

Sprecher, Arno

Working Paper — Digitized Version

Non-equivalent search strategies for resource-constrained project scheduling

Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 493

Provided in Cooperation with:

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

Suggested Citation: Sprecher, Arno (1998) : Non-equivalent search strategies for resource-constrained project scheduling, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, No. 493, Universität Kiel, Institut für Betriebswirtschaftslehre, Kiel

This Version is available at:

<https://hdl.handle.net/10419/147586>

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. 493

Non-Equivalent Search Strategies for
Resource-Constrained Project Scheduling ¹

Arno Sprecher



Manuskripte
aus den
Instituten für Betriebswirtschaftslehre
der Universität Kiel

No. 493

Non-Equivalent Search Strategies for
Resource-Constrained Project Scheduling¹

Arno Sprecher

May 1998, published November 1998

©Do not copy, publish or distribute without authors' permission.

Arno Sprecher, Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, Olshausenstraße 40,
24098 Kiel, Germany.

Email: Sprecher@bwl.uni-kiel.de

WWW : <http://www.wiso.uni-kiel.de/bwlinstitute/prod>

FTP : <ftp://www.wiso.uni-kiel.de/pub/operations-research>

¹Supported by the Deutsche Forschungsgemeinschaft

Abstract

Over the years numerous *branch-and-bound procedures for solving the resource-constrained project scheduling problem* (RCPSP) have been developed. Enumerating delaying alternatives, extension alternatives, feasible posets, feasible sequences, feasible completion times or feasible subsets, they all aim at finding as fast as possible a makespan minimal schedule among the resource and precedence feasible ones.

Some of the enumeration schemes have been modified to solve variants of the so-called resource-constrained project scheduling problem, like the resource-constrained project scheduling problem with multiple modes or with work content defined modes. We compare the enumeration of delaying alternatives and the enumeration of extension alternatives. Roughly considered the concepts that analyze only minimal delaying alternatives and the concept that analyze only maximal extension alternatives seem to be equivalent. Counterexamples will show that – in contrast to claims made in the literature – search tree reduction to minimal delaying alternatives and search tree reduction to maximal extension alternatives are not equivalent. While the former reduction preserves optimality the latter one is neither correct for the RCPSP with single execution modes nor for the RCPSP with work content defined modes.

Keywords: Project Scheduling, Resource Constraints, Branch-and-Bound, Delaying Alternatives, Extension Alternatives.

1 Introduction

As a generalization of the flow-shop, job-shop, and open-shop problem the resource-constrained project scheduling problem (RCPSP) is known as an NP-hard problem (cf. [6]). Therefore the main focus is on the development of branch-and-bound algorithms where different ideas have been presented to build the tree guiding the enumeration of the schedules. The schemes enumerate delaying alternatives (cf. [3], [4]), feasible completion times (cf. [19]), feasible extensions (cf. [18]), feasible posets (cf. [13]), feasible sequences (cf. [12], [14], [15]) and feasible subsets (cf. [11]), in order to find an optimal, i.e. makespan minimal, solution. The currently most studied procedure has been developed by Demeulemeester and Herroelen (cf. [3], [4]).

In the major part of business applications one can employ different resources and/or quantities to define alternative ways, i.e., modes, to execute the activities that comprise a project. The activity duration is a discrete function of the employed quantities. That is, the concept allows to model (a) acceleration of an activity by raising the quantities coming into operation (time-resource-tradeoff), and (b) resource substitution by raising the quantities of some resources and reducing the quantities of others (resource-resource-tradeoff). The problem derived is the multi-mode resource-constrained project scheduling problem (MRCPSP), which is commonly considered with makespan minimization as objective (cf. [20]). The discrete time/resource tradeoff problem (cf. [2]) considers only a single renewable resource and the discrete time/cost tradeoff problem (cf. [5]) only a single nonrenewable resource.

