,

An Integrated Perspective of Project Scheduling

Rainer Kolisch*
Institut für Betriebswirtschaftslehre
Christian-Albrechts-Universität, D-24098 Kiel, Germany
email: kolisch@bwl.uni-kiel.de

Rema Padman
The H. John Heinz III School of Public Policy and Management
Carnegie Mellon University, Pittsburgh, PA 15213, USA
email: rpadman@cmu.edu

July 3, 1997

1 Introduction

In recent years, several survey papers have summarized current and past research in project management and scheduling [41, 87, 42, 66, 96, 137, 90, 88]. These papers have primarily emphasized modeling and algorithmic developments in this area and bypassed research advances in the decision support area that facilitate use and deployment considerations. In this paper, we present an overview of deterministic project scheduling research that integrates methodologies, models, and data, that contribute toward building appropriate decision support capabilities in widely diverging project scheduling environments. While much of the early work in this area may already be summarized elsewhere, we briefly mention these for reasons of completeness and coherence. The particular emphasize here is on recent research results as well as the data generation and decision support issues. In addition, the unprecedented growth of the World Wide Web (WWW) has had a profound impact on the organizational use of information technology. We summarize the major research efforts, and present our thoughts on the use of this enabling technology in organizing and disseminating research results in the project management and scheduling domain for use by researchers and practitioners alike.

More specifically, we present a framework for integration that considers the multiple objectives that have been researched in the literature to evaluate project schedules, especially the objectives of minimizing makespan and maximizing net present value (NPV), highlighting the close relationship between their models and methods; problem representation issues, especially in the case of the NPV maximizing problem; new models and algorithmic procedures, both exact and heuristic, for solving project scheduling problems in constrained and unconstrained environments; extensions, relaxations, and modifications of the traditional problems and procedures; decision support and machine learning approaches in research and practice; benchmark data sets and their availability, advantages and limitations; and the promise of the WWW in providing the infrastructure and technologies for realizing this integration for the project scheduling problem. Multiproject scheduling is not included in this survey due to the length of the paper. We believe that this integrated perspective on deterministic project scheduling research will benefit both researchers and practitioners in identifying new areas of inquiry and application.

*This author has been supported by the German Science Foundation DFG.

The paper is organized as follows. Section 2 provides a brief description of the components of the project scheduling problem, and discusses representation issues, especially in the presence of cash flows and use of the net present value objective. Section 3 discusses the various randomly generated project scheduling problem data sets available in the literature to test exact algorithms and heuristic procedures. Section 4 provides an extensive summary of the methods for the makespan and NPV objectives, respectively, for unconstrained and resource–constrained projects. Section 5 looks at related problems such as the progress payment scheduling problem, problems with multiple execution modes and time lags. Section 6 highlights decision support approaches applied to this problem, including ongoing research efforts to use Internet– and Web–based technologies to support and extend the availability, reach, access, and use of the research described in the previous sections. Section 7 concludes with several directions for future research.

2 Problem Description and Representation

2.1 Elements of Project Scheduling Problems

Project scheduling problems (PSP) are made of activities, resources, precedence relations, and performance measures [171]. A classification of project scheduling models w.r.t. these four components has recently been proposed in Herroelen et al. [89]. In the following, each component is introduced. It is assumed that all data needed is available, deterministic, and integer valued.

2.1.1 Activities

A project consists of a number of activities, also known as jobs, operations, and tasks. In order to complete the project successfully, each activity has to be processed in one of several modes. Each mode represents a distinct way of performing the activity under consideration. The mode determines the duration of the activity, measured in number of periods, which indicates the time taken to complete the activity, the requirements for resources of various categories, as explained later in this section, and possible cash inflows or outflows occurring at the start, during processing, or at termination of the activities.

2.1.2 Precedence Relations

Often technological reasons imply that some activities have to be accomplished before others can start. This is handled by depicting the project as a directed graph where an activity is represented by a node and the precedence relation between two activities is represented by a directed arc. Other representation issues are discussed in Section 2.3.

2.1.3 Resources

Resources utilized by the activities are classified according to categories, types, and value [23]. The category classification includes resources that are renewable, nonrenewable, partially renewable and doubly constrained.

Renewable resources are constrained on a period basis only. That is, regardless of the project length, each renewable resource is available for every single period. Examples are machines, equipment, and manpower.

Nonrenewable resources are limited over the entire planning horizon, with no restrictions within each period. The classic example is the capital budget of a project.

Partially renewable resources, recently introduced by Böttcher et al. [29], limit utilization of resources within a subset of the planning horizon. An example is that of workers whose monthly working time, not the daily time, is limited by the working contract. It has been shown by [29] that both, renewable and nonrenewable resource types, can be depicted by partially renewable resources.

Doubly constrained resources are limited on a period basis as well as on a planning horizon basis. Budget constraints that limit capital availability for the entire project as well as limiting its consumption over each time period is an example of this type of resource. As formally shown by Talbot [186], each doubly constrained resource can be represented by one renewable and one nonrenewable resource, respectively. The type classification further distinguishes each category according to the function of the various resources. Finally, each resource type has a value associated with it, representing the amount of that resource needed to accomplish the specified activity.

Whenever there is at least one category of constrained resources, we term the resulting PSP a resource–constrained project scheduling problem (RCPSP).

2.2 Objectives of Project Scheduling

Makespan Minimization is probably the most researched and applied objective in the project scheduling domain. The makespan is defined as the time span between the start and the end of the project. Since the start of the project is usually assumed to be in $t = 0$, minimizing the makespan reduces to minimizing the maximum of the finish times of all activities. Makespan minimization is a *regular performance measure* [181]. When we have such a performance measure, we can compare two schedules for a given problem which differ only in the finish time of one activity and we can state that the schedule which has the smaller finish time for this activity is at least as good as the other schedule, i.e. the former dominates the latter.

Other regular performance measures. Other regular performance measures are the minimization of the (weighted) flow time of the activities or, if due dates are given, the minimization of the (weighted) delays [178, 170]. If only positive cash flows are considered, the maximization of the net present value, as treated below, is also a regular performance measure [89].

Maximize Net Present Value. When significant levels of cash flows are present in the project, in the form of expenses for initiating activities and progress payments for completion of parts of the project, the net present value (NPV) criterion is a more appropriate measure of project performance [22]. This criterion generates a cost–critical path and schedule of activities, in contrast to the well known time–critical path and schedule obtained by the makespan objective. Much of the research on the NPV project scheduling problem has concentrated on designing solution approaches for the resource–constrained extension, the RCPSP with cash flows where the problem is to maximize the NPV of the project subject to precedence and renewable resource constraints. The solution to the mathematical models provide both the optimal scheduled start time of each activity as well as the optimal project NPV. Other extensions include considerations of material and capital constraints and time–cost and multi–mode operational issues. Given the complex, combinatorial nature of these problems, optimal approaches, summarized later in this paper, have been successful only for small instances. This is due both to the difficulty in representing the problem in mathematical form, as well as the difficulty in solving the problem, once formulated.

Maximize Quality. Icmeli and Rom [99] were the first to explicitly consider the objective of quality within project scheduling. The stepping stone for modeling this objective was a nationwide empirical study in the U.S. performed by the same authors in [97]. One of the major results was the finding that maximizing the quality of the project is the most important objective of project managers. Hence, [99] presents a mixed–integer linear program (MILP) for project scheduling with this objective.

Minimize Costs. This objective has also attracted a lot of attention from the research community due to its practical significance. Cost-based objectives can be divided into (1) activity–cost and (2) resource–cost objectives.

In activity–cost objectives, the way activities are performed, i.e. the starting time and / or the chosen modes, results in *direct* costs which are minimized. Examples are the traditional, continuous time/cost trade-off problem [68] and its discrete extensions [56, 57, 64]. Recent work described in [4, 5] combines the multi–mode RCPSP and the time/cost-tradeoff problem to create the multi–mode RCPSP with crashable modes, where a given mode-duration can be reduced at some cost. The objective is the minimization of the project costs made up of the sum of activity costs and the penalty costs for not meeting a deadline.

In resource–cost objectives, the schedule of the activities does influence the cost *indirectly* via the resources. Examples are the classical resource leveling problem where the deviation between the resource requirements and the stated desirable levels are minimized [15, 31], and the resource investment problem [58, 127] where the cost associated with the constant level of resource capacity made available for the project is minimized.

2.3 Representation Issues

In general, two representations, activity–on–arc (AOA) and activity-on–node (AON), have been commonly used to capture project networks, resulting in an event–based or activity–based representation, respectively. In the AOA representation, nodes represent events and arcs represent activities. Dummy activities are used to preserve the precedence relations and dummy nodes capture the start and completion of the project. In the AON representation, activities and their associated parameter information are represented within the nodes and the precedence relations are represented by directed arcs. In contrast to Gantt–charts, both the AOA and AON representation offer a graphical depiction of the inter-relationships among activities.

Project scheduling problems with makespan objective are generally represented by AON networks. The most restrictive way of using AON networks is by allowing only arcs of the finish-to–start type with weight 0. Here, as soon as the last predecessor of an activity j is finished, activity j is, w.r.t. precedence relations, ready to start. Minimal time lags are introduced by assigning positive weights to arcs while maximal time lags are forced with negative valued weights. Beside finish-to-start arcs, finish–to–finish, start-to-start, and start–to–finish arcs can also be modeled. All four types can be transformed to one type as shown in [17]. Moder et al. [125] coin the use of different arcs as *precedence diagramming*.

Unlike project scheduling problems with minimum duration objective, appropriate problem representation is an important issue when the objective is to derive optimal activity schedules that maximize cash flows. The AOA representation provides intuitive simplicity and ease in capturing the logical flows. However, when cash flows are present, this method is ambiguous in its representation and interpretation of the distribution of cash flows. For example, as shown in Figure 1 (a) when several activities start or end at an event, it is unclear (1) which activity or sets of activities the cash value is associated with, (2) whether the cash flow is for initiating activities emanating from the event or for completing the activities ending at the event, due to the fact that an ending event of a previous activity also serves as a starting event for one or more succeeding activities, and (3) if the cash flow is the net of expenses and payments at the event, in which case, the data on the actual expenses and payments are lost to the model. Assumptions about these issues have to be stated in order that the solution to the model may be interpreted correctly.

More seriously, the event–based model assumes that some expenses have to be incurred much earlier than necessary to have the project completed on schedule. In addition, when expenses are associated with the beginning of activities, payments cannot be generated for completed work only. Advance payments are not allowed in some project environments. In such cases, event based models must be modified to associate the expenses for each activity with the terminal event for each activity. Padman et al. [144] used a modified version of the AOA representation, shown in Figure 1 (b), where each activity is preceded and succeeded by a dummy activity. This facilitates the representation of cash inflows and outflows separately. While the modified version of activity–on–arc representation is able to clearly capture the cash flows, it creates many more dummy activities which complicate the activity networks and may impact the efficiency of solution methods.

In contrast, the AON representation can handle complex relationships without introducing too many dummy activities [185, 110]. For large projects, this representation reduces computation considerably. This model also accommodates payments for either completed work or advance payments for partially completed work but for which expenses have already been incurred. Zhu [200] used a modified version, shown in Figure1 (c) that combines the advantages of AON and the modified version of AOA representations. This compact representation assumes that (1) expenses and payments are known and associated with individual activities, (2) the expenses occur at the start of the activity, and (3) the payments are made at the completion of the activity. However, if cash flows are associated with a set of activities, this representation has to be extended at the expense of added complexity. For this case, Dayanand [45] presents an event–based network with a modified structure in which every real activity is preceded by a dummy activity. The tail node of the dummy activity is the starting node for every real activity. Expenses are associated with these nodes. Since payments are associated with the completion of activities, they are associated only with tail nodes of real activities in such a network. It can also guarantee that payments are made only for completed work. Expenses and payments with groups of activities can also be modeled with equal ease. Thus this model provides the most flexibility in modeling cash flows under the varying conditions that exist in practical projects.

The basic assumption of event vs. activity representation of the project thus leads to significant differences in the assumptions for modeling of expenses, payments, and their distribution. In event–based models, since expenses are associated with events and several activities may commence at an event, expenses at any event may be assumed to be the total expenses incurred for all activities commencing at that event. In contrast, activity–based models may assume that expenses associated with each activity are known and payments are made for completed activities.

3 Data Sets and Problem Generation

With the development of efficient solution procedures arose the need for data instances in order to benchmark these methods. Collecting these instances have led to the creation of a bank of data sets. Recent years have also seen the development of methods and software to generate project instances based on many significant parameters of the problem.

3.1 Data Sets

Davis [41] was amongst the first to generate project scheduling instances for benchmarking and testing. He generated 83 instances of the RCPSP. Subsequently, 11 instances were generated by [150] and 10 more by [187]. Patterson [148] took the lead in collecting these instances and added 6 additional problems from Davis, resulting in a total of 110 instances. Furthermore, he also assembled the optimal solutions to these problems for the minimum makespan objective using all the state–of–the–art algorithms. These instances, known as the Patterson problems, were later used in several studies, such as [149, 151, 147, 20, 60, 161, 119, 106, 108, 118].

