

Kolisch, Rainer; Hempel, Kai

Working Paper — Digitized Version

Experimentelle Evaluation der methodischen Fundierung von Projektmanagementsoftware

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 372

Provided in Cooperation with:

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

Suggested Citation: Kolisch, Rainer; Hempel, Kai (1995) : Experimentelle Evaluation der methodischen Fundierung von Projektmanagementsoftware, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 372, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/155433>

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.

Nr. 372

**Experimentelle Evaluation der methodischen
Fundierung von Projektmanagementsoftware**

von

Rainer Kolisch und Kai Hempel

Mai 1995

Dr. Rainer Kolisch, Kai Hempel
Institut für Betriebswirtschaftslehre
Lehrstuhl für Produktion und Logistik
Christian-Albrechts-Universität zu Kiel
Olshausenstr. 40, D-24118 Kiel

1. Einleitung

EDV-gestütztes Projektmanagement (PM) ist eine weit verbreitete Planungsmethode auf allen Managementebenen.¹ Allein in Deutschland werden derzeit ca. 350 unterschiedliche kommerzielle PM-Programme angeboten.² Differenzierungen ergeben sich hinsichtlich der funktionalen Ausrichtung, beispielsweise für den F&E- oder den Fertigungsbereich, der benötigten Rechnerplattform und des Betriebssystems sowie der handhabbaren Projektgröße. Alle Programme besitzen jedoch mit den vier grundlegenden Planungsfunktionen Struktur-, Zeit-, Kapazitäts- und Kostenplanung eine gemeinsame modelltheoretische und methodische Grundlage. Vor dem Hintergrund der in den letzten Jahren drastisch reduzierten Kapazitäten und der gewachsenen Bedeutung von Lieferterminen hat insbesondere die Kapazitätsplanung eine rasante Weiterentwicklung erfahren. Während sich noch in den 80-er Jahren die Funktionalität auf die Darstellung von Kapazitätsbelastungsdiagrammen beschränkte und eine ressourcentreue Planung - wenn überhaupt - nur manuell durchführbar war,³ so ist heute die automatische Erzeugung ressourcenzulässiger Projektablaufpläne mit der Mehrzahl der PM-Programme möglich. Kern der Funktionalität ist das in *Abschnitt 2.2.1* näher dargestellte Grundmodell der Kapazitätsplanung. Programmabhängig werden darüber hinaus Erweiterungen angeboten, auf die in *Abschnitt 2.3* eingegangen wird.

Der vorliegende Beitrag untersucht die Qualität der Kapazitätsplanung von sieben kommerziellen PM-Programmen. Da von seiten der Softwarehersteller in der Regel keine Angaben zu den verwendeten Methoden gemacht werden, erfolgt die Beurteilung anhand einer experimentellen Untersuchung auf der Grundlage von 160 repräsentativen Testprojekten.

Der Beitrag ist wie folgt gegliedert: *Abschnitt 2* gibt zunächst einen kurzen Abriß der Aufgaben des Projektmanagements und eine vertiefende Darstellung der für die Kapazitätsplanung relevanten Modelle und Methoden. *Abschnitt 3* faßt den Stand bisheriger Arbeiten zur Evaluierung der Kapazitätsplanung von PM-Standardsoftware zusammen. Aufbau und Ergebnisse der experimentellen Untersuchung werden in *Abschnitt 4* dargestellt. Ein Resümee über Stand und zukünftige Entwicklung der Kapazitätsplanung innerhalb des Projektmanagements schließen den Beitrag.

¹ Vgl. Domschke/Drexel (1991b), S. 63 und Tobias (1991).

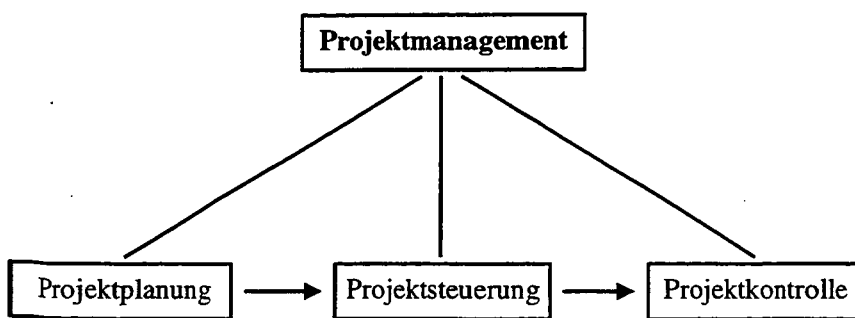
² Vgl. Dworatschek/Hayek (1992).

³ Vgl. Domschke/Drexel (1991a), S. 105.

2. Aufgaben und Methoden des Projektmanagements

Ein Projekt ist ein in der Regel einmaliges, zeitlich, sachlich und räumlich begrenztes, zielgerichtetes Vorhaben, das unter Verwendung knapper Ressourcen durchgeführt wird.⁴ Das Projektmanagement umfaßt alle Tätigkeiten, die eine im Sinn der Projektziele erfolgreiche Projektdurchführung herbeiführen. Als typische Projektziele gelten: Qualität, niedrige Projektkosten, Liefertermintreue, kurze Projektdauer sowie ein hoher Kapitalwert des Projekts.⁵ Die Gesamtaufgabe "Projektmanagement" läßt sich gemäß *Abbildung 1* in die Teilaufgaben der Projektplanung (Bestimmung eines Sollzustands), der Projektsteuerung (Umsetzung des Soll- in einen Istzustand) und der Projektkontrolle (Vergleich von Soll- und Istzustand sowie Durchführung einer Abweichungsanalyse) gliedern. Vom modell- und methodentheoretischen Standpunkt gibt es insbesondere im Rahmen der Projektplanung eine Reihe von anspruchsvollen Problemstellungen.

Abbildung 1: Teilaufgaben des Projektmanagements



2.1 Projektplanung

Die *Projektplanung* gliedert sich i.d.R. in die vier Stufen Struktur-, Zeit-, Kapazitäts- und Kostenplanung.⁶ In der *Strukturplanung*⁷ wird das Gesamtprojekt in einzelne Vorgänge $v=1,...,V$ zerlegt. Für jeden Vorgang v werden dessen Dauer d_v und die zu anderen Vorgängen existierenden technologischen Reihenfolgebeziehungen ermittelt, wobei N_v die Menge der Vorgang v direkt nachfolgenden Vorgänge bezeichnet. Abschließend wird das Projekt als Netzplan, in dem Vorgänge durch Knoten und technologische Reihenfolgebeziehungen durch

⁴ Zur Definition von Projekten vgl. auch *Schwarze* (1994), S. 11 ff., *Haberfellner* (1992), Sp. 2090 ff. sowie *Krüger* (1993), Sp. 3559 ff. Primär praxisorientierte Aspekte des Projektmanagements finden sich z.B. bei *Madauss* (1990).

⁵ Vgl. die empirische Untersuchung von *Icmeli* (1995).

⁶ Vgl. *Domschke/Drexel* (1991a). Die Kostenplanung ist nicht Gegenstand dieser Arbeit; vgl. hierzu z.B. *Nandi/Dutta* (1988).

⁷ Die Zerlegung des Projekts in Arbeitspakete und die Darstellung als Projektstrukturplan wird an dieser Stelle nicht gezeigt; vgl. hierzu *Schwarze* (1994), S. 31 ff.

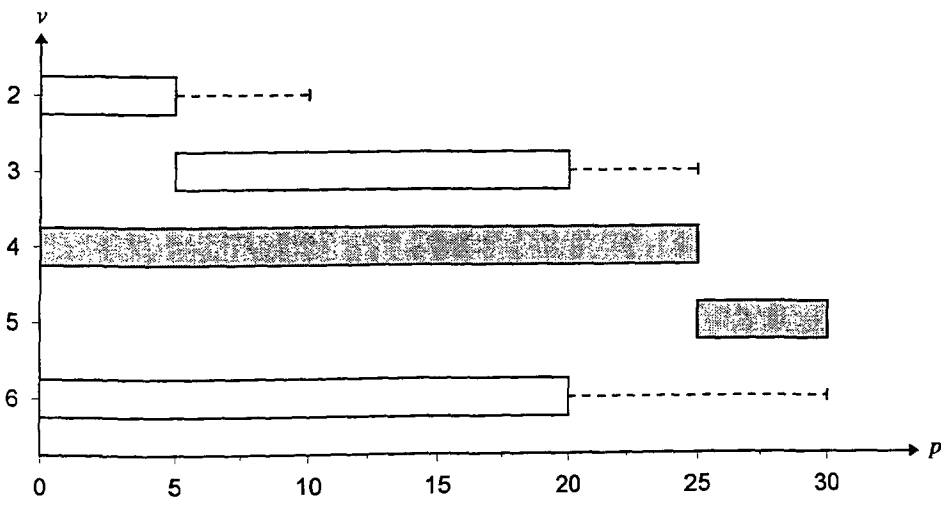
Pfeile symbolisiert werden, dargestellt.⁸ *Tabelle 1* gibt die Daten eines Beispielprojekts wieder. Die fiktiven Vorgänge $v=1$ und $v=7$ werden verwendet, um für das Projekt einen eindeutigen Start- sowie Endvorgang zu erhalten.