In this paper we will study search strategies developed to solve the single-mode resource-constrained project scheduling problem. First, the enumeration of delaying alternatives developed by Demeulemeester and Herroelen (cf. [4]). Second, the enumeration of extension alternatives developed by Stinson, Davis, and Kuhumawala (cf. [18]). Both approaches have been generalized to the MRCPSP (cf. [16], [7]). Roughly compared the related search tree reduction to minimal delaying alternatives and maximal extension alternatives seem to be both correct. We present counterexamples which show that – in contrast to claims made in the literature (cf. [2], [8], [9]) – search tree reduc-

tion to minimal delaying alternatives and maximal extension alternatives are not equivalent. The former reduction is feasible and preserves optimality and the latter one not.

We proceed as follows: In Section 2 we describe the problem setting. In Section 3 we summarize the search strategies and the search tree reduction. In Section 4 examples for the RCPSP with a single mode and the RCPSP with work content defined modes show that the reduction to minimal delaying alternatives is not equivalent to the reduction to maximal extension alternatives.

2 The Model

We consider a project which consists of J activities (jobs, tasks). Due to technological requirements, precedence relations between some of the activities enforce that an activity j , $1 = 2, \dots, J$, may not be started before all its predecessors h , $h \in \mathcal{P}_j$, are finished. The sets of successors of an activity j are referred to by \mathcal{S}_j , $j = 1, \dots, J$. The structure of the project is depicted by a so-called activity-on-node (AON) network where the nodes and the arcs represent the activities and precedence relations, respectively. The network is acyclic and numerically labeled, that is an activity j has always a higher number than all its predecessors. Without loss of generality activity 1 is the only start activity (source) and activity J is the only finish activity (sink).

The execution of the activities uses certain amounts of renewable resources from a set R and consumes certain amounts of nonrenewable resources from a set N (cf. [21], [22]). The renewable resources r , $r \in R$, have an availability of K_r units per period and the nonrenewable resources r , $r \in N$, have an availability of K_r units for the entire project. Combining different levels of usage and consumption the activities can be performed in different modes. If activity j , $j = 1, \dots, J$, is performed in mode m , $m = 1, \dots, M_j$, it takes d_{jm} periods, uses k_{jmr} units of renewable resource r , $r \in R$, each period activity j is in process, and consumes k_{jmr} units of nonrenewable resource r , $r \in N$. The activities may not be preempted, i.e., an activity once started in a certain mode has to be completed without interruption and change of mode. The objective is to find a makespan minimal schedule that meets the constraints imposed by the precedence relations, the limited per-period availability of the renewable resources and the limited overall availability of the nonrenewable resources. The problem obtained is the multi-mode resource-constrained project scheduling problem (MRCPSP). The MRCPSP contains the single-mode resource-constrained project scheduling problem (SRCPSP) where the activities have to be performed in a single pre-specified mode.

Demeulemeester et al. ([2]) introduced a special case of the MRCPSP. The problem considers a single renewable resource with a per-period availability of K units. The modes are defined by the so-called work contents W_j of the activities j , $j = 1, \dots, J$. The work content specifies the total number of units of the renewable resource that are used during the execution of activity j . The modes are defined to reflect time-resource trade-offs. An activity j performed in mode m has a duration of d_{jm} periods and uses k_{jm1} units of the renewable resource with (*) $d_{jm} \cdot k_{jm1} \geq W_j$. From the modes fulfilling (*) only the efficient ones are considered. That is, for two different modes m and m' , it is (a) $d_{jm} > d_{jm'}$ and $k_{jm1} < k_{jm'1}$ or (b) $d_{jm} < d_{jm'}$ and $k_{jm1} > k_{jm'1}$. The sub-problem of the MRCPSP is the discrete time/resource tradeoff problem (DTRTP).

3 Delaying and Extension Alternatives

In this section we will briefly summarize the enumeration schemes that rely on delaying alternatives and extension alternatives proposed for solving the RCPSP.