While the great benefit of the Patterson problems was the availability of optimal makespan values and the wide usage of the test set for both optimal and heuristic algorithm evaluation, a major disadvantage was the fact that the instances were not generated based on well defined problem parameters. This led Alvarez–Valdes and Tamarit [7] to generate a new test set. Alvarez and Tamarit utilized a full factorial design with five different problem parameters to generate 144 instances. These problems were employed in the research described in [7, 6, 19].

In response to the need for a structured approach to generating problem instances, Kolisch et al. [113, 114] proposed an instance generator for a general class of project scheduling problems. Together with the generator, they introduced a new problem parameter, called resource strength, which measures the availability of a specific resource as a convex combination of a lower bound and an upper bound of the resource availability. The resource strength can be applied to different categories and types of resources. By using the resource strength and other parameters within a full factorial experimental design, Kolisch et al. generated test sets for the single and the multi–mode RCPSP, the former with 480 and the latter with 640 instances, and 536 of them with feasible solutions. The single–mode set was employed in several studies, such as [124, 61, 107, 106, 108, 130, 33, 10]. The use of the multi–mode set has been cited by [109, 180, 179].

Kolisch and Sprecher [112] have added additional benchmark problems to the two sets mentioned above, resulting in more than 10,000 instances. All instances including the parametric characterization, and optimal and heuristic benchmark solutions are available on the WWW at *ftp://ftp.bwl.uni-kiel.de/pub/operations-research/psplib/*. This address also provides the sets of Patterson, Alvarez–Valdes/Tamarit and Boctor. The latter has been generated by [26]. It is made up of multi–mode problems with renewable resources only. The problems have been used in studies by [26, 27, 28].

Research on the project scheduling problem with the NPV maximization objective motivated R.A. Russell [160] to modify the Patterson problems by adding cash flows to the activities to represent expenses for initiating activities and payments for completed work. However, the presence of a very large cash flow at the end of these project instances prevented detection of significant differences between methods for minimizing makespan applied to the problem of maximizing NPV, and special procedures that were designed to exploit the nature of the NPV problem. In order to overcome this drawback, Padman et al. [144] generated 1440 instances of the RCPSP with cash flows for testing optimization–guided heuristics with the NPV objective. These instances were based on 10 replications each of 144 project scheduling environments generated using six parameters such as size, project network structure, frequency of cash flows, level of resource–constrainedness, profit margin, and cost of capital. These parameters are defined in [144]. This data set, called the PSD set, was later used in [142, 45, 46, 47, 48, 49, 51, 200, 202, 203].

3.2 Problem Generation

There have been many problem generators employed for the experimental testing of procedures. Only a few have been made available to the public. The first one, due to Demeulemeester et al. [59], focused on generating strongly random AOA networks for a given number of arcs and nodes.

As already mentioned above, Kolisch et al. were the first who programmed the general-purpose project generator ProGen for a broad class of different problems in the AON representation. They include different objective functions, single– and multi–mode activities, different categories of resources and single and multi–project scheduling problems. The program is capable of generating projects subject to well defined problem parameters.

Beside the use of the standard problems as generated by Kolisch et al., many researchers have used the flexibility of ProGen in order to generate specific project scheduling problems. De Reyck and Herroelen [52] generated assembly line balancing problems; De Reyck and Herroelen [53] used ProGen instances to analyze the impact of the network structure on solution times. Icmeli and Erenguc [95] generated RCPSP instances with cash flows in order to test their exact solution procedure. Icmeli and Ron [98] generated project planning problems with relaxed integrality assumptions on the project's time line and the activity durations, respectively. Finally, Ahn and Erenguc [4, 5] constructed a variant of the multi–mode and time/cost–tradeoff problem using this project generator, indicating its versatility in satisfying the requirements of many scheduling environments.

Recently, ProGen has been extended by Schwindt [165, 166] in order to incorporate minimal and maximal time lags as well as some additional problem parameters, such as the estimator of the network restrictiveness as suggested by Thesen [190].

Agrawal et al. [3] proposed an instance generator for AOA networks which employs the parameter complexity index (CI) as introduced by [18]. The CI measures how far a given network is from a parallel–serial network. Costs, duration and resource requirements associated with the activities are generated randomly from uniform distributions.

4 Algorithms

4.1 The Resource Unconstrained Case

4.1.1 Makespan Minimization

The resource–unconstrained project scheduling problem can be polynomially solved by a simple forward recursion procedure, where each activity is assigned its earliest precedence feasible start time [68].

4.1.2 Net Present Value Maximization

The literature on the NPV maximization problem was recently classified by Herroelen et al. [90], so we will only summarize the recent and major approaches in this area.

The pioneering work introducing the idea of maximizing the net present value of cash flows to schedule projects was presented by Russell [159]. His analysis considered cash outflows associated with expenses for initiating activities and cash inflows that occur over the duration of the project for completing sets of activities (shown in Figure 1 (a)). The problem was formulated as a nonlinear program using an AOA representation with known durations and precedence relations and net cash flows assigned to the event nodes. A first order Taylor series approximation of the nonlinear objective function results in a linear program whose dual is a transshipment network flow problem. The solution to the dual updates the occurrence times of the events. This procedure is repeated until the occurrence times converge. Russell thus showed that the cost–critical path is quite different from the time–critical path when monetary objectives are considered.

Grinold [80] extended this research by adding a project deadline and transforming the nonlinear program with linear precedence constraints and exponential objective function into an equivalent linear program that has the structure of a weighted distribution problem. This special structure is exploited by the solution procedure that determines the optimal solution by exploring the set of feasible trees on the project network such that all activities have zero slack. This procedure is also used to illustrate, with an example, the tradeoff between NPV and project duration.

Elmaghraby and Herroelen [70] used the Russell model to develop a simplified algorithm that gives the optimal schedule for the project scheduling problem with NPV objective. They show that, in general, it is optimal to schedule events with associated positive cash flows as early as possible, and events with net negative cash flows as late as possible subject to restrictions imposed by network structure. They also illustrate that net cash flows are dependent on the time of realization of cash flow nodes and in the absence of a project deadline, if the NPV is less than zero, the project will be delayed indefinitely. A computational evaluation of the procedure was given by Herroelen and Gallens [91]. Sepil [167] pointed out a flaw and corrected the procedure. Recently, Demeulemeester et al. [63] have proposed a new optimal algorithm that performs a recursive search on partial tree structures that utilize the concept of scheduling activities early if they bring in payments and delaying those activities that incur expenses. Computational tests report encouraging results in comparison to the Grinold procedure.

4.2 The Resource Constrained Case

4.2.1 Optimal Approaches

Makespan Minimization The methods applied so far to solve the RCPSP exactly are dynamic programming [34], zero–one programming [30, 157, 150, 151], and implicit enumeration with branch and bound. The majority of exact approaches utilize implicit enumeration with branch and bound. Enumeration schemes are, amongst others, reported by [14, 43, 86, 158, 183, 187, 36, 21, 60, 61, 124, 169, 33, 131]. We restrain from reporting on all of the above mentioned algorithms, instead we will focus on the ones which have been state of the art in the past or seem likely to be so in the future.

Talbot and Patterson [187] order the activities in an activity list such that precedence relations are taken into account. Then, they derive time windows for all activities by forward recursion and backward recursion from an upper bound of the makespan. Starting with the first activity on the list, the enumeration process tries to schedule the next activity on the list at the earliest precedence- and resource-feasible interval within the activity–specific time–window. If this is not possible, backtracking occurs to the last activity which is than scheduled one period later. The basic enumeration is enhanced by network cuts which allow to prune part of the enumeration tree. An implementation of this algorithm on parallel processors has recently been documented in [169].

The algorithm of Stinson [182, 183] is a breadth–first branch and bound approach based on the enumeration tree originally developed by [101] for the RCPSP with a single resource. Each node in the tree represents a feasible partial schedule, i.e., a schedule where only a subset of the activities has been scheduled [163]. The schedule time associated with a node measures the time elapsed in completing the partial schedule of the ancestral node. Offsprings of a node are created by enumerating all feasible combinations of not yet scheduled, precedence–feasible activities. The selection of nodes utilizes a set of six lexicographically ordered priority rules. To prune the solution tree, Stinson uses the two dominance rule (the stronger one is the left-shift–rule of Schrage [163]) as well as the precedence– and the resource–based lower bound (cf. e.g. [163]). Additionally, based on [197], a critical–sequence-based lower bound is employed which jointly takes into account precedence and resource constraints.

The algorithm presented by Demeulemeester and Herroelen [60, 61] is an extension of the depth-first branch and bound approach of [36]. The main difference with Stinson's procedure occurs in the enumeration tree, where new nodes are not created by considering sets of activities which are scheduled, but by considering sets of activities which are delayed. Two dominance rules are used to prune the enumeration tree. The first one is a variation of the left–shift dominance rule. The second one makes use of a "cutset", i.e. unscheduled activities for which all predecessors belong to the partial schedule. Bounding is performed with the precedence-based and the critical–sequence–based lower bound used by [183], and the weighted-node–packing–bound of [124].

Brucker et al. [33] applied insight gained from their work on the job shop problem for developing quite a novel solution method. They define four disjoint relations for pair of jobs: conjunctive relations for jobs which are precedence–related, disjunctive relations for jobs which cannot be processed simultaneously because of resource constraints, parallel relations for jobs which have to be simultaneously processed at least during one period, and flexibility relations for pair of jobs, where none of the other three relations holds. The branch–and–bound enumerates the set of flexibility relations and transfers them to disjunctive relations. The method utilizes three lower bounds. The first one is based on a LP–formulation which relaxes the precedence constraints and the non–preemption assumption. The The second one relaxes the RCPSP to a 2–job shop problem which can be efficiently solved by [32]. The The third lower bound extends the critical sequence based lower bound of Stinson [182].

The solution procedure of Mingozzi et al. [124] makes use of a new mathematical formulation which gives way to a number of new lower bounds and a new enumeration scheme. The latter relies on the concept of feasible subsets, i.e. subsets of activities which are note precedence related and where the sum of the resource requirements does not exceed the availability for any resource. The best lower bound formulates a weighted node packing problem which , given its NP–hard nature, is solved using heuristics.

Sprecher [177] has streamlined his exact procedure for the multi–mode problem (cf. Section 5.3) in order to efficiently solve the single–mode case.

Max NPV Doersch and Patterson [65] introduced a binary integer programming approach to the NPV project scheduling problem. This model included a constraint on capital for expenditure on activities in the project such that the available capital increased as progress payments were made. The objective function also included the cash flows associated with the completion of activities and any penalties incurred for late completion. The model was solved to optimality for projects involving 15 - 25 activities. The results indicated that at high cost of capital or long project duration, it is important to evaluate bonus/penalty and capital constraints while scheduling activities.

Smith-Daniels & Smith-Daniels [174] extended the Doersch and Patterson model to include material constraints and costs. The NPV of the project was maximized subject to material and capital constraints and solved to optimality on small problems. They concluded that not only do ordering and holding costs force activities with common requirements to start at the same time or close to each other, the additional constraints also result in lowering overall project costs even though they may cause activities, and hence the project, to be delayed.

Tavares [188] proposed a new dynamic programming formulation and solution method, where the optimality conditions were derived using calculus of variations for a set of interconnected projects. The objective function to be maximized included a net of the discounted sum of the benefits generated along the program, the discounted sum of the costs of project expenditures, and a term to penalize the variation in expenses over time. This program was applied successfully to a large railway construction project in Portugal.

Patterson et al. [152, 153] presented a zero–one programming model and a backtracking algorithm to maximize the NPV of the constrained project scheduling problem. It is unique in that it can also be used to minimize project duration. The solution methodology utilized the fact that the minimum duration problem is easier to solve than the max NPV problem and used it as a heuristic to generate starting solutions on which right-shifting of cash flows was applied to improve NPV. Ninety–one problems, ranging from 10 to 500 activities, were tested on both objectives using MINSLK and random rules, with optimal solutions found only for the smaller problems. The MINSLK rule generated higher NPV than the random rule.

Icmeli and Erenguc [94] developed a branch–and–bound algorithm for the RCPSP with cash flows which used the minimal delaying alternatives concept originally introduced in [60] for branching. This concept together with the rule that determines the node to branch from are used in bounding the size of the tree. The algorithm was tested on 50 test problems from the Patterson set with the number of activities ranging from 7 to 51, and 40 problems with 32 activities generated using ProGen, and with up to 3 resource types and was shown to be efficient in comparison to results in the literature.

4.2.2 Heuristic Approaches

Makespan Minimization Heuristic approaches for the RCPSP with makespan objective basically involve four different solution methodologies: (1) priority–rule–based scheduling, (2) truncated branch–and–bound, (3) disjunctive arc concepts, and (4) local search techniques. Furthermore, integer programming based heuristics have been used by [134].

