A MULTI-OBJECTIVE APPROACH TO THE RESOURCE CONSTRAINED PROJECT SCHEDULING

by
METE BAYYİĞİT
B.S. in I.E., Boğaziçi University, 1993

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science
in
Industrial Engineering

Boğaziçi University
July, 1994

A MULTI-OBJECTIVE APPROACH
TO
THE RESOURCE CONSTRAINED PROJECT SCHEDULING

APPROVED BY

Prof. Dr. Gündüz ULUSOY ....
(Thesis Supervisor)

Assoc. Prof. Linet ÖZDAMAR ....
(Marmara University)

Assoc. Prof. Gülay BARBAROSOĞLU ....

DATE OF APPROVAL ..20.7.1994............

To the best mother of the universe,
Sezer BAYYİĞİT

ACKNOWLEDGEMENTS

I would like to express my deeepest gratitude to my thesis supervisor Prof. Dr. Gündüz ULUSOY for his invaluable guidance, support and encouragement during my graduate study in Industrial Engineering department of Boğaziçi University and all the stages of this study.

Associate Prof. Linet ÖZDAMAR, who, I believe, is an expert in project management area, has been with me in every second of this thesis, like while modelling the problem, coding the computer program, running the code, performing the output analysis, writing the thesis report, etc. I can not forget the days and nights when we did not sleep working on this thesis. I am proud to have known such an academician, who never bores and gets tired while studying. I promise to be her colleague in the near future.

I would like to thank Associate Prof. Gülay BARBAROSOĞLU for her valuable comments and suggestions as a member of my thesis committee.

My special thanks go to, my girlfriend, Aysun COŞKUN, who could not have a sleep for nights while working on this thesis with me. I always appreciate her sincere attitudes towards me, patience and encouragement.

I would also like to thank Ebru KORULAR, one of the best industrial engineers in the industry, for her help while coding the computer program and E. Emel KURTULUŞ, whom I see as my sister, for her technical and moral support.

Finally, my deepest gratitude is for all of the members of my family, especially my mother, who respect and endure my life philosophy, for their support, encouragement and patience throghout all phases of my life.

ABSTRACT

Many examples where the resource constrained project scheduling framework is applicable for modelling and solving the scheduling problem under consideration can be found both in the literature and in practice. Recently, more realistic and accurate resource models and objectives have been developed. These models provide better means for coping with practical issues faced in real world applications.

In the resource constrained project scheduling, the heuristics that are existent in the literature aim at either minimizing the tardiness or maximizing the Net Present Value (NPV). In a project network, the constraints of the resource constrained project scheduling problem include precedence constraints representing the technological network and resource limitations. The resource constraints used in resource constrained project scheduling models have recently become more descriptive of real life situations. It is assumed that there is a project due date that is contractual and causes a penalty for each tardy period.

NPV is the exponential or discrete discounting of the cash flows accrued at the event occurrence times to the present. Tardiness in a project stands for the total number of late periods over the project due date. In real life, it is essential to make money while keeping the image of the company at its highest level on customers' eyes. Therefore, in this study new heuristics are generated and their performance according to NPV and tardiness criteria is statistically compared with that of the heuristics from the literature. The trade-off between NPV and tardiness is easily illustrated. The new heuristics are named as hybrid heuristics, because their priorities are based on the weighted combination of the two objectives, namely minimizing tardiness and maximizing NPV. The scheduling algorithm used in this study is iterative, making consecutive forward/backward scheduling passes. The advantages of using the dynamic activity time windows in pushing and pulling event times are shown.

ÖZET

Kaynak kısıtlı proje çizelgelemesi çerçevesinde, çizelgeleme problemini modellemeye ve çözmeye uygun hem literatürde hem de pratikte birçok örnek bulunmaktadır. Özellikle son zamanlarda daha gerçekçi kaynak modelleri ve amaç fonksiyonları geliştirilmiştir.

Kaynak kısıtlı proje çizelgelemede, literatürde varolan sezgisel yöntemler ya gecikmeyi enazlamayı ya da Net Şimdiki Değeri (NŞD) ençoklamayı amaçlamışlardır. Bir proje seriminde, kaynak kısıtlı proje çizelgeleme probleminin kısıtları, teknolojik serimi temsil eden öncelik kısıtlarını ve kaynak kısıtlarını ihtiva etmektedir. Bu modelde kontrata bağlı olabilen bir proje termini ve her geç kalınan zaman için bir ceza öngörülmektedir.

NŞD olayların meydana geliş zamanlarında oluşan nakit akışlarının üssel veya kesitli bir biçimde şimdiki zamana indirgenmiş halidir. Projedeki gecikme ise proje termininin üstüne toplam geç kalınan zaman sayısını göstermektedir. Gerçek yaşamda, şirketin müşteri gözündeki durumunu müşteri memnuniyetini proje terminine uyarak sağlamak ve para kazanmak esastır. Bu yüzden bu çalışmada bu iki amacı da gözönüne alan yeni sezgisel yöntemler önerilmiş ve NŞD ve gecikme kriterlerine göre bunların performansları, literatürdeki sezgisel yöntemlerin performanslarıyla istatiksel bir biçimde karşılaştırılmıştır. Bu sayede, NŞD ve gecikme arasındaki ödünleşim kolayca gösterilmektedir. Yeni geliştirilen sezgisel yöntemler melez olarak adlandırılmıştır; çünkü öncelikler, gecikmeyi enazlamayı ve Net Şimdiki Değeri ençoklamayı amaçlayan iki amaç fonksiyonunun ağırlıklı bileşimine bağlı kılınmıştır. Bu çalışmada kullanılan çizelgeleme algoritması, ardışık ileri/geri çizelgeleme geçişleri yapan iteratif bir algoritmadır. Olay zamanlarını ileri iterek ve öne çekerek faaliyet zaman aralıkları kullanmanın avantajları da gösterilmiştir.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

A project is a coordinated set of activities consuming resources which are generally limited in quantity. Since resources have costs, a project requires reimbursements during its progress. Reimbursements may be contractually bound to the occurrence of certain events. Consequently, an event which does not lead to a reimbursement has a cash outflow associated to the activities which are going to start once it occurs. Hence, when all activities leading to an event are completed, a cash inflow or outflow possibly exists according to the event's nature. In their efforts to maximize the net return on investments, project managers are faced with the difficult problem of timing the cash flows that take place at event occurrence times which are in turn dependent on how resources are allocated among activities. Furthermore, there might be a project due date by which the project must be completed. In case the due date is violated, management could be penalized for each tardy period.

Project Management is less difficult if only precedence relationships constrain the activity schedule. PERT (Program Evaluation and Review Technique) and CPM (Critical Path Method) techniques, which require polynomial time computations, readily provide allowable time windows for scheduling the activities. The objective is to complete the project by the minimal possible time permitted by the precedence relationships. However, in practice activities do not get completed on their own; rather, they consume resources during their progress. The scheduling problem becomes difficult to solve when the required resources are available in limited amounts, because the issue of allocating scarce resources among competing activities must be considered in optimizing a specific objective. This problem is known as the resource constrained project scheduling problem and is NP-hard [4]. Coping with this problem is a theoretical challenge and an important contribution for practitioners.

The resource constrained project scheduling problem does not represent an isolated area of research. On the contrary, it subsumes a wide area of scheduling problems.The constraints of the resource constrained project scheduling problem include precedence constraints representing the technological network and resource limitations. The resource constraints used in resource constrained project scheduling models have recently become more descriptive of real life situations. They are classified under the following categories: renewable, nonrenewable, and doubly constrained. Renewable resources are constrained on a period-by-period basis [see, for example, 26]. Labor can be considered as a renewable resource if it is used every day and limited on a daily basis. Nonrenewable resources are constrained on a project basis [29,30].The project budget or raw materials become nonrenewable resources if the total consumption over the whole project duration is limited to a certain value. Finally, doubly constrained resources are simultaneously constrained on a period and project basis. If, for example, the cash consumed is limited on a daily basis and also for the overall project, then it is a doubly constrained resource. Resource constraints complicate the representation of the problem, and the more accurately they describe the actual problem, the more difficult they become to handle.

In this study, we develop heuristic rules for dealing with the resource allocation problem where the project has a due date and resources are renewable. They have both tardiness and Net Present Value (NPV) considerations for enhancing both objectives. Furthermore, we apply the heuristic rules in an iterative scheduling algorithm which realizes consecutive forward/backward scheduling passes. A feature of the iterative algorithm is that its mechanics are based on the concept of dynamic time windows. Dynamic time windows are generated by the use of information pertaining to the schedule obtained in the previous iteration. This feature is well manipulated by the heuristic rules which use this information as well as NPV related information. We compare our heuristic rules with tardiness and NPV related rules from the literature (Dumond and Mabert [15], Ulusoy and Özdamar [36], Russell [29], Baroum and Patterson [2], Padman and Smith-Daniels [24]).

2. LITERATURE SURVEY

In the project scheduling literature the objective of minimizing the makespan is dealt with more frequently than the objective of maximizing the project's Net Present Value (NPV) (to cite a few project-duration oriented references see, e.g., Alvarez-Valdes and Tamarit [1], Bell and Park [3], Boctor [6, 7], Christofides et al. [8], Davis and Patterson [11], Pritsker et al. [27], Demeulemeester and Herroelen [12], Li and Willis [20], Stinson et al. [34], Ulusoy and Özdamar [36, 37]). However, a project with an optimal duration is not necessarily the most economically beneficial. Often, time-based objectives conflict with cost-based objectives such as maximizing a project's Net Present Value (NPV). The maximization of the project NPV is a more suitable objective function since a project is practically undertaken for making money. Unfortunately, fewer researchers deal with money-oriented objectives (Baroum and Patterson [2], Doersch and Patterson [13], Elmaghraby and Herroelen [16], Russell [28], Russell [29], Smith-Daniels and Aquilano [32], Padman and Smith-Daniels [24], Speranza and Vercellis [33], Yang et al. [39]).

Among the few references cited for money-oriented objectives above, Elmaghraby and Herroelen [16] and Russel [28] deal with the problem disregarding the resource constraints and considering only the precedence constraints among activities. Doersch and Patterson [11] treat cash as the single limited resource in their model constrained as a nonrenewable resource. In the other references, all models include renewable resource constraints with the exception of Speranza and Vercellis [33] which include nonrenewable resource constraints as well in their hierarchical multiple objective structure. The authors who omit the resource constraints demonstrate that finding the optimal NPV subject to precedence constraints is not difficult.

However, when resource consumption is limited, the project scheduling problem is shown to be NP-hard [4] for the objective function of minimizing makespan. Recent models including nonrenewable resource constraints and discrete time-resource functions for representing the activities are proposed by Weglarz et al. [38], Slowinski [30, 31], Talbot [35] and Patterson et al. [26]. Experimentation indicates that finding the optimal project NPV is more difficult than finding the optimal makespan ([Patterson et al. [26], Yang et al. [39]). The reason of this complexity is the intertwined relationship existing between resource bottlenecks and the NPV criterion. For example, suppose that an activity leading to a negative cash flow is scheduled on its late start time as suggested by Elmaghraby and Herroelen [16]. Due to resource limitations, this activity may impede the start of another activity leading to a large positive cash flow, resulting in a lower NPV. Hence, the optimal NPV can be discovered by examining every possible activity start time.This leads to an explosion of feasible solutions (Patterson et al. [26]).

Another problem that the project manager is faced with is the project's contractual due date. Due date related objective functions are discussed in the literature [17, 18, 19, 30]. Some authors (Bock and Patterson [5], Dumond [14] , Dumond and Mabert [15]) aim at assigning due dates to each project in a dynamic environment where projects arrive randomly over time. A general situation found in practice is the necessity of completing the project by its due date while maximizing its NPV. In recent references (Talbot [35], Patterson et al. [26]), NPV is maximized while a due date exists as a hard constraint. Multiple objective approaches where efficient solutions with respect to time- and cost-based objectives are generated also appear in the literature (Daniels [9], Slowinski [30,31], Speranza and Vercellis [33]).

The overview of the related studies according to the chosen objective are as follows:

2.1. The Objective of Maximizing Net Present Value

Having thoroughly exploited the objective of minimizing makespan, researchers realized that in practice a project cannot be isolated from cost considerations. Neither can a feasibility study prior to the launching of the project be carried out without considering the scheduling of activities. Since the costs incurred during the project depend on the actual progress of the scheduled activities and since the schedule itself is closely related to resource constraints other than cash, researchers have included the consideration of cash flows explicitly in resource-constrained project scheduling formulations.

2.1.1. Models with Precedence Constraints only

The pioneering work about the maximization of NPV is performed by Russell [27]. However, resource constraints are ignored. The nonlinear formulation model of Russell with the precedence constraints models is approximated by an LP whose dual results in a transshipment model where the nonlinear objective function is linearized using Taylor's series. This model will be explained in detail in "Theoretical Basis of NPV Related Heuristics" part of the thesis. In this model, the flows on the arcs provide updated occurrences of events which are used to provide an improved LP approximation. This process is repeated until successive occurring times are identical.