Stinson et al. (cf. [18]) provided a branch-and-bound algorithm that enumerates the extension alternatives (cf. [7]) of partial schedules. The partial schedules reflect scheduling decisions already made for a subset of the set of activities. The partial schedules are always feasible with respect to precedence and resource constraints. Given a partial schedule, the next decision point is determined by the time incrementing scheme introduced by Johnson (cf. [10], quoted in accordance with [18]). That is, assuming constant resource availability, it is sufficient to study the completion times of the activities already scheduled as candidates for start times of new activities. Consequently, the decision point can be determined as the minimal – unconsidered – completion time of the activities contained in the partial schedule. At the decision point it is tried to extend the partial schedule by activities that are eligible, i.e., by activities all the predecessors of which are in the partial schedule and finish at or before the decision point. From the set of eligible activities the subset of schedulable activities is built. The schedulable activities are the ones that can be selected on their own to be started at the decision point without violating the resource constraints. The extension alternatives are built by all the subsets of the schedulable activities that can be started simultaneously at the current decision point to extend the partial schedule without violating the resource constraints. They form the descendants of the current node of the branch-and-bound tree. The scheme starts with the empty partial schedule and successively extends the current partial schedule by determining the next decision point and selecting an extension alternative. Backtracking is performed if all the alternatives are evaluated at the current node of the branch and bound tree. The basic scheme is enhanced by dominance pruning as initially developed by Johnson (cf. [10], quoted in accordance with [18]). Moreover, (i) a precedence-based bound relying on MPM calculations of latest finish times of the activities for a given project duration, and (ii) a resource-based bound are used. Combining the effects of precedence and resource constraints Stinson et al. propose the critical sequence lower bound. Finally, left-shifts are studied to detect further dominance.

Demeulemeester and Herroelen (cf. [3]) presented an enumeration scheme relying on the idea of resolving resource conflicts by delaying some of the activities causing the conflict (cf. [1]). The procedure continues a given partial schedule by temporarily scheduling all the eligible activities at the decision point which is determined by Johnson (cf. [10]). If the cumulated resource requests at the decision point exceed the availability then it is branched to the next level. At this level the delaying alternatives are considered to resolve the conflict. Thereby, a delaying alternative is a subset of the set of activities in process at the decision point, the delay of which makes the partial schedule resource feasible. If no resource conflict occurs then the next decision point is determined. The algorithm tracks back if at the current level all the delaying alternatives are evaluated. Demeulemeester and Herroelen prove that it is sufficient to study only the minimal delaying alternatives, i.e., minimal subsets the delay of which resolves the resource conflict. Moreover, the basic algorithm is enhanced by the precedence-based bound and the critical sequence bound as well as additional dominance concepts. The concepts make use of left-shift dominance and, moreover, a cut-set rule similar to the dominance pruning used by Stinson et al. and the network-cuts as employed by Talbot and Patterson (cf. [19]). Finally, Demeulemeester and Herroelen introduce two immediate selection strategies.

Hartmann and Drexl ([7]) generalized the concept of extension alternatives to the multi-mode case. At a decision point determined as in the single-mode algorithm, for all the activities that become eligible, a so-called mode alternative is selected. A mode alternative assigns a mode to each of the new eligible activities. From the set of eligible activities the set of schedulable activities is built in accordance with the modes already assigned. The extension alternatives are then built by those subsets of the set of schedulable activities that can simultaneously be started – in the modes previously assigned – at the current decision point without causing a resource conflict. An extension alternative is selected and the next decision point is determined. At a certain level, the algorithm selects the next mode alternative if all the extension alternatives have been evaluated, and the algorithm tracks back if all the mode alternatives have been tested. The basic algorithm is enhanced by preprocessing rules and dominance concepts.

Sprecher et al. (cf. [16]) generalized the concept of delaying alternatives to the multi-mode case. At a decision point determined as in the single-mode algorithm, for all the activities that become eligible, a mode alternative – as defined above – is selected. Subsequent foregoing is as in the single-mode case, i.e. all the eligible activities, that is, the ones which have been delayed on a previous level and the ones that have become eligible at the current decision point, are put into process at the current decision point in the modes previously assigned. If a resource conflict occurs then a minimal subset of the set of activities in process at the decision point is delayed in order to solve the resource conflict. The next decision point is determined. At a certain level, the algorithm selects the next mode alternative if all the minimal delaying alternatives have been evaluated, and the algorithm tracks back if all the mode alternatives have been tested. The basic scheme is accelerated by preprocessing rules and dominance concepts.

