RESOURCE PORTFOLIO PROBLEM UNDER RESOURCE DEDICATION POLICIES IN MULTI-PROJECT ENVIRONMENTS

by

Umut Beşikci

B.S., Industrial Engineering, İstanbul Teknik University, 2004

M.S., Industrial Engineering, Boğaziçi University, 2006

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

Graduate Program in
Boğaziçi University
2012

ACKNOWLEDGEMENTS

I feel myself lucky for having an opportunity to work with Prof. Dr. Ümit Bilge. Her way of handling obstacles that we have faced throughout the dissertation was invaluable. Besides offering an unbelievable amount of her time for supervision of this dissertation, she has always been encouraging during my entire graduate study and helped me to build an academic vision.

I am grateful to my dissertation co-supervisor, Prof. Gündüz Ulusoy for giving me a chance to work with him and for his guidance and effort in this dissertation. Apart from his broad academic knowledge, his mentoring for my academic career was invaluable.

I would like to thank Prof. Necati Aras and Assoc. Prof. Ali Tamer Ünal for their interest and participation in this dissertation, they have offered me unbelievable amount of time during the development of this dissertation. I am also grateful to Prof. Bülent Durmuşoğlu not only for his participation but also for his helpful and encouraging commends.

I would also thank Erinç and other members of BUFAIM team for their support and help during my dissertation.

I do not think that I can express how grateful to my parents I am. I am sure that things would be much harder if I did not have their support and guidance. I also would like to thank Melike, for not only being a caring sister, but also for her friendship.

This dissertation is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) through Project Number MAG 109M571 and Boğaziçi University Scientific Research Projects (BAP) through Project Number O9HA302D.

ABSTRACT

RESOURCE PORTFOLIO PROBLEM UNDER RESOURCE DEDICATION POLICIES IN MULTI-PROJECT ENVIRONMENTS

Resource related decisions are one of the important aspects of multi-project environments, since the resource based considerations define the environment as a multi-project problem by coupling projects with corresponding conceptual and physical constraints. The characterization of the way resources are used by the individual projects in a multi-project environment is named as resource management policy in this dissertation. Resource management policies can differ with respect to the environment characteristics (e.g., geographical distribution of projects, specific resource characteristics, etc.). Thus, to identify and characterize different properties and aspects of the multi-project environment, different resource management policies need to be defined. Two different resource management policies are proposed in this dissertation. The first one is the Resource Dedication (RD) policy where resources cannot be shared among projects because of the characteristics of the multi-project environment. The second policy is an extension of RD, such that renewable resource transfers among projects are allowed when one of the projects finishes before the start of the another one. This resource management policy is called the Relaxed Resource Dedication (RRD) policy. These different resource management policies are investigated in a problem environment such that general resource capacities are included into the problem as another decision level. This problem is called Resource Portfolio Problem. The main contributions of this dissertation are the definitions for RPP under different resource management policies, corresponding mathematical models and the proposed solution approaches for multi-project scheduling problems.

ÖZET

ÇOKLU PROJE ORTAMLARINDA KAYNAK ADANMASI POLİTİKALARI ALTINDA KAYNAK PORTFÖYÜ PROBLEMİ

Çoklu proje çizelgeleme ve yönetim problemi proje çizelgeleme yazınında önemli bir yer tutmaktadir. Kaynaklarla ilgili kararlar problemdeki farklı projeleri kısıtlar üzerinden birbirine bağladığından, çoklu proje çizelgeleme problemleri için önemli bir olgudur. Bu doktora tezinde kaynakların projeler tarafından kullanılma biçimini karakterize eden kavrama kaynak yönetim politikası adı verilmiştir. Kaynak yönetim politikaları, çoklu proje çizelgeleme probleminin özellikerine göre değişiklikler gösterebilmektedir (projelerin geniş bir coğrafyaya dağılmış olması, kaynaklara ve projelere özel karateristikler vb.). Bu sebepten, değişik çoklu proje problem çerçevelerini tanımlayabilmek için, değişik kaynak yönetim politikaları sunulmalıdır. İki ayrı kaynak yönetim politikası bu doktora tezi çerçevesinde sunulmuştur. İlk politika Kaynak Adanması (KA) politikasıdır. KA politikasında yenilenebilir kaynaklar projelere paylaştırılmak yerine adanmaktadır öyleki adanma işleminden sonra kaynaklar projeler tarafından paylaşıla- mamaktadır. İkinci politika ise, KA politikasının bir uzanımıdır. Gevşetilmiş Kaynak Adanması (GKA) olarak adlandırılan bu politikada yenilenebilir kaynaklar projeler arasında taşınabilmektedir. Bu kaynak taşımasının gerçekleştirilmesi için bir projenin bitiş zamanının öteki projenin başlangıç zamanından önce olması gerekmektedir. Bu kaynak yönetim politikaları genel kaynak kapasitesi kararlarını dahil edildiği bir problem çerçevesinde incelenmiştir. Bu problem Kaynak Portföyü Problemi (KPP) olarak adlandırılmaktadır. Bu doktora tezinin genel katkıları değişik kaynak yönetim politikaları altındaki KPP tanımları ve bunlar için sunulan çözüm yöntemleridir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

λ_{kt}	Lagrangian coefficient for renewable resource constraint for resource k at time period t
μ_i	Lagrangian coefficient for nonrenewable resource constraint for resource i
π_{ij}	Imputed resource cost of activity j of project i
a_{iv}	Closeness of allowable upper bound to the dedicated nonrenewable resource i for project v
a_{kv}	Closeness of allowable upper bound to the dedicated renewable resource k for project v
a_{vj}	Arrival of job j, $j \in N_v$, of project i, $t \in G_v$ (arrivals occur at the beginning of periods)
AUB_{iv}	Allowable upper bound of nonrenewable resource constraint of resource i for project v
AUB_{ktv}	Allowable upper bound of renewable resource constraint of resource k at time period t for project v
b_i	Amount of nonrenewable resource i available for dedication to projects calculated from slack resource values
b_k	Amount of renewable resource k available for dedication to projects calculated from slack resource values
BR_{vk}	Total amount of dedicated renewable resource k to project v
BW_{vi}	Total amount of dedicated nonrenewable resource i to project v
c_v	Relative weight of project v
CP_i	Unconstrained completion time of project i
cr_k	Unit cost of renewable resource k
CT_i	Resource-constrained completion time of project i
cw_i	Unit cost of nonrenewable resource i
d_{jm}	Duration of activity j, $j \in N$, operating in mode m, $m \in M_j$
d_{vj}	Number of periods required to perform job j, $j \in N_v$, of project v, $i \in V$

d_{vjm}	Duration of activity j, operating on mode m
dd_v	Assigned due date for project v
$DRkv$	The assigned dedicated renewable resource k to project v
$DWiv$	The assigned dedicated nonrenewable resource i to project v
E_j	Early finish of activity j, $j \in N$
e_{vj}	The earliest possible period in which job j, $j \in N_i$, of project v, $v \in V$, can be completed
E_{vj}	Earliest finish time of activity j of project v
g_v	Desired due date for project i (project i is not late if it finishes before or on g_v), $v = 1 \ldots V$
G_v	Absolute due date defined for project i, $t \in G_v$
gr_{kt}	Amount of type k, $k \in \lvert K \rvert$, resource available in period t, $t \in G_v$
l_{ij}	Late start time of activity j of project i
L_j	Late finish of activity j, $j \in N$
l_{vj}	The latest possible period in which job j, $j \in N_i$, of project v, $v \in V$, can be completed
L_{vj}	Latest finish time of activity j of project v
LB	Lower bound
M_j	Set of modes for activity j, $m \in M_j$
M_{vj}	Set of modes for activity j of project v, $m \in M_{vj}$
N	Set of activities, $j \in N$
N_v	Set of jobs, $j \in N_v$ for project v
K	Set of renewable resources, $k \in K$
I	set of nonrenewable resources, $i \in I$
J_v	Set of activities of project v, $j \in J_v$
$\bar{p}$	Average processing duration of all activities
P	Set of all precedence relationships
P_{ij}	Priority of activity j of project i
p_{ij}	Duration of activity j of project i
p_{iv}	Preference of project v to nonrenewable resource i
p_{kv}	Preference of project v to renewable resource k

P_v	Set of all precedence relationships of project v
p_{vt}	Total tardiness cost incurred when project v, $i \in I$, is completed in period t, $t \in G_v$
r_{jkm}	Usage of renewable resource k, $k \in K$, by activity j, $j \in N$ with mode m, $m \in M_j$
r_{vjk}	Amount of type k resource required by job j, $j \in N_v$, of project v, $i \in V$, can be completed
r_{vjkm}	Renewable resource k usage of activity j of project v, operating on mode m
r_k	Total amount of available renewable resource k
r_{kt}	Available renewable resource k, $k \in K$, in period t, $t \in T$
$\bar{r_k}$	Average renewable resource allocation of activities over their corresponding time span
T	Set of time periods, $t \in T$
T_v	Set of time periods for project v, $t \in T_v$
tb	Total resource budget
U_i	Set of all unfinished activities of project i
ur_{vk}	No-delay resource requirement of project v for renewable resource k
uw_{vi}	No-delay resource requirement of project v for nonrenewable resource i
u_v	Earliest possible period in which project i could be completed, $v \in V$
V	Set of projects, $v \in V$
y_{iv}	Continuous variable for spare nonrenewable resource i dedicated to project v
y_{kv}	Continuous variable for spare renewable resource k dedicated to project v
Y_{vt}	Binary variable equals to 1 if all jobs of project v, $V \in V$, are completed before period t, $t \in G_v$
$Y_{vv'}$	Binary decision variable equals to 1 if project v finishes before the start of project v'
w_i	Total available nonrenewable resource i, $i \in I$, for the project

w_{jim}	Consumption of nonrenewable resource i, $i \in I$, by activity j, $j \in N$, in mode m, $m \in M_j$
w_{vjim}	Nonrenewable resource i usage of activity j of project v, operating on mode m
X_{jmt}	Binary variable equals to 1 if activity j, $j \in N$ operating in mode m, $m \in M$, is finished in period t, $t \in T$
X_{vjmt}	Binary variable equals to 1 if activity j of project v, operating in mode m, is finished in period t
X_{vjt}	Binary variable equals to 1 if job j, $j \in N_i$, in project i, $i \in I$, is completed in period t, $t \in G_i$

LIST OF ACRONYMS/ABBREVIATIONS

ART	Average Run Time
AWT	Average Weighted Tardiness
B&B	Branch and Bound
CPL	Critical Path Length
CPM	Critical Path Methodology
GA	Genetic Algorithm
GA-LA	Genetic Algorithm Employing Lagrangian Relaxation Based Preference Calculation
GA-LinR	Genetic Algorithm Employing Linear Relaxation Based Preference Calculation
LA	Lagrangian Relaxation
LA-RDP	Lagrangian Relaxation of Resource Dedication Problem
LA-SP	Lagrangian Relaxation of Single Project Scheduling Problem
LR	Linear Relaxation
MRCPSP	Multi-mode Resource Constrained Project Scheduling Problem
NA	Not Available
NS	No Solution
OS	Optimal Solution
RACP	Resource Availability Cost Problem
RCCP	Rough-Cut Capacity Planning
RCPSP	Resource Constrained Project Scheduling Problem
RCS	Resource Capacities Space
RD	Resource Dedication
RDP	Resource Dedication Problem
RDS	Resource Dedication Space
SO	Subgradient Optimization
R&D	Research and Development
RHS	Right Hand Side

RM	Resource Management
RS	Resource Sharing
R&M	Rachamadugu and Morton rule
RPP	Resource Portfolio Problem
RPP-RD	Resource Portfolio Problem under Resource Dedication Policy
RPP-RRD	Resource Portfolio Problem under Relaxed Resource Dedication Policy
RPP-RS	Resource Portfolio Problem under Relaxed Resource Sharing Policy
RRD	Relaxed Resource Dedication
SP-LR	Linear Relaxation of the Single Project Scheduling Problem
TS	Tabu Search

1. INTRODUCTION

Project scheduling problem can be defined as the sequencing of the activities subject to the precedence relations and resource constraints to optimize a predetermined objective or a set of objectives. The multi-project problem environments consist of a number of simultaneously executed projects which are, in general, competing for a set of common resources. In other words, different projects in the problem are coupled with resource related constraints. The way of this coupling is defined by the resource management (RM) policy for the multi-project environment. Apart from the resource management policy, the multi-project environment has different aspects related with the structure of the individual projects and activities which can affect the nature of the problem environment drastically.

First of all, the structure of activities is an important part of the problem. The activities can be executed with only one given predetermined recipe of resource usages (mode) or there can be a set of recipes each having different resource usages and corresponding durations. The first case characterizes the single mode project scheduling problem and the second case is known as the multi-mode project scheduling problem. In this dissertation, the projects in the multi-project environment will be of multi-mode case.

Another important aspect of the problem environment is the structure of the resources. There can be different types of resources in the project environment, namely, renewable, nonrenewable or doubly constrained resources. The renewable resources are available in limited quantities in each time period. On the other hand, if the total consumption of a resource is constrained over the time periods then the resource is called nonrenewable. Finally, resources are defined as doubly constrained, if both their per-period and total consumption are limited (Talbot, 1982). Detailed information about different resource types can be found in Demeulemeester and Herroelen (2002). In this dissertation, only renewable and nonrenewable resources will be considered.

Finally, the problem environment we investigate is a static environment without any stochastic component such that all the parameters in the problem environment are deterministic (e.g., activity mode durations, resource consumptions, general resource budget etc.). In static project environments, the project portfolio does not change for the planning horizon.