Elmaghraby and Herroelen [16] state that the activities with positive net cash flows should be scheduled as early as possible and the ones with negative cash flows as late as possible. The authors provide a simple procedure which is based on this property of the optimal solution for maximizing NPV, with only precedence constraints considered. The authors conclude that the earliest completion time schedule is not necessarily the optimal one when the time value of money is considered. They point out that while trying to optimize NPV, activities with negative cash flows might be delayed indefinitely, unless a project due date or large positive cash flows exists at the final phases of the project.

2.1.2. Models with Renewable Resource Constraints

Under the presence of resource constraints, maximizing NPV is not an simple task. A complex relationship exists between resource bottlenecks and the NPV criterion. For example, an activity with a negative cash flow may be scheduled on its late start time. Due to resource limitations, this activity may impede the start of another activity with a large positive cash flow, resulting in a lower NPV. Hence, for finding the optimal solution every possible start time for the activities must be taken into account, leading to a very detailed examination of feasible solutions.

Yang et al. [38] propose an integer programming approach which is an extension of the implicit enumeration procedure. The authors solve 10 problems with at most 21 activities under two cash flow models: one where each activity has a positive terminal value and one where terminal values are free to be positive or negative. There is also a bonus at the completion of the project, and a project due date is incorporated into the model. The authors note that as the project due date moves away from the optimal makespan, solution times explode exponentially. The model with general terminal values is more difficult to solve compared to the model with only positive terminal values. The computational experience reported in this paper does not encourage researchers to pursue optimization approaches for the NPV criterion.

Russell [28] approaches this problem heuristically through optimization guided dispatching rules which take input from the unconstrained (no resource constraints) problem solutions provided by the procedure in [27]. He has hypothesized that the heuristic rules utilizing information from the unconstrained cash flow analysis would be more effective than the heuristic rules whose objective was to minimize project makespan. Superior performance in this analysis means that a higher net present value of the project would be achieved subject to precedence relationships, cash flows and reource constraints.He tested this hypothesis using analysis of variance to test for equality of means in the heuristics' performance on test problems. General cash flows are assigned to events in an activity-on-arc representation of the project. The unconstrained solution represents an upper bound for this problem, and the optimal transshipment flows on arcs approximate the marginal costs of delaying the activities. Dispatching rules which are based on these marginal costs are compared with those designed for the makespan criterion, and the results demonstrate that in small problems (21-26 activities) no distinction can be made among makespan- and NPV-oriented dispatching rules with respect to an NPV objective. Furthermore, in larger problems where the resource constraints are not very tight, the makespan-oriented rule, MINSLACK, outperforms NPV-oriented rules.

Padman and Smith-Daniels [46] use the cash flow model described in [28] and apply optimization guided heuristics in a scheduling algorithm where at every scheduling decision time, the project with the resource constraints ignored is reoptimized by fixing the start times of completed and in progress activities. The authors investigated whether the NPV of project cash flows may be improved by releasing an activity to the queue of schedulable activities as soon as its precedence constraints are satisfied. Target release rules are also proposed. Immediate release rules imply that an activity becomes schedulable as soon as its predecessors ar completed. However, in a target release rule an activity does not become schedulable until its optimal starting time recommended by the unconstrained solution. These rules aim at minimizing earliness and tardiness with respect to the target dates. Their hypothesis was that the early release of activities to the schedule queue may not only result in an improved NPV, but also reduce potential bottlenecks induced by resource conflicts.

Therefore, increasing the range of scheduling alternatives by allowing all activities to join immediately the queue of schedulable activities may be preferred once they are precedence feasible.The heuristics they provide require the evaluation of earliness costs and tardiness penalties for each activity that are given by the relaxed optimization model. The earliness cost of an activity is the sum of the dual prices of its successors, whereas the tardiness cost is the sum of the dual prices of its predecessors. The authors compare their rules to the cash flow weight approach used by Baroum and Patterson[2] and the non-updated optimization-guided rules given in [28]. An experiment which covers a large number of problems with 48 and 110 activities demonstrates that the target release rules imbedded in the reoptimization algorithm are superior.

Smith-Daniels and Aquilano [62] consider a model where cash outflows occur at the beginning of each activity and a lump sum is received at the completion of the project. The authors propose a heuristic backward scheduling algorithm which obtains comparatively higher NPV values than the ones obtained by the earliest start forward scheduling algorithm due to the resulting right-shifted schedule. The backward algorithm treats the project duration provided by the earliest start schedule as the project due date and applies a late start scheduling procedure starting from this due date.

2.1.3. Models with Nonrenewable Resource Constraints

The simplest model representing nonrenewable resources is proposed by Doersch and Patterson [13]. In this model, the activity related inflows and outflows, which occur at discrete times during the progress of an activity, are added and subtracted respectively from the corresponding daily cash limit. The cash not utilized at the end of a day is carried on to the next day. The activities must be scheduled in such a manner that the daily net cash flow is not negative. Thus, cash is treated as the only nonrenewable resource. Renewable resource constraints are not considered in the model. The constraints also include precedence and activity completion constraints. The objective consists of maximizing NPV of the cash inflows and outflows as they occur during the progress of activities, and also the NPV of a penalty (reward) realized by the project depending on its lateness (earliness). Problems of less than 30 activities are reported to be solved to optimality using this formulation.

2. 2. The Objective of Minimizing Makespan

2.2.1. Renewable Resource Constraints

Intensive research has been conducted considering the objective of minimizing makespan subject to renewable resource constraints. More recent contributions are discussed here after a brief summary.

The zero-one integer programming formulation given by Pritsker et al. [27] applies to both single and multi-project problems under renewable resource constraints. This formulation has initiated studies [35] where improvements in the solution of the zero-one integer program are introduced. The formulation is also the backbone of more recent representations of the project scheduling problem [26] involving nonrenewable resources. However, in spite of the efforts to facilitate the solution of the integer program, the associated computational experience involves only small problems of at most 27 activities and 3 resource types.

There is an approach based on line balancing techniques where each activity is represented by a number of unit duration tasks equivalent to its duration. The resulting network is transformed into a shortest path network. This procedure, together with those due to Stinson et al. [34] and Talbot and Patterson [35], is evaluated by Patterson [25] with respect to computer storage, solution time and the number of problems solved optimally within a reasonable computation time.

The abundance of zero-one variables and the large number of constraints have led researchers to develop branch and bound procedures for the resource constrained project scheduling problem [8, 34]. The computational success of a branch and bound algorithm depends on its branching technique and the strength of its lower bound. Three recent contributions in this area [3, 8, 12] involve problem-specific branching techniques which are based on the resolution of resource conflicts. In other words, in these procedures, branching occurs only when a resource conflict is encountered. This seems to be the main explanation for their computational advantage over the standard branch and bound approach given by Stinson et al. [34]. It is shown that the procedure given by Christofides et al.[10] does not always produce the optimal solution. Hence, Demeulemeester and Herroelen [12] modify this algorithm to guarantee that it results in an optimal solution, employing a depth-first search in contrast to the breadth-first search used by Bell and Park [3]. The branching scheme in [12] consists of delaying some activities in progress in order to resolve resource conflicts which occur at a certain time point. A delay set consists of a group of activities in progress which, had they not been scheduled previously, release sufficient resources for the conflicting activity to be scheduled at the current time point. A minimal delay set is a set of activities no proper subset of which is a delay set. Each branch in the solution tree is represented by a minimal delay set. However, Bell and Park [3] represent a level of the tree by a minimal conflict set which is defined as a set of conflicting activities which are no longer in conflict if any element of the set is dropped. At each level, an element of this set represents a branch. This element is forced to be the successor of the earliest finishing activity among the remaining members of the same conflict set. Bell and Park [3] solve 110 standard test problems and report the number of generated and pruned nodes and CPU time. The same set of problems are solved by Demeulemeester and Herroelen [12], but only average CPU times are reported and they are not comparable with the results found in [3] due to differences in operating systems, coding and computer power. Christofides et al. [8] and Alvarez-Valdes and Tamarit [1] report results on some test problems, 40 with 25 activities and 3 resource types [8] and 38 with 51 and 103 activities with 6 resource types [1]. Özdamar and Ulusoy [22] experiment with both sets of problems: Patterson's 110 problems and the 78 problems provided in [8] and [1]. Özdamar and Ulusoy exercise a pruning heuristic using the branching scheme found in [3] with a depth-first search. The heuristic results in an average deviation of about 2 per cent from the optimum in both sets of problems.In the experimentation, it is observed that although the tree is pruned in a heuristic manner and the lower bound calculations are based on future resource conflicts and result in early pruning, the number of generated nodes is comparatively larger than those reported in [8] for problems with tight resource constraints. However, for the loosely constrained problems in [8] and for all of the 110 problems, the heuristic reduces the size of the tree drastically as compared to [8] and [3] respectively. Thus, the authors justify the conclusion that the branching scheme found in [3] is suitable for loose-moderate tightness of resource constraints, whereas the one found in [12] is specifically powerful in tightly constrained problems.

The effective lower bounds, dominance rules and branching techniques applied in the branch and bound procedures are observed to be insufficient for problems of practical size, however efficient they are.

The need to solve problems of practical size has motivated researchers to develop effective heuristics. This line of research was conducted by [1, 6, 68]. Heuristics are classified into two types [11]: serial heuristics, where activity priority is predetermined and remains fixed throughout the scheduling procedure, and parallel heuristics, where activity priority is updated each time an activity is scheduled. The network/resource characteristics of problems are analyzed to find out the conditions under which specific dispatching rules perform well consistently. Efforts are made to identify problem characteristics and their effect on dispatching rules [36]. The comparative performance of dispatching rules with respect to each other and to optimal results are evaluated [1, 6, 11]. The results show that although there are generally accepted good dispatching rules, their performance is problem dependent and they do not provide good solutions consistently. A strategy for using heuristics is given by Boctor [6] who proposes to run the problem each time with another well-reputed heuristic rule and keep the best solution obtained. Boctor also provides probabilities for obtaining optimal results for specific rule combinations.

Recent emphasis has been placed on developing heuristic procedures where problem dependency is eliminated. The common feature of these algorithms (Ulusoy and Özdamar [37]) is that they all consider resource and/or temporal conflicts among activities as the basic idea and progress in their scheduling procedures by resolving the conflicts through varying methods. For example, arbitrary disjunctive precedence arcs on conflicting activities are imposed to eliminate resource conflicts among concurrent activities in their single pass algorithm. Due to poor solution quality, a backtracking algorithm, which evaluates the effect of each added disjunctive arc on the makespan, is applied to the initial schedule. The backtracking algorithm is truncated when an improved solution cannot be found. Similar reasoning is used in the exact algorithm proposed by Bell and Park [3]. In an approach which does not require backtracking, precedence relationships are imposed on pairs and groups of conflicting activities in an effort to minimize the project duration. The result is a set of partial sequences feasible with respect to the imposed precedence constraints. They call their algorithm "Constraint Based Analysis" since the precedence setting rules are based directly on the temporal and resource constraints and are essential for minimizing makespan. Similar reasoning is utilized by Ulusoy and Özdamar [37] in their rule based algorithm which is called "Local Constraint Based Analysis". The difference in the two algorithms is that the former algorithm is applied on the whole project whereas Ulusoy and Özdamar [37] apply theirs on small segments of the project. The first approach requires a decision aid module to complete the set of feasible partial schedules, and is not time efficient since it has to reevaluate conflicts over all the activities of the project each time a precedence relationship is established. The second approach evaluates resource conflicts once and locally where a small number of schedulable activities are considered in a parallel scheduling algorithm. Consequently, it is time efficient and applicable to dynamic environments. The performance of such algorithms is less problem dependent than that of dispatching rules since they are not based on a priori insight but rather depend on the current temporal and resource constraints. The results of conflict based approaches are reported by Ulusoy and Özdamar [37]. The results show that on average, the project durations obtained by the procedure due to Bell and Han [4] is 2.44 per cent above the optimal solution, using the 110 test problems provided by Patterson[25]. Local Constraint Based Analysis, empowered by an iterative forward/backward scheduling algorithm, results in an average deviation of 1.14 per cent from the optimum on the same set of problems.

2.2.2. Nonrenewable Resource Constraints

A model where activities are represented by a discrete time-resource function is proposed by Slowinski [30] where time-based objectives, such as maximum lateness, or cost- based objectives, such as processing costs and total consumption of nonrenewable resources, are considered. However, the cost-based objectives do not take the time value of money into account, but consist of the minimization of total costs. The scheduling approach is preemptive and based on an LP formulation of the problem.