In the following section we will present examples illustrating that in contrast to the delaying concept where only minimal delaying alternatives have to be studied, the extension concept cannot be reduced to maximal extension alternatives.

4 Counterexamples

4.1 Maximal Extensions and the Single-Mode Case

We consider the example of Figure 1. The project consists of six activities. The usage of the single renewable resource is limited to $K = 4$ units per period. A lower bound of the makespan is given by the length of the critical path, i.e., 12 periods. The optimal solution is obtained by defining the start times $ST_1 = 0$, $ST_2 = 0$, $ST_3 = 3$, $ST_4 = ST_5 = 6$, and $ST_6 = 12$. The completion times are $CT_1 = 0$, $CT_2 = 3$, $CT_3 = 6$, $CT_4 = CT_5 = 12$, and $CT_6 = 12$. The optimal solution is unique.

The branch-and-bound tree related to the concept of maximal extensions alternatives consists of a single path. We denote a partial schedule by PS , and the maximal completion time of its activities by $CT(PS)$. As illustrated by the first Gantt chart of Figure 2, at $t = 0$, after scheduling the source activity 1, the only maximal extension alternative \mathcal{ME} is given by $\mathcal{ME} = \{2, 5\}$. At the next decision point $t = 3 = CT_2$ no activity can be added to the partial schedule without violating the resource or precedence constraints, that is, it is $\mathcal{ME} = \{\}$. The next decision point is $t = 6 = CT_5$. Only activity 3 is eligible at $t = 6$, and we obtain $\mathcal{ME} = \{3\}$ as the only maximal