Resource management policy is an important part of the multi-project scheduling problems, since it characterizes the problem environment by defining the coupling among different projects. The common resource policy applied in the multi-project scheduling literature is Resource Sharing (RS) policy where resources can be shared among project without any restrictions. Even though RS policy is employed in the multi-project scheduling literature frequently, it does not cover all possible cases. We have proposed Resource Dedication (RD) policy for the cases where resources cannot be shared among projects because of various reasons. The most common ones can be listed as follows:

- The characteristics of projects may not allow RS. This can be seen in various Research and Development (R&D) projects where the development process is highly technology intensive.
- The resource characteristics may force RD. For example, in software development projects, it is not desired to allocate developers to different projects because of the learning curve concept. Another example can be given as the heavy machinery equipment for certain cases where it becomes too costly to share them among projects because of installation.
- Another reason can be geographical limitations, where projects are distributed across the world such that it becomes unpractical to share the resources.

Resource dedication term is used in a couple of studies in project scheduling literature. Bianco *et al.* (1998) use term dedicated resources as a renewable resource with only unit capacity, such that when an activity utilizes the resource it becomes unavailable for other activities. This approach is different from our definition since the dedication term refers to the one unit capacity availability which forces the resource to be dedicated to the activity.

RD policy is extended further to Relaxed Resource Dedication (RRD) policy. In RRD policy, a resource can be transferred among projects, when the project using that resource is completed and another project is to start which requires that same resource. This concept is realized in the problem environment by allowing sequential relations between projects.

The general resource capacity will have different meanings according to the valid resource management policy in the problem environment. In the RS policy, the general resource capacity corresponds to the size of the shared pool for each renewable resource. Under the RD policy, on the other hand, it corresponds to the total amount of each renewable resource to be dedicated among the projects. In other words, resource management policy defines the way of coupling projects via general resource capacities. Naturally, general resource capacity values becomes an important part of the problem because of these considerations and with the Resource Portfolio Problem (RPP), these important considerations are included as another decision dimension of the problem, which we believe completes multi-project problem environments conceptually.

We have investigated RPP under these two different resource management policies which allowed us to incorporate general resource capacity decisions with the characteristics of the employed resource management policies. The multi-project environment for RPP under RD policy inherits different conceptual problems. First of all, the general resource capacities must be determined from the general resource budget. We call this conceptual problem as resource portfolio. The next step is determination of resource dedication values from the general resource capacity values. Finally, when resource dedication values are known the remaining problem is scheduling of individual projects with given resource dedication values which is in fact reduces solving the Multi-mode Resource Constrained Project Scheduling Problem (MRCPSP) for each project. The general problem environment and inherited problems are depicted in Figure 1.1.

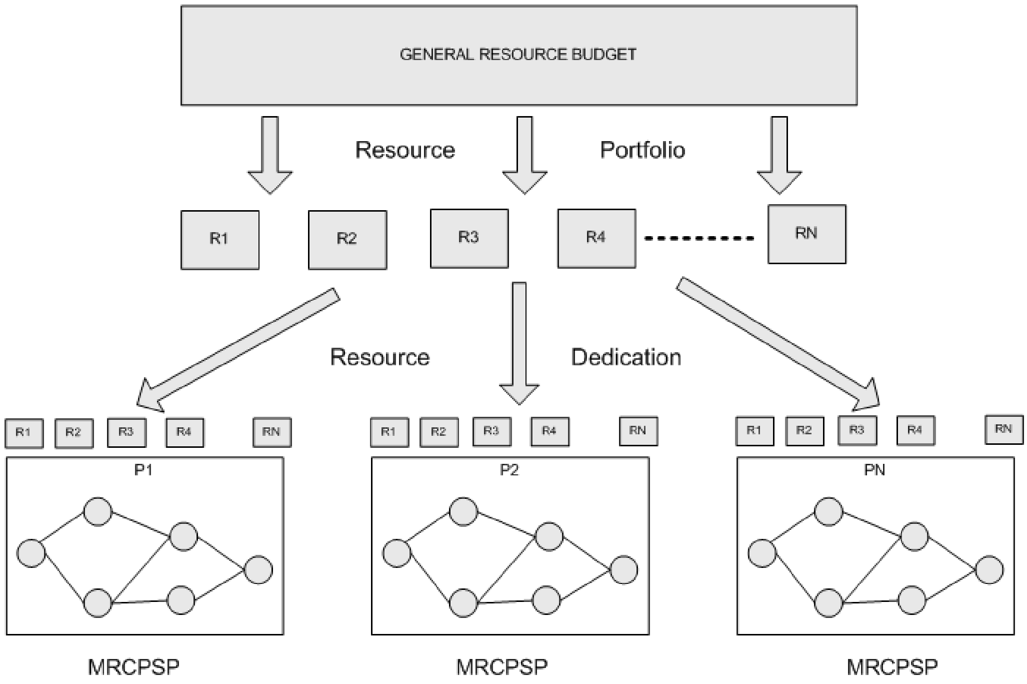

Figure 1.1. General problem environment.

This problem environment brings new issues in multi-project scheduling problem and introduces further complexities to the multi-project problem. To investigate the RPP under the RD policy, we will first look into a sub-problem, which does not include the allocation decision of the resource budget to resources. Hence, the subproblem is defined as a multi-project scheduling problem with given general resource capacities under the RD policy. This sub-problem is called the Resource Dedication Problem (RDP) in this dissertation. We use the insights gained from our research related with RDP and employ them for the cases encapsulating further complexities emerging from resource budget decisions and/or new resource management policies.

The dissertation is organized as follows: we will give an extensive literature review for the single project and multi-project scheduling in Chapter 2. In Chapter 3, RDP is introduced and proposed solution approaches for the problem are explained in detail with experimental results. Chapter 4 and Chapter 5 present RPP under the RD and RRD policies, respectively, with proposed solution approaches and an extensive experimental study. We conclude and point out some possible future research topics in Chapter 6.

2. LITERATURE REVIEW

MRCPSP characterizes the individual projects in this dissertation. Therefore, first a brief survey on MRCPSP will be given in Section 2.1. Then an extensive review for the multi-project scheduling will be presented in Section 2.2.

2.1. Multi-mode Resource-Constrained Project Scheduling Problem

In MRCPSP, the problem environment consists of activities, resources and different execution modes for each activity. The resources in the environment can be renewable, nonrenewable or doubly constrained. There can be different objectives used in project scheduling problems but generally the most common ones are; completion time of the project, project tardiness and net present value.

The most frequently employed mathematical formulation for MRCPSP is proposed by Talbot (1982) which is an integer program. The network has an activity on node representation with a start node and a termination node which can be dummy. Formulation requires the calculation of early finish times and late start times for the activities to decrease the number of decision variables for each activity. For this calculation, an upper bound for the project makespan T is either suggested or calculated with a heuristic approach. The early start and late finish values for each activity are calculated over the time horizon T without consideration of resource constraints. The early finish times are calculated employing the critical path method (CPM) and using the mode with the shortest duration for activities. The late finish times are calculated by backtracking the project network with last project having a late finish time of T and using the mode with the shortest duration. This guarantees that the optimal scheduling time of an activity can be represented within a relatively smaller time window. The mathematical programming model proposed by Talbot (1982) and the corresponding sets (Table 2.1), parameters (Table 2.2) and decision variables (Table 2.3) are given below.

Table 2.1. Sets for MRCPSP.

N	Set of activities, $j \in N$
M_j	Set of modes for activity j, $m \in M_j$
K	Set of renewable resources, $k \in K$
I	Set of nonrenewable resources, $i \in I$
T	Set of time periods, $t \in T$
P	Set of all precedence relationships

Table 2.2. Parameters for MRCPSP.

E_j	Early finish of activity j, $j \in N$
L_j	Late finish of activity j, $j \in N$
r_{kt}	Available renewable resource k, $k \in K$, in period t, $t \in T$
r_{jkm}	Usage of renewable resource k, $k \in K$, by activity j, $j \in N$ with mode m, $m \in M_j$
w_i	Total available nonrenewable resource i, $i \in I$, for the project
w_{jim}	Consumption of nonrenewable resource i, $i \in I$, by activity j, $j \in N$, in mode m, $m \in M_j$
d_{jm}	Duration of activity j, $j \in N$, operating in mode m, $m \in M_j$

Table 2.3. Decision variables for MRCPSP.

X_{jmt}	$\begin{cases} 1 & \text{if activity } j\text{, operating in mode } m \text{ is finished at period } t \\ 0 & \text{otherwise} \end{cases}$

Mathematical Model MRCPSP

$$min.\ z = \sum_{m \in M_N} \sum_{t=E_N}^{L_N} tX_{Nmt} \tag{2.1}$$

Subject to

$$\sum_{m \in M_j} \sum_{t=E_j}^{L_j} X_{jmt} = 1 \qquad j \in N \tag{2.2}$$

$$\sum_{m \in M_b} \sum_{t=E_b}^{L_b} (t - d_{bm}) X_{bmt} \geq \sum_{m \in M_a} \sum_{t=E_a}^{L_a} tX_{amt} \qquad \forall\ (a, b) \in P \tag{2.3}$$

$$\sum_{j \in N} \sum_{m \in M_j} \sum_{q=t}^{t+d_{jm}-1} r_{jkm} X_{jmq} \leq r_{kt} \qquad \forall\ k \in K\ \forall\ t \in T \tag{2.4}$$

$$\sum_{j \in N} \sum_{m \in M_j} \sum_{t=E_j}^{L_j} w_{jim} X_{jmt} \leq w_i \qquad \forall\ i \in I \tag{2.5}$$

$$X_{jmt} \in \{0, 1\} \qquad \forall\ j \in J, \forall\ m \in M_j \text{ and } \forall\ t \in T \tag{2.6}$$

The objective (Equation 2.1) is defined as minimizing the completion time of the project schedule. Note that the resulting schedule minimizes all regular objectives for the given scheduling problem. Constraint set (Equation 2.2) ensures that an activity is completed exactly once. Constraint set (Equation 2.3) ensures the precedence relations among activities. The precedence relations imposed here are of "finish to start with no time lag" type. Constraint sets (Equation 2.4) and (Equation 2.5) enforce renewable and nonrenewable resource capacities, respectively. Note that renewable resource capacities are time dependent in the above formulation.

Talbot (1982) suggests a two stage algorithm to solve MRCPSP. In stage one; the network is relabeled using one of the proposed heuristics to determine the order of the activities to consider for assignment. In addition to this, resources and modes are also sorted according to some criterion to determine the general requirement for resources. In particular, for each resource, mode, and activity the following index is calculated;

$$Index_{jkm} = d_{jm} \text{ for resource } k \text{ which maximizes } r_{jkm}/\bar{r_k}$$

where

$$\bar{r_k} = \sum_{t=E_j}^{L_j} r_{kt}/(L_j - E_j + 1) \tag{2.7}$$

The resources are sorted in decreasing order by summing these indices over all activities and modes. For sorting of modes, different rules are used according to the selected objective function. For example, if the objective is minimizing the makespan, then modes are sorted in increasing duration.

After these labeling of activities, resources and modes, one can proceed to stage two. In stage two, an implicit enumeration algorithm is used to develop a complete feasible schedule by using partial feasible schedules. First of all, activities are considered in increasing order of one of the given scheduling rules (e.g. minimum slack). The algorithm finds feasible solutions for the activities until an activity cannot be scheduled because of predecessor relations, resource constraints or its late finish time, L_j. At this point, a backtracking procedure is applied and the most recently selected activity is scheduled to a later time in order to free resources. This procedure is applied to all activities and modes till an improved solution is found or optimality is verified which is done in one of these two conditions are satisfied: either all possible job assignments have been explicitly or implicitly evaluated, or a solution is found which is equal to a proven theoretical bound such as the Critical Path Length (CPL).

Another solution approach for MRCPSP is proposed by Sprecher *et al.* (1997). The resources considered in the problem are renewable, nonrenewable and doubly constrained. The proposed methodology is a special branch and bound (B&B) procedure for the problem. Each node in the B&B tree is expressed by a time instant which can be the start time for any activity. In other words, a branch is created for the early finish time of the processing activities. On each node, a set of eligible activities is constructed. This set contains currently active activities and the activities that are delayed on previous branches. For such activities a mode is selected and if there is a resource conflict resulting from this mode selection, then a minimal delay activity is selected from the eligible set. A set is called a minimal delay set when another eligible activity cannot be added to the set by maintaining resource constraints. After that, a branch for the next level is created. If a complete schedule is created, then the next minimal delay alternative is considered. On the other hand, if the procedure returns to a level where all minimal delay alternatives are searched, then next mode alternative is selected.

There are different bounding rules proposed by Sprecher *et al.* (1997). Here, only the bounding rules which result with fathoming a branch will be explained. First one is associated with the time windows for activities. If an activity is assigned a starting time later than its late finish time, then the current partial schedule can not be completed with a total makespan less than the upper bound. Thus, this branch is fathomed. The second one is nonrenewable resource consumption of activities. If the total nonrenewable resource usage of the already assigned activities and minimum nonrenewable resource usage of unassigned activities exceeds any resource capacity, then that partial schedule is infeasible and the branch is fathomed. The study of Sprecher *et al.* (1997) is extended with new bounding rules by Sprecher and Drexl (1998).