Talbot [35] and Patterson et al. [26] also propose models where activities have multiple modes. Scheduling is nonpreemptive. Talbot [35] considers the objective of minimizing makespan under the presence of a given budget and minimizing total nonrenewable resource consumption under the presence of a project due date. A zero-one integer formulation of the doubly-constrained problem with a discrete time function for describing activities is also given. The problem is solved to optimality by initially ordering activities using a good dispatching rule and applying an enumeration algorithm with backtracking. If a feasible solution is found to be acceptable, then the procedure can be stopped without reaching the optimal solution. This feature is due to the fact that instead of evaluating lower bounds for each possible partial schedule, the algorithm starts with the late finish times of the incumbent best schedule as upper bounds for each individual activity. An iteration of the algorithm finds an improved schedule with activity late finish times one unit less than the best incumbent schedule. Thus, each iteration is started with a complete feasible solution which can be accepted as satisfactory. It is reported that if there exists a good starting solution, the procedure quickly finds improved feasible solutions. Computational experience covers 10-activity problems with up to three different modes per activity. Although 20- and 30-activity problems cannot always be solved to optimality good approximate solutions are obtained within a short time. This formulation and solution procedure are extended [26] to include predetermined cash inflows and outflows taking place during the progress of an activity and a delayed cash flow after its completion. The delayed cash flow provides funds for realizing the remaining activities. Consequently, special care must be taken in deciding on the activity modes to be selected to avoid a premature stoppage of the project. In these papers, the objectives of maximizing NPV or minimizing makespan under renewable and nonrenewable resource constraints are considered. When maximizing NPV a project due date is imposed to prevent the indefinite delay of activities. The objective of maximizing NPV is harder to handle because a search is made by right shifting the resource feasible starting time of each activity with a negative net cash flow to its latest possible resource feasible starting time. It is reported that no problem could be solved to optimality within a reasonable computation time. For the case of minimizing makespan, the size of problems solvable to optimality is small (10-30 activities). The computational results show the impact of a good starting solution in finding the optimal solution. Although it is noted that good feasible solutions are obtained early in the procedure, the quality of these early solutions are not reported.

A heuristic approach to the above problem is given by Özdamar and Ulusoy [21]. This approach is an extension of LCBA [37] discussed previously. It consists of evaluating the activities and their modes locally, that is, at every scheduling point in a single pass parallel forward scheduling algorithm. Activities which are conflicting with respect to available resources are evaluated by certain conditional decision rules which prevent an unnecessary extension of the current project duration. The conditional decision rules discard some of the activity modes from consideration, because scheduling them at that point in time will lead to a longer project duration extension as compared to the rest. Resulting from this evaluation is the choice of the combination of activity modes which extends the makespan least. Activities in this combination which have not been discarded by the decision rules are scheduled at that time. The procedure compares favorably with well reputed dispatching rules adapted from the single-mode case to this problem. The dispatching rules are adapted to the multiple-mode case in the following manner: schedulable activities are listed according to their priorities and the operating modes of each activity are sorted in order of ascending duration. Then, starting from the top of the activity list, activities are scheduled in the first mode on their mode lists, which is resource-feasible with regard to the remaining renewable and nonrenewable resources. In a study made independently, Boctor [9] reports that the mode-selection rule used in [21] is the best performing rule among others for minimizing project duration.

A heuristic approach to the project scheduling problem with nonrenewable resource constraints, where an activity is represented by a single operating mode, is given by Li and Willis [20]. In their formulation, variable resource requirements exist during an activity's duration, and the availability of nonrenewable resources is renewed at fixed periods over time. An iterative forward/backward type of scheduling is carried out where activity start times in the forward schedule are utilized in the construction of the backward schedule. Similarly, the information provided by the backward schedule is used in constructing the forward schedule. The iterations stop when no further improvement is obtained in the project duration. Hence, the project duration is squeezed while starting the activities as late as possible. The basic idea behind the procedure is to obtain a schedule as short as possible while implicitly considering the minimization of the net present value of financing costs. Hence, the two objectives of minimizing makespan and the net present value of costs are naturally merged into a single objective. The authors compare their heuristic procedure with dispatching rules, used in a forward scheduling algorithm, on five sets of 18 problems, each representing different types of resources (renewable with constant resource limits, renewable with resource constraints varying over time, nonrenewable, doubly-constrained, doubly-constrained with varying renewable resource limits) and find it to be promising. Since optimal results are available only for the set of problems with renewable resource constraints with resource limits constant over time, the authors compare their procedure with the optimal solutions for this set of problems and report an average deviation of 5.17 per cent from optimum.

3. THEORETICAL BACKGROUND

3.1. The Model

In our model, there exist cash inflows or outflows, F_i, taking place at event occurrence times, T_i. We assume an activity-on-arc representation. Activities have duration d_{ij}, and renewable resource requirements, r_{jik}, where ij is the activity index and k is the resource index. The resource limit is denoted by R_k. There is also a project due date, DD, which, when violated, causes a penalty, p, accrued each period that the project is tardy. The penalty is taken as a percentage of the last cash inflow which occurs at the completion time of the project. The objective is to maximize the NPV of cash flows including the effect of the due date penalty. However, the minimization of tardiness is also measured, since it is usually not possible to reflect the loss of customer good will in terms of a monetary penalty.

3.2. Theoretical Basis of NPV Related Heuristics

The objective of maximizing NPV is considered in the pioneering work of Russell [28] where resource constraints are ignored. Russell's nonlinear formulation considers only the precedence constraints among activities and is approximated by an LP through Taylor's expansion where known and feasible event occurrence times are substituted. The dual of the LP model is a transshipment model. In the dual model, the dual prices (flows) on the arcs indicate the marginal cost of delaying an activity beyond its unconstrained (resource constraints omitted) optimal completion time. By complementary slackness, an activity with a positive dual price has a zero total slack time. Hence, the flows imply the optimal occurrence times of the events which are then used to provide an improved LP approximation in the next iteration. The process of solving the transhipment model with updated event times is repeated until successive event occurrence times are identical and the optimal solution is at hand.

The mathematical formulation of Russell can be described as follows:

Suppose there are m activites (arcs), with durations d_k (k=1,...,m) and n events (nodes), occuring at times T_i with associated net cash flows F_i (i=1,...,n). The preceding and succeeding nodes of activity k are denoted by i(k) and j(k). The initial event is numbered 1 and the terminal event n. All times are referred to the initial event as the datum so that T_1=0. Given a discount rate α, the unconstrained cash flow problem can be formulated as:

$$\text{maximize} \sum_{i=1}^{n} F_i \exp(-\alpha T_i)$$
$$\text{subject to } T_{j(k)} - T_{i(k)} \leq d_k, \; , k = 1,...,m \qquad (3.1)$$

This is a nonlinear programming problem. The objective function of the above problem can be linearized by the approximation using Taylor's series. A current nonoptimum but feasible solution T' is assumed. T' is a vector of event times T', which satisfies the duration constraints but does not neceesarily optimize the objective function. It is assumed that the optimal event times Ti are sufficiently near to the current value T'_i that the approximation in taking only the first, linear terms in the Taylor expansion of the present value is valid. Thus

$$\Sigma_i F_i \exp(-\alpha T_i) = \Sigma_i F_i \exp(-\alpha T'_i) - \Sigma_i (T_i - T'_i) F_i \alpha \exp(-\alpha T'_i) + ...$$
$$= \Sigma_i F'_i + \Sigma_i T'_i \alpha F'_i - \Sigma_i T_i \alpha F'_i \text{ where } F'_i = F_i \exp(-\alpha T'_i) \qquad (3.2)$$

Therefore instead of maximizing the original nonlinear objective $\Sigma_i F_i \exp(-\alpha T_i)$, the appoximate procedure is to maximize the linear objective $-\Sigma_i T_i \alpha F'_i$. The coefficient of T_i in this approximation is the negative of the discount rate times the cash flow at event i discounted to the initial event from its current scheduled time T'_i.

The approximate linear programming (primal) problem can be rewritten as

$$\text{maximize} - \sum_{i=2}^{n} \alpha F'_i T_i$$
$$\text{subject to } \Lambda T \leq -d$$
$$T_m - T_1 \geq DD \qquad (3.3)$$

where $F'_i = F_i \exp(-\alpha T'_i)$, T is the vector of event times (T_i), d the vector of event durations (d_k), Λ is the node-arc incidence matrix with zeroes throughout the kth row except for a 1 in the $i(k)^{th}$ column and -1 in the $j(k)^{th}$ row (k=1,...,m) and DD is the project due date. For purposes of formulating the dual one is free to assume that the T_i's are constrained to be ≥ 0 or not as one chooses. (Since T_1=0), the precedence relationships will themselves ensure that the T_i's are $\geq$ 0.) It will be assumed that the T_i's are not constrained in sign. the dual problem then becomes

$$\text{minimize} - \sum_{k=1}^{n} d_k f_k - f_{n+1} DD$$

$$\text{subject to} \quad \Lambda' f = -\alpha F'$$

$$f_k \geq 0, \; k = 1,\ldots,m \tag{3.4}$$

where Λ', the transpose of Λ, is the arc-node incidence matrix with all entries in the ith row zero except for a 1 in columns corresponding to activities which proceeded from the i^{th} node and -1 in columns corresponding to activities directed in the j^{th} node. It should be noticed that the dual problem has the form of a transshipment problem. f_{n+1} is the dual variable representing the due date constraint. As realized, the dual of the problem is a minimum cost network problem. The right hand sides of the constraints of the dual problem involve the event occurence times. In Russell's optimization procedure, event occurence times are initially set to their earliest start times given by CPM. The dual is resolved and through the Complementary Slackness Theorem, new event times are obtained. After adjusting the right hand sides accordingly, the problem is reoptimized until the right hand sides in consecutive iterations are equal.

The unconstrained optimal solution provided by Russell's procedure [28] is used by Russell [29] and Padman and Smith-Daniels [24] in developing heuristic rules for dealing with the resource constrained problem. The heuristic approach is based on generating optimization guided dispatching rules which take the optimal flows on the activities and event occurrence times from the unconstrained problem solution provided by the Russell's procedure. The unconstrained solution represents an upper bound for the resource constrained problem. Russell [29] solves the unconstrained optimization problem once whereas Padman and Smith-Daniels [24] reoptimize the problem at every scheduling decision time by reflecting the start times of completed and in progress activities to the constraints. Padman and Smith-Daniels [24] define earliness and tardiness costs for activities and base their their rules on these concepts. The authors compare their immediate release and target release rules with those of Russell's [29] and Baroum and Patterson's [2] which use the unconstrained solution once. Padman and Smith-Daniels note that Russell's best performing rule (TS/LAN) is superior when the resource constraints are not tight. Among the 16 heuristic rules collected form the literature (Baroum and Patterson [2], Russell [29], Padman et al. [23], Padman and Smith-Daniels [24]), Eight heuristic rules are selected as the best performing under different resource tightness levels. Two of these heuristics are target release rules. An interesting result is that the familiar makespan-oriented MINSLACK rule is also present among the best eight rules. This result is also valid in the results reported by Russell [29].

The optimization-guided scheduling algorithm of Padman and Smith-Daniels [24] is explained in detail as follows:

Step 1: Solve the unconstrained NPV project scheduling problem using the minimum cost network flow approach.

Step 2: Schedule eligible activities (all activities that are precedence feasible) using a heuristic priority rule until available resources are exhausted or the resource requirements of the remaining activities exceed the quantity that is available. To derive the fair cost of delaying activities in a partially completed schedule, events associated with complete activities are tied to the initial node. If all activities have been scheduled, stop.

Step 3: Delay the start times of eligible activities that were not scheduled to the next possible time at which resources become available. Add new precedence feasible activities to the queue of eligible activities.

Step 4: Reoptimize the NPV network flow problem with the modified event times and durations from Step 3 and proceed to Step 2.

The updated unconstrained problem is solved using the dual simplex algorithm developed for the minimum cost network flow.

3.3. The Iterative Scheduling Algorithm

The heuristic rules, which enhance both tardiness and NPV criteria, are supported by an iterative scheduling algorithm applying consecutive forward and backward scheduling passes. The advantages of backward scheduling are reported by Smith-Daniels and Aquilano [32] for the criterion of maximizing NPV, and Li and

Willis [20], Özdamar and Ulusoy [21] demonstrate its benefits when the criterion of minimizing makespan is considered. Backward scheduling tends to pull activities to the right towards their late start times and disposes of the resource bottlenecks occurring at the beginning phases of the project. Consecutive forward / backward scheduling iterations lead to a smoother resource profile and hence to a smaller makespan. In the presence of a project due date, the backward iteration attempts at a JIT schedule. In maximizing NPV, the consecutive iterations have the effect of right- and left- shifting the events along with their respective cash flows. NPV related heuristic rules when combined with tardiness / makespan related rules make good use of the advantages provided by dynamic activity time windows in pushing and pulling event times according to their corresponding cash flows.

Our iterative scheduling algorithm is described as follows:

The initial forward scheduling iteration starts at time zero, and the set of schedulable activities are identified. An activity becomes shedulable when its predecessors are completed and there are sufficient resources to meet its resource requirements. If there are no resource conflicts, all schedulable activities are scheduled immediately. Otherwise, a preselected dispatching rule is used to allocate scarce resources among competing activities. The scheduling time is updated each time an activity is completed. This procedure is repeated until all activities are scheduled.