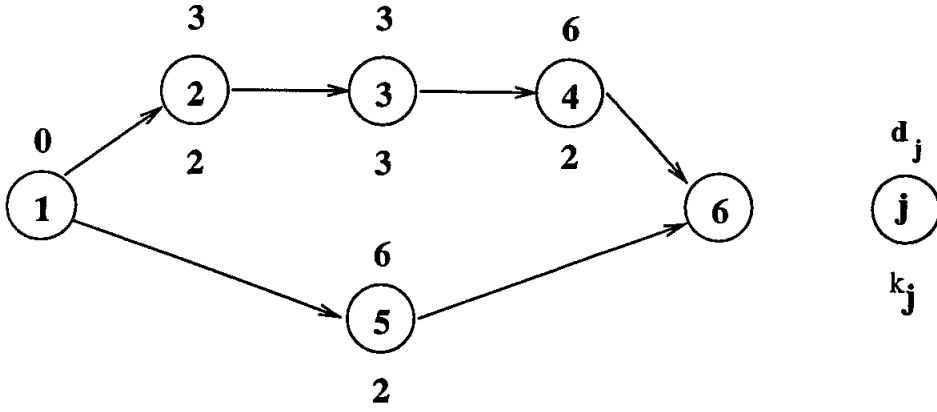


Figure 1: Maximal Extensions and the Single-Mode Case

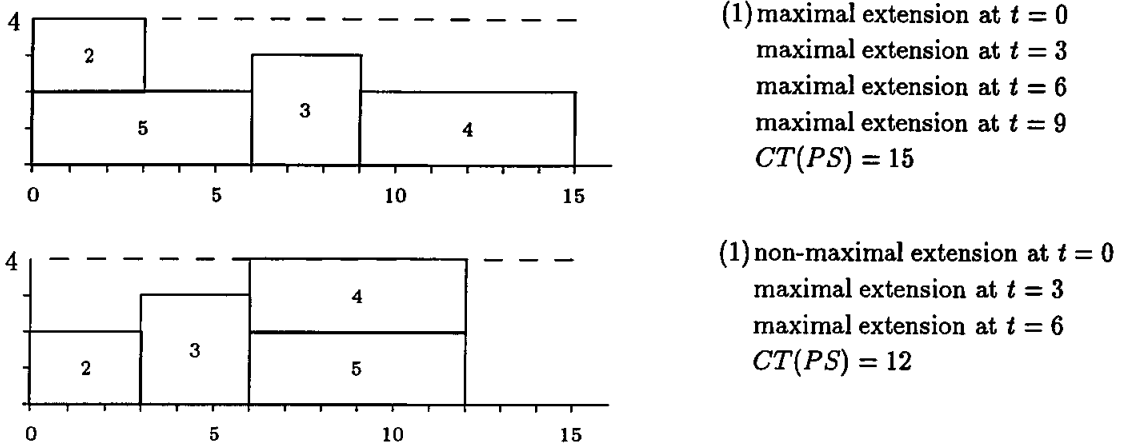


Figure 2: Partial Schedules of Project Instance - Single-Mode Case

extension alternative. The alternative is selected and the next decision point $t = 9$ is determined. At $t = 9$, again, we obtain a single alternative to be selected, i.e., $\mathcal{ME} = \{4\}$. Selecting the alternative and subsequently scheduling the sink activity 6, produces a schedule with $CT(PS) = 15$ periods. The schedule is not optimal. As illustrated by the second Gantt chart of Figure 2 the optimal schedule contains non-maximal extensions. Scheduling the source activity 1, selecting at $t = 0$ the non-maximal extension alternative $\mathcal{NME} = \{2\}$, at $t = 3$ the maximal extension alternative $\mathcal{ME} = \{3\}$, at $t = 6$ the maximal extension alternative $\mathcal{ME} = \{4, 5\}$, and subsequently scheduling the sink activity 6, produces a schedule with $CT(PS) = 12$ periods.

Employing minimal delaying alternatives the optimal solution is obtained by selecting minimal delaying alternative $\mathcal{MD} = \{5\}$ at $t = 0$, and $\mathcal{MD} = \{5\}$ at $t = 3$.

4.2 Maximal Extensions and Work Content Defined Modes

We consider an example. The project network and the work content is defined in Figure 3. We assume a constant resource availability of $K = 3$ units per period.

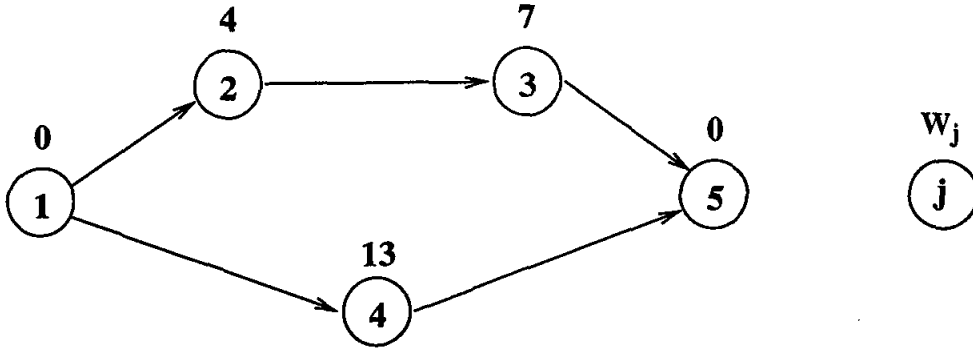


Figure 3: Example Network – Maximal Extensions and Work Content Defined Modes

The efficient modes that can be generated with respect to the specified work contents are listed in Table 1. The dummy activities 1 and 5 have only one mode. The activities 2, 3, and 4 have 2, 3, and 3 feasible modes, respectively.