Since an exact solution for MRCPSP is hard to obtain, there are various heuristic approaches to the problem in the literature. The first group of the heuristics is generally priority rule based heuristics where a set of rules are applied to the problem to efficiently obtain a feasible solution. One of the earliest rule based heuristic study is proposed by Boctor (1993). The general scheme of the proposed approach is quite simple where the objective is taken as the project makespan. If there are still non-scheduled activities, then the algorithm advances to the nearest point where there are some non-allocated resources and for that point the available activities are determined. From that activities list, the activity with the highest priority is selected and from the feasible modes of the corresponding activity, the mode with the highest priority is selected. The activity selection rules used in the study are minimum total slack, minimum late finish time, maximum number of immediate successors, maximum remaining work, maximum processing time, minimum processing time, maximum number of subsequent candidate. For mode selection these following rules are used: shortest feasible mode, least critical resource and least resource usage ratio. These rules are applied to different problems and results are evaluated with the deviation from the critical path solution, percentage increase over the CPL and the number of times the procedure has obtained an optimal solution. The results show that none of the rules is consistently successful over all experiments, but overall, min slack and shortest feasible mode combination were the most successful.

Another priority rule based heuristics approach for MRCPSP is proposed by Lova *et al.* (2006). In this study, authors only considered renewable resources and the objective is minimizing the project makespan. In addition to the priority rules for activity selection in Boctor (1993), there are seven more rules designed for the same decision. Since the computation time of a selection combination for a given problem is relatively small, the authors propose a multi pass scheme. For an s multi pass heuristic, s different rule combinations are considered and the rule combination giving the best results is selected.

The second group of heuristics is meta-heuristics, usually Genetic Algorithm (GA), Simulated Annealing (SA) or Tabu Search (TS) applications. Mori and Tseng (1997) propose a GA application for MRCPSP. The objective is to minimize the project duration and only renewable resources are considered in this study. Another important aspect of the proposed GA is the usage of the scheduling order intervals. For each activity, a scheduling interval is determined by forward and backward tracking. The start point of the interval is calculated with forward tracking and it is the order of the activity, if it does not have any predecessors and one plus maximum of end points of its predecessors otherwise. The end point of the interval is calculated with a backward tracking and it is the order of the activity, if it does not have any successors and one minus minimum of end point of its successors otherwise. The chromosome shows the selected mode for the activity (selected randomly), scheduling order interval and scheduling order (selected randomly between start and end points of scheduling order). Thus, it is not necessary to use a transformation procedure or scheduling builder. The fitness of a chromosome is measured by the project duration. Also note that only the resource and precedence feasible solutions are considered. The execution of GA is carried out applying the following steps. In initialization, a number of individuals are created by randomly assigning modes and scheduling orders. If a schedule is feasible, then it is included in the population. This procedure is executed until a predetermined number of feasible solutions are generated. The population is always sorted in ascending order of their project durations. For crossover, the best individual and a random one are selected. Then an activity is selected randomly. The activity string from that selected activity to the dummy start activity is taken from the individual which has the interval with the smaller start point. The missing activities are taken from the other individual. For mutation operation, an individual is selected randomly and random modes of a random number of activities are changed randomly. The experimental results show that GA gave better results than priority rule and stochastic based assignment procedures.

Hartmann (2001) proposes a GA for MRCPSP with renewable and nonrenewable resource types. The fitness calculation of the individuals allows the consideration of resource infeasible solutions in the algorithm. In case of violation of nonrenewable resource capacities, this total nonrenewable resource capacity violation is calculated and then the fitness value of the individual is set to maximum project duration limit plus the violation. Otherwise the fitness value is simply the project duration. Note that, a feasible individual always has a lower fitness value than an infeasible one with this fitness calculation approach. The chromosome representation of a solution is simply the order of scheduled activities and their selected modes in a two dimensional array. Schedule generation from this representation is fairly easy by assigning the activities with their selected modes as early as possible. Before executing the GA, a preprocessing procedure is applied on project data based on the bounding rules given in Sprecher *et al.* (1997). The initial population is generated by randomly selecting order of the activities and mode assignments such that they are always precedence feasible but they might be nonrenewable resource infeasible. For crossover operation, two individuals are selected randomly and then two children are generated by taking the partial scheduling order from one parent and using the order of the remaining activities from the other one, the mode assignments array is not changed in this procedure. Mutation is applied to newly generated individuals. First, the order of an activity is changed with one of its immediate neighbors keeping the mode assignment; and secondly, the mode assignment of an individual is changed with a given mutation probability. For selection of the next generation, the population is ranked according to fitness values and the fittest individuals are selected. On top of these operators, a local search procedure is also used. The local search technique is left shifting the activities by changing their mode assignments, if this move is feasible. This heuristic can be applied as a single pass or a multi-pass procedure. In the preliminary experiments it is seen that multi-pass heuristics do not improve the results of single pass heuristics. With an extensive experimental design, it is seen that the proposed GA outperforms the rule based heuristics and gives comparable results with other meta-heuristic approaches to MRCPSP.

Another meta-heuristic application for MRCPSP with renewable and nonrenewable resources is proposed by Jozefowska *et al.* (2001), which is a SA approach to the problem. There are a couple of important parametric considerations in the study. First of all, the initial temperature is calculated by using the number of cost decreasing and cost increasing transactions. Another one is the Markovian approach for the cooling scheme. The temperature is decreased with the standard deviation of the changes in the objective function value and a control parameter for the amount of decrement. The solutions are represented by a two dimensional list; one for the activities and one for their mode assignments. The resource infeasible solutions are included in the solution procedure with a similar approach as given by Hartmann (2001). Neighborhood generation is done in three different ways. First one is changing the scheduling order of an activity. To do this, an activity is selected and its nearest successor and predecessor are found. Then the activity is moved randomly between its nearest successor and predecessor and all activities are shifted left or right accordingly. The other method is randomly changing the mode assignment of an activity. The last one is the combination of these two methods. The numerical results show that the proposed approach does not give competitive results when it is compared to the results reported by Hartmann (2001) and Bouleimen and Lecocq (2003).

The SA approach of Bouleimen and Lecocq (2003) gives promising results for MRCPSP with renewable and nonrenewable resources. The solution representation is simply a two dimensional array; one for order of the activities and the other for mode selection of the activities. From this representation it is easy to generate a schedule by using a serial schedule generation scheme Kolish *et al.* (1995). The objective function is taken as the project duration. The proposed SA algorithm is based on a two stage approach. A neighborhood is generated with two different exploration techniques; one for the activities and one for the modes. Basically, with an outer SA iteration, a mode neighborhood is searched. The search is simply carried out by changing mode assignments of activities with feasible assignments. When a solution is accepted with a mode change, then an activity neighborhood search is executed. For activity neighborhood, simply order of an activity is changed considering the feasibility of the new assignment. Mode neighborhood acceptance is carried out with a probabilistic approach whereas activity neighborhood solutions are accepted only when they result in an improved solution. With this two stage neighborhood search, a better exploration of the solution space is achieved. Note that the neighborhood solutions are always feasible. In addition to these modifications, the authors generate a number of initial solutions to achieve a differentiation in the initial solutions used. Finally, they use another parameter to determine the iteration limit of both neighborhood searches which is also cooled during the run of the algorithm. The experiments carried out by Jozefowska *et al.* (2001) show that the study of Bouleimen and Lecocq (2003) gives most promising results for the MRCPSP.

Alcaraz *et al.* (2003) propose a GA for the MRCPSP with renewable and nonrenewable resources. The solution representation of the GA differs from the other meta-heuristic applications in the literature. The order of the activities and modes are represented by the conventional two dimensional array. In addition to this array, there is a label in all the chromosomes which states the direction of scheduling generation scheme; forward or backward. In forward schedule generation, activities are scheduled according to the given order and precedence graph. In backward schedule generation, activities are scheduled with the given order on the reversed precedence graph. In other words, an activity is scheduled in backward scheduling scheme, if all its successors are scheduled. These two schemes can give different schedules because of the left or right shifts. The fitness value is the makespan, if the solution is resource feasible. On the other hand, if the solution is resource infeasible, the fitness value is calculated as the sum of three components; the deviation between the feasible maximum makespan and possible minimum project makespan (by selecting the least duration modes without considering resource limitations) of the infeasible solution at hand, the over usage of the resources and finally the obtained objective function. The crossover operation is carried out by using the forward or backward label to determine the relative order of the activities. Two different moves are consecutively applied for; changing the order of the activities and changing the assignment of the modes. Experimental results show that the proposed algorithm generally gives competitive results with the recent studies of Hartmann (2001) and Jozefowska *et al.* (2001).

Mika *et al.* (2005) propose a tabu search for MRCPSP with positive discounted cash flows where a positive cash flow is associated with each activity. Four different payment methods are considered in the problem environment: lump-sum payment at the completion of the project, payments at activities' completion times, payments at equal time intervals and progress payments. A SA and a TS algorithm is proposed each of which uses the same neighborhood search moves. The first move is changing the assignment of an activity in the activity list and the second move is changing the selected mode of an activity. A combined move which changes the assignment of an activity and its selected mode is also defined. The tabu list includes the reverse of the moves executed which cannot be applied if inspiration criteria (the move results with a better objective value) is not satisfied. An extensive experimental research is conducted with different payment methods for both of the solution approaches.

Mika *et al.* (2005) investigates MRCPSP with schedule dependent set-up times. In this problem a set-up is required when resources in different locations are assigned to an activity. In this case, the time required for the set-up occurs because of the preparation of required resources which can be in different places. A TS algorithm is proposed for solution approach where a solution is represented with three lists: one for activity sequence, one for mode assignments of activities and finally one for locations of activities. The neighborhood search moves are based on changing these lists. Tabu list is updated similar to study of Mika *et al.* (2005). Proposed TS approach is compared with improvement heuristics and random sampling.

In Table 2.4 a summary of the literature for MRCPSP is given.

Table 2.4. The summary of literature review for MRCPSP.

Authors	**Objective**	**Solution Approach**	**Notes**
Talbot (1982)	Makespan	A two stage heuristic	A new mathematical formulation for MRCPSP
Sprecher *et al.* (1997)	Makespan	B&B procedure	Suggested different bounding rules
Sprecher and Drexl (1998)	Makespan	B&B procedure	Extended work of Sprecher *et al.* (1997)
Boctor (1993)	Makespan	Rule based heuristic	Rules for activity and mode selection
Lova *et al.* (2006)	Makespan	Rule based heuristic	A multi pass scheme
Mori and Tseng (1997)	Makespan	Genetic algorithm	Forward-backward tracking for scheduling intervals
Hartmann (2001)	Makespan	Genetic algorithm	Infeasibility is allowed
Jozefowska *et al.* (2001)	Makespan	Simulated annealing	A two dimensional list for activity and mode selection
?	Makespan	Simulated annealing	A two stage neighborhood search
Alcaraz et al. (2003)	Makespan	Genetic algorithm	Forward-backward schedule generation
Mika *et al.* (2005)	NPV	Tabu search	A combined neighborhood search
Mika *et al.* (2005)	Makespan	Tabu search	Includes sequence dependent set-up times

2.2. Multi-project Scheduling

The multi-project scheduling problem consists of several projects where the individual projects have the characteristics defined in Section 2.1. The literature for multi-project scheduling is summarized in three parts. The first part is related with studies which basically deal with multi-project scheduling problem under RS policy. The second part presents resource related works, since the problem under consideration includes the determination of resource levels (capacities) subject to a given total budget. The last part of this Section introduces the studies related with different resource management policies beyond RS policy.

2.2.1. Multi-project Scheduling under the Resource Sharing Policy

Most of the studies in the multi-project scheduling literature employ Resource Sharing policy. In this policy, the general resource pool of the multi-project environment can be shared among projects without any restrictions or additional costs. In other words, the allocated renewable resource for individual projects can vary throughout the time periods, which implies a resource flow among projects. This approach allows application of different solution approaches to the problem. For example, a common approach is combining all projects into a big project network and attending the problem with well known single project solution approaches with minor modifications. This can easily be done by adding a dummy start node such that the start nodes of all projects that do not have any predecessors are assigned as successors of this dummy start node. Further, a dummy end node is added such that the end nodes of all projects that do not have any successor nodes are assigned as predecessors of this dummy end node.

The mathematical model for RPP under the RS policy and the corresponding sets (Table 2.5), parameters (Table 2.6) and decision variables (Table 2.7) are given below:

Table 2.5. The sets for RPP-RS.

V	Set of projects, $v \in V$
J_v	Set of activities of project v, $j \in J_v$
P_v	Set of all precedence relationships of project v
M_{vj}	Set of modes for activity j of project v, $m \in M_{vj}$
K	Set of renewable resources, $k \in K$
I	Set of nonrenewable resources, $i \in I$
T	Set of time periods, $t \in T$

Table 2.6. The parameters for RPP-RS.

E_{vj}	Earliest finish time of activity j of project v
L_{vj}	Latest finish time of activity j of project v
d_{vjm}	Duration of activity j operating on mode m
r_{vjkm}	Renewable resource k usage of activity j of project v, operating on mode m
w_{vjim}	Nonrenewable resource i usage of activity j of project v, operating on mode m
dd_v	Assigned due date for project v
c_v	Relative weight of project v
cr_k	Unit cost of renewable resource k
cw_i	Unit cost of nonrenewable resource i
tb	Total resource budget

Table 2.7. The decision variables for RPP-RS.