Tabelle 1: Daten des Beispielprojekts

v	1	2	3	4	5	6	7
d_v	0	5	15	25	5	20	0
N_v	{2,4,6}	{3}	{5}	{5}	{7}	{7}	-

Die *Zeitplanung* verwendet die Ergebnisse der Strukturplanung, indem für jeden Vorgang v unter Berücksichtigung der technologischen Reihenfolgebeziehungen ein Zeitfenster $[FAZ_v; SAZ_v]$ möglicher Anfangszeitpunkte berechnet wird.⁹ Ausgehend vom geplanten Projektbeginn werden zunächst mit der Vorwärtsrekursion *frühestzulässige Anfangszeitpunkte* FAZ_v ermittelt. Anschließend werden, ausgehend von der Projektdauer FAZ_v , mit der Rückwärtsrekursion *spätestzulässige Anfangszeitpunkte* SAZ_v errechnet. Schließlich lassen sich noch verschiedene Arten von Pufferzeiten,¹⁰ die über die zeitliche Disponierbarkeit der Vorgänge Auskunft geben, ermitteln. Für das Beispielprojekt ist in *Abbildung 2* das Ergebnis der Zeitplanung als Gantt-Diagramm abgebildet. Auf der Abszisse sind die Perioden $p=1, \dots, 30$ skaliert; die Vorgänge sind auf der Ordinate als Balken abgetragen und befinden sich in der frühestzulässigen zeitlichen Lage. Gestrichelte Linien geben die gesamte Pufferzeit an; grau unterlegte Vorgänge befinden sich auf dem kritischen Pfad.

Abbildung 2: Vorgangs-Gantt



⁸ Eine weitere, jedoch weniger gebräuchliche Darstellungsform ist die Verwendung von Pfeilen für Vorgänge und Knoten für Ereignisse. Siehe dazu beispielsweise *Domschke/Drexel* (1991a), S. 82 ff.

⁹ Zu Details der Berechnung von frühest- und spätestzulässigen Anfangszeitpunkten vgl. *Domschke/Drexel* (1991a), S. 89 ff.

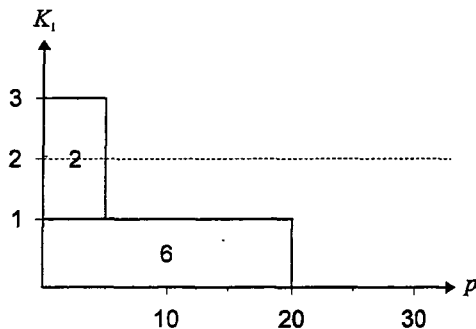
¹⁰ Zu unterschiedlichen Pufferzeiten und deren Berechnung vgl. *Altrogge* (1994) sowie *Ziegler* (1985).

2.2 Grundlagen der Kapazitätsplanung

2.2.1 Modell

Zur Durchführung der Kapazitätsplanung werden zunächst für jeden Vorgang Art und Umfang der benötigten Ressourcen (Material, Personal, Maschinen etc.) bestimmt. $r=1,\dots,R$ bezeichnet die mit K_r Einheiten je Periode vorhandenen Ressourcen. Vorgang v benötigt während der Bearbeitung je Periode k_{vr} Einheiten von Ressource r . Für das Beispielprojekt existiere $R = 1$ Ressource, die je Periode mit $K_1 = 2$ Einheiten verfügbar sei. Der Ressourcenbedarf der Vorgänge sei durch $k_{2,1} = 2$, $k_{6,1} = 1$ und $k_{1,1} = k_{3,1} = k_{4,1} = k_{5,1} = k_{7,1} = 0$ beschrieben. *Abbildung 3* zeigt, daß der in der Zeitplanung ermittelte Ablaufplan damit nicht mehr zulässig ist, weil in den Perioden $p=1,\dots,5$ die Kapazitätsnachfrage das vorhandene Kapazitätsangebot übersteigt.

Abbildung 3: Ressourcen-Gantt der Zeitplanung



Um nun einen Ablaufplan zu erhalten, der sowohl bezüglich der technologischen Reihenfolgebeziehungen als auch bezüglich der Kapazitätsrestriktionen zulässig ist, läßt sich ein binäres Entscheidungsmodell formulieren. Zunächst werden die Zeitfenster der Anfangszeitpunkte durch Addieren einer Konstanten zu SAZ_v so vergrößert, daß für jeden Vorgang ein kapazitätstreuer Start innerhalb seines Zeitfenster möglich ist.¹¹ Unter Verwendung der binären Entscheidungsvariable

$$x_{vp} = \begin{cases} 1, & \text{falls Vorgang } v \text{ am Ende von Periode } p \text{ gestartet wird} \\ 0, & \text{sonst} \end{cases}$$

erhält man dann das *Grundmodell der Kapazitätsplanung*:¹²

$$\text{Minimiere } Z(\underline{x}) = \sum_{p=FAZ_v}^{SAZ_v} p \cdot x_{vp} \quad (1)$$

unter den Nebenbedingungen

¹¹ Beispielsweise verwenden Domschke/Drexel (1991b) die Konstante $\sum d_v - FAZ_v$.

¹² Das Modell wurde erstmalig von Pritsker et al. (1969) vorgestellt. Zur Anwendung des Modells in der Produktionswirtschaft vgl. Günther/Tempelmeier (1995) und Kolisch (1995).

$$\sum_{p=FAZ_v}^{SAZ_v} x_{vp} = 1 \quad v=1,\dots,V \quad (2)$$

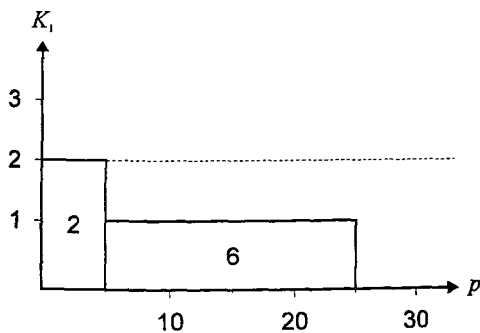
$$\sum_{p=FAZ_v}^{SAZ_v} (p+d_v) x_{vp} \leq \sum_{\pi=FAZ_n}^{SAZ_n} \pi x_{n\pi} \quad v=1,\dots,V-1, n \in N_v \quad (3)$$

$$\sum_{v=1}^V k_{vr} \sum_{\pi=\max\{1, p-d_v+1\}}^p x_{v\pi} \leq K_r \quad r=1,\dots,R, p=1,\dots,P \quad (4)$$

$$x_{vp} \in \{0,1\} \quad v=1,\dots,V, p=FAZ_v,\dots,SAZ_v \quad (5)$$

Die Nebenbedingungen (2) sichern, daß jeder Vorgang genau einmal bearbeitet wird. Mit den Nebenbedingungen (3) werden die technologischen Reihenfolgebeziehungen berücksichtigt, während die Nebenbedingungen (4) garantieren, daß je Ressource r und Periode p die kumulierte Kapazitätsnachfrage der in Bearbeitung befindlichen Vorgänge das Kapazitätsangebot K_r nicht überschreitet. Mit der Minimierung der Projektdauer berücksichtigt die Zielfunktion (1) nur eines der in *Abschnitt 2* genannten Ziele. Das Ziel der Projektdauerminimierung ist jedoch komplementär zu den Zielen niedrige Projektkosten, Liefertermintreue und hoher Kapitalwert. Eine kurze Projektlaufzeit verringert ceteris paribus die Kapitalbindungskosten sowie die Gefahr einer Lieferterminüberschreitung. Da in der Regel die Projektauszahlungen zu Beginn und die Projekteinzahlungen am Ende eines Projekts erfolgen, führt eine kurze Projektdauer außerdem zu einer geringeren Abdiskontierung der Einzahlungen und somit ceteris paribus zu einem höheren Kapitalwert. Weitere Gründe für die Verwendung der Zielfunktion (1) sind die Reduzierung der Datenunsicherheit und die frühere Bereitstellung von bis dato verwendeten Ressourcen für zukünftige Projekte.¹³

Abbildung 4: Ressourcen-Gantt der Kapazitätsplanung



Löst man (1) - (5) für das vorgestellte Beispielproblem mit einem Standardverfahren der ganzzahligen Optimierung, so erhält man die optimale Lösung $x_{1,0}=x_{2,0}=x_{3,5}=x_{4,0}=$

¹³ Vgl. Drexl/Kolisch (1993).