Job	Workload	Modes					
j	W_j	d_{j1}	k_{j1}	d_{j2}	k_{j2}	d_{j3}	k_{j3}
1	0	0	0	-	-	-	-
2	4	2	2	4	1	-	-
3	7	3	3	4	2	7	1
4	13	5	3	7	2	13	1
5	0	0	0	-	-	-	-

Table 1: Efficient Modes Defined by Workloads – $K = 3$

If all the activities are processed in their highest indexed mode then the precedence-based earliest start schedule is resource-feasible with a makespan of 13 periods. Applying the resource-based bound to the project instance we obtain a lower bound on the makespan of $\lceil (\sum_{j=1}^n W_j) / K \rceil = \lceil (4+7+13) / 3 \rceil = 8$ periods. To reduce the length of the earliest start schedule activity 4 has to be processed in a lower indexed mode. Doing so increases the resource-based bound to 9 periods. Consequently a schedule has a makespan of at least 9 periods.

We analyze the enumeration of the maximal extension alternatives at the decision point $t = 0$ after activity 1 is scheduled. The precedence feasible extension alternatives are listed in Table 2. The extensions are denoted by (pairs of) activity/mode combinations $[j, m]$.

	[2,1]	[2,2]	[4,1]	[4,2]	[4,3]	[2,1] [4,1]	[2,2] [4,1]	[2,1] [4,2]	[2,2] [4,2]	[2,1] [4,3]	[2,2] [4,3]
feasible	yes	yes	yes	yes	yes	no	no	no	yes	yes	yes
maximal	no	no	yes	no	no	-	-	-	yes	yes	yes

Table 2: Precedence Feasible Extensions at $t = 0$

The Gantt charts related to the feasible maximal extensions are illustrated in Figure 4. The first and second chart depict the maximal extension $\mathcal{ME} = \{[4, 1]\}$ at $t = 0$. If the partial schedule is continued at $t = 5 = CT_4$ by the maximal extension alternative $\mathcal{ME} = \{[2, 1], [3, 3]\}$ then the continuation has a makespan of $CT(PS) = 12$ periods as displayed in the first chart. If the partial schedule is continued at $t = 5$ by the non-maximal extension $\mathcal{NME} = \{[2, 1]\}$ and afterwards, at $t = 7$ by the