Once a scheduling iteration is completed, the allowable time windows of activities and event occurrence times are updated in the following manner so that a link is established between two scheduling iterations:

$$lft_{ij} = \max\{DD, M\} - pstart_{ij} \qquad (3.5)$$

$$T_i = \max\{DD, M\} - \min_{i \in \{IS_{ij}\}} \{pstart_{ij}\}, \text{ if the previous iteration is a forwards pass} \qquad (3.6)$$

lft_{ij} = late finish time of activity ij.
M = the best makespan obtained up to the current scheduling iteration.
$pstart_{ij}$ = scheduled start time of activity ij in the previous iteration.
IS_{ij} = the set of activities starting from event i. (if the previous iteration is forward pass)
= the set of activities ending at event i. (if the previous iteration is backward pass)

Hence, the time window of an activity which is specified by its lft and the current scheduling time, t_{now} , is related to its previous scheduled start time. This link has the effect of reordering the priorities assigned to the activities during the next scheduling iteration. Furthermore, event occurrence times which are used in considering the NPV of cash flows are also dynamic and attached to previously scheduled activity starting times.

Having updated activity late finish times, the backward iteration starts at time zero. In backward scheduling, activities become schedulable when their successors are completed. Other than the definition of schedulability, backward scheduling is no different than forward scheduling. The forward/backward iterations continue until neither NPV nor tardiness improve.

3.4. Heuristic Rules

We compare the seven NPV oriented heuristic rules identified as best performing by Padman and Smith-Daniels [24] with tardiness oriented rules (Dumond and Mabert [15], Lawrence and Morton [19], Özdamar and Ulusoy [21]) which are valid for the objective of minimizing makespan/tardiness. We also develop our own rules merging the basic concepts behind the two criteria to enhance both objectives. Our rules are not optimization guided and are based simply on cash flows and dynamic activity time windows. For a fair comparison we test all heuristic rules using the iterative scheduling algorithm.

The seven best performing NPV oriented heuristic rules are listed in the original names used by Padman and Smith-Daniels [24]:

1. ITS (Target Schedule; Immediate Release): This rule schedules all activities based on the difference between the current time and an activity's target schedule time, where all activities are scheduled according to increasing order of difference. The slack time available to all activities is used as the primary indicator of the delay or advance required in scheduling activity. This heuristic implicitly represents the early-tardy trade-off by attempting to schedule each activity as closely as possible to its target schedule date. (Padman and Smith-Daniels [24])

2. IOCS (Opportunity Cost Scheduling; Immediate Release): This rule is intended to capture the opportunity cost incurred for scheduling activity k instead of

other activities in the queue. The opportunity therefore has two components, the tardiness penalties incurred by the delay in scheduling activities in the queue other than k, as well as the earliness cost potentially incurred in scheduling k ahead of its target time. Activities are scheduled in ascending order of opportunity cost,

$$\text{priority} = \sum_{k \varepsilon A_k} P_k \cdot d_k + [EC_k (T_{i(k)} - t)] \tag{3.7}$$

where P_k of activity k is the tardiness penalty defined as the cost of delaying the start of an activity by one time period from its currently scheduled time of $T_{i(k)}$ to $T_{i(k)} + 1$ and computed as the sum of the dual prices of the immediately preceding activities, EC_k of activity k is the earliness cost evaluated as the sum of the dual prices of the immediate successor activities of k, A_k is the set of remaining activities in the schedule queue if activity k were to be performed and t is the current time period. (Padman and Smith-Daniels [24])

3. ILTP/LEC (Lowest Tardiness Penalty/Lowest Earliness Cost; Immediate Release): The objective of the ILTP/LEC rule is to identify the activities resulting in early progress payments that are not directly connected to the final completion of the project. This rule is implemented in two queues. Activities with positive tardiness penalties are scheduled first in ascending order of P_k and those with zero tardiness penalties are scheduled subsequently in ascending order of $EC_k \cdot I_k$, where

$$I_k = T_{i(k)} - t_{now} \tag{3.8}$$

Thus $EC_k \cdot I_k$ computes the reduction in NPV if the activity is considered ahead of its target schedule date by I_k units of time. Activities with positive tardiness penalties are scheduled first, since they result in immediate progress payments, while activities are not selected from the second queue until the priority queue is exhausted, since the scheduling of activities with early costs results in early incursion of cash flows. (Padman and Smith-Daniels [24])

4. LTP/LAN (Lowest Tardiness Penalty/Lowest Activity Number; Target Release): In this rule, activities are not released to the schedule queue until their target

schedule time in the revised unconstrained solution. Activities are then scheduled in ascending order of tardiness penalties, P_k. Activities with zero tardiness penalties are scheduled according to the lowest activity number (LAN). (Padman et al. [23])

5. CFW/OCC (Cash Flow Weight /Opportunity Cost of Cash Flow; Target Release): An activity's cash flow weight (CFW) is computed as the sum of the cash flows of all the activities that logically succeed in the precedence network. In this rule, activities in the priority queue are scheduled according to maximum cash flow weight. The OCC selection rule establishes priorities by evaluating the cost of delaying all other activities currently in the queue in terms of their cash flow weights. (Padman et al. [23])

6. CFW/LAN (Cash Flow Weight /Lowest Activity Number; Immediate Release): The activity with maximum CFW is scheduled first, while ties are broken using LAN. (Baroum and Patterson [2])

7. TS/LAN (Target Scheduling/Lowest Activity Number; Immediate Release): The target scheduling (TS) rule assigns priority based on the maximum difference between the current revised early finish time for activity k and the optimal unconstrained finish time for that activity defined by

$$(EF_k - T_{j(k)}) \tag{3.9}$$

Ties are broken with LAN. (Russell [29])

All the above rules use Russell's [28] optimization procedure except for CFW/LAN, which is based on the cash flows occurring at event times. Note that ITS and TS/LAN are concerned with the reoptimized event occurrence times in specifying priorities for activities. IOCS, LTP/LAN and ILTP/LEC are concerned with the reoptimized earliness and/or tardiness costs of activities as well as the optimal event occurrence times provided by Russell's [28] procedure. CFW/LAN and CFW/OCC use the cash flows related to the events which logically succeed the schedulable activity under consideration.

For using the above rules in backward scheduling, we reversed the concepts of earliness and tardiness costs. In backward scheduling, the immediate successors of a schedulable activity are actually its immediate predecessors in the original network. Additionally, in backward scheduling the target starting time of an activity,ij, becomes the unconstrained optimal occurrence time of the activity's ending event, j. Another point is that in backward scheduling we reoptimize the unconstrained problem, by reversing the direction of all arcs and reversing the sign of all cash flows related to the events. Thus, we obtain meaningful optimal event occurrence times for a backward schedule which also starts at time zero.

The NPV related rules obtain their solutions in two scheduling iterations consisting of a forward and a backward iteration. The reason is that in their existing form they do not use any information related to dynamic time windows.

The rules which are only tardiness oriented are listed as:

8. MINSLACK (Minimum Total Slack Time): Priority to activity ij is given by:

$$(lft_{ij}-t_{now}-d_{ij}) \qquad (3.10)$$

Activities are scheduled in ascending order of priority. Note that at the beginning of every scheduling iteration activity lft are updated as described in the previous section. (MINSLACK is redefined in this manner by Özdamar and Ulusoy [21].)

9. LFT (Minimum Late Finish Time): Priority to activity ij is given by: (lft_{ij}). Activities are scheduled in ascending order of priority.

10. WRUP (Weighted Resource Utilization and Precedence) : Priority to activity ij is given by:

$$w * suc_{ij} + (1-w) * \Sigma\, (r_{ijk} / R_k). \text{ (Ulusoy and Özdamar [36])} \qquad (3.11)$$

Here, w denotes a weight in the interval [0,1] and suc_{ij} denotes the number of immediate successors of activity ij. This rule is implemented by increasing w by increments of length 0.1 until w reaches one. The maximum NPV and the minimum tardiness schedules are then saved as the best schedules found in the initial forward iteration and then at the end of each subsequent iteration. (The two schedules are probably different in each iteration, since the best schedule with respect to tardiness need not be the one with the highest NPV.)

Hence, dynamic time windows are made use of by tardiness oriented rules, except for WRUP, which is a list rule obtaining a solution in exactly two scheduling iterations.

Apart from the above ten rules, we define four hybrid rules. The word "dynamic" belonging to the slack time is related with the calculation of activity late finish times, while that belonging to the net present value of cash flows to the calculation of event occurence times. At the beginning of each iteration, they have to be recalculated by the formulae 3.5 and 3.6, respectively (page 18). The new four hybrid rules are as follows:

11. D-SLACK-D-NPVCF (Dynamic Minimum Slack Time/Dynamic Net Present Value of Cash Flows):

Priority is given to activity ij by:

$$priority_{ij} = \frac{(lft_{ij} - t_{now} - d_{ij})}{\left(\sum_{m \, \varepsilon \, \{S_{ij}\}} e^{-\alpha Tm} * F_m\right)} \qquad (3.12)$$

$\{S_{ij}\}$ is the set of events which logically succeed activity ij. Activities are scheduled in ascending order of priority. The occurrence times of events, T_m, in dynamic NPVCF rule are determined as follows: In the first forward iteration, the occurrence times of events with negative cash flows are set to their late start times provided by CPM and the ones with positive cash flows are set to their early start times. In subsequent iterations, they assume the scheduled occurrence times provided by the solution with the least tardiness obtained in the previous scheduling iteration.

This rule attempts at minimizing tardiness by MINSLACK rule which reflects dynamic activity time windows and at maximizing NPV by updating the occurrence times of succeeding nodes at consecutive iterations and netting their respective cash flows to their present value. The activities with positive denominators constitute the first priority queue. Once the activities in the first queue are scheduled, the next priority queue which consists of activities with zero denominators are scheduled in ascending order of Slack Time. Then, the last priority queue, which is constituted of activities with negative denominators, is activated after negating the denominator.

12. W-D-SLACK-NPVCF (Weighted Dynamic Minimum Slack Time/Net Present Value of Cash Flows):

Priority is given to activity ij by:

$$priority_{ij} = (1\text{-}w) * (lft_{ij}\text{-}t_{now}\text{-}d_{ij}) - (w) * \left\lceil \left(\sum_{m \, \varepsilon \, \{S_{ij}\}} e^{-\alpha T m} * F_m \right) \div c \right\rceil^{-} \quad (3.13)$$

This rule is the weighted version of rule # 11. Again, w is a weight in the interval [0,1] and the rule is implemented in the same manner as WRUP. T_m are calculated using the early and late event times due to CPM analysis in the forward iteration. In the backward iteration, T_m are calculated by taking the difference between the due date, DD, and activity late finish times,lft, provided by the initial CPM analysis. Hence, T_m are fixed to certain values determined by CPM analysis and at each forward iteration they assume the same values. The same goes for backward iterations. Notice that the NPV of the succeeding nodes' cash flows are reduced by a factor, c, and then the resulting value is truncated to the nearest lower integer. The scaling factor is taken to be 1000 in our testing environment, since cash flows are of the order of magnitude of thousands. Consequently, the orders of magnitude of Slack Time and NPV are brought to the same level. Activities are scheduled in ascending order of priority.

13. W-D-SLACK-D-NPVCF (Weighted Dynamic Minimum Slack Time/Dynamic Net Present Value of Cash Flows):

Priority is given to activity ij by:

$$priority_{ij} = (1\text{-}w) * (lft_{ij}\text{-}t_{now}\text{-}d_{ij}) - (w) * \left\lceil \left(\sum_{m \, \varepsilon \, \{S_{ij}\}} e^{-\alpha T m} * F_m \right) \div c \right\rceil^{-} \quad (3.14)$$

This rule is the dynamic version of W-D-SLACK-NPVCF rule with respect to event occurrence times calculated at each iteration. The event occurrence times hence reflect the occurrence times found by the solution with the least tardiness obtained in the previous iteration.

14. W-SUC-D-NPVCF (Weighted Number of Immediate Successors-Dynamic Net Present Value of Cash Flows):

$$priority_{ij} = (1\text{-}w) * (suc_{ij}) + (w) * \lceil (\sum_{m \, \varepsilon \, \{S_{ij}\}} e^{-\alpha Tm} * F_m) \div c \rceil^{-} \qquad (3.15)$$

This rule is a weighted combination of WRUP and Dynamic NPVCF rules. Activities are scheduled in descending order of priority.

We also compare the results of dispatching rules with a procedure which has been tested previously for the makespan criterion and found to be significantly superior to all makespan oriented rules under investigation, including MINSLACK and LFT [21,36]. This procedure is called Local constraint Based Analysis (LCBA) and treats the problems in a different perspective from that of the dispatching rules in the sense that it evaluates activities according to relevant essential conditions which represent temporal and resource constraints. However, when the essential conditions are not sufficient to identify the complete sequence of schedulable activities, a priority rule is called to resolve the resource conflicts. In this context, we define the following two priority rules to be used by LCBA:

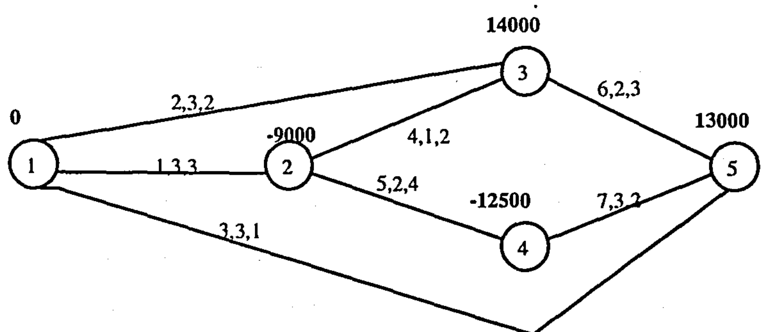
15. W-D-SLACK-D-NPVCF (As described above)

16. WRUP. (As described above)

Neither the above rules [# 11,12,13,14] nor LCBA [# 15,16] use Russell's (1970) optimization procedure which, when repeated at every activity completion time, leads to an explosion in solution time.