$x_{5,25} = x_{6,5} = x_{7,30} = 1$ mit einer gegenüber der Zeitplanung unveränderten Projektdauer von 30 Perioden. *Abbildung 4* stellt das korrespondierende und nun zulässige Ressourcen-Gantt dar.

Das Grundmodell (1) - (5) bildet bereits das Kernproblem der Kapazitätsplanung, die Konkurrenz der Vorgänge um knappe Ressourcen, ab. Die einschränkenden Annahmen erlauben die Erzeugung von optimalen Referenzlösungen und damit eine fundierte Bewertung der in den PM-Programmen verwendeten Methoden. Modellverallgemeinerungen, wie z.B. die Hinterlegung von Ressourcenkalendern, variable Kapazitätsangebots- und Nachfrageprofile sowie die Unterbrechbarkeit von Vorgängen, lassen sich ohne Schwierigkeiten durchführen.¹⁴ Allerdings ist - von Ausnahmen abgesehen - dann keine optimale Lösung des modifizierten Modells mehr möglich; heuristische Lösungen sind jedoch weiterhin ermittelbar.

2.2.2 Methoden

Das Grundmodell der Kapazitätsplanung gehört zur Klasse der *NP*-schweren Probleme.¹⁵ Der Lösungsaufwand steigt also exponentiell mit der Anzahl der Vorgänge. Optimale Verfahren können deshalb nur Probleme mit bis zu ca. 50 Vorgängen lösen. Aufgrund der in *Abschnitt 2.2.1* angesprochenen großen theoretischen und praktischen Relevanz des Modells sind seit Anfang der 60-er Jahre eine Vielzahl von Arbeiten zur optimalen und zur heuristischen Lösung des Problems erschienen. Einen Überblick über den jeweiligen Stand der Lösungsverfahren geben *Davis* (1966) und (1973), *Herroelen* (1972) sowie *Domschke* und *Drexel* (1991b).

Die Mehrzahl der optimalen und heuristischen Verfahren verwendet das folgende Konstruktionsschema:¹⁶ Beginnend mit der Periode $p=1$ und dem fiktiven Startvorgang $v=1$ werden in jedem Iterationsschritt die Vorgänge ermittelt, die jeweils einzeln in Bezug auf technologische Reihenfolgebeziehungen und verfügbare Kapazitäten einplanbar sind. Für das angeführte Beispiel sind das zu Beginn der Periode $p=1$ die Vorgänge $v=2, 4$ und 6 . Im Falle knapper Ressourcen können jedoch nicht alle einplanbaren Vorgänge auch gleichzeitig gestartet werden. So schließt sich im Beispiel die simultane Bearbeitung der Vorgänge $v=2$ und 6 aus. Es muß also entschieden werden, welche Vorgänge jetzt, das heißt in der aktuellen Periode, gestartet werden und welche Vorgänge dadurch erst in zukünftigen Perioden eingeplant werden können.

Das exakte Branch und Bound Verfahren von *Demeulemeester/Herroelen* (1992) enumeriert für jede Periode, in der aufgrund der Kapazitätsrestriktionen die einplanbaren Vorgänge nicht

¹⁴ Zur Integration von Ressourcenkalendern vgl. *Zhan* (1992); die Berücksichtigung von variablen Kapazitätsangebots- und Nachfrageprofilen ist bei *Kolisch* (1995), S. 20 ff. zu finden. Vorgangsunterbrechungen können durch Vorgangszерlegung, siehe z.B. *Sprecher et al.* (1995), berücksichtigt werden.

¹⁵ Vgl. *Blazcewicz et al.* (1983).

¹⁶ Zu Details vgl. *Kolisch* (1995), S. 66 ff.

gleichzeitig begonnen werden können, alle reihenfolge- und ressourcenzulässigen Startalternativen. Jede Alternative ist dabei durch die Menge der nicht gestarteten Vorgänge gekennzeichnet. Für das Beispiel ergeben sich die beiden Möglichkeiten, die Vorgänge $v=2$ und $v=4$ beziehungsweise die Vorgänge $v=4$ und $v=6$ zu starten. Die dazu gehörenden Mengen nicht gestarteter Vorgänge sind $\{6\}$ und $\{2\}$.

Heuristische Prioritätsregelverfahren treffen die Entscheidung darüber, welche Vorgänge gestartet werden, indem die Vorgänge anhand eines Dringlichkeits- oder Prioritätswerts sortiert werden. Vergleichsweise gute Ergebnisse werden erzielt, wenn man die Vorgänge absteigend nach dem spätestzulässigen Fertigstellungszeitpunkt sortiert. Für das Beispiel erhält man dann die Sortierung 2 vor 4 vor 6. Damit werden die Vorgänge $v=2$ und $v=4$ gestartet, während Vorgang $v=6$ verschoben wird.

Neben dem geschilderten Konstruktionsschema gibt es für Prioritätsregelverfahren noch eine weitere Möglichkeit: Vorgänge müssen *nur* hinsichtlich der technologischen Reihenfolgebeziehungen, *nicht* aber bezüglich der Ressourcenrestriktionen einplanbar sein. In der durch die Prioritätsliste definierten Reihenfolge werden anschließend alle Vorgänge in der jeweils hinsichtlich Reihenfolge- und Ressourcenrestriktionen frühestzulässigen Periode gestartet. Im Beispiel wird erst der Vorgang $v=2$ zu Beginn der Periode $p=1$, dann der Vorgang $v=4$ ebenfalls zu Beginn der Periode $p=1$, und schließlich der Vorgang $v=6$ zu Beginn der Periode $p=5$ gestartet.

Für heuristische prioritätsregelgestützte Verfahren stehen damit zwei Konstruktionsschemata und zahlreiche Prioritätsregeln zur Verfügung. Die in kommerzieller PM-Software realisierten Methoden zur Kapazitätsplanung verwenden in der Regel eine oder mehrere dieser Kombinationen aus Konstruktionsschema und Prioritätsregel.¹⁷ Meistens wird dem Anwender zusätzlich die Möglichkeit geboten, Prioritätsregeln explizit auszuwählen oder zu definieren, beziehungsweise durch die Vergabe von Rängen für die einzelnen Vorgänge, die Auswahl der einplanbaren Vorgänge direkt zu bestimmen. In den Arbeiten von *Johnson* (1992) und *Maroto et al.* (1994) werden die in der Standardeinstellung verwendeten Prioritätsregeln einiger PM-Programme aufgeführt.¹⁸

Die derzeit besten, (noch) nicht in kommerzieller PM-Software implementierten Heuristiken setzen über die geschilderten Verfahren hinaus eine oder mehrere der folgenden Methoden ein: verbesserte Prioritätsregeln, randomisierte Vorgangsauswahl sowie eine geeignete Kombination der beiden Konstruktionsschemata. *Kolisch* (1995) schlägt eine "adaptive" Lösungssuche vor. Dabei wird zuerst auf der Grundlage einer Projektdatenanalyse ein Konstruktionsschema ausgewählt. Anschließend wird eine zuvor spezifizierte Anzahl von Projektablaufplänen erzeugt. Im Gegensatz zu der deterministischen Vorgangsauswahl, bei

¹⁷ Vgl. *Johnson* (1992).

¹⁸ CA-SuperProject und Primavera Project Planner ordnen die einplanbaren Vorgänge aufsteigend nach der gesamten Pufferzeit; Time Line und Microsoft Project sortieren die Vorgänge aufsteigend nach FAZ.

der für jeden Vorgang ein Rang errechnet wird, werden Auswahlwahrscheinlichkeiten für alle einplanbaren Vorgänge ermittelt. Die Auswahlwahrscheinlichkeiten ergeben sich als Funktion einer neuen Prioritätsregel und der Anzahl bereits generierter Projektablaufpläne. *Leon/Balakrishnan* (1995) verwenden ein Hybrid der beiden Konstruktionsschemata. Die Sortierung der einplanbaren Vorgänge erfolgt absteigend nach einer für jeden Vorgang gezogenen gleichverteilten Zufallszahl aus dem Intervall $[0,25 d_v; 1,75 d_v]$. Damit werden tendenziell Vorgänge v mit längerer Bearbeitungsdauer d_v präferiert. Durch eine laufende Anpassung der Intervallgrenzen wird versucht, möglichst gute Lösungen zu erzeugen.