X_{vjmt}	$\begin{cases} 1 & \text{if activity } j\text{, operating on mode } m \text{ in project } v \text{ is finished at period } t \\ 0 & \text{otherwise} \end{cases}$
TC_v	Weighted tardiness cost of project v
R_k	Total amount of required renewable resource k
W_i	Total amount of required nonrenewable resource i

Mathematical Model RPP-RS

$$min.\ z = \sum_{v\in V} TC_v \tag{2.8}$$

Subject to

$$\sum_{m\in M_{vj}} \sum_{t=E_{vj}}^{L_{jv}} X_{vjmt} = 1 \quad \forall\ j \in N_v \text{ and } \forall\ v \in V \tag{2.9}$$

$$\sum_{m\in M_{vj}} \sum_{t=E_{vb}}^{L_{vb}} (t - d_{vbm})X_{vbmt} \geq \sum_{m\in M_{vj}} \sum_{t=E_{va}}^{L_{va}} tX_{vamt}$$
$$\forall\ (a, b) \in P \text{ and } \forall\ v \in V \tag{2.10}$$

$$\sum_{v\in V} \sum_{j\in N_v} \sum_{m\in M_{vj}} \sum_{q=t}^{t+d_{vjm}-1} r_{vjkm}X_{vjmq} \leq R_k \quad \forall\ k \in K\ \forall\ t \in T \tag{2.11}$$

$$\sum_{v\in V} \sum_{j\in N_v} \sum_{m\in M_{vj}} \sum_{t=E_{vj}}^{L_{vj}} w_{vjim}X_{vjmt} \leq W_i \quad \forall\ i \in I \tag{2.12}$$

$$\sum_{i\in I} cw_iW_i + \sum_{k\in K} cr_kR_k \leq tb \tag{2.13}$$

$$TC_v \geq C_v(\sum_{t=E_{vN}}^{L_{vN}} \sum_{m\in M_{vN}} tX_{vNmt} - dd_v) \quad \forall\ v \in V \tag{2.14}$$

$$X_{vjmt} \in \{0, 1\} \quad \forall\ j \in J, \forall\ t \in T, \forall\ m \in M_{vj} \text{ and } \forall\ v \in V \tag{2.15}$$

$$R_k, W_i, TC_v \in Z^+ \quad \forall\ v \in V, \forall\ k \in K \text{ and } \forall\ i \in I \tag{2.16}$$

Objective function (Equation 2.8) minimizes the total weighted tardiness cost over all projects. Constraint (Equation 2.9) ensures that all activities are scheduled once and only once for all projects. Constraint (Equation 2.10) implies predecessor relationships for all activities of all projects. Constraint sets (Equation 2.11) and (Equation 2.12) limit the renewable and nonrenewable resource usage and consumption, respectively. Note that in the mathematical programming model RPP-RS, all activities of all projects share the same resource pool represented by constraints (Equation 2.9) and (Equation 2.10). In constraint (Equation 2.13) the cost of the general resource pool is limited with total budget. The weighted tardiness values for individual projects are calculated in constraint set (Equation 2.14). Finally constraints (Equation 2.15) and (Equation 2.16) define feasible ranges for decision variables.

The renewable resource utilization of projects in a shared resource environment will change during the project duration but it will be restricted with the determined total resource capacity. In Figure 2.1 below, it can be seen that the utilized resource values during the project duration change from period to period for projects.

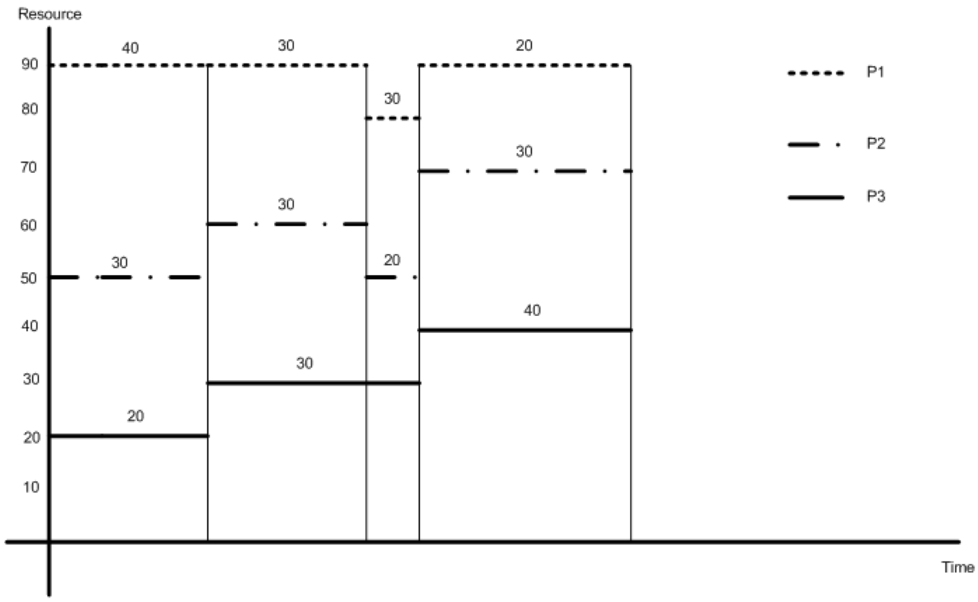

Figure 2.1. Example of resource allocation under shared resource policy.

One of the first studies on multi-project scheduling is a general integer programming formulation proposed by Pritsker *et al.* (1969). The proposed formulation covers most of the aspects of multi-project scheduling like renewable resource constraints, multiple modes and preemption. The terminology in the paper is as follows. Jobs are individual operations which require a set of resources and they can be executed in different ways. Jobs in turn form projects and can have precedence relations among themselves. Projects have due dates and earliest and latest possible completion times. This setting can be easily converted to a multi-project environment with multiple modes and renewable resource constraints. The proposed formulation and the corresponding sets (Table 2.8), parameters (Table 2.9 and Table 2.10) and decision variables (Table 2.11) is given below;

Table 2.8. The sets for multi-project scheduling formulation.

V	Set of projects, $v = 1 \ldots \lvert V \rvert$
N_v	Set of jobs, $j = 1 \ldots \lvert N_v \rvert$ for project v
K	Set of resources, $k = 1 \ldots \lvert K \rvert$
P_{vn}	Set of jobs that must immediately precede job n of project v

Table 2.9. The parameters for multi-project scheduling formulation.

G_v	Absolute due date defined for project v
g_v	Desired due date for project v (project v is not late if it finishes before or on g_v)
u_v	Earliest possible period in which project i could be completed, $v \in V$
a_{vj}	Arrival of job j, $j \in N_v$, of project i, $t \in G_v$ (arrivals occur at the beginning of periods)
d_{vj}	Number of periods required to perform job j, $j \in N_v$, of project v, $i \in V$
e_{vj}	The earliest possible period in which job j, $j \in N_i$, of project v, $v \in V$, can be completed

Table 2.10. The parameters for multi-project scheduling formulation (cont.).

l_{vj}	The latest possible period in which job j, $j \in N_i$, of project v, $v \in V$, can be completed
r_{vjk}	Amount of type k resource required by job j, $j \in N_v$, of project v, $i \in V$
gr_{kt}	Amount of type k, $k \in \lvert K \rvert$, resource available in period t, $t \in G_v$
p_{vt}	Total tardiness cost incurred when project v, $i \in I$, is completed in period t, $t \in G_v$

Table 2.11. The decision variables for multi-project scheduling formulation.

X_{vjt}	$\begin{cases} 1 & \text{if job } j \text{ in project } v \text{ is completed in period } t \\ 0 & \text{otherwise} \end{cases}$
Y_{vt}	$\begin{cases} 1 & \text{if all jobs of project } v \text{ are completed before period } t \\ 0 & \text{otherwise} \end{cases}$

Note that decision variable Y_{vt} takes a value of one for all periods following the period in which all jobs of the project i is completed.

Mathematical Model MPS2

$$z = min \sum_{i \in I} \sum_{t=g_i+1}^{G_i} p_{it}(1 - Y_{it}) \tag{2.17}$$

Subject to

$$\sum_{t=e_{ij}}^{l_{ij}} X_{ijt} = 1 \qquad \forall\, i \in I \text{ and } \forall\, j \in J \tag{2.18}$$

$$Y_{it} \leq (1/N_i) \sum_{j \in N_i} \sum_{q=e_{ij}}^{t-1} X_{ijq} \qquad \forall\, v \in V \text{ and } t = u_i, u_{i+1}, \ldots, G_v \tag{2.19}$$

$$\sum_{t=e_{im}}^{l_{im}} tX_{imt} + d_{in} \leq \sum_{t=e_{in}}^{l_{in}} tX_{int} \qquad \forall\, n \in N\ \forall\, m \in P_{in} \tag{2.20}$$

$$\sum_{i \in I} \sum_{j \in N_i} \sum_{q=t}^{t+d_{ij}-1} r_{ijk} X_{ijq} \leq gr_{kt} \qquad \forall\, t \in T \text{ and } \forall\, k \in K \tag{2.21}$$

$$X_{vjt} \in \{0,1\} \qquad \forall\, v \in V,\, \forall\, j \in J \text{ and } t = 1,2,\ldots,G_v \tag{2.22}$$

$$X_{vt} \in \{0,1\} \qquad \forall\, v \in V \text{ and } \forall\, t = 1,2,\ldots,G_v \tag{2.23}$$

Objective function (Equation 2.17) ensures that the total tardiness cost is minimized. Constraint (Equation 2.18) stands for job assignment whereas constraint (Equation 2.19) ensures the completion of a project. Constraint (Equation 2.20) satisfies precedence relations between jobs in a project. Constraint (Equation 2.21) limits resource usage with the given capacity. The mathematical model given above can be defined in a simpler way by eliminating Y_{it} decision variables and using the decision variable X_{ijt} corresponding to the last operation of projects. But to generalize the mathematical model for other concerns like preemption, authors choose this way of formulation. Substitution of resources or different modes for jobs is defined by creating decision variables for each mode and ensuring that only one of the modes will be active. Job splitting is applied by defining two different jobs that can not simultaneously be carried out. The proposed integer program is compared with priority rule based heuristics and it is seen that it can find an optimal solution for small problems.

Kurtuluş and Narula (1985) analyze the tardiness cost performance of scheduling rules for multi-project problem with different weights for projects. Note that in this study, activities have only one execution mode. The important aspect of the study is that the authors clearly show the distinction between solving the problem by combining projects into a large single project and solving them separately under the proposed objective functions. To see this, define; CT_i as the resource-constrained completion time of project i, CP_i as the unconstrained completion time of project i and finally M as the total number of projects.

The objective used for the multi-project approach;

$$z = \sum_{i=1}^{M} (CT_i - CP_i) \tag{2.24}$$

The objective used for the single project approach;

$$z = \max_i CT_i - \max_i CP_i \tag{2.25}$$

It is clear that these two different objective functions, (2.24) and (2.25) will give different schedules. The authors investigate the behavior of different scheduling rules for equal weight and unequal weight cases. Some of the scheduling rules used in the paper are weighted duration, maximum penalty and minimum total work content. In addition to this, Kurtuluş and Narula (1985) propose various parameters for the characterization of the project networks like Maximum Load Factor (MLF) and Average Utilization Factor (AUF) which are commonly used to characterize project networks in the project scheduling literature. Apart from these summary measures related with the characteristics of activities, the penalties for projects are defined with different functions which are related with the total work content and the critical path. It is observed that the scheduling rules behave distinctively for different ranges of the above defined project network characterizing parameters.

Authors generate a large number of problems which have different characteristics based on the summary measures and penalty functions. An experimental design is applied on the problem setting. Under unequal penalties, Maximum Penalty (MAXPEN) gives the best results. MAXPEN is followed with a large margin by Shortest Activity from Shortest Project (SASP). The results for equal penalties show that SASP and Minimum Slack (MINSLK) are close best rules.

Tsubakitani and Deckro (1990) apply the scheduling rules for multi-project problems proposed by Kurtuluş and Narula (1985) by creating an applicable framework for housing industry. First of all, the characteristics for the different projects of housing industry are determined by the measures proposed by Kurtuluş and Narula (1985) which in turn determine the specific scheduling rules used for the different settings. The proposed framework includes an update scheme for the ongoing projects which enables rescheduling.

Lawrance and Morton (1993) propose resource pricing based priority heuristic rules for multi-project scheduling. The aim is minimizing the total weighted tardiness cost of projects where weight is defined as the relative importance of a project. The heuristic rules are generally based on the following equation adopted from the well known Rachamadugu and Morton (R&M) rule;

$$\text{Activity Priority} = \frac{\text{Project Weight} \times \text{Activity Urgency}}{\text{Implicit Remaining Project Cost}} \tag{2.26}$$

Rachamadugu and Morton (R&M) rule is modified to employ the above formulation as follows:

$$P_{ij} = \frac{W_i}{\pi_{ij}} \exp\left(-\frac{(l_{ij} - t)^+}{k\bar{p}}\right) \tag{2.27}$$

where P_{ij} is the priority of activity j of project i, $\bar{p}$ is the average processing duration of all activities, π_{ij} is the imputed resource cost of activity j of project i, k is a look-ahead parameter and l_{ij} is the late start time of activity j of project i as calculated in the unconstrained version of the scheduling problem and W_i is the weight of the project.

The proposed priority rule is tested against more than 20 benchmark heuristics after a series of parameter estimation experiments. The experiments show that the different resource pricing heuristics do not differ significantly among themselves but give better results compared to the benchmark heuristic rules.