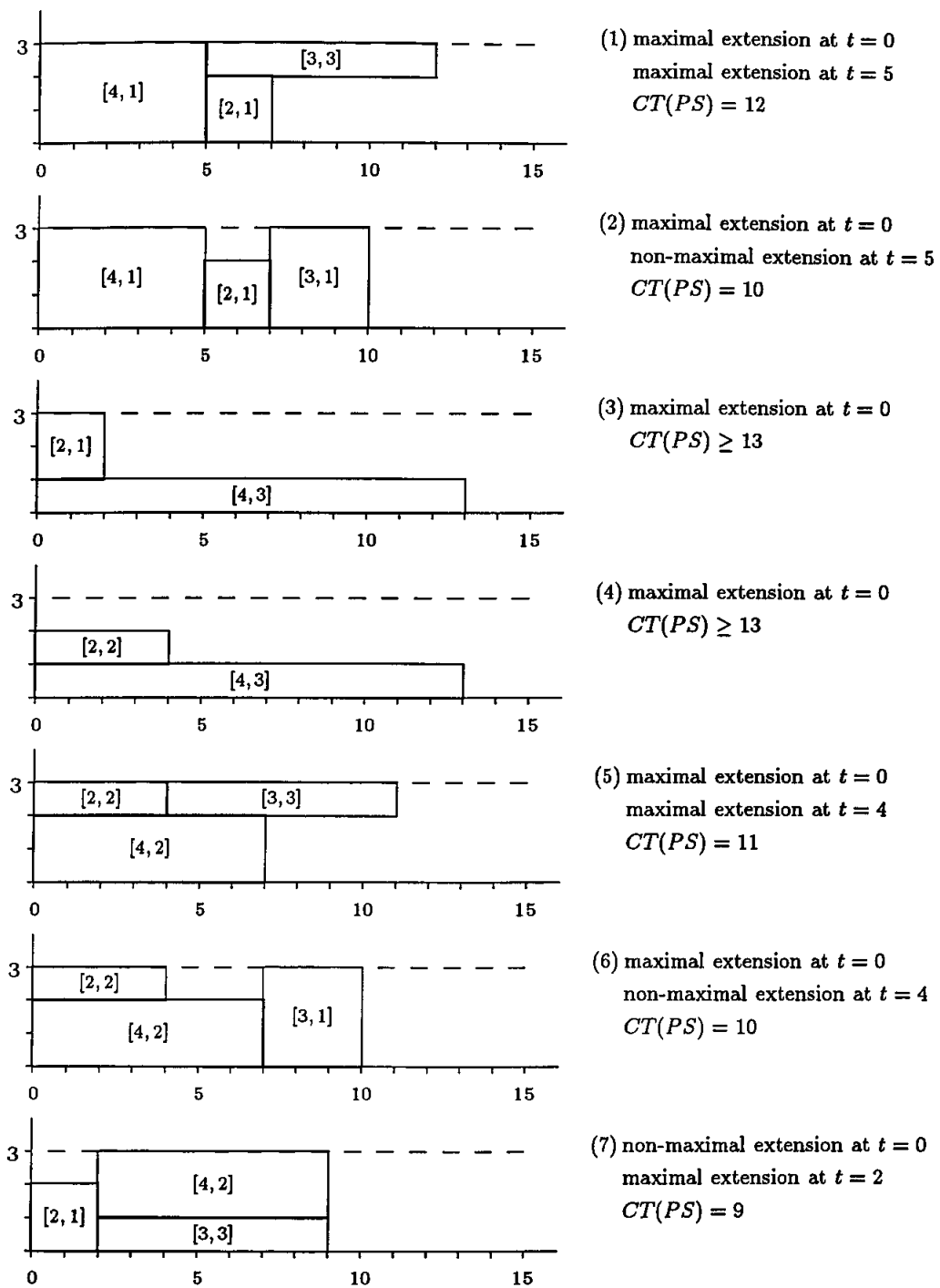


Figure 4: Partial Schedules of Project Instance - Work Content Defined Modes

maximal extension alternative $\mathcal{ME} = \{[3, 1]\}$ then $CT(PS) = 10$ is obtained. The third and the fourth chart consider the maximal extension $\mathcal{ME} = \{[2, 1], [4, 3]\}$ and $\mathcal{ME} = \{[2, 2], [4, 3]\}$ at $t = 0$, respectively. In both cases a continuation of the partial schedule has a makespan of $CT(PS) \geq 13$. The fifth and sixth chart consider continuations of the maximal extension $\mathcal{ME} = \{[2, 2], [4, 2]\}$ at $t = 0$. If the partial schedule is continued at $t = 4$ by the maximal extension $\mathcal{ME} = \{[3, 3]\}$ then the continuation has a makespan of $CT(PS) = 11$. If the partial schedule is continued at $t = 4$ by the empty non-maximal extension $\mathcal{NME} = \emptyset$ and afterwards at $t = 7$ by maximal extension $\mathcal{ME} = \{[3, 1]\}$ then $CT(PS) = 10$ is obtained. The

seventh chart shows the optimal solution with $CT(PS) = 9$. It is obtained by the non-maximal extension $\mathcal{NME} = \{[2, 1]\}$ at $t = 0$ and the maximal extension $\mathcal{ME} = \{[3, 3], [4, 2]\}$ at $t = 2$.

Using the concept of mode alternatives and minimal delaying alternatives we select mode alternative $\mathcal{MA} = \{[2, 1], [4, 2]\}$ and minimal delaying alternative $\mathcal{MD} = \{4\}$ at $t = 0$ and mode alternative $\mathcal{MA} = \{[3, 3]\}$ at $t = 3$ to obtain the optimal solution. We have presented examples to illustrate that the concepts enumerating minimal delaying alternatives and maximal extensions to determine makespan minimal schedules are not equivalent. The former strategy is optimal the latter one not.