Tabelle 2 dokumentiert den in den letzten vier Jahrzehnten bei der Lösung des Grundmodells der Kapazitätsplanung erzielten Fortschritt, der für Heuristiken anhand der mittleren prozentualen Überschreitung der minimalen Projektdauer und für optimale Verfahren anhand der rechenbaren Problemgröße sowie der dafür benötigten mittleren Rechenzeit in CPU-Sekunden angegeben wird.

Tabelle 2: Fortschritte bei der Lösung des Grundmodells der Kapazitätsplanung

Jahrzehnt		1960	1970	1980	1990
Heuristiken	Projektdauerverlängerung (%)	7% ^(a)	5,6% ^(b)	2,28% ^(c)	< 1 % ^(d)
	Problemgröße (V)	8 ^(e)	< 10 ^(f)	≤ 51 ^(g)	≤ 51 ^(h)
Optimale Verfahren	Rechenzeit (CPU-Sekunden)	2,5	3	< 5	< 1

^(a) *Pacoe* (1966)

^(b) *Davis/Patterson* (1975)

^(c) *Alvarez-Valdes/Tamarit* (1889)

^(d) *Kolisch* (1995) und *Leon/Balakrishnan* (1995)

^(e) *Pritsker et al.* (1969)

^(f) *Patterson/Huber* (1974)

^(g) *Patterson* (1984)

^(h) *Demeulemeester/Herroelen* (1992)

Im Vorgriff auf die in *Abschnitt 3* dargestellten Ergebnisse läßt sich feststellen, daß für das Grundmodell der Kapazitätsplanung die in der Literatur veröffentlichten Methoden den in kommerzieller PM-Software implementierten Verfahren deutlich überlegen sind. Während die derzeit besten Verfahren eine mittlere Projektdauerverlängerung von weniger als 1% erreichen, erzeugen die in dieser Untersuchung getesteten PM-Programme Projektdauerverlängerungen von ca. 4,4 % bis 9,8 %. Die Bedeutung dieses Unterschieds läßt sich daran abschätzen, daß eine Projektdauerverlängerung von 5% bei einer mittlere Projektdauer von ca.

2 Jahren¹⁹ das Projektende und damit die Einzahlung des Projekterlöses um ca. einen Monat verschiebt.

2.3 Erweiterungen des Grundmodells

Alle in dieser Untersuchung getesteten PM-Programme stellen dem Anwender das Grundmodell der Kapazitätsplanung zur Verfügung. In der Regel wird es dort als "lineares, konstantes oder gleichverteiltes Auslastungsmodell" bezeichnet. Darüber hinaus sind die in *Abschnitt 2.2.1* angesprochenen Erweiterungen bei den meisten Programmen möglich. Zwei weitere programmspezifische Optionen sind i) ein je Vorgang fixer Ressourcengesamtbedarf und ein variabler Ressourcenperiodenbedarf sowie ii) die Modellierung der Vorgangsdauer als Funktion des Ressourcenverbrauchs:

i) Im Gegensatz zum Grundmodell der Kapazitätsplanung wird die Annahme eines in jeder Bearbeitungsperiode konstanten Ressourcenbedarfs fallengelassen. Stattdessen fordert man, daß die Summe der während der Bearbeitungsdauer benötigten Kapazitäten einem zuvor spezifizierten Gesamtverbrauch entspricht. Beispielsweise kann für die in der *Abbildung 3* dargestellte unzulässige Lösung dann eine zulässige Lösung mit der gleichen zeitlichen Lage der Vorgänge erzeugt werden, indem der Vorgang $v=6$ in den ersten 5 Bearbeitungsperioden keine Kapazitätseinheiten, in den nächsten fünf Perioden 2 Kapazitätseinheiten und in den verbleibenden 10 Bearbeitungsperioden jeweils eine Kapazitätseinheit von Ressource $r=1$ zugewiesen bekommt. Eine solche Erweiterung des Grundmodells existiert in der Literatur bisher nicht, kann jedoch wie folgt vorgenommen werden: Statt des konstanten Ressourcenbedarfs k_{vr} je Periode führt man den konstanten Gesamtbedarf K_{vr} von Vorgang v an Ressource r ein. Der Periodenbedarf wird als Variable k_{vbr} abgebildet. k_{vbr} ist die Anzahl der von Ressource r in der $b=1, \dots, d_v$ -ten Vorgangsbearbeitungsperiode des Vorgangs v benötigten Kapazitätseinheiten. Prinzipiell kann dann in (1) - (5) k_{vr} durch k_{vbr} ersetzt werden und durch Hinzunahme der Nebenbedingungen

$$\sum_{b=1}^{d_v} k_{vbr} = K_{vr} \quad v=1, \dots, V, r=1, \dots, R \quad (6)$$

$$k_{vbr} \geq 0 \quad v=1, \dots, V, b=1, \dots, d_v, r=1, \dots, R \quad (7)$$

erhält man die erste Modellerweiterung. Leider ist dieses Modell jedoch in der Nebenbedingung (4) nichtlinear, womit die Erzeugung optimaler (Referenz-)Lösungen außerordentlich erschwert wird.

ii) Bei der Modellierung der Vorgangsdauer als Funktion des Ressourcenverbrauchs entspricht die Vorgangsdauer der Zeit, die benötigt wird, den vorher spezifizierten Gesamtressourcen-

¹⁹ Vgl. die empirische Untersuchung von Icmeli (1995).

verbrauch zu decken. Beispielsweise kann Vorgang $v=6$ statt in 20 auch in 10 Perioden bearbeitet werden, wenn der Vorgang in jeder Bearbeitungsperiode zwei Kapazitätseinheiten statt einer Kapazitätseinheit von Ressource $r=1$ zugewiesen bekommt. Diese Modellerweiterung ist für den diskreten Fall bereits in der Literatur bekannt und wird als Zeit-Ressourcen-Tradeoff bezeichnet.²⁰ Für jeden Vorgang v werden mehrere Bearbeitungsalternativen A_v gebildet, wobei jede Bearbeitungsalternative $a=1, \dots, A_v$ eines Vorgangs v durch eine Bearbeitungsdauer d_{va} und einen Ressourcenverbrauch k_{var} bezüglich jeder Ressource r bestimmt ist. Beispielsweise könnten für Vorgang $v=6$ die folgenden zwei Bearbeitungsalternativen verwendet werden: Alternative $a=1$ mit $d_{6,1}=20$, $k_{6,1,1}=1$ und Alternative $a=2$ mit $d_{6,2}=10$, $k_{6,1,2}=2$. Die Lösung des Zeit-Ressourcen-Tradeoff-Modells besitzt den gleichen (exponentiellen) Schwierigkeitsgrad wie das Grundmodell der Kapazitätsplanung. In der Literatur finden sich hierzu eine Reihe von exakten und heuristischen Verfahren.²¹

3. Vorliegende Untersuchungen zur methodischen Fundierung von PM-Software

Trotz der großen Bedeutung, die die Projektplanung im allgemeinen und die Kapazitätsplanung im besonderen für ein erfolgreiches Projektmanagement besitzen, sind bisher nur zwei Arbeiten zur methodischen Fundierung von PM-Software erschienen:²²

Johnson (1992) hat sieben in den USA vertriebene PM-Standardprogramme (in insgesamt 13 Versionen) auf der Basis der in der Literatur bekannten Testdatensätze von *Patterson* (1984) verglichen. Die verwendeten 110 Testdatensätze sind eine über mehrere Untersuchungen gewachsene, unsystematische Sammlung von Projektplanungsproblemen mit $V=7$ bis 51 Vorgängen und $R=1$ bis 3 Ressourcen.²³ *Johnson* hat jedes der 110 Probleme mit den 13 PM-Programmen gelöst und mit der optimalen Referenzlösung verglichen. Es ergab sich eine über alle Testdaten und Programme gemittelte Projektdauerverlängerung von 9,88%. Dabei variierte die Projektdauerverlängerung in Abhängigkeit der verwendeten PM-Software zwischen 5,03%, und 25,6%. Die beiden PM-Programme mit den besten Ergebnissen waren Time Line 2.0 und Hornet mit einer durchschnittlichen Projektdauerverlängerung von 5,03% und 5,53%; die PM-Programme mit den schlechtesten Planungsergebnissen waren Microsoft Project 1.0 und PertMaster Advanced mit einer durchschnittlichen Projektdauerverlängerung von 25,6% beziehungsweise 14,1%.

Maroto et al. (1994) haben MS Project 3.0 und drei Versionen von CA-SuperProject (V.2.0 für DOS, V.2.0 und V.3.0 für Windows) auf der Basis von 48 mit einem vollständig

²⁰ Vgl. u.a. *Talbot* (1982).

²¹ Zu einer Übersicht vgl. *Kolisch* (1995), S. 130 ff.