4. AN EXAMPLE

To illustrate the mechanics of the hybrid rules we solve a small example using W-D-SLACK-D-NPVCF rule in the iterative algorithm. The example network is found in Figure 4.1.



Legend:
On activities: (-,-):(activity index,duration,resource requirement)
On events: (-):(Cash flow at event occurrence time)

Figure 4.1. The Example

The interest rate is set at one per cent and the penalty per tardy period is $130. The resource limit is five units per period. The optimal resource constrained makespan is known to be 11 periods and the due date is set at this tight level. The solution procedure starts with the forward scheduling pass. This rule requires event occurrence times at each scheduling pass. In the first iteration, these are specified as late or early start times according to the sign of their cash flows. Event occurrence times are found in Table 4.1. To calculate the slack times, the rule requires activity lft times which are also displayed in Table 4.1. In Table 4.2. we observe the NPV of succeeding cash flows of all activities in the first forward iteration.

EVENT OCCURRENCE TIMES

EVENT	1	2	3	4	5
Forward # 1	0	6	4	8	8
Backward # 1	13	10	2	5	0
Forward # 2	0	3	8	5	11

ACTIVITY LFT

ACTIVITY	1	2	3	4	5	6	7
Forward # 1	6	9	11	9	8	11	11
Backward # 1	13	10	13	10	7	2	5
Forward # 2	3	8	11	6	5	11	9

TABLE 4.1. Event Occurrence Times and Activity LFT at Each Iteration.

ACTIVITY	1	2	3	4	5	6	7
NPV of Succeeding Events	5436	25451	12000	25451	461	12000	12000

TABLE 4.2. NPV of Logically Succeeding Events of Each Activity In the First Forward Iteration.

Table 4.3. displays the schedules generated in the first forward iteration by two distinctive weights, 0.0 and 0.2 along with activity slacks and priorities. When the weight given to the succeeding cash flows is zero, the priorities are determined by activity slack times only. This schedule results in an NPV of $3455 and a tardiness of two periods. However, when the weight is slightly increased to 0.2 we obtain a different schedule with an NPV of $4342 and a tardiness of three periods. Actually, the rule is implemented for 11 times in the first forward iteration starting from an NPV weight of zero and ending at a weight of one. Then the solution which provides the least tardiness is taken to calculate the next set of activity lfts and event occurrence times. Assuming that the solution with the least tardiness among 11 runs is obtained with an NPV weight of zero. Before we start with the backward iteration, we calculate event occurrence times, Ti, as the difference between the project duration of the selected solution and the earliest scheduled starting time of the activities starting from the event. The activity lft are also determined using their scheduled starting times as indicated in the previous section. These values are found in Table 4.1.

NPV WEIGHT=0.0

Schedule Time	Schedulable Activities #	Total Slack	Priority	Action
0	[1,2,3]			
	1	3	3	scheduled
	2	6	6	-
	3	8	8	scheduled
3	[2,4,5]			
	2	3	3	scheduled
	4	5	5	scheduled
	5	3	3	-
4	[ ]			
6	[5,6]			
	5	0	0	scheduled
	6	3	3	-
8	[6,7]			
	6	1	1	-
	7	0	0	scheduled
11	[6]	-2	-2	scheduled
13	complete			

NPV WEIGHT=0.2

Schedule Time	Schedulable Activities #	Total Slack	Priority	Action
0	[1,2,3]			
	1	3	1.4	-
	2	6	-0.2	scheduled
	3	8	4	scheduled
3	[1]			
	1	0	-1	scheduled
6	[4,5]			
	4	2	-3.4	scheduled
	5	0	0	
7	[5,6]			
	5	-1	-0.8	-
	6	2	-0.8	scheduled (random)
9	[5]			
	5	-3	-2.4	scheduled
11	[7]	-3	-4.8	scheduled
14	complete			

TABLE 4.3. Solution to The Example in the First Forward Iteration Using Weights 0.0 and 0.2.

Next, we start with the backward iteration with a weight of zero. The resulting schedule is observed in Table 4.4. This schedule results in zero tardiness and an NPV of $3934. Actually this is the best schedule obtained in the 11 runs with respect to NPV and tardiness criteria. Hence, both activity lft and event occurring times for the next forward iteration are calculated from this schedule and displayed in Table 4.1. Notice that even in this small example we obtain two schedules which are not dominated by the third with respect to both criteria. Hence, the flexibility of the weighted hybrid rules and the advantages of the iterative algorithm with its dynamic time windows are illustrated.

NPV WEIGHT=0.0

Schedule Time	Schedulable Activities#	Total Slack	Priority	Action
0	[3,6,7]			
	3	10	10	scheduled
	6	0	0	scheduled
	7	2	2	-
2	[2,4,7]			
	2	5	5	-
	4	7	7	-
	7	0	0	scheduled
3	[2,4]			
	2	4	4	scheduled
	4	6	6	-
5	[4]	4	4	scheduled
6	[5]	-1	-1	scheduled
8	[1]	2	2	scheduled
11	complete			

TABLE 4.4. The Backward Schedule Resulting from Weight Zero.

5. TESTING ENVIRONMENT

We use Patterson's [25] 85 problems for testing our heuristics. The size of the problems range from 10 activities and one resource type to 45 activities and three resource types. The complexity ratio, which is the ratio of activities to events, is between [1.2-1.8]. These problems have a resource Average Utilization Factor (AUF), which is defined by Kurtulus and Davis [17], in the interval [0.7-1.2]. However, this AUF level indicates that the resource constraints are loose (Padman and Smith-Daniels [24]). Since, resource tightness is reported to be an important factor which changes heuristic performance ranking (Padman and Smith-Daniels [24]), we create a second replicate of Patterson's 85 problems by tightening the resource constraints to result in an AUF between [1.8-2.5] and repeat our experiments on this set of problems. However, since the optimal resource constrained makespan values for these problems are not available, we replace the optimal value by the best makespan found by different dispatching rules and a truncated branch and bound algorithm (Özdamar and Ulusoy [22]).

Padman and Smith-Daniels [24] report in their extensive experimentation that the relative performance rank among different heuristic rules is preserved irrespective of factors such as problem structure, progress payment frequency, interest rate, project size and profit margin. The only important factor seems to be the tightness of the resource constraints. At the low level of resource tightness the rules ITS and TS/LAN are reported to be the best performing, whereas at the medium and high levels, the rules CFW-OCC and IOCS are the best respectively. Furthermore, IOCS and CFW-OCC perform best under all the levels of the remaining factors.

Cash outflows and inflows occurring at events are distributed uniformly in the interval [1000-20000]. We define and measure a characteristic that indicates the distribution of cash flows on the different phases of the project network. This characteristic, which we call cash skewedness, is important because it demonstrates whether the cash flows are biased towards the starting or ending phases of the project. In our testing environment, the skewedness for positive cash flows, described as the ratio of the sum of payments over the second half of the events to that of the first half, is in the range [1.2-1.4]. The skewedness for negative payments is between [0.8-1.0]. We remark that Patterson's problems have been converted to activity-on-arc networks with ordered event numbers and hence, half of the events almost represents mid-project phase. The profit margin in our experiments, which is measured as the ratio of cash inflows to cash outflows, is in the interval [%25-%35]. Furthermore, the ratio of the final payment to all cash outflows is also kept at a level lower than that of Padman and Smith-Daniels [24], i.e., at a level between [%10-%15], since a high payment at the end of the project leads to the trivial situation where a lower makespan implies a higher NPV. In such a case, the trade-off between the two criteria, if it exists, is lost in the experimentation. Under these conditions, the due date penalty formally becomes the link between the two criteria. In order to have an unbiased experimental design, the due date penalty is related to the final payment. It is taken as one per cent of the final payment and accrued each period that the project is tardy. Consequently, per period tardiness penalty is in the interval between [%0.20-%0.30] of all positive payments and it does not eliminate the trade-off between the two criteria.

The experiment is conducted at two levels of project due date. The first level represents the situation where the due date is set at the tightest possible value, that is, at the optimal resource feasible makespan value. At this level NPV is expected to be at its lowest level if the two criteria are strongly conflicting. The second level is where the due date is set at far into the future. At this level, the due date penalty has no effect on NPV and NPV is expected to assume its highest value. Our aim here is to specify the trade-off between the two criteria and identify the difference in the performance of NPV- and makespan-related rules and our hybrid heuristic rules under all conditions.

Consequently, our experiments include four testing environments: The first experiment involves tight due dates and loose resource constraints; the second experiment imposes loose due dates and loose resource constraints; the third experiment involves tight due dates and tight resource constraints while the fourth one imposes loose due dates and tight resource constraints.

6. RESULTS

We experiment with the dispatching rules and LCBA using the iterative scheduling algorithm and show the improvement in performance with respect to both tardiness and NPV criteria as a result of multiple scheduling passes. Furthermore, we compare makespan-related, NPV-related and the hybrid rules proposed here according to their performance in both criteria. We also partially repeat the experiment for certain rules to see the effects of using Russell's [28] unconstrained optimization procedure only once in the beginning of the scheduling pass.

In Tables A.1. and A.2., the results of the first and second experiments under loose resource constraints are given whereas in Tables A.3. and A.4., those under tight resource constraints are demonstrated. Tables A.1. and A.3., represent the case where the due dates are set equal to the optimal resource constrained makespan. Tables A.2. and A.4., on the other hand, demonstrate the results where the project due dates are set far in the future. In all tables, rules are identified by their indexes assigned to them earlier in the paper. In Tables A.1. and A.3., the first row conveys the average tardiness (with its standard deviation) found by the first forward iteration. Next, the average of the least tardiness realized at the end of the iterative scheduling algorithm is displayed for all the rules. These two measures are always zero in the second and fourth experiments since the due date is set to a value where it becomes ineffective. The third row demonstrates the average percentage deviation of the NPV from the unconstrained NPV given by Russell's [28] procedure. However, this result belongs to the solution providing the final tardiness value. The fourth row represents the average of the percentage deviation of the initial forward solution's NPV. The fifth row demonstrates the percentage deviation of the highest NPV found at the end of the iterations. The next two rows display the average tardiness belonging to the solutions represented in the fourth and fifth rows, i.e., they are the corresponding tardiness values of the solutions pertaining to the initial forward NPV and final NPV. In rules where more than one solution are obtained in the initial forward pass, (WRUP, W-D-SLACK-NPVCF, W-D-SLACK-D-NPVCF, W-SUC-D-NPVCF, LCBA's both options) the solution which has the best tardiness may not be the same with the one having the best NPV. Hence, it is also necessary to measure the best initial tardiness. In the final two rows we indicate the number of problems improved in tardiness and in NPV.

In Tables A.1. and A.3., the trade-off between the two criteria is observed by analyzing the average best tardiness (second row) and the average tardiness corresponding to the best NPV solution (seventh row). We observe that this trade-off is more significant under tight resource constraints than that of loose resource constraints. Furthermore, when we compare the average NPV percentage deviation belonging to the best tardiness solution (third row) with the average percentage deviation of the best NPV (fifth row) in Tables A.1. and A.3., we observe that the values in the third row fall between the initial forward and final average NPV percentage deviations. Actually, the best tardiness values are always better than the ones corresponding to the best NPV. In fact, when NPV is plotted against tardiness, for each rule, there are three coordinates on the graph provided by the three average value pairs: point a(second row, third row), point c(fourth row, sixth row), point b(fifth row, seventh row). Although it is not exactly correct to make such a generalization on empirically obtained average values, we observe that the function defined by NPV versus tardiness seems to be a unimodular one. When the individual problems rather than average values are viewed from the same perspective, we reach the same conclusion. The relationship between these pairs is illustrated on the graph below.