(1) Priority–Rule–Based Scheduling is made up of two components: a schedule generation scheme and a priority rule. Two different schemes for the generation of feasible schedules can be distinguished [107]: the serial and the parallel method, respectively. Both generate a feasible schedule by extending a partial schedule in a stage–wise fashion. In each stage the generation scheme forms the set of all schedulable activities, called the decision set. A specific priority rule is then employed in order to choose one activity from the decision set which will be scheduled. While the decision set of the serial method is made up of all currently unscheduled activities whose predecessors have already been scheduled, the parallel method defines the set as including all the precedence–feasible unscheduled activities which can be started at the scheduled time, given the resource constraints. Both methods schedule in each stage, respectively iteration, exactly one of the $j = 1, \ldots, J$ activities. Hence, both perform J iterations. The serial method is widely known as list scheduling in the area of machine scheduling [164, 103]. It has been shown by Kolisch [106] that the serial method generates active schedules, whereas the parallel method generates non–delay schedules. When minimizing a regular measure of performance such as the makespan the set of active schedules does always contain an optimal schedule while this may not be case for the set of non–delay schedules. Dependent on the number of passes performed and hence the number of schedules generated, single–pass and multi–pass approaches are distinguished. Results utilizing the serial method in a single–pass environment were

published by many authors [38, 39, 25, 194, 107]. Multi-pass applications of the serial method are documented by [38, 25, 120, 156, 107]. Computational experiments conducted with the single–pass version of the parallel method are more frequent than those with the serial method and are, amongst others, reported by [146, 40, 149, 147, 189, 196, 71, 117, 192, 7, 6, 25, 194, 136, 138, 107, 106, 191]. Arora and Sachdeva [9] report an implementation of the parallel method on parallel processors. Multi–pass efforts on the basis of the parallel method are reported by [198, 117, 6, 192, 25]

Pollack–Johnson [156] uses a so–called depth–first, jumptracking branch and bound search of a partial solution tree. The algorithm is essentially a parallel scheduling heuristic. Instead of scheduling the activity with the highest priority value it branches on certain occasions such that one branch has the activity with the highest priority value and the other branch has the activity with the second highest priority value, which is scheduled next.

Kolisch and Drexl [108] proposed a special multi–pass approach, called adaptive search procedure. The method makes use of the serial and parallel schedule generation scheme employing a deterministic as well as a sampling method. Based on an analysis of the problem at hand and the number of iterations already performed, the procedure decides on the specific method to apply. The use of bound lowers the computational effort.

Mausser and Lawrence [123] use block structures to improve the makespan of projects. They start by generating a feasible solution with a parallel scheduling scheme. Following this, they identify blocks which represent contiguous sets of time periods that completely contain all jobs occurring within it. Each such block can be considered independent of the other blocks. The methods essentially reschedule individual blocks in order to shorten the overall project length.

Özdamar and Ulusoy [193, 138] embed "local constraint based analysis" into a single–pass parallel scheduling scheme in order to decide via feasibility checks and so–called essential conditions which activities have to be scheduled and which activities have to be delayed at any given time.

(2) Truncated Branch and Bound methods employ the enumeration scheme of the procedures given in Section 4.2.1 but instead of exploring the entire enumeration tree, only a partial exploration is performed. Alvarez–Valdes and Tamarit [7]. make use of the enumeration tree as presented in [36] which uses essentially the same enumeration tree as employed in [60], i.e., nodes consist of sets of activities which have to be delayed. Instead of enumerating all offspring nodes, the heuristic implicitly or explicitly chooses one node.

(3) Disjunctive–Arc–Based Heuristics make use of the methodology developed by Balas [13] for solving the job shop problem. Heuristics solution methods for the RCPSP were proposed by [168, 7, 20, 12]. The basic idea of the disjunctive–arc–based approaches is to extend the precedence relations (the set of conjunctive arcs) by adding additional arcs (the disjunctive arcs) such that the minimal forbidden sets, i.e. sets of technologically independent activities which cannot be scheduled simultaneously due to resource constraints, are destroyed and thus the earliest finish schedule is feasible with respect to (precedence and) resource constraints.

Shaffer et al. [168] restrict the scope, within their "resource scheduling method", to those forbidden sets for which all activities in the earliest finish schedule are processed at the same time. The disjunctive arc which produces the smallest increase in the earliest finish time of the unique sink is introduced and the earliest finish schedule is recalculated. The algorithm terminates as soon as a (precedence– and) resource–feasible earliest finish schedule is found.

Alvarez–Valdes and Tamarit [7] propose four different ways of destroying the minimal forbidden sets. The best results were achieved by applying the following strategy: Beginning with the minimal forbidden sets of lowest cardinality, one set is arbitrarily chosen and destroyed by adding the disjunctive arc for which the earliest finish time of the unique dummy sink is minimal.

Bell and Han [20] present a two–phase algorithm for this problem. The first phase is very similar to the approach of Shaffer et al. However, phase 2 tries to improve the feasible solution obtained by phase one as follows: after removing redundant arcs, each disjunctive arc that is part of the critical path(s) is temporarily cancelled and the phase 1 procedure is applied again.

(4) Local Search Techniques for project scheduling problems have only emerged in recent years. All approaches, except the one of Sampson and Weiss [161] encode the solution as activity list which can be mapped into a schedule by means of one of the priority-rule-based scheduling schemes outlined above.

In the procedure of Sampson and Weiss [161], a solution is represented by a shift vector whose length provides a count of the number of jobs; each element of the shift vector is an integer number. A schedule is obtained as follows: each activity is started at the maximum of the finish times of immediate predecessors plus the integer as defined in the shift vector. Consequently, a solution is always precedence feasible schedule. Infeasibility with respect to resource constraints is handled by penalizing the objective function with a measure taking into account the number of renewable resources used in excess of the period limit. The neighborhood of a solution is defined as all solutions that differ in exactly one element of the shift–vector.

Leon and Balakrishnan [119], Naphade et al. [130], Lee and Kim [118], and Kolhlmorgen et al. [104] extended the approach originally introduced by Storer et al. [184] for the job shop problem in order to apply local search methods to the RCPSP. Basically, they encode a solution as a string of numbers which assigns each job a priority value. By using these priority values within a schedule generation scheme, one obtains a feasible schedule and the associated objective function value. This encoding can be adapted to different type of local search methods such as simulated annealing [72], tabu search [77, 78], and genetic algorithms [79].

Baar et al. [12] propose two different neighborhoods for their tabu search procedure. The first one encodes a solution as an activity list which can be mapped to a schedule with the serial scheduling scheme. The neighborhood is defined as all activity list which can be reached by shifting a resource critical job to a new position. The second neighborhood builds up on the exact solution procedure of Brucker et al. [33]. Essentially, activity pairs are either forced or freed to be (for at least one time period) processed in parallel via so–called parallelity relations. For a given problem with fixed parallelity relations a schedule is obtained by simple forward recursion (cf. Section 4.1.1).

NPV-Maximization Heuristic procedures for the NPV problem can be broadly classified into (1) optimization–guided, (2) parameter–based, and (3) metaheuristic approaches. Optimization guided approaches use the solution values from the unconstrained NPV model which is a relaxation of the RCPSP to develop single-pass heuristics. In contrast, parameter–based heuristics utilize information from the critical path and cash flows, such as the sum of cash flows over successor paths, to develop the priority rules. Distinct from these approaches is the metaheuristic strategy that guides activity scheduling using general search procedures.

(1) Optimization–guided approaches began with R.R. Russell [160] who initiated a stream of research on such heuristic methods for the RCPSP with NPV objective. He used both insights from the relaxed resource–unconstrained NPV problem and methods designed for the minimum duration problem to develop six heuristics. They were tested on 80 problems ranging from small scale problems with 30 activities to large scale problems with 1461 activities. One of the heuristics, based on random selection of activities for scheduling (RAND–50), was used as a benchmark to select the best out of 50 randomly generated solutions. It was observed that no one specific heuristic performed best in all situations. For the small scale problems, the heuristics had similar performance and were within 5-10% of the optimal solution. As the project size increased, the level of resource–constrainedness determined the efficient heuristics. MINSLK/LAN rule, a good rule for the minimum duration problem, was found to perform best for large projects when the resource constraints were not tight. In contrast, when resources are tight, rules based on the relaxation of the RCPSP provided better performance, additionally reinforcing the fact that max NPV problem requires new approaches compared to the minimum duration problem.

Padman, Smith–Daniels and Smith–Daniels [144] used insights gained from the solution of the relaxed optimization model, as in R.R. Russell [160], in developing heuristic procedures to schedule projects with multiple constrained resources. They showed that a heuristic procedure with embedded priority rules that uses information from the repeated solution of the relaxed optimization model, unlike Russell's procedure which solved the relaxed model once, increased project net present value. The optimization–guided heuristic procedure and nine different embedded priority rules were tested in a variety of project environments that considered different network structures, levels of resource–constrainedness, and cash flow parameters, called the PSD data set. Extensive testing on the PSD data set, showed that the new heuristic procedures dominate heuristics using information from the Critical Path Method (CPM) and in most cases outperform heuristics from previous research. The best performing heuristic rules classified activities into priority and secondary queues according to whether they led to immediate progress payments, thus front loading the project schedule.

Padman and Smith–Daniels [142] extended previous work using the relaxed optimization model to evaluate trade–offs between early and tardy penalties in the scheduling of activities. They embedded eight heuristics in the greedy procedure discussed in [144] to test whether releasing activities to the schedule queue as soon as their predecessor activities were completed could result in improved project NPV. Extensive testing on the PSD data indicated the success of this approach.

(2) Parameter–based approaches were applied by Smith–Daniels & Aquilano [172] who compared the duration and NPV of a late–start critical path schedule to that of an early–start critical path schedule. It was assumed that cash outflows occurred at the beginning of the period and a single project payment was received on completion of the project. Their assumptions were tested using the 110 Patterson problems [148]. The authors found improved average NPV and lower average duration for late–start schedules than early–start schedules.

Ulusoy and Özdamar [139] presented an iterative scheduling algorithm with the objective of improving both the project duration and net present value. The consecutive forward/backward scheduling passes made by the iterative algorithm result in a smoother resource profile, which, along with right–shifting of activities, improves both the project duration and NPV. In the cash flow model assumed here, activity expenditures occur at their starting times and payment is made at the completion of the project. The algorithm was tested on two sets of problems from the literature. The results demonstrated that under the assumed cash flow model, the iterative scheduling algorithm improved both criteria.

Baroum and Patterson [16] evaluated several heuristic approaches used by project and contract managers involving single– and multi–pass procedures. Their single–pass procedures used priority weights based upon cumulative future cash flows for all successor activities. Multi–pass procedures were enhancements to improve upon the single–pass solution obtained. A full factorial experimental design was used to assess the performance of the heuristic procedures. The computational results demonstrated the efficacy of the discounted cash–flow, positional weight heuristics over more traditional methods.

(3) Metaheuristic strategies have had considerable success in recent years. Icmeli and Erenguc [93] applied tabu search procedure to a starting feasible solution generated using a simple single–pass algorithm. The initial solution was improved over several iterations by moving each activity one time unit early or late from its current completion time, with the restriction that the resulting completion time should not violate earliest and latest completion times for the activity. They also investigated the use of long term memory within tabu search to further improve the results. Computational results on 50 problems from the Patterson set indicated that these procedures were both efficient and close to optimal.

Motivated by the success of adaptive strategies in solving complex routing and manufacturing problems, [154] and [204] adapted multi–heuristic combination for solving project scheduling problems. In [204], six simple rules that capture different aspects of the scheduling problem, such as resource–constrainedness and network topology, are called upon randomly to schedule activities. The underlying premise is that over a number of iterations, the rules will exploit the changing conditions in the project environment. Extensive experimentation conducted using the Patterson and PSD data sets reveal the superior performance of the combination method in comparison with the individual participants. Learning strategies, the most natural extension, are not incorporated in this study.

Zhu and Padman [203] also reported on the design, implementation, and experimentation of a local search enhancement strategy for schedule improvement using tabu search. The procedure, using cash flow based move generation strategies, helps to overcome the problems associated with getting trapped in local optima and is equally useful as a repair heuristic. Several parameters within tabu search, such as novel candidate generation strategies, are examined and their impact on solution methods and project NPV are evaluated. Unlike previous heuristics, the metaheuristic approach dominates in over 85% of the PSD problems, a significant improvement over heuristics in the literature. The results illustrate that problem–independent, metaheuristic approaches are better able to exploit the complex interactions of the many critical parameters of the RCPSP in comparison to the single–pass, parameter–based, problem dependent heuristics that are commonly used.

Zhu and Padman [201] also applied distributed computing concepts to the RCPSP through the use of an Asynchronous Team (A–Team) approach. An A–team is a software organization that facilitates cooperation amongst multiple heuristic algorithms so that together they produce better solutions than if they were acting alone. They embedded several simple heuristics for solving the RCPSP within the iterative, parallel structure of A–Team which provides a natural framework for distributed problem solving. Preliminary results on small randomly generated project networks indicated that the combination of multiple, simple heuristics outperform many single pass, complex optimization–based heuristics proposed in the literature.

5 Related Problems

5.1 Capital Constrained Problem

Smith–Daniels et al. [173] argued that the capital constrained project scheduling problem presented a unique managerial challenge as compared to the RCPSP since, in large projects, it is frequently the case that a capital constraint limits the value of work that may be put in progress at any time. In contrast to the RCPSP, additional quantities of capital, the constrained resource, become available for use as progress payments are received for completed work. Since the objective is to maximize project net present value, it is important for the project schedule to arrive at a balance between early receipt of progress payments, which improve NPV and increase the capital balance available, and delay of particular large expenditures. Heuristic methods, using information from the solution to the unconstrained NPV problem, are tested on large project networks, presenting the first results on this practical problem.