Speranza and Vercellis (1993) propose a two stage approach for multi-project scheduling where all the projects can have precedence relations among themselves. The first stage of the approach aims to define projects as “activities” with multiple modes, where each mode is determined by solving a mathematical model for the project with a given budget that yields a finish time for the project. The authors assume that a set of possible budget limitations for a project can be estimated which in turn constitute different modes for a project, which are called macro modes. The network formed in this manner is expressed as an RCPSP where the objective is the maximization of the net present value. The solution to this problem gives the required start and finish times of the projects, and the total nonrenewable and renewable resource capacities that a project can use in a given period. In the second stage of the framework, this information is used for detailed scheduling of individual projects for makespan minimization, and the activity mode assignments of each project within the given amount of renewable and nonrenewable resources are obtained. It has been shown by Hartmann and Sprecher (1996) that the exact solution procedure by Speranza and Vercellis (1993) might not lead to an optimal solution for two or more renewable resource types and for instances with nonrenewable resources it might not even reach a feasible solution. Speranza and Vercellis (1993) carry out an experimental study with real world data from a construction company. Generally, the execution time is determined by the number of macro modes defined. To find a good trade-off between the solution quality and execution time authors conduct an iterative approach by increasing the number of macro modes till a reasonable result is obtained.

Work of Speranza and Vercellis (1993) is extended by Can and Ulusoy (2011). For macro mode generation, a new mathematical formulation is proposed where the results of macro mode generation model of Speranza and Vercellis (1993) is used. Macro-project scheduling phase is improved with different procedures. First of all, the time horizon is determined with a mathematical model which directly decreases number of decisions for macro-project scheduling phase. Another improvement is achieved with a post-processing procedure which distributed the left over renewable capacity. To solve the macro-project scheduling problem, a GA is proposed. Note that this GA also is used for initial solution generation for exact solution approaches. The reported test results show that the modifications and new procedures proposed for the problem are efficient.

Another hierarchical approach for multi-project scheduling is proposed in Kim and Leachman (1993). The problem environment consists of multiple projects with both renewable and nonrenewable resource constraints. The scheduling of activities is based on project selection and activity selection from the selected project. The overall scheduling procedure is an iterative one. It starts with an initial early and late finish times for projects and updates these values at each iteration. The procedure iterates till predetermined value is reached for the total weighted tardiness over all projects.

Yang and Sum (1997) approach the multi-project scheduling with various considerations such as due date assignment, resource allocation, project release and activity scheduling. The authors see multi-project problems as a dual level management problem in a dynamic environment where new projects arrive during the execution of the current set of projects. The problems in the first level are related with individual project scheduling tasks, which are activity selection and resource allocation. The upper management decisions are related with release dates of projects and due date assignments. For these problems the authors propose priority rules. For due date assignment, workload related and CPL related priority rules are defined. The release of projects is handled by limiting the number of active projects. Yang and Sum (1997) is one of the first studies that pointed out RS assumption is not a strong generalization in multi-project scheduling.

Lova *et al.* (2000) study multi-project scheduling with different objectives at different levels. The authors' approach to the problem consists of two levels and two different objective types associated with each level. First, the problem is solved with the objective of minimizing mean project duration or maximum project duration; namely, time related objectives. A forward-backward heuristic is applied to achieve these objectives. With a starting feasible solution, activities are left and right shifted according to different priority rules for different objectives. When no further improvement can be achieved with forward-backward heuristics, resource related objectives are considered and heuristics such as maximum total work content are applied to minimize idle resources and to achieve resource leveling.

Gonçalves *et al.* (2008) use a GA approach to multi-project scheduling problems where only renewable resources are considered. The basic approach of the authors is combining all activities of the projects into a big project network and solving this combined project network using GA. The chromosome structure is different from the generally used structures in the project scheduling literature. The genes corresponding to activities are priority coefficients for the activities and are used in the schedule generation procedure. In addition to these genes, there are different genes for delay times of activities and start times for individual projects for schedule generation. The performance measure of an individual in the GA is a composition of tardiness, earliness and deviation of project flow time from the makespan calculated for the unconstrained case using CPM.

Mittal and Kanda (2009) propose a two-phase heuristic for multi-project scheduling. Activities do not have alternative modes and only renewable resources are present. The objective is minimizing the makespan and deviation from CPL. The rationale behind the two-phase heuristic is the distinction between project and activity selection. Basically, project selection is done first (phase one) and an activity is selected from the selected project (phase two) with defined priority rules for both project and activity selection. The priority rules for activity selection are the common priority rules used in the literature like minimum duration or minimum slack. The priority rules used for project selection are CPL, remaining CPL, total work content and total work remaining.

The literature for multi-project scheduling is summarized in Table 2.12 below.

Table 2.12. The summary of literature review for multi-project scheduling.

Authors	**Objective**	**Solution Approach**	**Notes**
Pritsker *et al.* (1969)	Total project duration	Exact	Presents a general formulation
Kurtuluş and Narula (1985)	Weighted tardiness	Rule based	Presents different measures for multi-project scheduling
Tsubakitani and Deckro (1990)	Makespan	Rule based	An application for the study of Kurtuluş and Narula (1985)
Lawrance and Morton (1993)	Weighted tardiness	Rule based	Based on Rachamadugu and Morton rule
Speranza and Vercellis (1993)	NPV and makespan	Two stage	Introduced macro mode concept
Can and Ulusoy (2011)	NPV and makespan	Two stage	Extended macro mode concept
Kim and Leachman (1993)	Makespan	Rule based	Iterative approach
Yang and Sum (1997)	Due date based	Rule based	Introduces a dynamic environment
Lova *et al.* (2000)	Makespan and resource leveling	Rule based	A two stage approach
Gonçalves *et al.* (2008)	Due date based	Genetic algorithm	Activity priority based chromosome structure
Mittal and Kanda (2009)	Makespan	Rule based	A two-phase approach

2.2.2. Resource Related Decisions in Multi-Project Scheduling

In this Section, the studies addressing the determination of resource requirements of projects and other resource related concerns such as rough-cut capacity planning or resource availability cost problem are presented in detail.

Resource related decisions are an important part of the multi-project scheduling problems since the coupling of individual projects in the problem environment generally is achieved with the resource related constraints. Although studies in multi-project scheduling literature generally assume predefined general resource capacity set, the multi-project problem environment considered in this dissertation includes the determination of resource levels (capacities).

Möhring (1984) investigates the resource requirement problem for project scheduling with a graph theoretical approach. The author defines two important problem classes for project scheduling. The first problem, *problem A*, is defined as a problem of scarce resources where the objective is finding the shortest project time with a given amount of resources. The second problem, *problem B*, is defined as a problem of scarce time where the objective is finding the least cost resource requirements within a given project duration. The author uses graph theory to represent and solve these project scheduling problems. Apart from the solution methodologies for these problems, the important part of the study is the realization of the interaction between the two different problems. Since solution procedures for problem A have a better algorithm behavior, one can find better solution procedures for *problem B*, if a way of transformation can be found among the problems. The author notes that the objective of *problem A* is presented as a constraint in problem B and vice versa. Using this rationale, a special “dual” relation between these two problems is defined and used in the solution procedures proposed. Note that, the problem under study in this dissertation is of the form of *problem A*.

Another study in the form of *problem B* is presented by Demeulemeester (1995) and it is defined as resource availability cost problem (RACP). The general idea is determining the least cost resource requirements for a single project scheduling problem. The proposed formulation for RACP is similar to the formulations for resource constrained project scheduling problem (RCPSP) with the basic differences being in the objective function and in the constraint for project duration. The solution methodology proposed for RACP is based on decision problems defined with resource limits. These definitions serve as cuts for the feasible region for the problem, and the algorithm employs efficient capacity points (resource availability value in the solution that is greater than the other resource availabilities) to obtain the optimal solution. The experiments have demonstrated that the solution procedure is very effective.

An approach dealing with the determination of resource requirements of projects is the rough-cut capacity planning (RCCP) for a multi project environment. A recent example is provided in Gademann and Schutten (2004). It is assumed that a project can be broken into jobs that stand for activity groups, which are not known yet. For each of these estimations, a precedence relation structure is constructed. In addition to this, a multi-project environment is modeled as a single project problem with all the estimated activity groups. Two variants of the RCCP problem are defined in line with *Problem A* and *B*, namely resource driven and time driven. In the resource driven RCCP, only the regular capacity can be used and the objective is minimizing the total lateness. In the time driven RCCP, a project delivery time is to be met by determining the non-regular capacity (i.e., the additional capacity beyond the already available capacity) and the objective is minimizing the cost of this non-regular capacity. The authors propose a linear program for the time driven RCCP problem. The model allows for precedence infeasible solutions and preemptions and heuristic procedures are proposed to repair such solutions. To overcome the precedence infeasibility, the mathematical model is changed slightly to determine the time windows for activities to ensure feasibility. The solutions of the modified linear program are improved by local search heuristics based on dual relations between the time windows and the decision variables.

2.2.3. Multi-project Scheduling under Different Resource Management Policies

Even though RS policy is a very common assumption in the multi-project scheduling literature , there are studies pointing out this issue and/or propose different resource management heuristics. Yang and Sum (1997) was one of the first studies underlining the weakness of generalizing RS policy. Apart from that, Krüger and Scholl (2008) classified different resource transfers for different multi-project environments. Authors defined three different approaches for dealing with resource transfers; transfer neglecting, transfer reducing and transfer using approach. Transfer neglecting approach is conceptually RS policy where there is not any restriction on resource flow between projects whereas transfer reducing approach does not allow RS and resources are dedicated to projects. And finally in transfer using approach resources can be transferred between projects with explicit costs. Note that the individual projects in the problem environment inherit the characteristics of an RCPSP. The transfer using approach is further investigated in Krüger and Scholl (2009). Authors propose different heuristic rules for the parallel and serial scheduling schemes modified for the resource transfer concept.

3. RESOURCE DEDICATION PROBLEM

3.1. Problem Definition and Mathematical Model for Resource Dedication Problem

RPP is a complex problem with different conceptual problems it inherits; determination of general resource capacities from the general resource budget, dedication of resource to projects and scheduling of individual projects with the given resource dedication values. Thus, to understand the nature of the problem, first we have narrowed the problem domain and tried to identify the characteristics of RD policy by excluding resource portfolio part (determination of general resource capacity values from the general budget). The multi-project scheduling problem with the given general resource capacities under the RD policy is called the Resource Dedication Problem (RDP) in this study. The basic aim of defining and solving this problem is to gain insights for RD policy and employ these to RPP under the RD policy.

RDP is the dedication of a set of limited resources to a set of projects in a multi-project environment in such a way that individual project schedules would result in an optimal schedule for a specified objective. Here, the objective is the minimization of the total weighted tardiness over all projects. Once the resources are dedicated to individual projects, they are not anymore allowed to be shared with the other projects and the problem becomes solving an MRCPSP for each project with the dedicated resource amounts. The proposed formulation for RDP and the corresponding sets (Table 3.1), parameters (Table 3.2) and decision variables (Table 3.3) are given below;

Table 3.1. The sets for RDP.

N	Set of activities, $j \in N$
M_j	Set of modes for activity j, $m \in M_j$
I	Set of nonrenewable resources, $i \in I$
P	Set of all precedence relationships
T_v	Set of time periods for project v

Table 3.2. The parameters for RDP.

E_{vj}	Earliest finish time of activity j of project v
L_{vj}	Latest finish time of activity j of project v
d_{vjm}	Duration of activity j operating on mode m
r_{vjkm}	Renewable resource k usage of activity j of project v, operating on mode m
w_{vjim}	Nonrenewable resource i usage of activity j of project v, operating on mode m
dd_v	Assigned due date for project v
c_v	Relative weight of project v
cr_k	Unit cost of renewable resource k
cw_i tb	Unit cost of nonrenewable resource i Total resource budget
r_k	Total amount of available renewable resource k
w_i	Total amount of available nonrenewable resource i

Table 3.3. The decision variables for RDP.

X_{vjmt}	$\begin{cases} 1 & \text{if activity } j \text{ of product } v\text{, operating on mode } m \text{ is finished} \\ & \text{at period } t \\ 0 & \text{otherwise} \end{cases}$
TC_v	Weighted tardiness cost of project v
BR_{vk}	Total amount of dedicated renewable resource k to project v
BW_{vi}	Total amount of dedicated nonrenewable resource i to project v

Mathematical Model RDP

$$min.\ z = \sum_{v \in V} TC_v \tag{3.1}$$

Subject to

$$\sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{jv}} X_{vjmt} = 1 \quad \forall\ j \in N_v \text{ and } \forall\ v \in V \tag{3.2}$$

$$\sum_{m \in M_{vj}} \sum_{t=E_{vb}}^{L_{vb}} (t - d_{vbm}) X_{vbmt} \geq \sum_{m \in M_{vj}} \sum_{t=E_{va}}^{L_{va}} t X_{vamt}$$

$$\forall\ (a, b) \in P \text{ and } \forall\ v \in V \tag{3.3}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{q=t}^{t+d_{vjm}-1} r_{vjkm} X_{vjmq} \leq BR_{vk}$$

$$\forall\ v \in V, \forall\ k \in K \text{ and } \forall\ t \in T_v \tag{3.4}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{vj}} w_{vjim} X_{vjmt} \leq BW_i \qquad \forall\ i \in I \tag{3.5}$$

$$\sum_{v \in V} BR_{vk} \leq r_k \qquad \forall\ k \in K \tag{3.6}$$

$$\sum_{v \in V} BW_{vi} \leq w_i \qquad \forall\ i \in I \tag{3.7}$$

$$TC_v \geq C_v \Big(\sum_{t=E_{vN}}^{L_{vN}} \sum_{m \in M_{vN}} t X_{vNmt} - dd_v \Big) \quad \forall\ v \in V \tag{3.8}$$

$$X_{vjmt} \in \{0, 1\} \quad \forall\ j \in J, \forall\ t \in T, \forall\ m \in M_{vj} \text{ and } \forall\ v \in V \tag{3.9}$$

$$BR_{vk},\ BW_{vi} \text{ and } TC_v \in Z^+ \quad \forall\ v \in V, \forall\ k \in K \text{ and } \forall\ i \in I \tag{3.10}$$

Objective function (Equation 3.1) minimizes total weighted tardiness cost over all projects. Constraint set (Equation 3.2) ensures that all activities are scheduled once and only once for all projects. Constraint set (Equation 3.3) implies predecessor relationships for all activities of all projects. Constraint set (Equation 3.4) sets the renewable resource dedications for each project. Constraint set (Equation 3.5) determines nonrenewable resource dedications for each project. Constraint sets (Equation 3.6) and (Equation 3.7) limit the total dedicated renewable and nonrenewable resources according to the given general resource capacities, respectively. Constraint set (Equation 3.8) calculates the weighted tardiness cost for project v. Note that general renewable and nonrenewable resource capacities are given as parameters in this problem. The resource dedication concept is achieved with constraint set (Equation 3.4) where the corresponding decision variable does not have a time index.