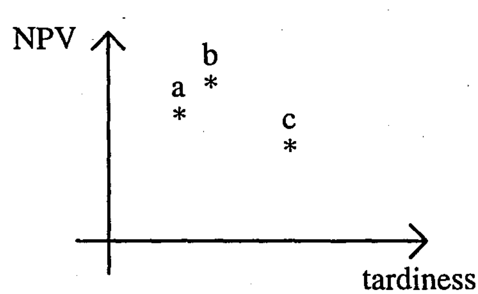


Figure 6.1. The Graph of Average NPV vs. Average Tardiness

As expected, when the due dates are not tight, NPV 's are relatively higher than the ones under tight due dates irrespective of resource tightness. Yet, the difference in average NPV's is not statistically significant at one per cent level of significance. This result is due to the low level of the tardiness penalty. To show the effects of a higher tardiness penalty, set at 10% of the final payment accrued per period tardy, we repeat the first experiment on four dispatching rules, W-D-SLACK-NPVCF, W-SUC-D-NPVCF, IOCS, TS/LAN, under tight due dates and loose resource constraints and display the results in Table A.5. Under a loose due date the effect of increasing the tardiness penalty is not effective since no schedule can possibly be tardy. Hence, the results in Table A.5. can be compared with the ones in Table A.2. Comparing the tight and loose due date results, we observe that there exists a statistically significant difference between the average percentage deviations of NPV, demonstrating the effect of a higher tardiness penalty. Furthermore, the average tardiness values corresponding to the initial and final NPV solutions in Table A.5. are remarkably close to those observed as the average initial and final best tardiness values (the first two rows) in Table A.1. under all four rules, indicating the complete elimination of the trade-off between the two criteria.

In analyzing Tables A.1. to A.4., we can classify the rules into three groups in terms of performance. The first and best performing group with respect to both criteria includes the hybrid rules (# 12,13,14) and LCBA (# 15/16). The second best performing group includes the optimization-guided rule TS/LAN (# 7) and the makespan-oriented rule WRUP (# 10). The final group includes the remaining optimization-guided rules and makespan-oriented rules (# 1, 2 , 3, 4, 5, 6, 8, 9, 11). Under tight resource constraints, the rule CFW/LAN (# 6) is upgraded to the second best performing group when NPV criterion is considered.

Let us first analyze performance with respect to the criterion of NPV. Under both loose and tight due date and resource constraints, i.e., observing Tables A.1. to A.4., the best four rules with respect to NPV criterion are rules # 15, 12, 13, 14, namely, LCBA (option 1), W-D-SLACK-NPVCF, W-D-SLACK-D-NPVCF and W-SUC-D-NPVCF in any order with the exception that LCBA's second option replaces W-SUC-D-NPVCF in the first experiment. The performance ranking with respect to NPV criterion is conveyed for each experiment in Table A.6. It is observed in Tables A.1. to A.4. that the superior performance of the hybrid rules and LCBA options becomes comparatively more impressive under tight resource constraints. For example, we cannot discover a statistical difference between the average percentage deviation of final NPV given by LCBA's first option (first best performing group element) and TS/LAN (second best performing group element) under loose resource constraints, though such a difference exists between the two rules at both five per cent and one per cent levels of significance under tight resource constraints. In fact, TS/LAN which is used in the iterative algorithm where the unconstrained problem is reoptimized at each scheduling time, seems to be the best among NPV-related rules, under both loose and tight resource constraints. This result is contradictory to that of Padman and Smith-Daniels [24] who claim that TS/LAN performs rather poorly under tight resource constraints. Surprisingly, the performances of WRUP and TS/LAN are quite close to each other, except that WRUP's tardiness performance is considerably superior to that of TS/LAN. Hence, TS/LAN is the closest to hybrid rules and LCBA among optimization-guided rules in its performance with respect to both criteria. The remaining optimization-guided rules are removed away from the best performing hybrid rules and LCBA and shown to be consistently poorer in performance. Furthermore, the makespan-oriented rules MINSLACK and LFT are not worse than the optimization-guided rules considering the criterion of NPV. In the first experiment, there is no statistically significant difference between MINSLACK and IOCS and CFW/LAN, which are the best optimization-guided rules in the third group. As we noted previously, although in the first experiment there seems to be no statistically significant difference among LCBA and TS/LAN, there certainly exists a difference between the best performing rules # 15, 12 ,13, and the group of optimization-guided rules, represented by IOCS, CFW/LAN and ILTP/LEC, (# 2, 6, 3) which are the best in their own group. Under tight resource constraints, the performance of CFW/LAN improves considerably and this rule becomes as good as TS/LAN. Hence, we can conclude that there is a slight change in the ranking of optimization-guided rules when the resource constraints switch from loose to tight. The first three ranks under loose constraints are held by TS/LAN, CFW/LAN and IOCS (# 7, 6, 2) whereas under tight resource constraints CFW/LAN, TS/LAN and LTP/LAN become (# 6, 7, 4) superior. However, it is important to note that the performance of LCBA's second option, and the hybrid rules are not affected by the experimental conditions and they are quite reliable.

When we analyze the results with respect to the criterion of tardiness, the best performing rules are the rules # 16, 15, 13, 12, namely, LCBA's second and first option, W-D-SLACK-D-NPVCF and W-D-SLACK-NPVCF. In Table A.7. the ranking of heuristic performance with respect to the criterion of tardiness is displayed. The performance of the first group including LCBA (# 15/16) and the hybrid rules (# 12, 13) are significantly superior as compared to that of the third group including the optimization-guided rules except for TS/LAN. The second best performing group now includes MINSLACK, LFT , TS/LAN (# 8, 9, 7) and the hybrid rules, # 11, 14. We note that WRUP (# 10) belongs to the second group under loose resource constraints whereas it becomes a member of the first group under tight resource constraints. There exist statistically significant differences between the second and third group elements and among the first and third groups. We observe that the elements of the groups formed for NPV and tardiness criteria are nearly the same with a few exceptions.

The first four positions in the ranking displayed in both Tables A.6. and A.7., are held by hybrid rules (# 12, 13, 14 in Table A.6. only), LCBA's first option or second option (# 15/16) and in Table A.7., the makespan-related rule, WRUP (# 10 in Table A.7. only). To summarize, we can conclude that the best rules with respect to both criteria are the hybrid ones which account for the Net Present Value of cash flows of logically succeeding events as well as activity total slack time. Furthermore, LCBA, as a robust scheduling procedure, has a superior performance with respect to both criteria if its first option is utilized. We remark that the earliness and tardiness costs defined by Padman and Smith-Daniels [24] implicitly rely on slack times. In our experimentation we observe that using slack time explicitly results in better performance. Another point is that the cash flow distribution and tardiness penalty used in the tests result in an unbiased environment where both criteria are observed independently. In spite of this fact, the tardiness and NPV ranking displayed in Tables A.6. and A.7. follow each other closely leading to the nontrivial result that a rule which is good for one criterion is also good for the other. This information is valuable for practitioners who are anxious to keep their promised project delivery date while making a profit on the project.

In all testing environments the iterative algorithm based on dynamic time windows and event occurrence times has an ameliorating effect on both tardiness and NPV criteria. Apart from the improvements of about 25 per cent in the average NPV percentages and upto 50 per cent in the average tardiness, we observe that the number of problems improved in NPV as well as in tardiness due to the iterative algorithm is quite large (upto 68 problems out of 85 problems in some rules with respect to NPV and upto 51 problems with respect to tardiness). We remark that the difference between the initial and best NPV is statistically significant for all rules at the five per cent level of significance. This result is also true for the average tardiness values.

In our experimentation we observe that the optimization-guided heuristics take a considerable amount of CPU time. Table A.8. demonstrates the average CPUsec. taken by each rule to solve a single problem in the fourth experiment. The experimentation is conducted on an 486 IBM compatible PC and the programming language is Turbo Pascal. Notice that the CPU times taken by the makespan-related rules, MINSLACK, WRUP and LFT and by LCBA's both options are less than one CPUsec. whereas the hybrid rules take about 30-40 CPUsec. However, the optimization guided rules proposed by Padman and Smith-Daniels [24] take about 130-140 CPUsec. per problem. Hence, in order to investigate the effect of using Russell's optimization algorithm only once on both performance and CPU time, we test four optimization-guided rules, IOCS, ILTP/LEC, CFW/LAN, TS/LAN (# 2, 3, 5, 7) and apply the iterative scheduling algorithm without updating the information from the optimal unconstrained solution every time an activity is completed. The results of the experiments carried out under the first and fourth testing environments are conveyed in Table A.9. It is observed and statistically proven that neither the NPV nor the tardiness results are significantly different from the corresponding results found in Tables A.1. and A.4. respectively. In other words, it does not matter whether or not the unconstrained problem is reoptimized. Naturally, the CPU times taken by these rules drop down to a level of 4-7 seconds.

The results of some statistical tests are consistent with the above explanations and displayed in Appendix B in Tables B.1., B.2., B.3., B.4., B.5., B.6., B.7., B.8., B.9. In Table B.1. (experiment # 1), the null hypothesis is that there is not a significant difference between the performance of the rules # 2, 7, 12, 15, namely IOCS, TS/LAN, W-D-SLACK-NPVCF, LCBA's first option respectively according to the final percentage of NPV deviation under the loose resource constraints. The observed F value is 5.425 while the tabulated F value with 3 numerator degrees od freedom and ∞ denominator degrees of freedom (dof) at the level of significance, α=0.01 is 3.78. Hence, we reject the null hypothesis. Applying the Duncan's Multiple Range Test at α =0.01 to come up with the different rules according to the NPV criterion, we find that there are significant differences between the rule pairs (# 2, # 15) and (# 2, # 12).

In Table B.2., the purpose is the same, but the rules used are the ones # 3, 5, 7, 8, namely ILTP/LEC, CFW/OCC, TS/LAN and MINSLACK respectively. We note that they are the rules found in the literature. In this case, the null hypothesis is not rejected, meaning that there is not a significant difference between the performance of the rules # 3, 5, 7, 8. However, Duncan's Multiple Range Test shows that there are significant differences between the rule pairs (# 3, # 5) and (# 5, # 8) at α=0.01. Table B.3. also shows that there is a significant difference according to the same performance criterion in rules # 7, 14, 15, namely TS/LAN, W-SUC-D-NPVCF and LCBA's first option under tight resource constraints at the same level of significance and we find that there are significant differences between the rule pairs (# 7, # 14) and (# 7, # 15).

In Tables B.4. and B.5., the null hypothesis is that there is not a significant difference among the 16 heuristic rules according to the average per cent of NPV deviation of 85 problems at two levels of project due date, respectively. In both cases, the null hypothesis is not rejected with these observed F values at α=0.01 and α=0.025.

In Tables B.6., B.7., B.8., B.9., the null hypothesis is that there is not a significant difference in performance of 85 problems according to the NPV criterion between the optimization-guided case and the case where the optimization procedure is used only once in rules # 7, 2, 3, 6 respectively. At many levels of significance, namely α= 0.1, 0.05, 0.025, 0.01, 0.005, the observed F values are less than the tabulated F values with one numerator degrees of freedom and 168 denominator degrees of freedom. Therefore, there is not a sufficient evidence to indicate a statistically significant difference.

7. CONCLUSION

This paper is focused on developing robust heuristic rules which aim at achieving good quality solutions with respect to the criteria of maximizing the project NPV and minimizing tardiness in the presence of a project due date. The proposed heuristic rules merge two classical concepts: activity slack time and the net present value of the cumulative cash flows on succeeding events. The rules are basically the weighted average of these two magnitudes. The iterative algorithm in which these rules are imbedded supports the rules by providing a link between consecutive forward/backward scheduling passes. This link involves an update on activity time windows and event occurrence times based on the scheduled starting times of activities obtained in the previous schedule. These updated values are used in giving priorities in the next scheduling iteration. All tested rules benefit considerably from the iterative scheduling algorithm.

The proposed heuristic rules are compared with previously published NPV- and tardiness-oriented rules. They are found to be superior in quality with respect to both criteria. Furthermore, they are robust in the sense that they are not affected by the changing testing environment. In the experimentation, the trade-off between the two criteria is made visible and for all rules three coordinate pairs are obtained in terms of NPV and tardiness average values. We remark that the functional relationship with respect to these average values is unimodular. It is also observed that the rules which achieve the minimum tardiness are the ones which achieve the maximum project NPV. For example, the optimization-guided rules which have the worst tardiness performance also have the worst NPV performance. This observation is emphasized, because the testing environment, which is prepared in an unbiased manner, enables an almost independent analysis of each criterion.

Due to the combinatorial nature of the resource constrained project scheduling problem, it is not possible to generate all efficient solutions with respect to both criteria for large size problems. Further research in this area might be aimed at investigating the functional relationship between the two criteria by generating the efficient frontier at least for small problems.

