a function of J .

| J | Artemis Schedule Publisher | CA Super- Project | Microsoft Project | Primavera Project Planner | Project Manager Workbench | Project Scheduler6 | Time Line |
|-----|----------------------------------|----------------------|----------------------|---------------------------------|---------------------------------|-----------------------|-----------|
| 10 | 7.79 | 1.83 | 2.36 | 3.67 | 5.21 | 2.17 | 2.60 |
| 20 | 10.73 | 4.85 | 6.08 | 4.59 | 6.88 | 6.10 | 4.52 |
| 30 | 9.79 | 6.11 | 6.91 | 4.73 | 7.79 | 7.34 | 6.30 |

Table A2: Estimated percent increase of the benchmark makespan as a function of R .

| R | Artemis Schedule Publisher | CA Super- Project | Microsoft Project | Primavera Project Planner | Project Manager Workbench | Project Scheduler6 | Time Line |
|-----|----------------------------------|----------------------|----------------------|---------------------------------|---------------------------------|-----------------------|-----------|
| 1 | 6.84 | 2.93 | 3.02 | 3.01 | 4.45 | 2.52 | 2.75 |
| 2 | 12.08 | 5.17 | 5.82 | 4.99 | 6.67 | 6.48 | 4.86 |
| 3 | 8.04 | 4.37 | 6.74 | 4.59 | 8.97 | 6.23 | 5.47 |

Table A3: Estimated percent increase of the benchmark makespan as a function of RS .

| RS | Artemis Schedule Publisher | CA Super- Project | Microsoft Project | Primavera Project Planner | Project Manager Workbench | Project Scheduler6 | Time Line |
|------|----------------------------------|----------------------|----------------------|---------------------------------|---------------------------------|-----------------------|-----------|
| 0,2 | 14.71 | 8.25 | 9.76 | 8.31 | 12.40 | 11.67 | 5.86 |
| 0,5 | 8.63 | 3.83 | 4.43 | 3.76 | 5.54 | 4.15 | 4.25 |
| 0,7 | 7.07 | 1.73 | 2.79 | 1.75 | 3.27 | 1.75 | 3.59 |

Table A4: Estimated percent increase of the benchmark makespan as a function of RF .

| RF | Artemis Schedule Publisher | CA Super- Project | Microsoft Project | Primavera Project Planner | Project Manager Workbench | Project Scheduler6 | Time Line |
|------|----------------------------------|----------------------|----------------------|---------------------------------|---------------------------------|-----------------------|-----------|
| 0,5 | 8.00 | 2.60 | 2.89 | 3.44 | 4.85 | 2.29 | 3.53 |
| 1 | 11.52 | 6.22 | 7.82 | 5.35 | 8.53 | 8.57 | 5.44 |

Table A5: Estimated percent increase of the benchmark makespan as a function of NC .

| NC | Artemis Schedule Publisher | CA Super- Project | Microsoft Project | Primavera Project Planner | Project Manager Workbench | Project Scheduler6 | Time Line |
|------|----------------------------------|----------------------|----------------------|---------------------------------|---------------------------------|-----------------------|-----------|
| 1,5 | 9.87 | 4.98 | 5.83 | 4.80 | 7.01 | 5.56 | 4.55 |
| 2 | 9.65 | 3.84 | 4.88 | 3.98 | 6.36 | 5.29 | 4.42 |