²² Eine Untersuchung zur Messung der Neuheit von Software im allgemeinen findet sich bei *Brockhoff/Zanger* (1993).

²³ Eine genaue Beschreibung der Testdatensätze wird in *Patterson* (1984) sowie in *Demeulemeester/Herroelen* (1992) gegeben.

faktoriellen Versuchsdesign generierten Testprojekten verglichen. Das Versuchsdesign berücksichtigte $R=1, 2$ und 3 Ressourcen sowie Probleme mit starken und geringen Kapazitätsrestriktionen, das heißt *ceteris paribus* niedrigen und hohen K_r -Werten in Nebenbedingung (4). Für jede Faktorausprägungskombination wurden 8 Testprobleme generiert, wobei die Anzahl der Vorgänge jeweils zufällig zwischen 30 und 100 gewählt wurde. Als Maß für die Güte der Planungsergebnisse wurde die Differenz zwischen der in der Kapazitätsplanung und der in der Zeitplanung berechneten Projektdauer verwendet. Mit Hilfe einer Varianzanalyse ermittelten *Maroto et al.* den Einfluß der Faktoren und den Einfluß der Interaktion zwischen den Faktoren. Die verwendete PM-Software, die Stärke der Kapazitätsrestriktionen und alle zweifachen Interaktionen zwischen den Faktoren PM-Software, Anzahl der Ressourcen und Stärke der Kapazitätsrestriktionen besitzen bei einer Irrtumswahrscheinlichkeit von $\alpha=1\%$ einen signifikanten Einfluß. Die Anzahl der Ressourcen ist bei einer Irrtumswahrscheinlichkeit von $\alpha=5\%$ signifikant.

Zusammenfassend kann festgestellt werden, daß derzeit kein methodischer Vergleich der Kapazitätsplanung von PM-Software vorliegt, der auf der Grundlage eines systematischen Testdesigns und unter Berücksichtigung einer repräsentativen Anzahl aktueller PM-Programme durchgeführt wurde. Ein solcher Vergleich ist daher Gegenstand des folgenden *Abschnitts 4*.

4. Experimentelle Untersuchung

4.1 Untersuchungsobjekte

Bei der Auswahl der zu untersuchenden PM-Standardsoftware standen drei Gesichtspunkte im Vordergrund: *i)* Die PM-Programme sollten eine weite Verbreitung besitzen, *ii)* die Funktionalität sollte das in *Abschnitt 2.2.1* dargestellte Grundmodell der Kapazitätsplanung beinhalten und *iii)* die Rechner- und Betriebssystemanforderungen der PM-Programme sollten vergleichbar sein. Unter Berücksichtigung dieser Kriterien wurden die in *Tabelle 3* aufgeführten sieben PM-Programme ausgewählt. Bei allen handelt es sich um PC-basierte Versionen für die Benutzeroberfläche Windows.

Tabelle 3: Geteste PM-Standardsoftware

Programm	Vertrieb
Artemis Schedule Publisher V.4.1	Lucas Management Systems, Neuss
CA-SuperProject V.3.0 C	Computer Associates, Darmstadt
Microsoft Project V.4.0	Microsoft, Unterschleißheim
Primavera Project Planner V.1.0	Intec, Landshut
Project Manager Workbench V.1.1.02w	Applied Business Tech., Hamburg
Project Scheduler 6.0 V.1.02	Scitor, Frankfurt/Main
Time Line V.6.0.0	Symantec, Düsseldorf

4.2 Untersuchungsdesign

Um die Qualität der Kapazitätsplanung bei einer experimentellen Untersuchung reliabel zu messen, ist eine repräsentative Auswahl an Testprojekten notwendig. Auf der Grundlage eines faktoriellen Versuchdesigns wurden in der vorliegenden Arbeit 160 Testprojekte generiert. Jedes Testprojekt kann durch die Ausprägung systematisch und zufällig variierten Projektstrukturparameter, wie zum Beispiel die Anzahl der Vorgänge und die Anzahl der Ressourcen, charakterisiert werden. In der Literatur wurden bisher eine Vielzahl von Vorschlägen gemacht, wie Projektstrukturparameter geeignet zu definieren sind.²⁴ In der vorliegenden Arbeit wurden die Projektstrukturparameter Anzahl der Vorgänge V , Anzahl der Ressourcen R , Netzwerkkomplexität (NK), Ressourcenfaktor (RF) und Ressourcenstärke (RS) systematisch variiert.²⁵

$$NK = \frac{1}{V} \sum_{v=1}^V |N_v| \quad (8)$$

$$RF = \frac{1}{(V-2) \cdot R} \sum_{v=1}^V \sum_{r=1}^R \begin{cases} 1, \text{ falls } k_{vr} > 0 \\ 0, \text{ sonst} \end{cases} \quad (9)$$

$$RS_r = \frac{K_r - K_r^{\min}}{K_r^{\max} - K_r^{\min}} \quad r=1, \dots, R \quad (10)$$

Die Netzwerkkomplexität (8) ist die durchschnittliche Anzahl der einem Vorgang v direkt nachfolgenden Vorgänge N_v . Viele (wenige) technologische Reihenfolgebeziehungen innerhalb des Projekts implizieren einen hohen (niedrigen) NK -Wert. Der Ressourcenfaktor (9) reflektiert den mittleren Prozentsatz der zur Durchführung eines Vorgangs benötigten Ressourcen. Bei einem RF -Wert von 1 benötigt jeder Vorgang $v=2, \dots, V-1$ zu seiner Durch-

²⁴ Zur ausführlichen Diskussion von Projektstrukturparametern vgl. Kolisch et al. (1995).

²⁵ Vgl. Kolisch et al. (1995).

führung alle Ressourcen; ein RF -Wert von 0 tritt dann ein, wenn keiner der Vorgänge Ressourcen zu seiner Durchführung erfordert. Während der Ressourcenfaktor die Tatsache mißt, *ob* Ressourcen benötigt werden, reflektiert die Ressourcenstärke (10) das Verhältnis von Ressourcenangebot und Ressourcennachfrage. K_r^{min} ist die von Ressource r mindestens benötigte Kapazität, damit jeder Vorgang einzeln bearbeitet werden kann; K_r^{max} ist die von Ressource r benötigte Kapazität für den innerhalb der Zeitplanung ermittelten Ablaufplan. Der RS -Wert ist ebenfalls auf das Intervall $[0,1]$ normiert. Mit kleiner werdendem RS -Wert sind die Projektplanungsprobleme zunehmend stärker kapazitätsbeschränkt. *Tabelle 4* stellt die realisierten Ausprägungen der systematisch variierten Faktoren dar.

Tabelle 4: Systematisch variierte Faktoren

Faktor	V	R	RS	RF	NK
Ausprägung	10	2	0.2	0.5	1.5
	20	3	0.5	1	2
	30	4	0.7		

Die Projektstrukturparameter Vorgangsdauer d_v , Höhe der Ressourcennachfrage k_{vr} sowie die Anzahl direkter Vorgänger V_v und Nachfolger N_v eines Vorgangs v wurden jeweils zufällig aus den in *Tabelle 5* definierten gleichverteilten Intervallen generiert.

Tabelle 5: Zufällig variierte Faktoren

Faktor	d_v	k_{vr}	V_v	N_v
Minimale Ausprägung	1	1	1	1
Maximale Ausprägung	10	10	3	3

Die Projektstrukturparameter Anzahl der direkten Nachfolger des fiktiven Startvorgangs $v=1$ und Anzahl der direkten Vorgänger des fiktiven Endvorgangs $v=V$ wurden jeweils auf den konstanten Wert 3 fixiert.

Bei einem vollständig faktoriellen Testdesign und 10 Replikationen je Faktorausprägungskombination würde man $10 \cdot 3^3 \cdot 2^2 = 1.080$ Testdatensätze erhalten, die mit jedem der sieben PM-Programme gelöst werden müßten. Um den damit verbundenen Aufwand zu reduzieren, wurde ein fraktioniertes faktorielles Testdesign verwendet. Dem Vorteil einer Reduzierung der Faktorausprägungskombinationen steht jedoch der Nachteil gegenüber, daß keine Interaktionseffekte zwischen den Faktoren gemessen werden können.²⁶ Das fraktionierte Testdesign wurde auf der Grundlage orthogonaler Matrizen erstellt.²⁷ Die Anzahl der Faktorausprägungskombinationen konnte dadurch auf 16 reduziert werden. Bei zehnfacher

²⁶ Vgl. Green (1974), S. 61.

²⁷ Es wurde das Programm Conjoint Designer V.2.0 von Bretton-Clark verwendet. Zu einer ausführlichen Beschreibung des Verfahrens vgl. Addelman (1962).