3.2. Solution Methodologies

The given mathematical model conceptually contains two different problems: (i) Dedication of the resource capacities to projects and (ii) scheduling of the individual activities of the corresponding project. Scheduling of activities with multi-modes under nonrenewable and renewable resource constraints as a generalization of resource constrained project scheduling problem is shown to be NP hard by Kolish *et al.* (1995). The given multi-project problem not only encapsulates a multiple number of this problem but also the decision for resource dedications for the projects. Thus, the exact solution methodologies for the given formulation become unpractical for reasonable problem sizes. Therefore, heuristic solution approaches will be employed here for the solution of the Model RDP above. Besides the exact solution approach, two heuristic procedures are introduced here: (i) A GA with a new local improvement heuristic called combinatorial auction (CA) for RDP; and (ii) a Lagrangian relaxation based heuristic.

3.2.1. A Genetic Algorithm Based on Combinatorial Auction for the Resource Dedication Problem

In this dissertation, GA is used as an intelligent search infrastructure supported by the new local improvement heuristic CA for RDP. Below detailed information about CA for RDP is given.

<u>3.2.1.1. Combinatorial Auction for the RDP.</u> CA for RDP is a local improvement heuristic based on the preferences of the projects for the resources. The preference of a project for a resource can be defined as the value of the resource for that project according to the current state of the resources of the project. Hence, the preference of a project for a resource can be taken as an indication for a possible improvement that would result in the objective when a unit of that resource is obtained by some means by that project. If one can calculate the preferences of the projects for the resources for a given resource dedication state for the projects, then the current solution can be moved to a more preferable solution with respect to the preferences. If the preference calculation reflects the importance of the resources for the projects correctly, then this move will improve the objective. In this dissertation, two different preference calculation approaches are used: (i) Linear relaxation (LR) based and (ii) Lagrangian relaxation (LA) based.

Linear Relaxation Based Preference Calculation

If the linear relaxation of the single project scheduling model (Equation 2.1)-(Equation 2.6) Talbot (1982) is solved for each project with some set levels of dedicated resources, the solution can be used to determine the preferences of the projects for the resources. LR of the given formulation (SP-LR) results in two basic information: (i) Dual variable values (or shadow prices) and (ii) allowable upper and lower bounds for the right hand sides (RHS) of the constraints. According to the duality theory, dual variable values can be interpreted as the amount of the improvement in the objective function, if the RHS of the constraint is increased by one unit. On the other hand, allowable upper and lower bounds represent the limits of the RHS values where the optimal basis will not change. There are two basic groups of constraints in the project scheduling model. First group is the sets of technology constraints (activity assignment (Equation 2.2) and precedence constraints (Equation 2.3)), and the other group consists of the sets of resource capacity constraints (renewable (Equation 2.4) and nonrenewable (Equation 2.5) resource constraints). Resource capacity constraints are not generally binding in the solution of SP-LR, this in turn results with distinct allowable upper and lower bounds from the RHS values of resource capacity constraints. If the resource constraints are increased to their allowable upper bounds, the new basis should give, at worst, as good a feasible solution as the previous basis, since the capacity is increased for the resource. The allowable upper bounds for the RHS of the resource constraints can be used to determine the sensitivity of the projects for resources and can be used as preferences for these resources. In other words, the resource levels that cause a change in the optimal basis in the LR are interpreted as a direction for the resource levels that can improve the objective of the original problem. The preference for a resource by a project is defined as the closeness of the allowable upper bound to the current RHS value as follows:

For a renewable resource k

$$a_{kv} = \frac{1}{\min\limits_{t} \{AUB_{ktv}\} - DR_{kv}} \tag{3.11}$$

$$p_{kv} = \frac{a_{kv}}{\sum\limits_{v}^{V} a_{kv}} \tag{3.12}$$

For a nonrenewable resource i

$$a_{iv} = \frac{1}{AUB_{iv} - DW_{iv}} \tag{3.13}$$

$$p_{iv} = \frac{a_{iv}}{\sum_{v}^{V} a_{iv}} \tag{3.14}$$

where a_{kv} and a_{iv} are the closeness of the allowable upper bounds to the dedicated renewable and nonrenewable resource levels, respectively; AUB_{ktv} is the allowable upper bound value for renewable resource k resulting from the usage constraint for time period t (constraint set (Equation 2.4)); AUB_{iv} is the allowable upper bound value for nonrenewable resource i resulting from the usage constraint for the project (constraint set (Equation 2.5)); DR_{kv} and DW_{iv} are the dedicated renewable and nonrenewable resources for project v, respectively; p_{kv} and p_{iv} are the preferences of project v for renewable resource k and nonrenewable resource i, respectively.

Lagrangian Relaxation Based Preference Calculation:

Another preference calculation approach can be defined by employing Lagrangian relaxation. Consider the Lagrangian relaxation of the modified MRCPSP formulation, given below, where both nonrenewable and renewable resource constraints are relaxed:

Mathematical Model LA-SP

$$\begin{aligned} min.\ z_{LA-SP} = TC + \sum_{k=1}^{K}\sum_{t=1}^{T} \lambda_{kt} \left\{ \sum_{j=1}^{N}\sum_{m=1}^{M_j}\sum_{q=t}^{t+d_{jm}-1} r_{jkm}X_{jmq} - DR_k \right\}^{+} \\ + \sum_{i}^{I} \mu_i \left\{ \sum_{j=1}^{N_v}\sum_{m=1}^{M_j}\sum_{t=E_j}^{L_j} w_{jim}X_{jmt} - DW_i \right\}^{+} \end{aligned} \tag{3.15}$$

Subject to

(2.2), (2.3), (2.6)

$$TC \geq C\left(\sum_{t=E_N}^{L_N}\sum_{m=1}^{M_N} X_{Nmt} - DD\right) \tag{3.16}$$

$$TC = TC_{min} \tag{3.17}$$

where DR_k and DW_i are dedicated renewable and nonrenewable resource values respectively, TC_{min} is the minimum reachable weighted tardiness for the project, which can be easily calculated from the project network employing CPM without considering any resource constraints.

This formulation is exactly the Lagrangian relaxation model of MRCPSP except constraint (Equation 3.17), which sets the weighted tardiness equal to the calculated minimum tardiness for the project. In addition to this, in the calculation of the objective function LA-SP, penalty values are added if they are nonnegative. In other words, only the renewable and nonrenewable resource usages exceeding the given capacities DR_k and DW_i are included in the summations in (Equation 3.15).

First, note that this Lagrangian relaxation formulation is not for solving the MR-CPSP. Rather, it is used to calculate the preferences of the projects for the resources. The rationale of these modifications is to define a methodology to accurately calculate preferences. Since the total weighted tardiness cost part of the objective is set to the minimum reachable value, the formulation will accept the necessary resource infeasibilities to reach that minimum possible weighted tardiness value. Thus, when the subgradient optimization approach is applied to this formulation, after a number of iterations, the values of λ_{kt} and μ_i can be used as an estimation for the value of the corresponding resource to the project, or in other words, as the preference of the project for that resource to reach the minimum possible weighted tardiness. In addition to this, penalizing only the resource over-utilizations prevents the abuse of the Lagrangian objective function by under-utilization of the resources . Note that, especially for the renewable resource case, where only the constraint for the time period with the maximum usage will be active, the abuse will be excessive.

For each resource, when a series of sub-gradient optimization iteration is executed, the preference for renewable (Equation 3.18) and nonrenewable (Equation 3.19) resources can be calculated for each project v can as follows:

$$p_{kv} = \max_{t} \{\lambda_{vkt}\} \tag{3.18}$$

$$p_{iv} = \mu_{vi} \tag{3.19}$$

Moving to a More Preferable Solution:

After calculating the preferences of the projects for the resources, the next issue is moving to a more preferable solution using these preferences. This is carried out by calculating the slack resources (resources that are not used for the current solution of the projects) and distributing these slack resources among the projects according to the projects' preferences. The amount of a resource that a project can give up can easily be determined from the solution of the individual project scheduling problem. The smallest slack for each renewable resource observed over the project duration and the slacks of the nonrenewable resource capacity constraints will become the surpluses of the corresponding resources for a project. Preference and slack values can be used to form a continuous knapsack problem to distribute the calculated slack resources for maximizing the total gain for all projects. The knapsack models for renewable and nonrenewable resources are given below.

For nonrenewable resources

$$max.\ z = \sum_{i} \sum_{v} p_{iv} y_{iv} \tag{3.20}$$

Subject to

$$\sum_{v} y_{iv} \leq b_i \quad \forall i \in I \tag{3.21}$$

$$y_{iv} \in R^{+} \tag{3.22}$$

For renewable resources

$$max.\ z = \sum_k \sum_v p_{kv} y_{kv} \tag{3.23}$$

Subject to

$$\sum_v y_{kv} \leq b_k \quad \forall k \in K \tag{3.24}$$

$$y_{kv} \in R^+ \tag{3.25}$$

where p_{iv} (p_{kv}) is the preference of project v for resource i (k), b_i (b_k) is the amount of resource i (k) available for dedication to different projects, and y_{iv} (y_{kv}) is the non-negative continuous decision variable for the amount of spare resource i (k) dedicated to project v. This formulation tries to maximize the total preference value gained by dedicating the spare resources to projects.

Solving MRCPSP for each Project:

When resource dedication values for each project are determined the problem reduces to MRCPSP for each project where the resource dedication value for each project becomes the renewable and nonrenewable resource capacities. To solve these problems *CPLEX 11.2* is employed with the mathematical formulation of Talbot (1982). Since CA for RD is applied over and over through the execution of GA, some modifications are needed to facilitate the solution of MRCPSP for each project. There are two basic tasks when solving an MRCPSP with *CPLEX*, model generation and executing the solution procedure.

There is one decision variable in MRCPSP formulation which stands for the activity finish times. When this decision variable set is kept same, the basic difference between two resource dedication instances for a given project is only the renewable and nonrenewable resource capacity values. Thus, a generic model can be used to represent a project and when a solution is needed for a given resource dedication instance for the corresponding project, then the only required task is changing the right hand side of the renewable and nonrenewable resource constraints. This can easily be achieved by defining a decision variable for each renewable and nonrenewable resource and setting these decision variables to the given renewable and nonrenewable resource dedication values with equality constraints. The related constraints for these tasks is deleted and re-added to the generic models before the solution procedure. Note that to use generic models, a large time period which can cover all the possible renewable and nonrenewable resource capacity values is needed. This time period is calculated by determining minimum renewable and nonrenewable resource requirements for each resource and applying the Simulated Annealing approach proposed by **?**. The minimum renewable resource requirement is calculated by finding the minimum renewable resource requirement for each project over all modes and taking maximum of these values over all projects. Then minimum nonrenewable resource requirement can be calculated from the selected modes in the aforementioned procedure. Note that, this time period calculation results with a very large number of activity finish decision variable set.

Generic models for each project reduces the model generation time but has a negative effect on execution time of solution procedure of *CPLEX 11.2* because of the large activity finish decision variable set. To compensate this inefficiency a couple of modifications are used. First of all, all solved cases for each problem is stored, and when a resource dedication value is not changed for a project then the stored solution is used for that case. In addition to this the activity finish decision variables are reduced by giving upper bounds to the corresponding generic model employing resource dominant cases. When a resource dedication instance for a given project has higher resource capacity values for all renewable and nonrenewable resources than one of the stored solved problems, then the makespan of the corresponding solution is given as an upper bound to the generic model. This is achieved by limiting the makespan of the project with this upper bound, where *CPLEX* cleans the redundant decision variables in pre-processing phase.

3.2.1.2. The Basics of the Genetic Algorithm Applied. An individual in the proposed GA is defined so as to reflect a multi-project environment with dedicated resource values. Only resource feasible individuals are present in the population. In Table 3.4 the general view of an individual is given. Note that a chromosome represents a resource dedication instance with the determined resource allocations to the projects. This representation can also be interpreted as a single string that represents resource dedication values for projects.

Table 3.4. General representation of an individual

	Projects		
Resources	**Project1**	**Project2**	**Project3**
R1	15	12	20
R2	8	14	21
R3	10	21	11
R4	15	12	13

There are three different ways to generate individuals for the initial population.The first two methods employ the no-delay resource requirements of the projects. No-delay resource requirement of a project is calculated by solving the resource unconstrained project scheduling problem and calculating the resource requirement from the resulting schedule. In the first method, resource dedications are determined proportional to the no-delay resource requirements of the projects with respect to the given general resource capacities. One individual is generated in this manner. In the second method, for each project, the no-delay resource requirement is satisfied from the general resource capacities, and resource dedications of the remaining projects are determined proportional to the no-delay resource requirements of the corresponding projects. And finally, resource dedications are randomly generated for each project. Based on the preliminary test runs the population size is selected as ten.