APPENDIX A

TABLES OF RESULTS OF EXPERIMENTS

RULE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Initial tardiness	4.82/ 3.25	3.18/ 2.36	3.79/ 3.00	3.80/ 2.96	3.86/ 2.75	3.61/ 2.83	2.49/ 2.23	1.84/ 1.91	2.05/ 1.75	1.85/ 1.60	2.05/ 1.87	1.36/ 1.38	1.39/ 1.49	2.36/ 1.86	1.80/ 2.18	1.11/ 1.25
Final tardiness	3.34/ 2.28	2.12/ 1.81	2.65/ 2.05	2.69/ 2.18	2.50/ 2.02	2.53/ 1.89	1.65/ 1.49	1.05/ 1.12	1.29/ 1.16	1.19/ 1.26	1.48/ 1.42	0.84/ 0.91	0.85/ 0.94	1.71/ 1.49	0.84/ 1.42	0.64/ 0.97
Corresponding NPV of final tardiness	20.22/ 11.44	16.91/ 8.88	18.07/ 9.83	18.18/ 10.06	18.89/ 11.24	18.44/ 10.50	13.73/ 7.60	17.11/ 13.10	16.27/ 8.78	16.34/ 12.10	15.13/ 8.47	13.85/ 7.85	13.73/ 7.85	16.03/ 9.89	12.94/ 7.85	13.02/ 8.08
Inilial % of NPV deviation	26.83/ 15.00	20.37/ 10.62	21.52/ 12.70	21.64/ 12.80	22.98/ 13.14	17.24/ 8.72	17.53/ 9.25	18.64/ 10.06	19.76/ 10.10	17.41/ 9.44	16.56/ 8.8	14.18/ 7.08	14.61/ 7.97	16.32/ 8.17	16.45/ 11.35	14.04/ 8.63
Final % of NPV deviation	18.25/ 10.25	15.64/ 8.43	16.27/ 8.65	16.41/ 9.13	17.89/ 10.76	15.9/ 8.79	13.07/ 7.30	14.88/ 8.85	15.25/ 8.38	13.58/ 7.54	14.75/ 8.50	12.09/ 7.16	12.18/ 7.16	13.58/ 7.37	11.27/ 7.03	12.41/ 7.96
Corresponding tardiness of initial NPV	4.82/ 3.25	3.18/ 2.36	3.79/ 3.00	3.80/ 2.96	3.86/ 2.75	3.61/ 2.83	2.49/ 2.23	1.84/ 1.91	2.05/ 1.75	2.24/ 1.91	2.05/ 1.87	1.76/ 1.61	1.81/ 1.89	2.59/ 2.09	1.96/ 2.19	1.36/ 1.42
Corresponding tardiness of final NPV	3.85/ 2.66	2.66/ 1.94	3.02/ 2.19	3.15/ 2.49	3.23/ 2.59	3.29/ 2.58	1.94/ 1.64	1.28/ 1.27	1.61/ 1.34	2.12/ 2.10	1.68/ 1.48	1.66/ 1.50	1.54/ 1.49	2.46/ 1.99	1.49/ 1.73	1.11/ 1.39
No of problems improved in tardiness	39	42	31	30	39	32	31	28	36	32	25	26	26	29	39	24
No of problems improved in NPV	58	53	46	48	48	22	53	61	64	52	37	44	45	39	58	38

TABLE A.1. The Results of Experiment 1. (Legend for Tables A.1. - A.4., Table A.9.: average/standard deviation)

RULE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Inilial % of NPV deviatio	23.73/ 13.84	18.32/ 10.04	19.07/ 11.82	19.25/ 11.79	20.46/ 12.18	14.84/ 8.06	15.86/ 8.44	18.26/ 9.61	18.68/ 10.11	17.09/ 9.12	21.20/ 14.71	12.60/ 6.92	12.83/ 7.00	13.71/ 7.23	16.63/ 11.32	13.38/ 7.79
Final % of NPV deviation	15.69/ 9.44	13.89/ 7.98	14.41/ 8.37	14.33/ 8.43	15.86/ 10.42	13.71/ 8.21	11.77/ 6.75	14.05/ 8.04	14.08/ 8.03	11.95/ 7.49	15.29/ 10.79	10.82/ 6.85	11.02/ 6.89	10.14/ 9.56	10.38/ 6.26	11.99/ 7.38
No of problems improved in NPV	60	53	48	48	48	23	56	66	65	62	49	39	40	48	58	32

TABLE A.2. The Results of Experiment 2.

RULE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Initial tardiness	5.23/ 3.85	3.63/ 4.09	4.67/ 4.84	4.62/ 3.87	4.32/ 3.85	4.13/ 3.58	2.51/ 2.75	2.47/ 2.57	1.76/ 2.22	1.17/ 1.54	2.67/ 2.89	1.64/ 2.13	1.63/ 2.06	2.06/ 2.19	2.73/ 2.41	1.10/ 1.70
Final tardiness	3.32/ 2.65	2.12/ 2.47	2.58/ 2.71	2.31/ 2.76	2.87/ 2.89	2.73/ 2.34	1.44/ 1.79	1.27/ 1.84	1.01/ 1.75	0.74/ 1.28	1.14/ 1.82	0.77/ 1.51	0.99/ 1.62	1.40/ 1.74	1.05/ 1.71	0.37/ 1.18
Corresponding NPV of final tardiness	43.46/ 19.64	42.38/ 20.53	43.34/ 20.81	43.19/ 20.87	45.68/ 24.13	42.17/ 24.77	35.57/ 17.51	39.19/ 19.15	41.36/ 18.96	38.71/ 17.90	42.35/ 22.83	35.02/ 19.19	30.82/ 14.28	29.53/ 13.79	34.26/ 18.71	37.30/ 17.49
Inittial % of NPV deviation	53.76/ 25.30	46.04/ 20.91	47.93/ 23.14	47.47/ 21.41	46.52/ 24.19	32.84/ 15.27	38.46/ 16.73	45.12/ 20.67	47.10/ 21.59	39.30/ 18.18	45.02/ 25.42	30.50/ 14.92	31.35/ 14.39	33.01/ 16.21	43.92/ 19.31	36.98/ 18.41
Final % of NPV deviation	39.76/ 17.95	39.89/ 17.94	39.63/ 19.69	38.00/ 17.68	41.39/ 20.36	32.54/ 15.15	32.72/ 14.59	39.23/ 18.69	40.55/ 18.64	32.59/ 14.83	40.01/ 21.62	29.09/ 14.68	29.62/ 14.11	27.21/ 12.39	27.57/ 13.17	33.06/ 13.90
Corresponding tardiness of initial NPV	5.23/ 3.85	3.63/ 4.09	4.67/ 4.84	4.62/ 3.87	4.32/ 3.85	4.13/ 3.58	2.51/ 2.75	2.47/ 2.57	1.76/ 2.22	2.85/ 3.62	2.67/ 2.89	3.11/ 3.10	2.58/ 2.65	3.20/ 3.30	3.82/ 2.81	2.52/ 3.04
Corresponding tardiness of final NPV	4.26/ 3.45	2.73/ 2.81	3.73/ 3.52	3.73/ 3.70	3.85/ 3.49	4.05/ 3.64	1.93/ 2.21	1.35/ 1.97	1.35/ 1.97	2.37/ 2.81	1.49/ 2.06	2.46/ 2.77	2.07/ 2.05	3.12/ 2.88	2.45/ 2.43	1.73/ 2.50
No of problems improved in tardiness	40	34	34	39	31	28	31	40	35	23	39	27	28	25	51	32
No of problems improved in NPV	60	45	45	50	33	5	51	62	64	48	33	28	28	47	68	28

TABLE A.3. The Results of Experiment 3.

RULE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Inittial % of NPV deviation	50.71/ 23.76	44.84/ 21.98	45.63/ 22.41	45.15/ 21.04	44.34/ 23.63	30.70/ 15.20	37.12/ 16.63	43.86/ 20.66	45.96/ 21.66	37.55/ 17.96	53.92/ 25.91	29.01/ 14.35	30.99/ 14.34	31.60/ 15.51	43.76/ 20.40	35.82/ 18.58
Final % of NPV deviation	37.35/ 17.69	38.88/ 18.86	37.69/ 19.19	35.91/ 17.16	39.34/ 19.78	30.46/ 15.03	31.68/ 14.44	38.46/ 18.81	39.76/ 18.66	31.12/ 14.82	36.72/ 18.52	27.78/ 3.16	28.73/ 13.58	25.30/ 11.90	27.57/ 13.59	31.93/ 14.80
No of problems improved in NPV	62	47	45	50	33	5	52	64	64	48	62	29	36	55	68	28

TABLE A.4. The Results of Experiment 4.

RULE	2	7	12	14
Inlitial % of NPV deviation	38.42/ 19.86	32.13/ 19.98	23.77/ 14.39	30.77/ 17.08
Final % of NPV deviation	28.84/ 15.77	24.20/ 14.78	18.64/ 10.66	25.73/ 15.43
Corresponding tardiness of initial NPV	3.18/ 2.36	2.49/ 2.23	1.37/ 1.39	2.39/ 1.99
Corresponding tardiness of final NPV	2.21/ 1.83	1.68/ 1.51	0.92/ 1.03	1.89/ 1.78
No of problems improved in NPV	48	44	43	35

TABLE A.5. The Results Obtained in Experiment 1 Using a Higher Tardiness Penalty.

RANK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
EXPERIMENT #1 (TABLE 1)																
RULE #	15	12	13	16	7	14/10	10/14	11	8	9	2	6	3	4	5	1
EXPERIMENT #2 (TABLE 2)																
RULE #	14	15	12	13	7	10	16	6	2	8	9	4	3	11	1	5
EXPERIMENT #3 (TABLE 3)																
RULE #	14	15	12	13	6	10	7	16	4	8	3	1	2	11	9	5
EXPERIMENT #4 (TABLE 4)																
RULE #	14	15	12	13	6	10	7	16	4	11	1	3	8	2	5	9

TABLE A.6. Heuristic Performance Ranking with respect to NPV Criterion. (Legend for Tables A.6. and A.7.: rule1/rule2 implies the rules share the position)

RANK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
EXPERIMENT #1 (TABLE 1)																
RULE #	16	15/12	12/15	13	8	10	9	11	7	14	2	5	3	6	4	1
EXPERIMENT #3 (TABLE 3)																
RULE #	16	10	12	13	9	15	11	8	14	7	2	4	3	6	5	1

TABLE A.7. Heuristic Performance Ranking with respect to Tardiness Criterion.

RULE	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
EXPERIMENT # 4																
Average CPU sec.	134.56	139.48	140.28	138.8	141.47	2.83	138.78	0.12	0.12	0.81	4.02	33.96	37.75	38.45	0.98	0.21

TABLE A.8. Average CPU Times in Experiment 4.

	EXPERIMENT # 1				***EXPERIMENT # 4***			
RULE	**2**	**3**	**5**	**7**	**2**	**3**	**5**	**7**
Initial tardiness	3.29/ 2.83	3.62/ 2.60	3.89/ 2.78	2.54/ 2.40	-	-	-	-
Best tardiness	2.05/ 1.69	2.49/ 2.11	2.55/ 2.03	1.98/ 1.60	-	-	-	-
Iniltial % of NPV deviation	20.33/ 11.59	22.32/ 12.05	22.97/ 13.65	17.15/ 9.22	47.87/ 25.80	45.48/ 24.03	46.33/ 24.82	36.90/ 15.97
Final % of NPV deviation	15.60/ 9.06	16.40/ 9.27	17.51/ 10.30	12.89/ 7.13	40.07/ 20.61	38.30/ 19.56	37.90/ 18.05	32.11/ 14.18
Corresponding tardiness of initial NPV	3.29/ 2.83	3.62/ 2.60	3.89/ 2.78	2.54/ 2.40	-	-	-	-
Corresponding tardiness of final NPV	2.56/ 1.86	2.95/ 2.25	3.33/ 2.72	2.06/ 1.77	-	-	-	-
No of problems improved in tardiness	37	35	32	35	-	-	-	-
No of problems improved in NPV	51	49	50	47	43	46	40	47
Average CPU sec.	-	-	-	-	4.36	7.27	6.59	4.28

TABLE A.9. Results of Optimization-Guided Rules where the Optimization Procedure is Applied Only Once.

APPENDIX B
STATISTICAL ANALYSIS OF EXPERIMENTS

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	916.021	3	305.340	5.425
ERROR	18912.906	336	56.288	

TABLE B.1. The ANOVA Table for Comparing the Final Percentage of NPV Deviation of Rules # 2, 7, 12, 15 in Experiment 1.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	701.009	3	233.670	3.195
ERROR	24570.735	336	73.127	

TABLE B.2. The ANOVA Table for Comparing the Final Percentage of NPV Deviation of Rules # 3, 5, 7, 8 in Experiment 1.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	1598.010	2	799.005	4.495
ERROR	44795.885	252	177.761	

TABLE B.3. The ANOVA Table for Comparing the Final Percentage of NPV Deviation of Rules # 7, 14, 15 in Experiment 3.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
TIGHTNESS	19.290	1	19.290	4.003
ERROR	144.540	30	4.818	

TABLE B.4 The ANOVA Table to Test whether There Exists a Difference Between Experiments 1 and 2 in the Performance of the 16 Rules According to the Final Percentage of NPV Deviation.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
TIGHTNESS	19.245	1	19.245	0.558
ERROR	1034.853	30	34.495	

TABLE B.5. The ANOVA Table to Test whether There Exists a Difference Between Experiments 3 and 4 in the Performance of the 16 Rules According to the Final Percentage of NPV Deviation.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	1.157	1	1.157	0.022
ERROR	8699.773	168	51.784	

TABLE B.6. The ANOVA Table to Test whether There Exists a Difference Between the Optimization-Guided Case and the Case where the Optimization is Applied Only Once in the Performance of 85 Problems of Rule # 7 According to NPV Criterion.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	1.743	1	1.743	0.023
ERROR	12778.856	168	76.064	

TABLE B.7. The ANOVA Table to Test whether There Exists a Difference Between the Optimization-Guided Case and the Case where the Optimization is Applied Only Once in the Performance of 85 Problems of Rule # 2 According to NPV Criterion.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	2.960	1	2.960	0.037
ERROR	13287.143	168	79.090	

TABLE B.8. The ANOVA Table to Test whether There Exists a Difference Between the Optimization-Guided Case and the Case where the Optimization is Applied Only Once in the Performance of 85 Problems of Rule # 3 According to NPV Criterion.