5.2 Time/Cost Trade–Off Resource Constrained Project Scheduling Problem

Erenguc et al. [73] pointed out that in previous formulations of the RCPSP with cash flows, the activity durations are assumed to be fixed and reductions in the activity durations are not allowed. They present the time/cost trade–off problem where the durations can be reduced from their normal requirements by allocating more resources, assumed to be unlimited, with associated crashing costs that are included in the NPV objective function. They develop a Generalized Benders decomposition procedure for obtaining an optimal solution. This procedure is tested on 56 problems with reasonable computational effort.

5.3 Multiple Execution Modes

As pointed out in Section 2.1.1 the multi–mode problem is characterized by the fact that activities can be performed in different discrete modes. Hence, a solution has to determine the *timing* of activities as traditional in scheduling and the *assignment* of modes. This adds further complexity to the already complex case of renewable resource constraints, an $\mathcal{NP}$–hard optimization problem. Even worse, if more than two nonrenewable resources are taken into account, the problem of finding a feasible solution becomes $\mathcal{NP}$–hard [109].

5.3.1 Optimal Approaches

Optimal procedures for solving the multi–mode RCPSP with makespan objective have been presented by [186, 152, 175, 176, 180, 178, 179]. All of these are extensions of branch and bound procedures originally proposed for the single-mode RCPSP.

Based on [187], Talbot [186] suggests a two-phase solution approach. In phase one, activities, modes, and renewable resources are sorted in order to speed up the enumeration procedure applied in phase two. The enumeration procedure employs the activity–list and schedules the next activity on the list in the shortest mode at the earliest time. If an activity cannot be scheduled in its time window, the algorithm tracks back and tries to schedule the successor in the activity list at a later time or the next mode. Patterson et al. [152] refined this solution method and reported computational results in [153]. Most noteworthy is the introduction of a "precedence tree" which allows a systematic enumeration of mode–assignments and start times. Sprecher [176] improved the method in three aspects: He corrected some flaws, he introduced the notion of an i-partial schedule which uniquely describes a node i of the enumeration tree and the associated partial schedule, and he added four dominance and one feasibility bounding rule. Further refinements of this procedure, including new and powerful bound, are given in [178, 179].

Speranza and Vercellis [175] proposed a depth–first oriented branch and bound procedure which enumerates the set of active schedules. The precedence–based lower bound is employed to prune portions of the enumeration tree. Subsequently it was shown by Hartmann and Sprecher [85] that the method might fail to produce optimal or feasible solutions. Sprecher et al. [180] extended the enumeration scheme of Demeulemeester and Herroelen [60] for the single–mode RCPSP. Hartmann and Drexl [84] generalized the exact procedure of Stinson et al. [183] to the multi–mode context. Furthermore, they made an in-depth comparison of the three branch–and–bound strategies of Sprecher [176], Demeulemeester and Herroelen [60], and Stinson et al. [183] to solve the multi–mode RCPSP.

5.3.2 Heuristic Approaches

Heuristic solution methodologies for (special cases of) the multi–mode RCPSP use (1) single–pass and multi–pass priority–rule–based scheduling [171, 26, 28, 67, 136, 109], (2) Simulated Annealing [171, 27], and (3) Genetic Algorithms [135, 82, 128].

(1) Priority–rule–based scheduling. Boctor [26] employs a modified parallel scheduling scheme, where an activity is in the decision set if it is at least resource feasible in one mode. Activities are chosen with the MSLK–rule, and modes are chosen on account of the minimum duration. A multi–pass variant uses five ordered pairs of activity– and mode–priority rules. In Boctor [28], instead of choosing one activity from the decision set, the set of non–dominated schedulable activities is chosen by calculating a lower bound of the prolongation of the resource–unconstrained makespan.

Drexl and Grünewald [67] propose a regret–based biased random sampling approach which jointly employs a serial scheduling scheme and the SPT priority rule. Slowinski et al. [171] solve the multi–mode RCPSP with multiple objectives. First, a (precedence–feasible) priority list of the activities is derived with one of 12 priority rules. In the order of the priority list, (precedence–feasible) activities are scheduled in the mode with shortest resource–feasible duration at the earliest period possible. The procedure is extended to multi–pass approach by randomly selecting from the ranked activities instead of scheduling the first activity on the list.

Özdamar and Ulusoy [136] broadened their "local constraint based analysis"–approach to solve the multi–mode RCPSP. They report results which are consistently better than the single–pass priority rule based approaches and a multi–pass approach, respectively.

Kolisch and Drexl [109] suggested a local search procedure which especially takes into account scarce nonrenewable resource. The method employs a look–ahead strategy to obtain an initial feasible mode–assignment, i.e. an assignment of each activity to one of its modes, followed by a basic local search performed on the mode–assignments. Every feasible mode–assignment is evaluated by running the adaptive search algorithm of Kolisch and Drexl [108].

(2) Simulated Annealing. Slowinski et al. [171] were the first to try their hand on simulated annealing to solve the multi–mode RCPSP. Based on the scheduling list they propose a pairwise interchange neighborhood where a new list is generated by exchanging the positions of two randomly chosen activities which are not precedence–related. The list can than be converted to a schedule by employing the serial scheduling scheme.

Boctor [27] also suggests a simulated annealing approach which operates on the scheduling list. In contrast to Slowinski et al., he favors a shift–neighborhood where one randomly chosen activity is shifted to a new precedence feasible position on the list.

(3) Genetic Algorithms. Özdamar [135] proposes a genetic algorithm which builds on the ideas of [184]. A solution is represented by two lists, each with length J. The first specifies an execution mode for each activity. The second list specifies, for each of the $n = 1, \ldots, J$ iterations of the parallel scheduling scheme, the priority rule which selects the activity from the decision set. This activity is subsequently scheduled. New individuals are created with a two–point crossover where the first n_1 elements are taken from the mother list, the next $n_2 - n_1$ elements are taken from the father list, and the final $J - n_2$ elements are taken again from the mother list.

Hartmann [82] reports excellent results with a genetic algorithm. An individual is represented by two lists: a precedence–feasible activity list and a mode–assignment list. The latter selects a mode for each activity. With this information the former list can, by employing the serial scheduling scheme, be converted into a schedule. For the crossover, a new individual is created by taking the first part of the activity list from a father individual while ordering the remaining activities as given by the mother individual. Also, the first part of the mode-assignment is taken from the father and the mode–assignment of the remaining activities is taken from the mother. The method of Mori and Tseng [128] employs similar ideas.

5.4 Scheduling with Maximal Time Lags

All of the models and methods treated allow only the incorporation of minimal time lags between pair of activities (cf. Section 2.3). If additionally, maximal time lags are allowed, the problem becomes much more complicated and standard RCPSP algorithms generally fail to obtain solutions. Hence, special purpose methods which incorporate maximal time lags have been proposed. Amongst them are the exact procedures of [17, 54, 62] and the heuristic algorithms of [199, 132, 75].

5.5 The Payment Scheduling Problem

All the NPV models discussed earlier maximize the contractor's NPV and are based on the assumption that cash flows associated with completion of activities are known. However, when bids are submitted for projects, the amount and timing of payments are important variables that can be negotiated to improve financial performance. In practice, the contractor usually knows the expenses associated with project activities. He can use this information combined with the knowledge of other project parameters such as activity durations to negotiate the payments received for completed work so that the project achieves the maximum level of financial returns. Bey, Doersch and Patterson [22] argued that since the decision to organize on a project basis often is an indication that a firm is committing substantial portions of its financial resources to relatively few projects, the effective timing of cash receipts and outlays can have a significant impact on the ultimate profitability of the endeavor. And even in the case of a relatively small contractor, opportunities do exist for increasing profitability through the judicious use of progress payments.

As pointed out by Elmaghraby [69], the use of network models as aids in the preparation of project bids has received little research attention, even though cost estimation and bidding have been popular topics with practitioners for a fairly long time. In his paper, Elmaghraby suggests a method of arriving at the project cost based on the expenses associated with each activity in the project and the activity schedule. Each milestone event in the project is allocated some of the cost of all activities that precede the event, and the activity schedule is used to adjust for the time value of money.

Gilbreath [76], Hendrickson and Au [83] and Clough and Sears [37] are excellent references and provide a good introduction to the general project contracting literature. The recent survey by Herroelen et al. [90] provide a summary of the various contracting arrangements common in the project management domain, such as lump–sum contracts, cost–plus contracts, and so on. [45, 49] also summarize prevalent contracting schemes as they highlight the significant role of the nature of the contracts in scheduling progress payments and activities for both contractor and client.

Dayanand and Padman [50] examine, in the first such effort in the literature, the problem of simultaneously determining the amount, location and timing of progress payments. They present several models that can be used by the contractor and client for managing cash flows, setting milestones, and negotiating critical contract parameters, in addition to identifying the location, timing, and amount of progress payments. Solutions for the models that highlight assumptions, and representation and tractability issues are illustrated with the aid of an example. Based on the assumptions made, the models are applicable in different project environments.

In [47], the authors discuss the problem of determining the amount and timing of payments from a contractor's perspective. Optimal and heuristic payment schedules to an integer programming model are shown to be affected by a number of factors such as project deadlines, the number of payments, profit margins, cost of capital, pattern of expenses and the structure of the network. In particular, when progress payments are based on expenses incurred by the contractor, the percentage of expenses recovered with each payment and the number of payments have a significant impact on payment schedules. Dayanand and Padman [48], also propose a multistage heuristic to determine a set of payments using simulated annealing in the first stage. In the second stage, activities are rescheduled to improve project NPV. The performance of this general purpose heuristic is compared with other problem–dependent heuristics from [47], with significant improvement in schedules and NPV.

A client and contractor enter into an agreement for a project either by competitive bidding or negotiation. While there has been much discussion in the literature of the contractor's requirements and problems, very little research exists to help the client manage complex project contracts and schedules. In contrast to earlier research, Dayanand and Padman [49] consider the problem of simultaneously determining the amount and timing of progress payments in projects from a client's perspective. They develop several mixed integer linear programming models based on some practical methods of determining payment schedules, and discuss properties of the models as well as characteristics of optimal payment schedules obtained with each of the models. These models are shown to be valid for a wide variety of contract options. The impact of these payment schedules offered by the client on the contractor's schedule of activities is also explored. Contrary to popular practice, regularly scheduled payments increase the client's expenses on the project. In [51], the authors also develop heuristics for solving the client's payment scheduling problem. Finally, Dayanand [45, 46] proposes models and solution methods for the joint payment scheduling problem that can used by the contractor and client in negotiating payments and payment parameters in projects.

6 Deployment Issues

This section summarizes research efforts that describe, develop and implement decision support systems (DSS) for project scheduling support from the various components discussed earlier. Many of the methods described above have been implemented by the authors and are available upon request. Nevertheless, despite some exceptions, these implementations can be used mainly for purposes of testing, and cannot be judged as ready–to–use, robust software systems.

6.1 Decision Support Systems

Effective and efficient project management requires the ability to identify and integrate the vast store of information and knowledge that has been developed over the past several decades in this area. Given the relentless advances in both technology and computers as well as models and methods, decision support systems provide users with the ability to manage the knowledge and apply it for better decision making. While many of the tools and products currently available support individual decision making, there is a growing array of decision aids for group and organizational decision support. In addition, these tools are extending the frontiers of research and application by integrating artificial intelligence capabilities into the day–to–day decision making on contract management, scheduling, cost management, and other such project–related activities.

The fact that project scheduling is a prominent area for decision support has been pointed out by Anthonisse et al. in 1988 [8]. Since then a number of academic prototypes have emerged. The successful efforts at combining machine learning and optimization approaches in providing valuable and realistic decision support in domains such as manufacturing and routing have prompted application of the same in project scheduling and management. Tavares [188] developed a multicriteria analysis of a multi–project scheduling problem. Möhring et al. [126] proposed an advanced DSS for scheduling which makes use of eigenmodels. Further DSS approaches for project scheduling are reported in [44, 102].

Özdamar and Ulusoy [139] established a DSS-system which consists of a modeling module and a scheduling module. The modeling module allows the user to model a single project consisting of activities with multiple execution modes, different types of resources (renewable, nonrenewable, doubly constrained), different type of attributes going along with activity modes (cash flows, resource requirements, durations), different type of objective functions (minimize makespan, maximize net present value, minimize tardiness). The scheduling module utilizes an interactive version of the heuristic proposed in [136, 193]. Additionally, the system offers the opportunity to reduce planning nervousness in a revised planning environment by freezing parts of schedule.

Drexl and Kolisch [?] report on the successful implementation of project scheduling models and methods in a commercial scheduling system, called LEITSTAND (cf. [155]), for the scheduling of machine tool assembly. Slowinski et al. [171] proposed a multi–objective project scheduling system which can handle many categories of resource constraints. Problems can be defined by a user–friendly model editor. The system generates a set of feasible schedules using different heuristic scheduling techniques such as single– and multi–pass priority rule based scheduling and simulated annealing (cf. Section 5.3), which can then be evaluated with multi–objective analysis. Furthermore, schedules can be revised by a constructive, interactive process. Visualization of the solutions can be done with the help of Gantt–charts and resource profiles.

The DSS of Norbis and Smith [133] takes into account seven different objectives, including two time–related, four due–ate–related and one related to resource utilization. The different objectives are taken into account by an interactive schema which works through the algorithmic design of the heuristic to navigate the tradeoffs among the objectives.