Replikation je Faktorausprägungskombination ergaben sich damit $10 \cdot 16 = 160$ Testprojekte. Um einen Referenzwert Z_j^* für jedes der $j=1, \dots, 160$ Testprobleme zu erhalten, wurde zunächst versucht, diese mit dem optimalen Verfahren von *Demeulemeester/Herroelen* (1992) innerhalb einer Stunde CPU-Rechenzeit optimal zu lösen. Für 153 Testprobleme konnte der optimale Zielfunktionswert errechnet werden, für die restlichen 7 Probleme ergab sich eine obere Schranke der minimalen Projektdurchführungsdauer. Die Errechnung der Zielfunktionswerte Z_{ij} der $i=1, \dots, 7$ getesteten Softwarepakete für die $j=1, \dots, 160$ Testprobleme erfolgte durch das Einlesen der Testprojekte in das jeweilige PM-Programm und die anschließende Durchführung der als „Kapazitätsabgleich“ oder „Ressourcenausgleich“ bezeichneten Funktion zur Lösung des Grundmodells der Kapazitätsplanung. Dabei wurde, soweit dies überhaupt vom Benutzer beeinflussbar war, stets die Standardeinstellung des jeweiligen Programms verwendet. Die Güte der Projektplanung des i -ten Programms für das j -te Testprojekt wurde anhand der prozentualen Verlängerung der minimalen Projektdauer Π_{ij} gemessen:

$$\Pi_{ij} = \frac{Z_{ij} - Z_j^*}{Z_j^*} \cdot 100 \quad (11)$$

4.3 Gesamtergebnis

Abbildung 5 stellt die Häufigkeitsverteilung der $7 \cdot 160 = 1.120$ Π_{ij} -Werte dar. Der Median beträgt 3,57 %, der Mittelwert 5,79 % bei einer Standardabweichung von 7,51 % und einer Spannbreite von [0%; 51,85%]. In 45% der Fälle ermitteln die in den PM-Programmen implementierten Verfahren die kürzestmögliche Projektdauer. Der Verlauf der Häufigkeitsverteilung deutet darauf hin, daß die Mehrzahl der Programme die einplanbaren Vorgänge absteigend nach der gesamten Pufferzeit oder dem spätestzulässigen Fertigstellungszeitpunkt sortiert.²⁸ *Tabelle 6* gibt den Mittelwert μ und die Standardabweichung σ der Projektdauerverlängerung Π_i für jedes der $i=1, \dots, 7$ PM-Programme wieder. Die mittlere Projektdauerverlängerung aller PM-Programme ist mit $\Pi=5,79$ % deutlich besser als der in der Untersuchung von *Johnson* (1992) dokumentierte Wert von 9,88 %. Eine auf die Verwendung einfacherer Testprojekte zurückzuführende Verbesserung der Ergebnisse kann ausgeschlossen werden, weil das in dieser Untersuchung eingesetzte Testdesign eher schwierigere Testprojekte als die von *Johnson* (1992) verwendeten Probleme generiert.²⁹

²⁸ Vgl. *Davis/Patterson* (1975). Die beiden Autoren zeigen auch, daß sich die beiden Regeln bei Verwendung des in *Abschnitt 2.2.2* dargestellten ersten Konstruktionsschemas entsprechen.

²⁹ Vgl. *Kolisch et al.* (1995).

Abbildung 5: Häufigkeitsverteilung der Projektdauerverlängerung Π_{ij}

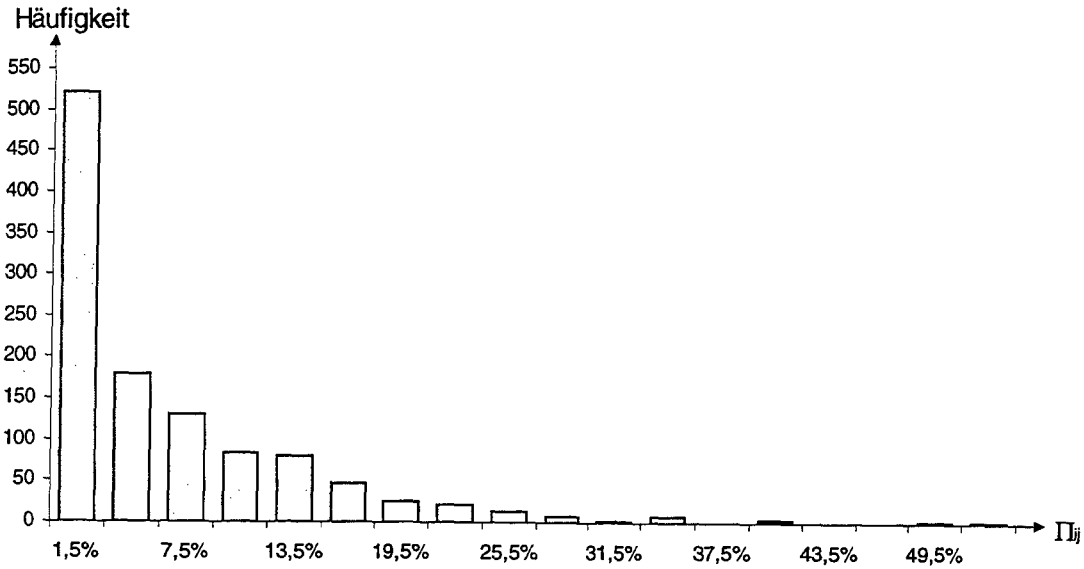


Tabelle 6: Mittlere Projektdauerverlängerung Π_i der PM-Programme

Programm	Projektdauerverlängerung	
	$\mu(\Pi_i)$	$\sigma(\Pi_i)$
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 %

4.4 Einfluß der Faktoren

Im folgenden soll der Einfluß der systematisch variierten Projektstrukturparameter auf die Lösungsgüte der getesteten PM-Programme untersucht werden. Zunächst werden Aussagen auf Basis der gewählten Stichprobe von 160 Projektplanungsproblemen getroffen. Diese Beobachtungen dienen anschließend dazu, Hypothesen über den generellen Einfluß der Faktoren zu definieren und zu testen.

Die Tabellen A1 - A5 im Anhang geben das arithmetische Mittel und die Standardabweichung der PM-Programme für die Teilstichproben, die jeweils durch die Ausprägung der systematisch variierten Faktoren definiert werden, wieder. Die folgenden Beobachtungen können gemacht werden:

- Mit zunehmender Vorgangszahl verschlechtern sich die Ergebnisse der Kapazitätsplanung. Eine Ausnahme bildet das Programm Artemis Schedule Publisher. Das Ergebnis für 30 Vorgänge ist geringfügig besser als das Ergebnis bei 20 Vorgängen.
- Drei der sieben PM-Programme erzeugen schlechtere Ergebnisse für Probleme mit einer größeren Zahl knapper Ressourcen. Von den vier verbleibenden PM-Programmen verbessern drei Programme (CA-SuperProject, Primavera Project Planner und Project Scheduler 6) die Ergebnisse beim Übergang von $R=2$ zu $R=3$ Ressourcen geringfügig; ein Programm (Artemis Schedule Publisher) zeigt deutlich bessere Ergebnisse. Für alle Programme gilt jedoch, daß die Kapazitätsplanung bei $R=3$ Ressourcen schlechtere Ergebnisse als bei nur einer Ressource erzeugt.
- Zunehmende Ressourcenknappheit, also steigende RS -Werte, verschlechtern die Planungsergebnisse.
- Die Projektdauerverlängerung nimmt zu, wenn die durch den RF -Wert gemessene Anzahl der von den Vorgängen benötigten Ressourcen ansteigt.
- Ein hoher Grad an technologischer Verflechtung zwischen den Vorgängen bewirkt bessere Planungsergebnisse.

Offenbar besitzt jeder der fünf Faktoren einen in Abhängigkeit von der getesteten PM-Software mehr oder minder starken Einfluß. Es wird deshalb für jede Faktor-Programmkombination (i, j) die folgende Nullhypothese H_0 formuliert:

H_0 : Die Güte des von der i -ten PM-Software erzeugten Kapazitätsplans ist nicht von der Ausprägung des j -ten Faktors abhängig.