SOURCE	SUM OF SQUARES	DEGREES OF FREEDOM	MEAN SQUARE	F RATIO
RULE	0.448	1	0.448	0.006
ERROR	12998.652	168	77.373	

TABLE B.9. The ANOVA Table to Test whether There Exists a Difference Between the Optimization-Guided Case and the Case where the Optimization is Applied Only Once in the Performance of 85 Problems of Rule # 6 According to NPV Criterion.

REFERENCES CITED

[1] Alvarez-Valdes, A. and J.M. Tamarit, "Heuristic Algorithms for Resource Constrained Project Scheduling: A Review and an Empirical Analysis," in *Advances in Project Scheduling,* (R. Slowinski and J. Weglarz, Ed.), Elsevier, the Netherlands, pp. 113-134, 1989.

[2] Baroum, S. and J.H. Patterson, "A Heuristic Algorithm for Maximizing the Net Present Value of Cash Flows in Resource Constrained Project Schedule," Working Paper, Indiana University, 1989.

[3] Bell, C.E. and K. Park, "Solving Resource Constrained Project Scheduling Problems by A* Search," *Naval Research Logistics*, Vol. 37, pp. 61-84, 1990.

[4] Blazewicz, J., J.K. Lenstra and Rinnooy Kan, A.H.G., "Scheduling Subject to Resource Constraints: Classification and Complexity," *Discrete Applied Mathematics*, Vol. 5, pp. 11-24, 1983.

[5] Bock, D.B. and J.H. Patterson, "A Comparison of Due Date Setting, Resource Assignment and Job Preemption Heuristics for the Multiproject Scheduling Problem," *Decision Sciences*, Vol. 21, pp. 387-402, 1990.

[6] Boctor, F.F., "Some Efficient Multi-Heuristic Procedures for Resource Constrained Project Scheduling," *European Journal of Operational Research*, Vol. 49, pp. 3-13, 1990.

[7] Boctor, F.F., "Heuristics for Scheduling Projects with Resource Restrictions and Several Resource-Duration Modes," *International Journal of Production Research*, Vol. 31, pp. 2547-2558, 1993.

[8] Christofides, N., R. Alvarez-Valdes, and J.M. Tamarit, "Project Scheduling with Resource Constraints: A Branch and Bound Approach," *European Journal of Operational Research*, Vol. 29, pp. 262-273, 1987.

[9] Daniels, R.L., "Resource Allocation and Multi Project Scheduling," in: R. Slowinski and J. Weglarz (ed.), *Advances in Project Scheduling,* Elsevier, pp. 87-113, 1989.

[10] Daniels, R.L. and J.B. Mazzola, "Flow Shop Scheduling with Resource Flexibility," to appear in *Operations Research.*

[11] Davis, E.W. and J.H. Patterson, " A Comparison of Heuristics and Optimum Solutions in Resource Constrained Project Scheduling," *Management Science*, Vol. 21, pp. 944-955, 1975.

[12] Demeulemeester, E. and W. Herroelen, "A Branch and Bound Procedure for the Multiple Resource Constrained Project Scheduling Problem," *Management Science*, Vol. 38, pp. 1803-1818, 1992.

[13] Doersch, R.H. and J.H. Patterson, "Scheduling a Project to Maximize its Present Value: A Zero-One Programming Approach," *Management Science*, Vol. 23, pp. 882-889, 1977.

[14] Dumond, J.," In a Multi-Resource Environment How Much Is Enough?," *International Journal of Production Research*, Vol. 30, pp. 411-431, 1992.

[15] Dumond, J. and V.A. Mabert, "Evaluating Project Scheduling and Due Date Assignment Procedures: An Experimental Analysis," *Management Science*, Vol. 34, pp. 101-118, 1988.

[16] Elmaghraby, S.E. and W.S. Herroelen, "The Scheduling of Activities to Maximize the Net Present Value of Projects," *European Journal of Operational Research*, Vol. 49, pp. 35-40, 1990.

[17] Kurtuluş, I.S. and E.W. Davis, "Multi Project Scheduling: Categorization of Heuristic Rule Performance," *Management Science*, Vol. 28 , pp. 161-172, 1982.

[18] Kurtuluş, I.S. and S.C. Narula, "Multi Project Scheduling: Analysis of Project Performance," *IIE Transactions*, Vol. 17, pp. 58-66, 1985.

[19] Lawrence, S.R. and T.E. Morton, "Resource-Constrained Multi-Project Scheduling with Tardy Costs: Comparing Myopic, Bottleneck, and Resource Pricing Heuristics," *European Journal of Operational Research*, Vol. 64, pp. 168-187, 1993.

[20] Li , K.Y. and R.J. Willis, "An Iterative Scheduling Technique for Resource-Constrained Project Scheduling," *European Journal of Operational Research*, Vol. 56, pp. 370-379, 1992.

[21] Özdamar, L. and G. Ulusoy, " A Local Constraint Based Analysis Approach to Project Scheduling under General Resource Constraints," to appear in *European Journal of Operational Research* .

[22] Özdamar, L. and G. Ulusoy, "A Heuristic with a Branch and Bound Structure for the Resource Constrained Project Scheduling Problem," Working Paper, Marmara University, Department of Industrial Engineering, Istanbul, Turkey, 1993.

[23] Padman, R., D.E. Smith-Daniels and V.L. Smith-Daniels, "Heuristic Scheduling of Resource Constrained Projects with Cash Flows: An Optimization Based Approach," Working Paper 90-6, Carnegie Mellon University, 1990. .

[24] Padman, R. and D.E. Smith-Daniels, " Early-Tardy Cost Trade-Offs in Resource Constrained Projects with Cash Flows: An Optimization-Guided Heuristic Approach," *European Journal of Operational Research*, Vol. 64, pp. 295-311, 1993.

[25] Patterson, J.H. , "A Comparison of Exact Approaches for Solving the Multiple Constrained Resource, Project Scheduling Problem," *Management Science*, Vol. 30, pp. 854-867, 1984.

[26] Patterson, J. H., R. Slowinski, F.B. Talbot and J. Weglarz, "Computational Experience with a Backtracking Algorithm for Solving a General Class of Precedence and Resource Constrained Scheduling Problems," *European Journal of Operational Research*, Vol. 49 , pp. 68-79, 1990.

[27] Pritsker, A.A.B., L.J. Watters and P.M. Wolfe, "Multi-Project Scheduling with Limited Resources: A Zero-One Programming Approach," *Management Science*, Vol. 16, pp. 93-108, 1969.

[28] Russell, A. H., "Cash Flows in Networks," *Management Science*, Vol. 16, pp. 357-373, 1970.

[29] Russell, R. A., "A Comparison of Heuristics for Scheduling Projects with Cash Flows and Resource Restrictions," *Management Science*, Vol. 32 , pp.1291-1300, 1986.

[30] Slowinski, R., "Two Approaches to Problems of Resource Allocation among Project Activities - A Comparative Study," *Journal of the Operational Research Society*, Vol. 31, pp. 711-723, 1980.

[31] Slowinski, R., "Multiobjective Project Scheduling under Multiple Category Resource Constraints," in: R. Slowinski and J. Weglarz (ed.), *Advances in Project Scheduling,* Elsevier, pp. 151-167, 1989.

[32] Smith-Daniels, D.E., and N.J. Aquilano, "Using a Late Start Resource Constrained Project Schedule to Improve Project Net Present Value," *Decision Sciences*, Vol. 18, pp. 617-630, 1987.

[33] Speranza, M.G. and C. Vercellis, "Hierarchical Models for Multi-Project Planning and Scheduling," *European Journal of Operational Research*, Vol. 64, pp. 312-325, 1993.

[34] Stinson, J.P., E.W. Davis, and B.W. Khumawala, " Multiple-Resource Constrained Scheduling Using Branch and Bound," *AIIE Transactions*, Vol. 10, pp. 252-259, 1978.

[35] Talbot, F.B., "Resource Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case," *Management Science*, Vol. 28 , pp. 1197-1210, 1982.

[36] Ulusoy, G. and L. Özdamar, "Heuristic Performance and Network/Resource Characteristics in Resource-Constrained Project Scheduling," *Journal of the Operational Research Society*, Vol. 40, pp. 1145-1152, 1989.

[37] Ulusoy, G. and L. Özdamar, "A Constraint Based Perspective in Resource Constrained Project Scheduling," to appear in *International Journal of Production Research.*

[38] Weglarz, J., J. Blazewicz, J. Cellary and R. Slowinski, "An Automatic Revised Simplex Method for Constrained Resource Network Scheduling," *ACM Transactions on Mathematical Software*, Vol. 3, pp. 295-300, 1977.

[39] Yang, K.K., F.B. Talbot and J.H. Patterson, "Scheduling a Project to Maximize its Net Present Value: An Integer Programming Approach," *European Journal of Operational Research*, Vol. 64, pp. 188-198, 1993.

REFERENCES NOT CITED

Bell, C.E. and J. Han, "A New Heuristic Solution Method in Resource Constrained Project Scheduling Problems," *Naval Research Logistics*, Vol. 38, pp. 315-331, 1991.

Cooper, D.F., "Heuristics for Scheduling Resource Constrained Scheduling Projects: An Experimental Investigation," *Management Science*, Vol. 22, pp. 1186-1194, 1976.

Davies, E.M., "An Experimental Investigation of Resource Allocation in Multiactivity Projects," *Journal of the Operational Research Society*, Vol. 24, pp. 587-591, 1973.

Davis, E.W., "Project Network Summary Measures in Constrained-Resource Scheduling," *AIIE Transactions*, Vol. 7, pp. 132-142, 1975.

Davis, E.W. and G.E. Heidorn, "An Algorithm for Optimal Project Scheduling under Multiple Resource Constraints," *Management Science*, Vol. 17, pp. 803-816, 1971.

Demeulemeester, E., W.P. Simpson, S. Baroum, J.H. Patterson and K. Yang, "Comments on a Paper by Christofides et al. for Solving the Multiple-Resource Constrained Single Project Scheduling Problem," Working Paper, 1991.

Erschler, J. and F. Roubellat, "An Approach for Real-Time Scheduling of Activities with Time and Resource Constraints," In *Advances in Project Scheduling,* (R. Slowinski and J. Weglarz, Ed.), Elsevier , the Netherlands, pp. 237-277, 1989.

Hastings, N.A.J., " On Resource Allocation in Project Networks," *Journal of the Operational Research Society*, Vol. 23, pp. 217-221, 1972.

Khattab M. and F. Choobineh,"A New Heuristic for Project Scheduling with a Single Resource Constraint," *Comp. Ind. Eng.*, Vol. 19, pp. 514-518, 1990.

Khattab M. and F. Choobineh, "A New approach for Project Scheduling with a Limited Resource," *Int. J. Prod. Res.*, Vol. 29, pp. 185-198, 1991.

Patterson, J.H. ,"Project Scheduling: The Effects of Problem Structure on Heuristic Performance," *Naval Research Logistics Quarterly*, Vol. 23, pp. 95-123, 1976.

Patterson, J. H. and W.D. Huber, "A Horizon-Varying, Zero-One Approach to Project Scheduling," *Management Science*, Vol. 20, pp. 990-998, 1974.

Patterson, J. H. and G.W. Roth, "Scheduling a Project under Multiple Resource Constraints: A 0-1 Programming Approach," *AIEE Transactions*, Vol. 8, pp. 449-455, 1976.

Patterson, J. H., R. Slowinski, F.B. Talbot and J. Weglarz, "An Algorithm for a General Class of Precedence and Resource Constrained Scheduling Problems," in: R. Slowinski and J. Weglarz (ed.), *Advances in Project Scheduling*, Elsevier, pp. 3-28, 1989.

Talbot, F.B., "Resource Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case," *Management Science*, Vol. 28 , pp. 1197-1210, 1982.

Tavares, L.V., "A Multi-Stage Nondeterministic Model for Project Scheduling under Resource Constraints," *European Journal of Operational Research* Vol. 49, pp. 92-101, 1990.

Thesen, A., "Heuristic Scheduling of Activities under Resource and Precedence Restrictions," *Management Science*, Vol. 23, pp. 412-422, 1976.

Thesen, A., "Measures of Restrictiveness of Project Networks," *Networks*, Vol. 7, pp. 193-208, 1977.

Willis, R.J. and N.A.J. Hastings, "Project Scheduling with Resource Constraints Using Branch and Bound Methods," *Journal of the Operational Research Society*, Vol. 27, pp. 341-349, 1976.

Willis, R.J., "Critical Path Analysis and Resource Constrained Project Scheduling - Theory and Practice," *European Journal of Operational Research*, Vol. 21, pp. 149-155, 1985.