As discussed earlier, the inherently difficult nature of the constrained project scheduling problem with cash flows precludes the development of optimal schedules for projects of reasonable size. Therefore, heuristics have found wide application. Recent research has addressed a number of questions regarding the deployment and appropriate use and management of these heuristics as well as the design of new ones.

Padman and Zhu [143] proposed a problem space computational model that integrates the multiple knowledge sources, such as optimization and statistical models, exact and heuristic solution procedures, and heuristic selection and repair techniques, associated with the resource–constrained project scheduling task. While integrated application of these knowledge sources is required to effectively support scheduling, previous research has focussed on developing and implementing them in isolation. The problem space computation model presented in this paper addresses this shortcoming by integrating these various knowledge sources, thus enabling the development of decision support systems for resource–constrained project scheduling. The authors conclude that this modeling system can accommodate a wide variety of constraints and objectives that occur in practical project environments.

Heuristics in the literature have been proposed for the RCPSP with cash flows based on differing information about the project such as activity duration, tardiness penalty, opportunity cost, and so on, as discussed earlier, to improve project schedules and NPV. However, the problem of structuring this information, identifying the important parameters that can best characterize the project, and developing tools to aid the selection of the best heuristic for a specific project environment are equally challenging from an information management perspective. Padman [140] and Zhu and Padman [202] develop a neural network system to induce the relationship between project parameters and heuristic performance and compare the results of this approach with those from traditional statistical procedures. They demonstrate that neural network methodology can be employed both to extract information about project conditions as well as to provide predictions about appropriate categories of heuristics to use for novel cases.

Padman and Roehrig [145] discuss the use of genetic programming (GP) for heuristic selection, and compare it directly to alternative methods such as OLS regression and neural networks. Genetic programming [116], a variant of genetic algorithms, employs strings of arbitrary length to represent solutions as tree structures. The study indicates that the GP approach yields results that are an improvement on earlier methods. The GP solution also gives valuable information about project environments where a given heuristic is inappropriate. In addition, this approach has no problem evolving complex nonlinear functions to capture the relationship between problem parameters and heuristic performance. Thus the results given in this paper shed light on the logical domains of applicability of the various heuristics, while at the same time provide an improved heuristic selection process.

6.2 Assessment of Commercial Project Management Systems

Research on the assessment of commercial project management systems can be roughly divided into general evaluations of project management programs and the evaluation of its resource allocation capabilities. Work in the first area has been performed by [195, 11, 55, 111, 162]. Noteworthy is a multi–attribute–value theory–based approach for selecting project management software pursued in [111].

Research in the second field has only emerged in recent years [100, 122, 74, 105]. This is due to the fact that commercial project management systems did not offer the capability of resource allocation until the beginning of the nineties. Additionally, researchers lacked powerful, exact methods to derive optimal benchmark solutions for problem instances of real–world size. The most current evaluation study has been conducted by [105] where the quality of seven commercial project management systems was assessed on a set of 160 problems generated by factorial design. As a benchmark, all problems were solved with a state–of–he–art optimal procedure of [60]. Statistical tests were performed in order to investigate the impact of well defined problem parameters and to detect significant differences between the quality of the software packages. The average deviation from the benchmark solution ranged between 4.39 % for the best and 9.76 % for the worst package, with significant differences in scheduling performance. Several problem parameters had a critical effect on the performance of the software systems. The scheduling performance deteriorated with increasing number of activities and increasing level of constrained resources, respectively. An increasing scarcity of the resources and an ascending number of requested resources also led to poor scheduling performance. Finally, the scheduling results were (slightly) improved for projects with many precedence constraints per activity. All parameters, except the number of precedence constraints, showed a significant impact on the scheduling performance of the majority of the project management systems. These observations are in accordance with experimental results reported by [107] using well known priority rules for the minimum duration problem and by Padman et al. [144]in the case of the max NPV problem.

6.3 Internet and Web-based Project Scheduling Systems

The use of the World Wide Web (WWW) and Internet technologies to collect, organize, and deploy decision support components for research and practice has only begun to be explored by the project scheduling community [121, 141]. The dramatic increase in the capabilities of the enabling technologies, and the wide reach of the Internet, facilitate seamless integration of data, models, solvers, and conversions among multiple representation requirements to researchers worldwide. At the same time, solver benchmarking and evaluation efforts, previously time–consuming and tedious due to incompatible data formats and representations, are also made easier. From a practitioner perspective, the extensive time lags between successful research efforts and availability of these tools and methods in practice can be reduced considerably.

Such end user requirements, emphasizing the need for the sort of connectivity offered by information networks, include platform independent access to DSS components (i.e., the unbundling of data, models and the interface), the need to incorporate data from remote, and possibly, dynamically changing, data sources (e.g., project data in a reactive scheduling environment), communicate customized reports based on DSS analysis to a distributed group of users (e.g., current schedule, costs, payment invoices for work completed, and so on), and integration with other organizational applications (e.g., order placement for materials and other project resources) using messaging standards such as electronic data interchange (EDI) [92].

Given that several alternative architectures may be realized using Web–based technologies, how should these end user requirements for project scheduling decision support be satisfied? What should be the criteria used to evaluate these alternatives and how do some important alternative architectures compare with one another? [115] outline these architectural choices and evaluation criteria, and [141] has extended these choices to the project scheduling problem.

Efforts such as the MMM project (*http://macke.wiwi.hu-berlin.de/mmm/*), MatSe project (*http://macke.wiwi.hu-berlin.de/matse/index2.html*), and the DecisionNet project (*http://dnet.sm.nps.navy.mil/*) are examples and prototypes of systems that can provide distributed authoring, distribution of software components, decision technologies, and so on for users. The ftp site mentioned earlier, (*ftp://ftp.bwl.uni-kiel.de/pub/operations-research/psplib/*), is an example of a repository of data sets and problem generation programs that can be executed for benchmarking and computational testing purposes.

MMM [81]explores issues in the distributed creation and dissemination of statistical software modules, available on distributed mathematical servers, in a dynamic, interoperable, and seamless environment. The MMM software implements a distributed computing environment with Web browsers as user interface to MMM middleware, which links consumers to services via interoperable access to mathematical servers.

MatSe is another example of how to connect to remotely installed software services to provide high–end user interfaces to link to mathematical servers on the Internet. The objective of MatSe is to support mathematical research on new and efficient algorithms. MatSe is developed for an association of research groups at several universities in Germany. In addition to standard Internet technologies, MatSe uses technologies that provide access to algorithms and algorithmic meta-knowledge, developed at the University of Tübingen, Germany, and the MMM system mentioned earlier. MatSe also enables project management using ProGen for generating project instances, illustrating the integration capability underlying the development of such systems.

DecisionNet, a digital library of executable decision technologies, is a distributed collection of decision technologies that are accessible and executable over the World Wide Web; these technologies are used by consumers, and are maintained by their providers at their local sites. DecisionNet does not contain the technologies themselves but has meta-information about the technologies and has the software infrastructure that facilitates transactions between consumers and providers of decision technologies [24].

Extending these ideas to the project scheduling models and methods results in two architectural choices for delivering decision support, a server–centric and a client–centric alternative, respectively [115, 141]. The server–centric architecture, shown in Figure 2, requires that the data be sent to the algorithm, residing on a remote server. The algorithm is published in executable format and the server contains the interface to accept the data in the desired format, pass it to the solver, and either notify the user of completion of the execution or publish the results. While the advantages are that the user only needs a web browser to access and work in this environment, which also provides platform independence, there are some drawbacks. Depending on the load on the server and the network, the processing could be delayed. In addition, if the project is very large, large amounts of data need to be shipped to the server. A more critical problem exists in the context of confidential data, such as in high security government projects or while bidding on highly competitive projects. The exchange of data on the network may compromise confidentiality.

The client–centric architecture, illustrated in Figure 3 gets the method to the data, thus shifting processing from server to the client. This requires the algorithm or heuristic to be implemented as a Java applet [2, 121] or ActiveX control [35], the web browser should be capable of executing this technology, and the user needs to trust the provider of the solver, especially in the case of ActiveX control, since it executes on the user's machine. The advantages are that the confidential data remains secure, and the size of the data does not pose a problem. However, the client side needs more software and tools to handle execution of the scheduling programs, and if these are ActiveX controls, there needs to be a verification mechanism that prevents the programs from corrupting the user's machine.

Users can make choices regarding these architectures depending on the size of the project data, confidentiality requirements, frequency of scheduling (e.g., high frequency in a reactive scheduling environment), platform independence of the software used to implement the algorithms, and ability to use and integrate local data with the remote application. As researchers develop and implement new and more efficient solvers for the project scheduling problem classes outlined in earlier sections, it becomes imperative to make these available for comparison, benchmarking, and evaluation studies as well as provide practitioners with state–of–the–art techniques that add value and can be executed with considerable ease.

7 Conclusions and Future Research

This paper has summarized an extensive array of research on the various aspects of the project scheduling problem. We have also compiled, for the first time, information about all the data sets used by researchers for testing and evaluation purposes. We believe that the integration of the optimal solution algorithms and heuristic procedures for the various classes of problems with the data generation methods and data representation issues, and deployment capabilities via commercial software, prototype decision support systems, and the WWW, provide a comprehensive view of the state of research and practice in the domain of deterministic, single project scheduling.

Several research directions are indicated by this overview. First, while minimizing makespan and maximizing NPV have been the dominant objectives of much of the research to date, there are considerable opportunities in investigating other objectives that combine these with costs, leveling resources, and so on. Many of these problems, being difficult combinatorial optimization problems, require heuristic approaches for solving problems of practical size and optimal approaches for validating, comparing and benchmarking the solutions given by heuristics.

Second, given the increasingly global nature of project bidding, evaluation, and implementation, integrating these objectives and constraints with the resulting models and methods into supply chain management issues is still to be explored. This requires considerable investigation of the issues related to the payment scheduling problem, multi-mode problems, and time/cost, time/resource and cost/quality trade-off problems. Extensions of these to different types of contracts, application of resource, capital and other types of constraints, and addition of bonuses, penalties and deadlines, can significantly improve the practical value and facilitate the integration of project scheduling problems into supply chain management concerns.

A critical third direction is the deployment issue. As seen from the summary of commercial software, it is obvious that much of the research has not yet found its way into practice. This has implications for the realism, validity and value of the research. Active practice can inform research while, at the same time, improving the efficiency and effectiveness of project bidding, contract negotiation, project scheduling, and implementation phases. Research issues include WWW–based integration concerns that take into account confidentiality and size of project data, remote and on–ite execution of methods for reactive scheduling and schedule repair, and so on. This is especially important in helping to manage the "virtual team", on which very little research exists. Adams and Adams [1] report that sharing adequate levels of information across distances and the lack of a common system to transmit and discuss this information is crucial to the success of virtual projects in the future.

Fourth, methodologies for solving project scheduling type problems have definitively left the realm of constructive approaches such as simple priority rule–based heuristics and enumeration–based exact procedures. Instead, the door has been swung open to modern machine learning approaches from the field of artificial intelligence. Examples are the new powerful local search methods which, by operating on activity list–based neighborhoods, escaped the problem of feasibility. In the field of exact methods, parallel computations and innovative new concepts from the field of classical job shop scheduling mark the way to new benchmark frontiers.

Finally, widely accepted problem generators and benchmark sets including the state of the art solutions are by the virtue of the Internet beginning to be available for every researcher. This should initiate fruitful competition and collaboration between the research and practitioner communities based on well accepted standards which will bring forth additional and valuable new developments in areas such as problem classification, extensions of the generator capabilities to cover all these classifications, and applications in new domains.

References

[1] Adams, J.R. and Adams, L.L. The virtual project: Managing tomorrow's team today. *PM Network*, 11(1):37-42, 1997.

[2] Arnold, K. and Gosling, J. The Java Programming Language. Addison Wesley Publishing Company, 1996.

[3] Agrawal, M.K. and Elmaghraby, S.E. and Herroelen, W.S. DAGEN: A generator of testsets for project activity nets. *European Journal of Operational Research*, 90:376–382, 1996.

[4] Ahn, T. and Erenguc, S.S. Resource constrained project scheduling problem with multiple crashable modes. Technical report, College of Business Administration, University of Florida, FL 32611 Gainesville, USA, 1995.

[5] Ahn, T. and Erenguc, S.S. The resource constrained project scheduling problem with multiple crashable modes: A heuristic procedure. Technical report, College of Business Administration, University of Florida, FL 32611 Gainesville, USA, 1995.

[6] Alvarez–Valdés, R. and Tamarit, J.M. Algoritmos heurísticos deterministas y aleatorios en secuenciacón de proyectos con recursos limitados. *Qüestiió*, 13:173–191, 1989.

[7] Alvarez–Valdés, R. and Tamarit, J.M. Heuristic algorithms for resource–constrained project scheduling: A review and an empirical analysis. In Słowiński, R. and Węglarz, J., editor, *Advances in project scheduling*, pages 113–134. Elsevier, Amsterdam, 1996.

[8] Antonisse, J.M. and Hee, K.M. and Lenstra, J.K. Resource–constrained project scheduling: An international exercise in DSS development. *Decision Sciences*, 4(??):??–??, 1988.

[9] Arora, R.K. and Sachdeva, R.K. Distributed simulation of resource constrained project scheduling. *Computers & Operations Research*, 16:295–304, 1989.