Here, three basic procedures are employed to create new individuals in the proposed GA; namely (i) crossover, (ii) mutation and (iii) CA for RDP.

Crossover Operations: Generally speaking, a crossover operator selects two different individuals from the population in a predetermined way and exchanges sub-strings from the selected individuals to generate two new offsprings. There are three different types of crossover operations defined for the proposed GA. The first crossover operation takes two projects from different individuals and changes sub-strings of resource dedications for the selected projects in different individuals. The second crossover operation similarly selects a resource and exchanges the resource dedications for all projects of the parents. The last crossover operation selects a sub-string where the cut off points do not have to be project or resource changing positions of the chromosome, and then exchanges these sub-strings on the parents. The general idea behind these crossover operations is the generation of individuals inheriting good parts of their parents.

Mutation Operators: Mutation operators are used to establish the diversity in the population by changing an individual. There are three different mutation operations employed here. The first mutation operation selects two projects and a resource, and then resource dedications of these two selected projects are exchanged. In the second mutation operation one project and two resources are selected, and then the resource dedications of selected project for these two resources are exchanged. The last mutation operation selects two different resource dedications of two projects and exchanges them.

Repair Function: The individuals generated from GA operators can have resource infeasibility caused by corresponding resource dedication values. These infeasibilities (if any exists) are corrected by decreasing resource dedications of the projects according to the general resource capacities. These repair mechanisms require at worst (number of projects x number of resources) or (number of projects) operations for different crossover operators and as many as (number of projects) operations for mutation operators and hence consume a negligible amount of CPU time.

CA for RDP: In GA, CA for RDP is used on randomly selected individuals. Based on preliminary test runs, the probabilities for each crossover and mutation type operators as well as CA for RDP are set to 0.1.

The general execution of GA is given below.

Initialization: Generate initial population as described above

Run: Apply the steps 1-4 until the allowed execution time limit is reached

Step 1 (Crossover): For each crossover operator check crossover probability for each individual in the current population. If crossover operator is issued to be applied, select a mate individual randomly and apply crossover operator. Check resource dedication infeasibility, if any infeasibility exists, restore feasibility.

Step 2 (Mutation): For each mutation operator, check mutation probability for each individual in the current population. If mutation operator is issued to be applied, execute mutation operator on the individual. Check resource dedication infeasibility, if any infeasibility exists, restore feasibility.

Step 3 (Combinatorial Auction for RDP): Check CA for RDP probability for each individual. If CA for RDP is issued to be applied, execute CA on RDP for the individual.

Step 4 (Passing to the Next Generation): Select the best ten individuals as the next generation, delete the remaining individuals.

3.2.2. Lagrangian Relaxation Based Heuristic for the Resource Dedication Problem

The proposed solution approach employs a Lagrangian relaxation of RDP and the subgradient optimization methodology to search for the Lagrangian multipliers. In the following Sections, Lagrangian relaxation formulation and the details of the subgradient optimization approach are given.

3.2.2.1. Lagrangian Relaxation of the RDP. The *Model RDP* has two basic groups of constraints. The first group is related with the scheduling of individual projects, namely the activity assignment constraint sets (Equation 3.2), precedence constraint sets (Equation 3.3) and weighted tardiness calculation (Equation 3.8) for each project. The other group of constraints is related with the resource dedication. Constraint sets (Equation 3.4) and (Equation 3.5) limit the renewable and nonrenewable resource usage with the corresponding dedicated resource limits, respectively. And finally constraint sets (Equation 3.6) and (Equation 3.7) limit the total resource dedication to projects with the available overall resource capacities.

Relaxing constraint sets (Equation 3.4) and (Equation 3.5) leads to the following Lagrangian relaxation designated as LA-RDP:

Mathematical Model LA-RDP

$$min.\ z_{LA-RDP} = \sum_{v=1}^{V} TC_v + \sum_{v=1}^{V}\sum_{k=1}^{K}\sum_{t=1}^{T} \lambda_{vkt} \left\{ \sum_{j=1}^{N_v}\sum_{m=1}^{M_{vj}}\sum_{q=t}^{t+d_{vjm}-1} r_{vjkm} X_{vjmq} - BR_{vk} \right\}$$
$$+ \sum_{v=1}^{V}\sum_{i=1}^{I} \mu_{vi} \left\{ \sum_{j=1}^{N_v}\sum_{m=1}^{M_{vj}}\sum_{t=E_{vj}}^{L_{vj}} w_{vjim} X_{vjmt} - BW_{vi} \right\} \tag{3.26}$$

Subject to

(3.2), (3.3), (3.6), (3.7), (3.8), (3.9), (3.10)

Note that the objective function of *LA-RDP* has two main parts, one in terms of the activity finish time decision variables and the other in terms of the resource dedication decision variables. If this separation is taken as the basis, the constraint sets can also be decomposed. Constraint sets (Equation 3.2), (Equation 3.3) and (Equation 3.8) contain activity finish decision variables; whereas constraint sets (Equation 3.4) and (Equation 3.5) contain renewable and nonrenewable resource dedication decision variables, respectively. Thus, model LA-RDP decomposes into three subproblems SP1, SP2 and SP3 related with activity finish, renewable resource dedication and nonrenewable resource dedication decision variables, respectively.

SP1

$$min.\ z_{SP1} = \sum_{v=1}^{V} TC_v + \sum_{v=1}^{V}\sum_{k=1}^{K}\sum_{t=1}^{T} \lambda_{vkt} \left\{ \sum_{j=1}^{N_v}\sum_{m=1}^{M_{vj}}\sum_{q=t}^{t+d_{vjm}-1} r_{vjkm}X_{vjmq} \right\} + \sum_{v=1}^{V}\sum_{i}^{I} \mu_{vi} \left\{ \sum_{j=1}^{N_v}\sum_{m=1}^{M_{vj}}\sum_{t=E_{vj}}^{L_{vj}} w_{vjim}X_{vjmt} \right\} \tag{3.27}$$

Subject to

(3.2), (3.3), (3.9)

SP1 is basically scheduling the individual projects without any resource constraints and can be further decomposed with respect to individual projects. The resulting schedules will be precedence feasible but can be resource infeasible for each project according to the results of resource subproblems SP2 and SP3.

SP2

$$min.\ z_{SP2} = \sum_{v=1}^{V}\sum_{k=1}^{K}\sum_{t=1}^{T} \lambda_{vkt}(-BR_{vk}) \tag{3.28}$$

Subject to

$$BR_{vk} \leq ur_{vk} \quad \forall\ k \in K \text{ and } \forall\ v \in V \tag{3.29}$$

(3.6), (3.10)

SP3

$$min.\ z_{SP3} = \sum_{v=1}^{V}\sum_{i}^{I} \mu_{vi}(-BW_{vi}) \tag{3.30}$$

Subject to

$$BW_{vi} \leq uw_{vi} \quad \forall\ i \in I \text{ and } \forall\ v \in V \tag{3.31}$$

(3.7), (3.10)

where in SP2 and SP3, ur_{vk} and uw_{vi} are resource usage values for renewable and nonrenewable resources, respectively when CPM is applied to individual project networks. Thus, they are no-delay resource requirements for individual projects.

SP2 and SP3 are for resource dedication decision variables. They are further decomposable in terms of individual renewable and nonrenewable resources, respectively. The additional constraints (Equation 3.29) and (Equation 3.31) limit the dedicated renewable and nonrenewable resources, respectively. Note that if these constraints are not added to SP2 and SP3, all the general resource capacity would be dedicated to the project, which has the largest λ_{vkt} summed over t and similarly to the one with the largest μ_{vi}. Thus, these constraints prevent the abuse of the general resource capacities by the corresponding project. The resulting resource dedications for the projects will be feasible for the general resource limits (r_k, w_i).

The combination of optimal solutions of the subproblems SP1, SP2, and SP3 is a relaxation for RDP, since the additional constraints (Equation 3.29) and (Equation 3.31) limit the resource dedication decision variables with their possible maximum value (no-delay resource requirements). Note that this relaxation also gives a tighter lower bound than the original relaxation of the problem with the additional constraints (Equation 3.29) and (Equation 3.31) to SP2 and SP3, respectively, since these additions prevent the abuse of the renewable and nonrenewable resource dedication decision variables.

3.2.2.2. Subgradient Optimization for RDP. Subgradient optimization approach is used to obtain a solution to RDP. There are three basic steps in an iteration of subgradient optimization: (i) obtaining a lower bound (LB) with a given set of Lagrangian multipliers, (ii) obtaining an upper bound (UB) and (iii) updating the Lagrangian coefficients. By solving the three subproblems, an LB for the RDP can be obtained. SP1 can be solved using exact solution approaches. The resource subproblems are continuous knapsack problems, which are also easy to solve.

An efficient UB is calculated using the results of the Lagrangian relaxation. The solutions of resource subproblems could lead to a resource dedication, which is infeasible for some projects. Thus, resource dedication results cannot be used directly for an UB calculation. In order to overcome this deficiency for UB calculation, resource dedication values for renewable (Equation 3.32) and nonrenewable (Equation 3.33) resources for each project is calculated as follows:

$$BR_{vk}^{UB} = r_k \frac{BR_{vk}}{\sum_{v=1}^{V} BR_{vk}} \tag{3.32}$$

$$BW_{vi}^{UB} = w_i \frac{BW_{vi}}{\sum_{v=1}^{V} BW_{vi}} \tag{3.33}$$

This resource dedication calculation approach normalizes the resource dedication over the resource dedication values of the Lagrangian relaxation problem. After setting BR_{vk}^{UB} and BW_{vi}^{UB} values, RDP reduces to solving individual MRCPSP for each project. The UB calculation for RDP is carried out by solving individual MRCPSP for each project with the given BR_{vk}^{UB} and BW_{vi}^{UB} values and Lagrangian coefficients are calculated accordingly.

3.3. Experimental Results

To test the solution approaches proposed here for RDP a series of test problems are used. Test problems are grouped according to measures proposed by Kurtuluş and Narula (1985); the network complexity (NC) and the maximum utilization factor (MUF). NC is calculated as the total number of arcs divided by total number of nodes in the project network. For the multi-project scheduling problems presented in this study, for a determined network complexity value for the multi-project problem, the NC values of all the projects in the problem are set to this value (i.e. if the multi-project problem has a NC value of 1.4 then all the individual projects in the multi-project problem has a NC value of 1.4).

MUF is the ratio of the resource requirement of no-delay schedule of the project to available resource, i.e., resource capacity. Thus, if MUF is less than or equal to one, the project can be scheduled without any delays. Since the resources cannot be shared, combining the projects and calculating the MUF values are not suitable for the proposed multi-project problem environment. In order to take into account resource dedication concept, the no-delay resource requirement of the multi-project problem is calculated as the sum of no-delay resource requirements of the individual projects. With this approach, if MUF value is less than or equal to one, the multi-project problem can have a no-delay schedule for the dedicated resources case. Similarly when MUF value is increased, resources become tight for the multi-project problem.

Multi-project problems are created with activity-on-node representation combining six different projects from *j20* and *j30* sets in PSPLIB (http://129 .187.106.231/ psplib), developed in Kolisch and Sprecher (1996). Two different levels of NC (1.4 and 1.8) and three different levels of MUF (1.2, 1.4, and 1.5) are selected and a full factorial design with 10 problems in each combination is created. In order to compare different methods, for each combination a total of ten base problems are used. For example, the problems in combination (NC 1.4 - MUF 1.5) and (NC 1.8 - MUF 1.5) have the same network structure, but the projects in the first combination have some of their arcs deleted to achieve an NC value of 1.4. Similarly, (NC 1.8 - MUF 1.4) and (NC 1.8 - MUF 1.5) have exactly the same network structure (same number of arcs, nodes and same modes for activities) but the general resource capacities differ. With this approach, when an optimal solution is found for a problem, it can be used as a lower bound for the other cases with larger NC and/or MUF values. In other words, the weighted tardiness values for the combination (NC 1.4 - MUF 1.2) can be used as a lower bound for the other test cases.

To have a positive weighted tardiness value for projects in the multi-project problem environments, the following approach is used. The due date of the project with the highest weight is set as the makespan calculated for the unconstrained case using CPM (no-delay due date). As the weight decreases, projects are assigned tighter due dates than their no-delay due date. As a result, the minimum possible total weighted tardiness becomes a fixed value for the multi-project problem as shown in Table 3.5. This value can be considered as a lower bound for the multi-project problem. The test problems can be downloaded from the link "http://www.bufaim.boun.edu.tr/flexset.zip".

Table 3.5. Minimum possible total weighted tardiness values for projects.

Project	Due date	Weight	Possible Least Weighted Tardiness
Project1	Nodelay due date	6	0
Project2	Nodelay due date - 1	5	5
Project3	Nodelay due date - 2	4	8
Project4	Nodelay due date - 3	3	9
Project5	Nodelay due date - 4	2	8
Project6	Nodelay due date - 5	1	5
Minimum Possible Total Weighted Tardiness			35

Solution approaches are coded with *Microsoft Visual Studio 2010 C#*. For all the problems that are solved with an exact solution approach, *ILOG CPLEX Callable Library* is used employing *CPLEX 11.2*. For the exact solution approach of RDP, a total of 4096 Mb working memory is allocated and the hard drive is used when this allocation is exceeded. Test runs are carried out on an *Intel Xeon X 5492, 3.40 Ghz* processor.