Um die Hypothese für jede der 35 Faktor-Programm-Kombinationen zu testen, wurde die einfaktorielle Varianzanalyse (VA) angewendet. Bei der Überprüfung der Testvoraussetzungen ergab sich jedoch, daß weder die Annahme der Normalverteilung noch die der Varianzhomogenität erfüllt ist. Deshalb wurden zusätzlich parameterfreie Tests durchgeführt. Der Einfluß der Faktoren mit nur zwei Ausprägungen wurde mit dem U-Test von Mann und Whitney (MW) überprüft; für alle Faktoren mit drei Ausprägungen wurde der H-Test nach Kruskal und Wallis (KW) verwendet. Die Signifikanzniveaus der Varianzanalyse und der parameterfreien Tests sind in *Tabelle 7* angegeben. Grau hinterlegte Felder symbolisieren einen signifikanten Einfluß für die jeweilige Faktor-Programm-Kombination bei einer Irrtumswahrscheinlichkeit von $\alpha \leq 5\%$. Bis auf eine Ausnahme (der Einfluß der Faktors R auf das PM-Programm CA-SuperProject) zeigen die Varianzanalyse und die parameterfreien Tests für die gleichen Faktor-Programm-Kombinationen signifikante Ergebnisse. Im einzelnen lassen sich die folgenden Aussagen treffen: Die Qualität der Planungsergebnisse aller Programme wird signifikant von der Anzahl der nachgefragten Ressourcen (RF) beeinflusst. Gleiches gilt mit Ausnahme von Time Line für die Knappheit der Ressourcen (RS). Die Zahl

der berücksichtigten Ressourcen (R) führt nur bei ca. der Hälfte der Programme, nämlich bei Artemis Schedule Publisher, Project Scheduler 6 und Time Line, zu einer signifikanten Änderung der Projektdauerverlängerung. Das gleiche gilt für die Anzahl der Vorgänge (V). Diesmal wird die Qualität der Planungsergebnisse der Programme CA-SuperProject, Microsoft Project, Project Scheduler 6 und Time Line signifikant beeinflusst. Der Grad der technologischen Verflechtung NK ist der einzige Faktor, der die Planungsgüte keines Programms signifikant beeinflusst.

Tabelle 7: Signifikanzniveaus für alle Faktor-Programm-Kombinationen

Faktor Test	R		V		NK		RF		RS	
	KW	VA	KW	VA	MW	VA	MW	VA	KW	VA
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

4.5 Programmvergleich

Wie aus Tabelle 6 hervorgeht, erzielen die sieben PM-Programme $i=1,...,7$ für die 160 Testdatensätze unterschiedlich mittlere Projektdauerverlängerungen Π_i . Um zu überprüfen, ob generell von einer unterschiedlichen Planungsgüte der PM-Programme ausgegangen werden kann, wurde mit dem parameterfreien Friedman-Test die Hypothese

H_0 : Die Güte der von den sieben PM-Programmen erzeugten Kapazitätsplanungsergebnisse ist identisch.

überprüft. Mit einer Irrtumswahrscheinlichkeit von 0,00 % konnte die Nullhypothese abgelehnt und damit die Alternativhypothese angenommen werden. Mit dem ebenfalls parameterfreien Wilcoxon-Test wurde anschließend die Frage untersucht, welche der sieben Programme sich in der Güte der Kapazitätsplanung signifikant unterscheiden. Die Hypothese lautet:

H_0 : Die Güte der von der h -ten und der i -ten PM-Software erzeugten Kapazitätsplanungsergebnisse unterscheidet sich nicht.

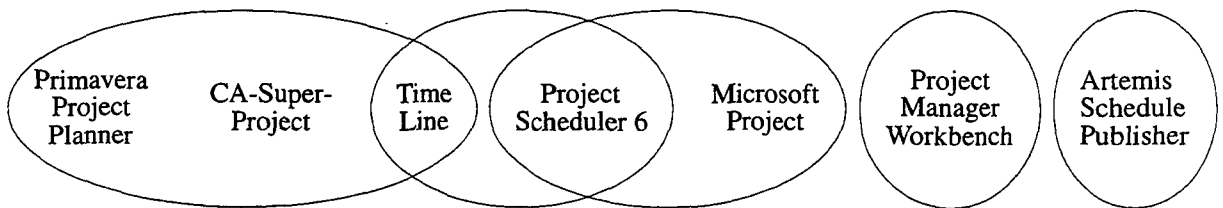
In Tabelle 8 sind die Ergebnisse der 21 Tests aufgeführt. Grau hinterlegte Felder symbolisieren, daß sich die zwei betrachteten Programme bei einer Irrtumswahrscheinlichkeit von $\alpha \leq 5\%$ unterscheiden. Für nur fünf der 21 Programmpaare konnte kein signifikanter

Unterschied festgestellt werden. *Abbildung 5* faßt die Ergebnisse zusammen, indem PM-Programme, die sich nicht signifikant unterscheiden, jeweils innerhalb eines Kreises dargestellt werden. Es ergeben sich fünf Planungsergebnis-Klassen; innerhalb einer Klasse besteht kein signifikanter Unterschied zwischen den PM-Programmen; zwischen zwei Programmen aus unterschiedlichen Klassen besteht jeweils ein signifikanter Unterschied.

Tabelle 8: Signifikanzniveaus der Wilcoxon-Tests für alle Programm-Kombinationen

	CA-Super- Project	Microsoft Project	Primavera Project Planner	Project Manager Workbench	Project Scheduler 6	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 %

Abbildung 5: Planungsergebnis-Klassen



5. Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde die methodische Fundierung der Kapazitätsplanung von PM-Standardsoftware auf der Basis einer experimentellen Studie verglichen. Die Ergebnisse bisheriger Untersuchungen (signifikant unterschiedliche Planungsgüte der Programme, signifikanter Einfluß der Ressourcenknappheit auf die Planungsgüte) wurden bestätigt. Darüber hinaus konnte ein signifikanter Einfluß weiterer Faktoren (Anzahl der Vorgänge, Anzahl der Ressourcen, Ressourcenfaktor) nachgewiesen werden.

Insgesamt zeigen die Ergebnisse, daß die Hersteller von PM-Standardsoftware bei der methodischen Entwicklung der Kapazitätsplanung wesentliche Fortschritte erzielt haben. Während noch in den 80-er Jahren die Funktion einer ressourcentreuen Projektablaufplanung von kaum einer PM-Standardsoftware abgedeckt wurde, ist sie heute fester Bestandteil der meisten PM-Programme. Die Güte der Planung - gemessen an der Verlängerung der kürzest-

möglichen Projektdauer - hat sich seit der Untersuchung von *Johnson* (1992) von einem Mittelwert von 9,88% auf einen Mittelwert von 5,79% verbessert.

Dennoch verbleibt eine große Differenz zu den Ergebnissen der in *Abschnitt 2.2.2* skizzierten derzeit besten Lösungsverfahren. Hier ist also noch ein erheblicher Spielraum für weitere Produktverbesserungen und den damit erreichbaren Wettbewerbsvorteilen für Hersteller und Nutzer von PM-Standardsoftware gegeben. Gleiches gilt für die nicht explizit in dieser Arbeit angesprochenen Funktionen wie z.B. der Zeit- und Kapazitätsplanung unter Berücksichtigung von Höchstabständen,³⁰ der Ressourcenbedarfsnivellierung³¹ und der Kostenplanung.³²

Aus wissenschaftlicher Sicht ergeben sich aus der vorliegenden Untersuchung zwei interessante Anknüpfungspunkte:

- Anhand einer empirischen Untersuchung sollten die Anforderungen von Projektmanagern an PM-Standardsoftware erhoben werden. Einen Ansatz hierzu bietet die Untersuchung von *Icmeli* (1995).
- Auf dieser Grundlage könnte dann eine gezielte Entwicklung von Modellen und Methoden sowie eine über den Rahmen dieser Arbeit hinausgehende Evaluierung der in kommerzieller PM-Software eingesetzten Verfahren durchgeführt werden.

Literatur

- Addelman, S. (1962): Orthogonal main-effect plans for asymmetrical factorial experiments, *Technometrics*, Vol. 4, S. 21-58.
- Altrogge, G. (1994): Netzplantechnik, 2. Auflage, Oldenbourg, München.
- 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.
- Bandelloni, M., M. Tucci und R. Rinaldi (1994): Optimal resource leveling using non-serial dynamic programming, *European Journal of Operational Research*, Vol. 78, S. 162-177.
- 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.
- Brinkmann, K. und K. Neumann (1994): Heuristic procedures for resource-constrained project scheduling with minimal and maximal time lags: The minimum project duration and the resource-levelling problem, Arbeitspapier, Universität Karlsruhe.
- Brockhoff, K. und C. Zanger (1993): Meßprobleme des Neuheitsgrades - dargestellt am Beispiel von Software, *Zeitschrift für betriebswirtschaftliche Forschung*, Jg. 45, S. 835-851.
- Davis, E.W. (1966): Resource allocation in project network models - A survey, *The Journal of Industrial Engineering*, Vol. 17, S. 177-188.
- Davis, E. W. (1973): Project scheduling under resource constraints - Historical review and categorization of procedures, *AIIE Transactions*, Vol. 5, S. 297-313.
- 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.