[10] Artigues, C. and Roubellat, F. and Billaut, J.C. Characterization of a set of schedules in a resource–constrained multi–project scheduling problem with multiple modes. Technical Report LAAS Report 96097, Labratoire d'Analyse et d'Archtiecture des Systémes du C.N.R.S., Toulouse, 1996.

[11] Assad, A.A. and Wasil, E.A. Project management using a microcomputer. *Computers & Operations Research*, 13(2/3):231–260, 1986.

[12] Baar, T. and Brucker, P. and Knust, S. Tabu–search algorithms for the resource–constrained project scheduling problem. Technical Report ???, Osnabrücker Schriften zur Mathematik, Fachbereich Mathematik/Informatik, 49069 Osnabrück, 1997.

[13] Balas, E. Machine sequencing via disjunctive graphs: An implicit enumeration algorithm. *Operations Research*, 17(??):941–957, 1969.

[14] Balas, E. Project scheduling with resource constraints. In Beale, E.M.L., editor, *Applications of mathematical programming techniques*, pages 187–200. English University Press, London, 1971.

[15] Bandelloni, M. and Tucci, M. and Rinaldi, R. Optimal resource leveling using non–serial dynamic programming. *European Journal of Operational Research*, 78:162–177, 1994.

[16] Baroum, S.M. and Patterson, J.H. A comparative evaluation of cash flow weight heuristics for maximizing the net present value of a project. Technical report, Indiana University, 1993.

[17] Bartusch, M. and Möhring, R.H. and Radermacher, F.J. Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, 16:201–240, 1988.

[18] Bein, W.W. and Kamburowski, J. and Stallmann, M.F.M. Optimal reduction of two–terminal directed graphs. *SIAM Journal on Computing*, 221(6):1112–1129, 1992.

[19] Beldiceanu, N. and Bourreau, E. and Rivreau, D. and Simonis, H. Solving resource–constrained project scheduling problems with CHIP. Technical report, Cosytec, Parc Club Orsay–Université, 4 rue Jean–Rostand, 91893 Orsay Cedex, France, 1996.

[20] Bell, C.E. and Han, J. A new heuristic solution method in resource–constrained project scheduling. *Naval Research Logistics*, 38:315–331, 1991.

[21] Bell, C.E. and Park, K. Solving resource–constrained project scheduling problems by A*–search. *Naval Research Logistics*, 37:61–84, 1990.

[22] Bey, R.B. and Doersch, R.H. and Patterson, J.H. The net present value criterion: Its impact on project scheduling. *Project Management Quarterly*, 12(2):35–45, 1981.

[23] Błażewicz, J. and Cellary, W. and Słowiński, R. and Węglarz, J. *Scheduling under resource constraints — Deterministic Models*. Baltzer, Basel, 1986.

[24] Bhargava, H. and Krishnan, R. and Muller, R. Decision support on demand: Emerging electronic markets for decision technologies. *Decision Support Systems*, forthcoming.

[25] Boctor, F.F. Some efficient multi–heuristic procedures for resource–constrained project scheduling. *European Journal of Operational Research*, 49:3–13, 1990.

[26] Boctor, F.F. Heuristics for scheduling projects with resource restrictions and several resource–duration modes. *International Journal of Production Research*, 31(11):2547–2558, 1993.

[27] Boctor, F.F. An adaptation of the simulated annealing algorithm for solving resource–constrained project scheduling problems. Technical Report 94–48, Faculte des Sciences de l'Administration, Universite Laval, Quebec, 1994. erscheint in IJPR.

[28] Boctor, F.F. A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. *European Journal of Operational Research*, 90:349–361, 1996.

[29] Böttcher, J. and Drexl, A. and Kolisch, R. and Salewski, F. Project scheduling under partially renewable resource constraints. Technical Report 398, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1996.

[30] Bowman, E.H. The schedule–sequencing problem. *Operations Research*, 7:621–624, 1959.

[31] Brinkmann, K and Neumann, K. Heuristic procedures for resource–constrained project scheduling with minimal and maximal time lags: The resource leveling and the minimum project–duration problem. *Journal of Decision Systems*, 5:129–155, 1996.

[32] Brucker, P. An efficient algorithm for the job–shop–problem with two jobs. *Computing*, 40:353–359, 1988. nicht in Literaturordner enthalten.

[33] Brucker, P. and Schoo, A. and Thiele, O. A branch & bound algorithm for the resource–constrained project scheduling problem. Technical Report 178, Osnabrücker Schriften zur Mathematik, Fachbereich Mathematik/Informatik, 49069 Osnabrück, 1996.

[34] Carruthers, J.A. and Battersby, A. Advances in critical path methods. *Operational Research Quarterly*, 17(4):359–380, 1966.

[35] Chappell, D Understanding ActiveX and OLE. Microsoft Press, 1996.

[36] Christofides, N. and Alvarez–Valdés, R. and Tamarit, J.M. Project scheduling with resource constraints: A branch and bound approach. *European Journal of Operational Research*, 29:262–273, 1987.

[37] Clough, R.A. and Sears, G.A. *Construction project Management*. John Wiley and Sons, 1991.

[38] Cooper, D.F. Heuristics for scheduling resource–constrained projects: An experimental investigation. *Management Science*, 22(11):1186–1194, 1976.

[39] Cooper, D.F. A note on serial and parallel heuristics for resource–constrained project scheduling. *Foundations of Control Engineering*, 2(4):131–133, 1977.

[40] Davies, E.M. An experimental investigation of resource allocation in mulitactivity projects. *Operational Research Quarterly*, 24:587–591, 1973.

[41] Davis, E.W. *An exact algorithm for the multiple constrained project scheduling problem*. PhD thesis, Yale University, 1969.

[42] Davis, E.W. Project scheduling under resource constraints — Historical review and categorization of procedures. *AIIE Transactions*, 5:297–313, 1973.

[43] Davis, E.W. and Heidorn, G.E. An algorithm for optimal project scheduling under multiple resource constraints. *Management Science*, 17:803–816, 1971.

[44] Davis, K.R. and Stam, A. and Grzybowski, A. Resource constrained project scheduling with multiple objectives: A decision support approach. *Computers & Operations Research*, 19(7):657–669, 1992.

[45] Dayanand, N. Scheduling payments in projects: An optimization based approach. Unpublished PhD. Dissertation, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213, 1996.

[46] Dayanand, N. and Padman, R The payment scheduling problem in project networks. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[47] Dayanand, N. and Padman, R Payments in projects: A contractor's model. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[48] Dayanand, N. and Padman, R A simulated annealing approach for scheduling payments in projects. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[49] Dayanand, N. and Padman, R Project contracts and payment schedules: The client's problem. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[50] Dayanand, N. and Padman, R On modeling payments in projects. *Journal of the Operational Research Society*, forthcoming.

[51] Dayanand, N. and Padman, R Progress payments in projects: Scheduling heuristics for the client. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[52] de Reyck, B. and Herroelen, W. Assembly line balancing by resource–constrained project scheduling techniques — A critical appraisal. Technical report, Department of Applied Economics, Katholieke Universiteit Leuven, 1995.

[53] de Reyck, B. and Herroelen, W. On the use of the complexity index as a measure of complexity in activity networks. *European Journal of Operational Research*, 91:347–366, 1996.

[54] de Reyck, B. and Herroelen, W. A branch–and–bound procedure for the resource–constrained project scheduling problem with generalized precedence relations. *European Journal of Operational Research*, ??:??–??, 1997. to appear.

[55] de Wit, J. and Herroelen, W. An evaluation of microcomputer-based software packages for project management. *European Journal of Operational Research*, 49:102–139, 1990.

[56] Deckro, R.F. and Hebert, J.E. Resource constrained project crashing. *OMEGA International Journal of Management Science*, 17(1):69–79, 1989.

[57] Deckro, R.F. and Hebert, J.E. A multiple objective programming framework for tradeoffs in project scheduling. *Engineering Costs and Production Economics*, 18:255–264, 1990.

[58] Demeulemeester, E. Minimizing resource availability costs in time–limited project networks. *Management Science*, 41(10):1590–1598, 1995.

[59] Demeulemeester, E. and Dodin, B. and Herroelen, W.S. A random activity network generator. *Operations Research*, 41:972–980, 1993.

[60] Demeulemeester, E. and Herroelen, W. A branch–and–bound procedure for the multiple resource–constrained project scheduling problem. *Management Science*, 38(12):1803–1818, 1992.

[61] Demeulemeester, E. and Herroelen, W. New benchmark results for the resource–constrained project scheduling problem. Technical report, Department of Applied Economics, Katholieke Universiteit Leuven, 1995. to appear in MS.

[62] Demeulemeester, E. and Herroelen, W. A branch–and–bound procedure for the generalized resource–constrained project scheduling problem. *Operations Research*, ??(??):??–??, 1997.

[63] Demeulemeester, E. and Herroelen, W. and Van Dommelen, P. An optimal recursive search procedure for the deterministic unconstrained max–npv project scheduling problem. Technical report, Department of Applied Economics, Katholieke Universiteit Leuven, 1996.

[64] Demeulemeester, E.L. and Herroelen, W.S. and Elmaghraby, S.E. Optimal procedures for the discrete time/cost trade–off problem in project networks. *European Journal of Operational Research*, 88:50–68, 1996.

[65] Doersch, R.H. and Patterson, J.H. Scheduling a project to maximize its present value: A zero–one programming approach. *Management Science*, 23(8):882–889, 1977.

[66] Domschke, W. and Drexl, A. Kapazitätsplanung in Netzwerken. *OR Spektrum*, 13:63–76, 1991.

[67] Drexl, A. and Grünewald, J. Nonpreemptive multi–mode resource–constrained project scheduling. *IIE Transactions*, 25(5):74–81, 1993.

[68] Elmaghraby, S.E. *Activity networks: Project planning and control by network models.* Wiley, New York, 1977.

[69] Elmaghraby, S.E. Project bidding under deterministic and probabilistic activity durations. *European Journal of Operational Research*, 49:14-34.

[70] Elmaghraby, S.E. and Herroelen, W.S. The scheduling of activities to maximize the net present value of projects. *European Journal of Operational Research*, 49:35–49, 1990.

[71] Elsayed, E.A. Algorithms for project scheduling with resource constraints. *International Journal of Production Research*, 20(1):95–103, 1982.

[72] Egelse, R.W. Simulated annealing: A tool for operational research. *European Journal of Operational Research*, 46:271-281, 1990.

[73] Erenguc, S.S. and Tufekci, S. and Zappe, C.J. The solution of the time/cost trade-off problem with discounted cash flows using generalized benders decomposition. Working Paper, University of Florida, 1991.

[74] Farid, F. and Manoharan, S. Comparative analysis of resource–allocation capabilities of project management software package. *Project Management Journal*, 24(2):35–44, 1996.

[75] Franck, B. Empirische Untersuchung von Prioritätsregeln für die ressourcenbeschränkte Projektplanung mit zeitlichen Minimal– und Maximalabständen. In Kleinschmidt, P. and Bachem, A. and Derigs, U. and Fischer, D. and Leopold–Wildburger, U. and Möhring, R., editor, *Operations Research Proceedings 1995*, pages 144–149. Springer, Berlin, 1996.

[76] Gilbreath, R.D. *Managing Construction Contracts.* John Wiley and Sons, Inc., 1992.

[77] Glover, F. Tabu Search, Part I. *ORSA Journal on Computing*, 1(3):190-206, 1989.

[78] Glover, F. Tabu Search, Part II. *ORSA Journal on Computing*, 2(1):190-206, 1990.

[79] Goldberg, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA, 1989.

[80] Grinold, R.C. The payment scheduling problem. *Naval Research Logistics Quarterly*, 19(1):123–136, 1972.

[81] Gunther, O., Muller, R., Schmidt, P., Bhargava, H. And Krishnan, R. MMM: A web-based system for sharing statistical computing modules. *Internet Computing*, 1(3):59– 68.

[82] Hartmann, S. Project scheduling with multiple modes: A genetic algorithm. Technical Report 435, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1997.

[83] Hendrickson, C. And Au, T. *Project Management for Construction.* Prentice Hall, Englewood Cliffs, New Jersey.

[84] Hartmann, S. and Drexl, A. Project scheduling with multiple modes: A comparison of exact algorithms. Technical Report 430, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1997.

[85] Hartmann, S. and Sprecher, A. A note on "'Hierarchical models for multi–project planning and scheduling"'. *European Journal of Operational Research*, 94:377–383, 1996.

[86] Hastings, N.A.J. On resource allocation in project networks. *Operational Research Quarterly*, 23(2):217–221, 1972.

[87] Herroelen, W. Resource–constrained project scheduling — The state of the art. *Operational Research Quarterly*, 23:168–173, 1972.

[88] Herroelen, W. and Demeulemeester, E. and de Reyck, B. Resource–constrained project scheduling — A survey of recent developments. Technical report, Department of Applied Economics, Katholieke Universiteit Leuven, 1996.

[89] Herroelen, W. and Demeulemeester, E. and de Reyck, B. A classification scheme for project scheduling problems. Technical report, Departement of Applied Economics, Katholieke Universiteit Leuven, 1997.