References

- [1] CHRISTOFIDES, N.; R. ALVAREZ-VALDES AND J.M. TAMARIT (1987): Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, Vol. 29, pp. 262-273.
- [2] DEMEULEMEESTER, E.; B. DE REYCK AND W. HERROELEN (1997): The discrete time/resource trade-off problem in project networks – A branch-and-bound approach. Research Report, Department of Applied Economics, University Leuven, Belgium.
- [3] DEMEULEMEESTER, E. AND W. HERROELEN (1992): A branch-and-bound procedure for the multiple resource-constrained project scheduling problem. *Management Science*, Vol. 38, pp. 1803-1818.
- [4] DEMEULEMEESTER, E. AND W. HERROELEN (1997): New benchmark results for the resource-constrained project scheduling problem. *Management Science*, Vol. 43, pp. 1485-1492.
- [5] DEMEULEMEESTER, E.; W. HERROELEN AND S. ELMAGHRABY (1996): Optimal procedures for the discrete time/cost trade-off problem in project networks. *European Journal of Operational Research*, Vol. 88, pp. 50-68.
- [6] GAREY, M.R. AND D.S. JOHNSON (1979): *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, CA.
- [7] HARTMANN, S. AND A. DREXL (1998): Project scheduling with multiple modes: A comparison of exact algorithms. *Networks*, to appear.
- [8] HERROELEN, W.; E. DEMEULEMEESTER AND B. DE REYCK (1996): Resource-constrained project scheduling: A survey of recent developments. In: Zimmermann, U.; U. Derigs; W. Gaul; R.H. Möhring and K.-P. Schuster (Eds.): *Operations Research Proceedings 1996*, pp. 197-202.
- [9] HERROELEN, W.; E. DEMEULEMEESTER AND B. DE REYCK (1996): Resource-constrained project scheduling: A survey of recent developments. Research Report, University Leuven, Belgium. *Computers & Operations Research*, to appear.
- [10] JOHNSON, T.J.R. (1967): An algorithm for the resource-constrained project scheduling problem. PhD Dissertation, Massachusetts Institute of Technology, USA.
- [11] MINGOZZI, A.; V. MANIEZZO; S. RICCIARDELLI AND L. BIANCO (1998): An exact algorithm for the resource constrained project scheduling problem based on a new mathematical formulation. *Management Science*, to appear.
- [12] PATTERSON, J.H.; R. SLOWINSKI; F.B. TALBOT AND J. WEGLARZ (1989): An algorithm for a general class of precedence and resource constrained scheduling problems. In: Slowinski, R. and J. Weglarz (Eds.): *Advances in project scheduling*. Elsevier, Amsterdam, pp. 3-28.
- [13] RADERMACHER, F.J. (1985/86): Scheduling of project networks. *Annals of Operations Research*, Vol. 4, pp. 227-252.
- [14] SPRECHER, A. (1997): Scheduling resource-constrained projects competitively at modest memory requirements. *Manuskripte aus den Instituten für Betriebswirtschaftslehre*, No. 425 – revised, Kiel, Germany.

- [15] SPRECHER, A. AND A. DREXL (1998): Solving multi-mode resource-constrained project scheduling problems by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, to appear.
- [16] SPRECHER, A.; S. HARTMANN AND A. DREXL (1997): An exact algorithm for project scheduling with multiple modes. *OR Spektrum*, Vol. 19, pp. 195-203.
- [17] SPRECHER, A.; R. KOLISCH AND A. DREXL (1995): Semi-active, active and non-delay schedules for the resource-constrained project scheduling problem. *European Journal of Operational Research*, Vol. 80 , pp. 94-102.
- [18] STINSON, J.P.; E.W. DAVIS AND B.M. KHUMAWALA (1978): Multiple resource-constrained scheduling using branch and bound. *AIIE Transactions*, Vol. 10, pp. 252-259.
- [19] TALBOT, F.B. AND J.H. PATTERSON (1978): An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems. *Management Science*, Vol. 24, pp. 1163-1174.
- [20] TALBOT, F.B. (1982): Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science*, Vol. 28, pp. 1197-1210.
- [21] WEGLARZ, J. (1979): Project scheduling with discrete and continuous resources. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, pp. 644-650.
- [22] WEGLARZ, J. (1980): On certain models of resource allocation problems. *Kybernetics*, Vol. 9, pp. 61-66.