³⁰ Vgl. *Zhan* (1994).

³¹ Vgl. *Bandelloni et al.* (1994) und *Brinkmann/Neumann* (1994).

³² Vgl. *Nandi/Dutta* (1988).

- 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.
- Domschke, W. und A. Drexl (1991a): Einführung in Operations Research, 2. Auflage, Springer, Berlin.
- Domschke, W. und A. Drexl (1991b): Kapazitätsplanung in Netzwerken - Ein Überblick über neuere Modelle und Verfahren, *OR Spektrum*, Jg. 13, S. 83-76.
- Drexl, A. und R. Kolisch (1993): Produktionsplanung und -steuerung bei Einzel- und Kleinserienfertigung, *Wirtschaftswissenschaftliches Studium*, Jg. 22, Heft 2, S. 60-66.
- Dworatschek, S. und A. Hayek (1992): Marktspiegel Projektmanagement-Software: Kriterienkatalog und Leistungsprofile, 3. Auflage, Verlag TÜV Rheinland, Köln.
- Green, P.E. (1974): On the design of choice experiments involving multifactor alternatives, *Journal of Consumer Research*, Vol. 1, S. 61-68.
- Günther, H.-O. und H. Tempelmeier (1995): Produktion und Logistik, 2. Auflage, Springer, Berlin.
- Haberfellner, R. (1992): Projektmanagement, in: Frese, E. et al. (Hrsg.), *Handwörterbuch der Organisation*, 3. Auflage, Poeschel, Stuttgart, Sp. 2090-2102.
- Herroelen, W. (1972): Resource-constrained project scheduling - The state of the art, *Operational Research Quarterly*, Vol. 23, S. 261-275.
- Icmeli, O. (1995): Analysis of the characteristics of projects in diverse industries, Arbeitspapier, College of Business Administration, Cleveland State University.
- 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. (1995): Project scheduling under resource constraints - Efficient heuristics for several problem classes, *Physica*, Heidelberg.
- Kolisch, R., A. Sprecher und A. Drexl (1995): Characterization and generation of a general class of resource-constrained project scheduling problems, erscheint in: *Management Science*, Vol. 41, Nr. 9.
- Krüger, W. (1993): Projektmanagement, in: Wittmann, W. et al. (Hrsg.), *Handwörterbuch der Betriebswirtschaft*, Teilband 2, 5. Auflage, Poeschel, Stuttgart, Sp. 3559-3570.
- Leon, V.J. und R. Balakrishnan (1995): Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling, erscheint in *OR Spektrum*.
- Madaüss, B.-J. (1990): Handbuch Projektmanagement, 3. Auflage, Poeschel, Stuttgart.
- Maroto, C., S. Balasch, P. Tormos, C. Capilla und F. Crespo (1994): Project management software: Capabilities and efficiency in resource constrained scheduling, Arbeitspapier, 4-th International Workshop on Project Management and Scheduling, Leuven.
- Nandi, S.N. und B.K. Dutta (1988): Trends in project cost control systems, *OMEGA International Journal of Management Science*, Vol. 16, S. 513-520.
- Pascoe, T.L. (1966): Allocation of resources C.P.M., *Revue Francaise Recherche Operationelle*, Nr. 38, S. 31-38.
- 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.
- Patterson, J.H. und W.D. Huber (1974): A horizon-varying, zero-one approach to project scheduling, *Management Science*, Vol. 20, S. 990-998.
- Pritsker, A.A.B., L.J. Watters und P.M. Wolfe (1969): Multiproject scheduling with limited resources: A zero-one programming approach, *Management Science*, Vol. 16, S. 93-107.
- Schwarze, J. (1994): Netzplantechnik: Eine Einführung in das Projektmanagement, 7. Auflage, Verlag Neue Wirtschafts-Briefe, Herne.
- Sprecher, A., R. Kolisch und A. Drexl (1995): Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem, *European Journal of Operational Research*, Vol. 80, S. 94-102.
- Talbot, B. (1982): Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case, *Management Science*, Vol. 28, S. 1197-1210.
- Tobias, A. (1991): O.R. techniques for use in redesigning manufacturing and associated business systems, *European Journal of Operational Research*, Vol. 51, S. 168-178.
- Zhan, J. (1992): Calendarization of time planning in MPM Networks, *ZOR - Methods and Models of Operations Research*, Vol. 36, S. 423-438.
- Zhan, J. (1994): Heuristics for scheduling resource-constrained projects in MPM networks, *European Journal of Operational Research*, Vol. 76, S. 192-205 ff.
- Ziegler, H. (1985): Minimal and maximal floats in project networks, *Engineering Costs and Production Economics*, Vol. 9, S. 91-97.

Summary

This study assesses the quality of seven commercial project management packages to schedule precedence- and resource-constrained projects. We introduce the basic project scheduling model and give a short survey of optimal and heuristic solution methods. Furthermore, we outline extensions of the basic model which are implicitly employed in some of the packages. Since the implemented methods are proprietary, we measure the quality of the software in terms of the deviation from the optimal project length. 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.

Anhang: Einfluß der systematisch variierten Faktoren auf die Projektdauerverlängerung

Tabelle A1: Mittelwert und Standardabweichung der Projektdauerverlängerung in Abhängigkeit der Ausprägungen des Faktors V

V	Artemis Schedule Publisher		CA-Super- Project		Microsoft Project		Primavera Project Planner		Project Manager Workbench		Project Scheduler 6		Time Line	
	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$
10	7,79	9,11	1,83	4,04	2,36	4,30	3,67	6,49	5,21	7,87	2,17	4,34	2,60	4,44
20	10,73	10,60	4,85	6,34	6,08	6,52	4,59	6,02	6,88	9,01	6,10	7,80	4,52	5,05
30	9,79	8,72	6,11	6,30	6,91	7,48	4,73	5,68	7,79	8,43	7,34	10,06	6,30	5,19

Tabelle A2: Mittelwert und Standardabweichung der Projektdauerverlängerung in Abhängigkeit der Ausprägungen des Faktors R

R	Artemis Schedule Publisher		CA-Super- Project		Microsoft Project		Primavera Project Planner		Project Manager Workbench		Project Scheduler 6		Time Line	
	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$
1	6,84	7,10	2,93	4,65	3,02	4,05	3,01	4,63	4,45	6,67	2,52	4,42	2,75	4,03
2	12,08	11,25	5,17	5,74	5,82	6,57	4,99	6,38	6,67	7,69	6,48	8,38	4,86	5,19
3	8,04	7,94	4,37	7,46	6,74	7,85	4,59	6,48	8,97	11,24	6,23	9,26	5,47	5,50

Tabelle A3: Mittelwert und Standardabweichung der Projektdauerverlängerung in Abhängigkeit der Ausprägungen des Faktors RS

RS	Artemis Schedule Publisher		CA-Super- Project		Microsoft Project		Primavera Project Planner		Project Manager Workbench		Project Scheduler 6		Time Line	
	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$
0,2	14,71	10,49	8,25	8,12	9,76	8,74	8,31	7,13	12,40	11,17	11,67	11,06	5,86	6,29
0,5	8,63	9,87	3,83	4,96	4,43	5,14	3,76	5,60	5,54	7,30	4,15	5,61	4,25	5,04
0,7	7,07	7,09	1,73	2,94	2,79	3,89	1,75	3,26	3,27	4,49	1,75	4,00	3,59	3,41

Tabelle A4: Mittelwert und Standardabweichung der Projektdauerverlängerung in Abhängigkeit der Ausprägungen des Faktors RF

RF	Artemis Schedule Publisher		CA-Super- Project		Microsoft Project		Primavera Project Planner		Project Manager Workbench		Project Scheduler 6		Time Line	
	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$
0,5	8,00	9,50	2,60	4,49	2,89	4,08	3,44	6,04	4,85	6,60	2,29	4,31	3,53	4,84
1	11,52	9,87	6,22	6,79	7,82	7,52	5,35	5,92	8,53	9,91	8,57	9,45	5,44	5,18

Tabelle A5: Mittelwert und Standardabweichung der Projektdauerverlängerung in Abhängigkeit der Ausprägungen des Faktors NK

NK	Artemis Schedule Publisher		CA-Super- Project		Microsoft Project		Primavera Project Planner		Project Manager Workbench		Project Scheduler 6		Time Line	
	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$	$\mu(\Pi_i)$	$\sigma(\Pi_i)$
1,5	9,87	10,68	4,98	6,16	5,83	6,65	4,80	5,85	7,01	9,43	5,56	7,77	4,55	5,24
2	9,65	8,93	3,84	5,85	4,88	6,39	3,98	6,23	6,36	7,71	5,29	8,21	4,42	4,96