[90] Herroelen, W.S. and van Dommelen, P. and Demeulemeester, E.L. Project network models with discounted cash flows — A guided tour through recent developments. *European Journal of Operational Research*, 100:97–121, 1997.

[91] Herrolen, W. and Gallens, E. Computational experience with an optimal procedure for the scheduling of activities to maximize the net present value of projects. *European Journal of Operational Research*, 65:274–277, 1993.

[92] Hendry, M. Implementing EDI. Artech House Publisher, 1993.

[93] Icmeli, O. and Erenguc, S. A tabu search procedure for the resource constrained project scheduling problem with discounted cash flows. *Computers & Operations Research*, 21(8):841–853, 1994.

[94] Icmeli, O. and Erenguc, S.S. A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows. Technical report, Decision and Information Sciences Departement, University of Florida, 1995.

[95] Icmeli, O. and Erenguc, S.S. A branch and bound procedure for the resource constrained project scheduling problem with discounted cash flows. *Management Science*, 42(10):1395–1408, 1996.

[96] Icmeli, O., Erenguc, S.S. and Zappe, C.J. Project scheduling problems: A survey. *International Journal of Operations and Production Management*, 13(11):80-91, 1993.

[97] Icmeli, O. and Rom, W. Analysis of the characteristics of projects in diverse industries. Technical report, Cleveland State University, Department of Operations Management and Business Statistics, 1995.

[98] Icmeli, O. and Rom, W. Solving the resource constrained project scheduling problem with optimization subroutine library. Technical report, Cleveland State University, Department of Operations Management and Business Statistics, 1995.

[99] Icmeli, O.T. and Rom, W.O. Ensuring quality in resource constrained project scheduling. Technical report, Operations Management and Business Statistics Departement, Cleveland State University, Cleveland, OH 44115, USA, 1996.

[100] Johnson, R.V. Resource constrained scheduling capabilities of commercial project management software. *Project Management Journal*, 22(4):39–43, 1992.

[101] Johnson, T.J.R. *An algorithm for the resource–constrained project scheduling problem.* PhD thesis, Massachusetts Institute of Technology, 1967.

[102] Jüngen, F.J. and Kowalczyk, W. An intelligent interactive project management support system. *European Journal of Operational Research*, 84:60–81, 1995.

[103] Kim, Y.-D. A backward approach in list scheduling algorithms for multi–machine tardiness problems. *Computers & Operations Research*, 22:307–319, 1995.

[104] Kohlmorgen, U. and Schmeck, H. and Haase, K. Experiences with fine–grained parallel genetic algorithms. *Annals of Operations Research*, ??(??):??–??, 1997.

[105] Kolisch, R. Resource allocation capabilities of commercial project management systems. Technical report, Christian–Albrechts–Universitäet zu Kiel, Kiel, Germany, 1997. submitted to Interfaces.

[106] Kolisch, R. Efficient priority rules for the resource–constrained project scheduling problem. *Journal of Operations Management*, 14(3):179–192, 1996.

[107] Kolisch, R. Serial and parallel resource–constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90:320–333, 1996.

[108] Kolisch, R. and Drexl, A. Adaptive search for solving hard project scheduling problems. *Naval Research Logistics*, 43:23–40, 1996.

[109] Kolisch, R. and Drexl, A. Local search for nonpreemptive multi–mode resource– constrained project scheduling. *IIE Transactions*, ??(??):??–??, 1997.

[110] Krishnamoorthy, M.S. and Deo, N Complexity of the minimum–dummy-activities problem in a PERT network. *Networks*, 9:189-194, 1979.

[111] Kolisch, R. and Hempel, K. Auswahl von Standardsoftware, dargestellt am Beispiel von Programmen für das Projektmanagement. *Wirtschaftsinformatik*, 38(4):399–410, 1996.

[112] Kolisch, R. and Sprecher, A. PSPLIB — a project scheduling problem library. *European Journal of Operational Research*, 96:205–216, 1997.

[113] Kolisch, R. and Sprecher, A. and Drexl, A. Characterization and generation of a general class of resource–constrained project scheduling problems: Easy and hard instances. Technical Report 301, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1992.

[114] Kolisch, R. and Sprecher, A. and Drexl, A. Characterization and generation of a general class of resource–constrained project scheduling problems. *Management Science*, 41(10):1693–1703, 1995.

[115] Krishnan, R. and Padman, R. On using web technologies to architect DSS: The case of support requirements planning. *Proceedings of the International Society of Decision Support Systems Conference*, July 1997.

[116] Koza, J.R. *Genetic programming: On the programming of computers by natural selection*. MIT Press, 1992.

[117] S.R. Lawrence. Resource constrained project scheduling – A computational comparison of heuristic scheduling techniques. Technical report, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, 1985.

[118] Lee, J.–K. and Kim, Y.–D. Search heuristics for resource constrained project scheduling. *Journal of the Operational Research Society*, 47:678–689, 1996.

[119] Leon, V.J. and Balakrishnan, R. Strength and adaptability of problem–space based neighborhoods for resource–constrained scheduling. *OR Spektrum*, 17(2/3):173–182, 1995.

[120] Li, R.K.–Y. and Willis, J. An iterative scheduling technique for resource–constrained project scheduling. *European Journal of Operational Research*, 56:370–379, 1992.

[121] Ly, E. Distributed java applets for project management on the web. *Internet Computing*, 1(3):21-26, 1997.

[122] Maroto, C. and Tormos, P. Project management: an evaluation of software quality. *International Transactions on Operational Research*, 1(2):209–221, 1994.

[123] Mausser, H.E. and Lawrence, S.R. Exploiting block structure to improve resource–constrained project schedules. Technical report, University of Colorado, Graduate School of Business Administration, 1995.

[124] Mingozzi, A. and Maniezzo, V. and Ricciardelli, S. and Bianco, L. An exact algorithm for project scheduling with resource constraints based on a new mathematical formulation. Technical Report 32, Department of Mathematics, University of Bologna, 1994.

[125] Moder, J.J. and Phillips, C.R. and Davis, E.W. *Project management with CPM, PERT and precedence diagramming*. Van Nostrand Reinhold, New York, 3. edition, 1983.

[126] Möhring, R. and Müller, R. and Radermacher, F.J. Advanced DSS for scheduling: Software engineering aspects and the role of Eigenmodels. *Annals of Operations Research*, ??(??):??–??, 1997. to appear.

[127] Möhring, R.H. Minimizing costs of resource requirements in project networks subject to a fixed completion time. *Operations Research*, 32:89–120, 1984.

[128] Mori, M. and Tseng C.C. A genetic algorithm for multi–mode resource constrained project scheduling problem. *European Journal of Operational Research*, 100:134–141, 1997.

[129] Müller, R. and Krishnan, R. and Schmidt, P. Accessing "computational" services over the WWW: The MMM project, 1995.

[130] Naphade, K.S. and Wu, S.D. and Storer, R.H. Problem space search algorithms for the resource–constrained project scheduling problem. Technical report, Lehigh University, Department of Industrial & Manufacturing Systems Engineering, 1995.

[131] Nazareth, T. and Verma, S. and Bhattacharya, S. and Bagchi, A. The multiple resource–constrained project scheduling problem: A breadth–first approach. Technical report, Indian Institute of Management, Calcutta, 1996.

[132] Neumann, K. and Zhan, J. Heuristics for the minimum project duration problem with minimal and maximal time lags under fixed resource constraints. *Journal of Intelligent Manufacturing*, 6:145–154, 1995.

[133] Norbis, M. and Smith, J.M. An interactive decision support system for the resource constrained scheduling problem. *European Journal of Operational Research*, 94:54–65, 1996.

[134] Oğuz, O. and Bala, H. A comparative study of computational procedures for the resource constrained project scheduling problem. *European Journal of Operational Research*, 72:406–416, 1994.

[135] Özdamar, L. A genetic approach to a general category project scheduling problem. Technical report, Department of Industrial Engineering, Marmara University, Göztepe, 81040 Istanbul, 1996.

[136] Özdamar, L. and Ulusoy, G. A local constraint based analysis approach to project scheduling under general resource constraints. *European Journal of Operational Research*, 79:287–298, 1994.

[137] Özdamar, L. and Ulusoy, G. A survey on the resource–constrained project scheduling problem. *IIE Transactions*, 27:574–586, 1995.

[138] Özdamar, L. and Ulusoy, G. An iterative local constraint based analysis for solving the resource constrained project scheduling problem. *Journal of Operations Management*, 14(3):193–208, 1996.

[139] Özdamar, L. and Ulusoy, G. A note on an iterative forward/backward scheduling technique with reference to a procedure by Li and Willis. *European Journal of Operational Research*, 89:400–407, 1996.

[140] Padman, R. Choosing Solvers in Decision Support Systems: A Neural Network Application in Resource-Constrained Project Scheduling. *Recent Developments in Decision Support Systems*, eds. C. Holsapple and A. Whinston, Springer Verlag, Berlin, 101, 539-574.

[141] Padman, R. Web technologies for project scheduling decision support systems. Presented at the Scheduling and Heuristic Search Workshop, University of California, Riverside, May 8, 1997.

[142] Padman, R. and Smith–Daniels, D.E. Early-tardy cost trade-offs in resource constrained projects with cash flows: An optimization guided heuristic approach. *European Journal of Operational Research*, 64:295–311, 1993.

[143] Padman, R. and Zhu, D. On integrating knowledge sources using problem spaces: An application in project scheduling. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[144] Padman, R. and Smith–Daniels, D.E. and Smith–Daniels, V.L. Heuristic scheduling of resource–constrained projects with cash flows. *Naval Research Logistics*, 44:364–381, 1997.

[145] Padman, R. and Roehrig, S. A genetic programming approach for heuristic selection in constrained project scheduling. *Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies* eds. R.S. Barr, R.V. Helgason, and J.L. Kennington, Kluwer Academic Publishers, Norwell, MA, 405-421.

[146] Patterson, J.H. Alternate methods of project scheduling with limited resources. *Naval Research Logistics Quarterly*, 20:767–784, 1973.

[147] Patterson, J.H. Project scheduling: The effects of problem structure on heuristic performance. *Naval Research Logistics Quarterly*, 20:95–123, 1976.

[148] Patterson, J.H. A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem. *Management Science*, 30(7):854–867, 1984.

[149] Patterson, J.H. and Davis, E.W. A comparison of heuristic and optimum solutions in resource–constrained project scheduling. *Management Science*, 21:944–955, 1975.

[150] Patterson, J.H. and Huber, W.D. A horizon–varying, zero–one approach to project scheduling. *Management Science*, 20:990–998, 1974.

[151] Patterson, J.H. and Roth, G.W. Scheduling a project under multiple resource constraints: A zero–one programming approach. *AIIE Transactions*, 8:449–455, 1976.

[152] Patterson, J.H. and Slowinski, R. and Talbot, F.B. and Weglarz, J. An algorithm for a general class of precedence and resource constrained scheduling problems. In Slowinski, R. and Weglarz, J., editor, *Advances in project scheduling*, pages 3–28. Elsevier, 1989.

[153] Patterson, J.H. and Slowinski, R. and Talbot, F.B. and Weglarz, J. Computational experience with a backtracking algorithm for solving a general class of precedence and resource–constrained scheduling problems. *European Journal of Operational Research*, 49:68–79, 1990.

[154] Pesch, E. *Learning in automated manufacturing*. Physica, Heidelberg, 1994.

[155] Pinedo, M. *Scheduling — Theory, algorithms, and systems*. Prentice Hall, Englewood Cliffs, 1995.

[156] Pollack–Johnson, B. Hybrid structures and improving forecasting and scheduling in project management. *Journal of Operations Management*, 12:101–117, 1995.

[157] A.A.B. Pritsker, L.J. Watters, and P.M. Wolfe. Multiproject scheduling with limited resources: A zero–one programming approach. *Management Science*, 16:93–107, 1969.

[158] Radermacher, F.J. Scheduling of project networks. *Annals of Operations Research*, 4:227–252, 1985/86.

[159] Russell, A.H. Cash flows in networks. *Management Science*, 16(5):357–373, 1970.

[160] Russell, R.A. A comparison of heuristics for scheduling projects with cash flows and resource restrictions. *Management Science*, 32(10):1291–1300, 1986.

[161] Sampson, S.E. and Weiss, E.N. Local search techniques for the generalized resource constrained project scheduling problem. *Naval Research Logistics*, 40:665–675, 1993.

[162] Schneider, W.G. and Hieber, D. Software zur ressourcenbeschränkten Projektplanung. Technical Report 494, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 1997.

[163] Schrage, L. Solving resource–constrained network problems by implicit enumeration — Nonpreemptive case. *Operations Research*, 18:263–278, 1970.

[164] Schutten, J.M.J. List scheduling revisited. *Operations Research Letters*, 18:167–170, 1996.

[165] Schwindt, C. ProGen/max: Ein Problemgenerator für ressourcenbeschränkte Projektplanungsprobleme mit zeitlichen Minimal– und Maxmialabständen. Technical report, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 1995.

[166] Schwindt, C. Generation of resource–constrained project scheduling problems with minimal and maximal time lags. Technical Report 489, Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe, 1996.

[167] Sepil, C. Comment on Elmaghraby and Herroelen's "The scheduling of activities to maximize the net present value of projects". *European Journal of Operational Research*, 73:185–187, 1994.

[168] Shaffer, L.R. and Ritter, J.B. and Meyer, W.L. *The critical–path method.* McGraw Hill, New York, 1965.

[169] Simpson, W.P. and Patterson, J.H. A multiple–tree search procedure for the resource–constrained project scheduling problem. *European Journal of Operational Research*, 89:525–542, 1996.

[170] Słowiński, R. Multiobjective project scheduling under multiple–category resource constraints. In Słowiński, R. and Węglarz, J., editor, *Advances in Project Scheduling*, pages 151–167. Elsevier, Amsterdam, 1989.

[171] Słowiński, R. and Soniewicki, B. and Węglarz, J. DSS for multiobjective project scheduling. *European Journal of Operational Research*, 79:220–229, 1994.

[172] Smith–Daniels, D.E. and Aquilano, N.J. Using a late–start resource–constrained project schedule to improve project net present value. *Decision Sciences*, 18:617–630, 1987.

[173] Smith–Daniels, D.E. and Padman, R. and Smith–Daniels, V.L. Heuristic scheduling of capital constrained projects. *Journal of Operations Management*, 14(3):241–254, 1996.

[174] Smith–Daniels, D.E. and Smith–Daniels, V.L. Maximizing the net present value of a project subject to materials and capital constraints. *Journal of Operations Management*, 7:33–45, 1987.

[175] Speranza, M.G. and Vercellis, C. Hierarchical models for multi–project planning and scheduling. *European Journal of Operational Research*, 64:312–325, 1993.

[176] Sprecher, A. *Resource constrained project scheduling — Exact methods for the multi–mode case.* Springer, Berlin, 1994.

[177] Sprecher, A. Solving the RCPSP efficiently at modest memory requirements. Technical Report 425, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1996.

[178] Sprecher, A. and Drexl, A. Solving multi–mode resource–constrained project scheduling by a simple, general and powerful sequencing algorithm. Part I: Theory. Technical Report 385, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1996.

[179] Sprecher, A. and Drexl, A. Solving multi–mode resource–constrained project scheduling problems by a simple, general and powerful sequencing algorithm. Part II: Computation. Technical Report 386, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, 1996.

[180] Sprecher, A. and Hartmann, S. and Drexl, A. An exact algorithm for project scheduling with multiple modes. *OR Spektrum*, ??(??):??–??, 1997.

[181] Sprecher, A. and Kolisch, R. Semi–active, active, and non–delay schedules for the resource–constrained project scheduling problem. *European Journal of Operational Research*, 80:94–102, 1994.

[182] Stinson, J.P. *A branch and bound algorithm for a general class of multiple resource–constrained scheduling problems*. PhD thesis, Graduate School of Business Administration, University of North Carolina, 1976.

[183] Stinson, J.P. and Davis, E.W. and Khumawala, B.M. Multiple resource–constrained scheduling using branch and bound. *AIIE Transactions*, 10:252–259, 1978.

[184] Storer, R.H. and Wu, S.D. and Vaccari, R. New search spaces for sequencing problems with application to job shop scheduling. *Management Science*, 38:1495–1509, 1992.

[185] Syslo, M.M. On the computational complexity of the minimum–dummy–activities problem in a PERT network. *Networks*, 14:37-45, 1984.

[186] Talbor, B. Resource–constrained project scheduling with time–resource tradeoffs: The nonpreemptive case. *Management Science*, 28(10):1197–1210, 1982.

[187] Talbot, F.B. and Patterson, J.H. An efficient integer programming algorithm with network cuts for solving resource–constrained scheduling problems. *Management Science*, 24(11):1163–1174, 1978.

[188] Tavares, L.V. Multicriteria scheduling of a railway renewal program. *European Journal of Operational Research*, 25:395–405, 1986.

[189] Thesen, A. Heuristic scheduling of activities under resource and precedence restrictions. *Management Science*, 23(4):412–422, 1976.

[190] Thesen, A. Measures of the restrictiveness of project networks. 7:193–208, 1977.

[191] Thomas, P.R. and Salhi, S. An investigation into the relationship of heuristic performance with network–resource characteristics. *Journal of the Operational Research Society*, 48(1):34–43, 1997.

[192] Ulusoy, G. and Özdamar, L. Heuristic performance and network/resource characteristics in resource–constrained project scheduling. *Journal of the Operational Research Society*, 40(12):1145–1152, 1989.

[193] Ulusoy, G. and Özdamar, L. A constraint–based perspective in resource constrained project scheduling. *International Journal of Production Research*, 32(3):693–705, 1994.

[194] Valls, V. and Pérez, M.A. and Quintanilla, M.S. Heuristic performance in large resource–constrained projects. Technical Report 92–2, Departament D'Estadistica I Invecigacio Operativa, Universitat de Valencia, 1992.

[195] Wasil, E.A. and Assad, A.A. Project management on the PC: Software, applications, and trends. *Interfaces*, 18:75–84, 1983.

[196] Whitehouse, G.E. and Brown, J.R. GENRES: An extension of Brooks algorithm for project scheduling with resource constraints. *Computers & Industrial Engineering*, 3:261–268, 1979.

[197] Wiest, J.D. Some properties of schedules for large projects with limited resources. *Operations Research*, 12(??):395–418, 1964.

[198] Wiest, J.D. A heuristic model for scheduling large projects with limited resources. *Management Science*, 13(6):B–359–B–377, 1967.

[199] Zhan, J. Heuristics for scheduling resource–constrained projects in MPM networks. *European Journal of Operational Research*, 76:192–205, 1994.

[200] Zhu, D. Intelligent decision support strategies for constrained project scheduling. Unpublished PhD dissertation, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

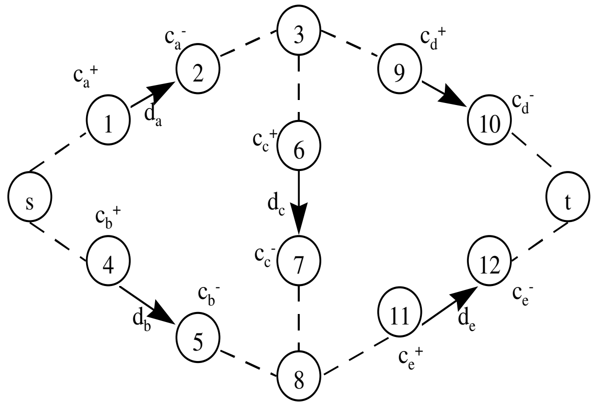
[201] Zhu, D. and Padman, R. Tabu search for scheduling resource-constrained projects with cash flows. Working paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[202] Zhu, D. and Padman, R. Connectionist approaches for solver selection in constrained project scheduling. *Annals of Operations Research*, forthcoming.

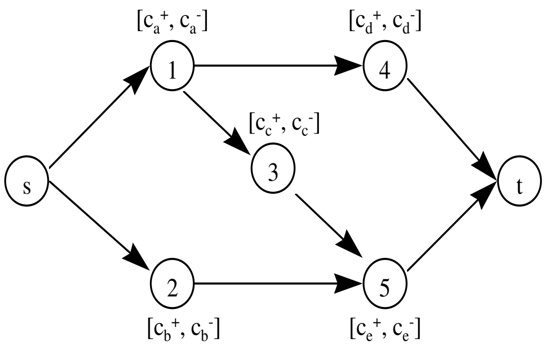
[203] Zhu, D. and Padman, R. A multiheuristic combination model for scheduling resource–constrained projects with cash flows. Working Paper, The Heinz School, Carnegie Mellon University, Pittsburgh, PA 15213.

[204] Zhu, D. and Padman, R. A cooperative multi-agent approach to constrained project scheduling. *Interfaces in Computer Science and Operations Research: Advances in Metaheuristics, Optimization, and Stochastic Modeling Technologies* eds. R.S. Barr, R.V. Helgason, and J.L. Kennington, Kluwer Academic Publishers, Norwell, MA, 367-381.

(a) Activity-on-arc model with cash flows (A.H. Russell, 1970)



(b) Modified activity-on-arc model with cash flows (Padman et al., 1997)



(c) Activity-on-node model with cash flows (Zhu, 1995)

Figure 1: Problem Representation

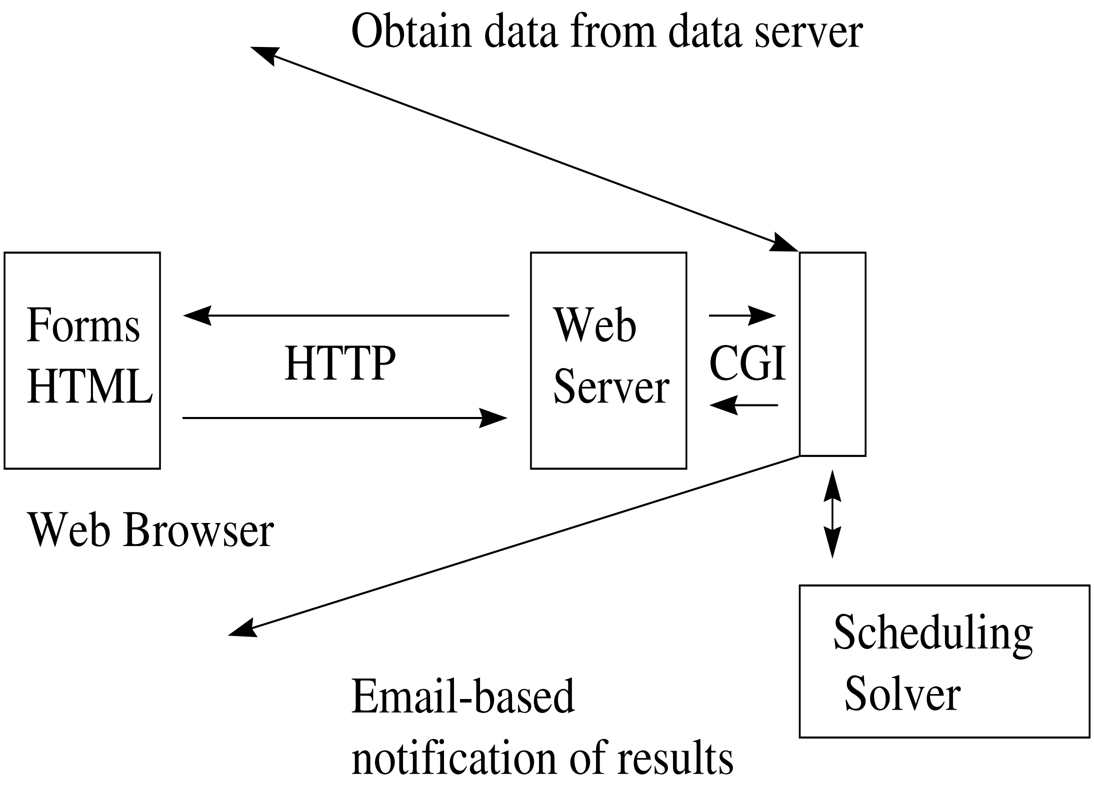


Figure 2: Server–Centric Architecture

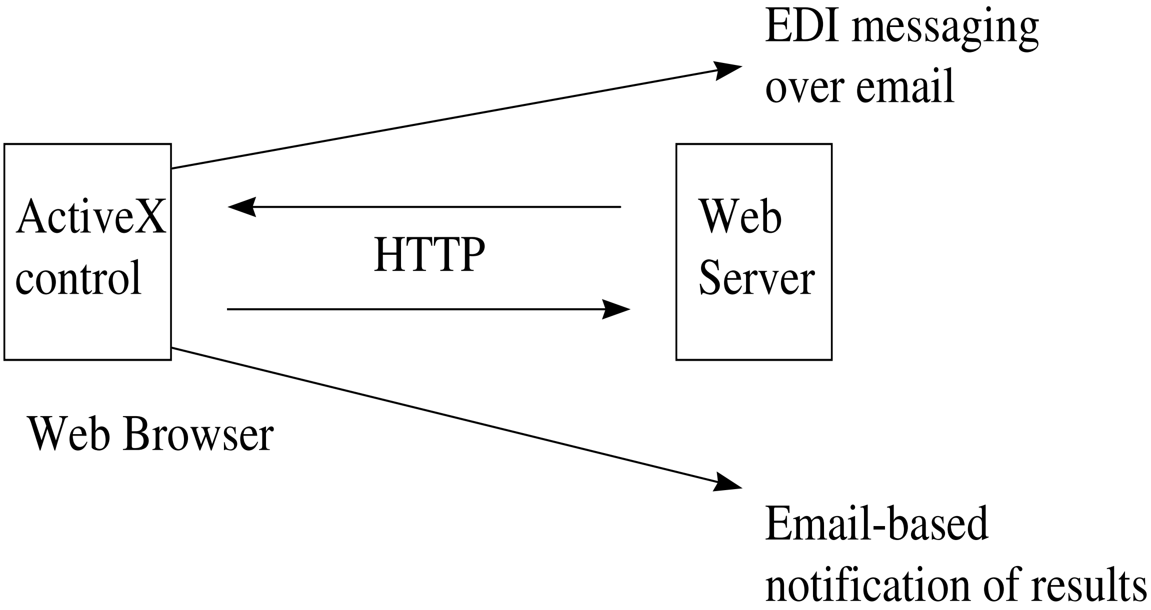


Figure 3: Client–Centric Architectures