Results for multi-project problems consisting of projects with 22 and 32 activities are presented in Table 3.6 and Table 3.7, respectively, where GA-LA and GA-LinR refer to GA with CA based on Lagrangian relaxation and CA based on linear relaxation, respectively. SO column refers to the Subgradient Optimization based solution procedure. Exact column is for the exact solutions. The NC-MUF column shows the corresponding network complexity and maximum utilization factors used for the problem groups. As it is mentioned earlier, there are ten problem instances in a problem group. The AWT column reports the average weighted tardiness for a problem group, whereas the values in the ART column show the average execution time of the solution approaches in minutes for a problem group. If a solution approach cannot reach a feasible solution for an instance within the execution time limit, then AWT value is set to NA (not available). All of the solution approaches have an execution time limit of 120 minutes. OS column shows the number of instances for which optimal solution is found in a problem group whereas NS column shows the number of instances where no solution could be found in a problem group within the execution time limit. When the sum of OS and NS is less than ten, it shows that the exact solution approach could only find an incumbent solution (feasible but not proven optimal) in the given execution time limit for the remaining problem instances.

Table 3.6. The results for the problem groups with 22 activities.

	GA-LA				GA-LinR				SO				Exact			
NC-MUF	AWT	ART	OS	NS	AWT	ART	OS	NS	AWT	ART	OS	NS	AWT	ART	OS	NS
1.4-1.2	35	6.19	10	0	35	6.19	10	0	35	4.65	10	0	35	1.39	10	0
1.4-1.4	42.6	94.4	3	0	46.6	100.6	3	0	56.6	111.76	2	0	NA	101.17	4	6
1.4-1.5	52.9	117.29	3	0	66.9	120	1	0	86.7	120	0	0	NA	120	0	10
1.8-1.2	35	3.38	10	0	35	4.42	10	0	35	16.02	10	0	35	1.44	10	0
1.8-1.4	41.4	98.99	5	0	45.6	100.07	4	0	64.6	120	0	0	NA	95.38	4	5
1.8-1.5	50.6	120	2	0	67.9	120	0	0	101.7	120	0	0	NA	112.1	1	9

Table 3.7. The results for the problem groups with 32 activities.

	GA-LA				GA-LinR				SO				Exact			
NC-MUF	AWT	ART	OS	NS	AWT	ART	OS	NS	AWT	ART	OS	NS	AWT	ART	OS	NS
1.4-1.2	35	14.81	10	0	35	18.22	10	0	35	5.98	10	0	35	12.03	10	0
1.4-1.4	56	74.07	5	0	155	103.11	3	0	131.9	120	0	0	NA	90.11	3	5
1.4-1.5	181	110.08	1	0	269	109.24	1	0	314.5	120	0	0	NA	102.23	2	8
1.8-1.2	35	8.73	10	0	35	17.59	10	0	38.6	90.05	2	0	35	12.07	10	0
1.8-1.4	82.2	88.45	5	0	131	112.77	1	0	132	123	0	0	NA	70.26	5	4
1.8-1.5	148	110.15	1	0	247	110.06	1	0	302.9	120	0	0	NA	104.82	2	7

The results for RDP are examined for solution quality and solution times and compared using *paired t-test* with 0.05 level of significance. First of all, when MUF values are closer to one, all solution approaches give optimal values for all or most of the problems in a problem group. But as MUF values increase, exact solution approach begins to fail finding solutions in the given solution runtime limit and falls behind the other approaches. Both GA approaches give results above the given lower bound.

Thus, problem groups with MUF values closer to one can be thought as relatively easy problems whereas problem groups with higher MUF values can be thought of as relatively hard problems, which is in fact an expected result.

SO approach falls behind GA approaches even though it can find feasible solution for all instances in all problem groups. Both of the GA approaches GA-LA and GA-LinR give overall good results for all problem groups. GA-LA is significantly better than the other solution approaches according to the solution quality aspect. Especially when the resources are tight, Lagrangian relaxation based preference calculation is significantly better than linear relaxation based preference calculation. The statistical test results also show that network complexity is not a significant factor for solution quality and only have an additional initialization load for solution time.

The run times are compared for the cases where all solution approaches reach the optimum objective values, because otherwise the procedures are stopped at the run time limit. When these values are compared, no statistically significant difference is observed among the solution approaches. But note that solution times are reasonable for relatively easy problems for all approaches.

Another issue related with the solution quality is the lack of a strong lower bound for the cases. A strong lower bound can help us to measure the solution quality of the proposed approaches more precisely. To obtain a strong lower bound for the problems, we have added an initial solution, which is the best solution obtained by GA-LA procedure, to the *CPLEX* and used the reported lower bound. The Gap column shows the distance of the weighted tardiness value obtained by GA procedures to these lower bounds. WT and WTD columns show the the total weighted tardiness and deviation between GA procedures and *CPLEX* feed by initial solution, respectively. The average row show the average values for the respective columns. We have tested 12 cases from the problem set with different NC and MUF values and obtained the following average gaps for GA-LA and GA-LR in Table 3.8. As it can be seen from the results, GA-LA has a very small weighted tardiness deviation and gap value in average.

Table 3.8. The results for the gaps for GA procedures for selected cases.

	Exact		GA-LA			GA-LR		
Problems	LB	UB	WT	WTD	Gap	WT	WTD	Gap
Problem1	35	35	35	0	0	35	0	0
Problem2	35	35	35	0	0	35	0	0
Problem3	35	38	41	3	0.17	48	10	0.37
Problem4	35	35	35	0	0	35	0	0
Problem5	35	35	35	0	0	35	0	0
Problem6	39	39	41	2	0.05	65	26	0.67
Problem7	35	35	35	0	0	35	0	0
Problem8	35	35	35	0	0	35	0	0
Problem9	35	35	35	0	0	48	13	0.37
Problem10	35	35	35	0	0	35	0	0
Problem11	35	35	35	0	0	45	10	0.29
Problem12	35	35	36	1	0.03	55	20	0.57
Averages	35.33	35.58	36.08	0.5	0.02	42.17	6.58	0.19

Before concluding this Chapter, it should be remarked that the combinatorial auction approach employing the lagrangian relaxation based preference calculation, which is shown to work very well for the RDP problem introduced in this Chapter, will be repeatedly used as an integral module of the solution approaches proposed in the following Chapters as we generalize RDP to RPP .

4. RESOURCE PORTFOLIO PROBLEM UNDER THE RESOURCE DEDICATION POLICY

4.1. Problem Definition and Mathematical Model for Resource Portfolio Problem under the Resource Dedication Policy

In this Chapter, Resource Portfolio Problem under the RD policy will be investigated. This multi-project problem environment includes multiple projects with assigned due dates; with activities that have alternative resource usage modes; a RD policy that does not allow sharing of resources among projects throughout the time periods; and a total budget. There are three issues to face when investigating the aforementioned multi-project problem. First, the total budget should be distributed among different resource types to determine the general resource capacities which correspond to the total amount for each renewable resource to be dedicated to the projects. With the general resource capacities at hand, the next issue to be handled is the resource dedication to individual projects in certain amounts to be employed throughout the planning horizon. With the resources dedicated to the individual projects, the activity scheduling reduces to the multi-mode resource constrained project scheduling problem (MRCPSP) for each individual project. Note that when general resource capacity values are determined, the problem reduces to RDP, which is investigated extensively in Chapter 3. The solution approaches for RPP under the RD policy will employ the insights gained from the solution approaches developed for RDP.

RPP in a multi-project environment is the determination of general resource capacities for a given total resource budget, dedication of a set of resources to a set of projects with assigned due dates according to the determined general resource capacities in such a way that individual project schedules would result in an optimal solution for a predetermined objective. The objective for the problem environment is taken as the minimization of the total weighted tardiness of the projects. The problem under investigation can be thought as an integrated capacity planning and multi-project scheduling problem under the RD policy. The multi-project environment of RPP has a high internal dependency among the projects in the sense of Hans *et al.* (2007) because of the general resource budget. On the other hand, for a given resource dedication within RDP, in the resulting MRCPSP there is no internal dependency among the projects. The mathematical formulation for RPP under the RD policy and the corresponding sets (Table 4.1), parameters (Table 4.2 and Table 4.2), and decision variables (Table 4.4) are given below;

Table 4.1. The sets for RPP under RD policy.

V	Set of projects, $v \in V$
J_v	Set of activities of project v, $j \in J_v$
P_v	Set of all precedence relationships of project v
M_{vj}	Set of modes for activity j of project v, $m \in M_{vj}$
K	Set of renewable resources, $k \in K$
I	Set of nonrenewable resources, $i \in I$
T	Set of time periods of project v

Table 4.2. The parameters for RPP under RD policy.

E_{vj}	Earliest finish time of activity j of project v
L_{vj}	Latest finish time of activity j of project v
d_{vjm}	Duration of activity j operating on mode m
r_{vjkm}	Renewable resource k usage of activity j of project v, operating on mode m
w_{vjim}	Nonrenewable resource i usage of activity j of project v, operating on mode m
dd_v	Assigned due date for project v

Table 4.3. The parameters for RPP under RD policy (cont.).

c_v	Relative weight of project v
cr_k	Unit cost of renewable resource k
cw_i	Unit cost of nonrenewable resource i
tb	Total resource budget

Table 4.4. The decision variables for RPP under RD policy.

X_{vjmt}	$\begin{cases} 1 & \text{if activity } j \text{ of product } v\text{, operating on mode } m \text{ is finished at period } t \\ 0 & \text{otherwise} \end{cases}$
BR_{vk}	Amount of renewable resource k dedicated to project v
BW_{vi}	Amount of nonrenewable resource i dedicated to project v
TC_v	Weighted tardiness cost of project v
R_k	Total amount of required renewable resource k
W_i	Total amount of required nonrenewable resource i

Mathematical Model RPP-RD

$$min.\ z = \sum_{v \in V} TC_v \tag{4.1}$$

Subject to

$$\sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{jv}} X_{vjmt} = 1\ \forall\ j \in N_v \text{ and } \forall\ v \in V \tag{4.2}$$

$$\sum_{m \in M_{vj}} \sum_{t=E_{vb}}^{L_{vb}} (t - d_{vbm}) X_{vbmt} \geq \sum_{m \in M_{vj}} \sum_{t=E_{va}}^{L_{va}} t X_{vamt}$$

$$\forall\ (a,b) \in P \text{ and } \forall\ v \in V \tag{4.3}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{q=t}^{t+d_{vjm}-1} r_{vjkm} X_{vjmq} \leq BR_{vk}\ \forall\ k \in K\ \forall\ t \in T\ \forall\ v \in V \tag{4.4}$$

$$\sum_{j\in N_v}\sum_{m\in M_{vj}}\sum_{t=E_{vj}}^{L_{vj}} w_{vjim}X_{vjmt} \leq BW_{vi}\ \forall\ i\in I \text{ and } \forall\ v\in V \tag{4.5}$$

$$\sum_{v\in V} BR_{vk} \leq R_k\ \forall\ k\in K \tag{4.6}$$

$$\sum_{v\in V} BW_{vi} \leq W_i\ \forall\ i\in I \tag{4.7}$$

$$\sum_{i\in I} cw_iW_i + \sum_{v\in V} cr_kR_k \leq tb \tag{4.8}$$

$$TC_v \geq c_v(\sum_{t=E_{vN}}^{L_{vN}}\sum_{m\in M_{vN}} tX_{vNmt} - dd_v)\ \forall\ v\in V \tag{4.9}$$

$$X_{vjmt}\in\{0,1\}\ \forall\ j\in J,\ \forall\ t\in T,\ \forall\ m\in M \text{ and } \forall\ v\in V \tag{4.10}$$

$$BR_{vk},\ BW_{vi},\ R_k,\ W_i,\ TC_v\in Z^+\ \forall\ v\in V,\ \forall\ k\in K \text{ and } \forall\ i\in I \tag{4.11}$$

Objective function (Equation 4.1) minimizes the total weighted tardiness over all projects. Constraint set (Equation 4.2) ensures that all activities are scheduled once and only once for all projects. Constraint set (Equation 4.3) reflects the precedence relationships among the activities of all projects. Constraint set (Equation 4.4) sets the maximum level for the renewable resource usage over all projects and resource types. Constraint set (Equation 4.5) imposes the maximum level for the nonrenewable resource usage over all projects and resource types. Constraint sets (Equation 4.6) and (Equation 4.7) calculate the required renewable and nonrenewable resource capacities according to the dedicated renewable and nonrenewable resources, respectively. Constraint set (Equation 4.8) ensures that the total cost of the general renewable and nonrenewable resources does not exceed the total budget available. Constraint set (Equation 4.9) calculates the weighted tardiness values for each project. Constraint sets (Equation 4.10)-(Equation 4.11) specify the feasible ranges for the decision variables.

4.2. Solution Methodologies

In chapter 3, efficient solution approaches for RDP are developed including a GA application employing a new local improvement heuristic called CA for RD. The insights gained from those studies constitute the basis for the proposed solution approaches for RPP under RD policy. The RPP formulation includes capacity allocation dimension beyond RDP, namely, determination of the general resource capacities from a total budget, which will be dedicated to individual projects. In other words, in addition to the resource dedication space (RDS), the whole solution space of RPP has another dimension: the general resource capacities. Thus, a search algorithm for RPP should explore resource capacities space (RCS) (different general resource capacities instances) and RDS (different resource dedication instances which are constrained by a general resource capacity instance), and further, for each general resource capacity and corresponding dedication instance, an MRCPSP should be formulated and solved for each project. A *two-phase GA algorithm* is proposed for searching this complex solution space. For comparison purposes, a so-called *monolithic GA* approach is also suggested such that operators defined for the two phases are applied simultaneously in a single phase.

4.2.1. A Two-Phase GA for the Resource Portfolio Problem

Recognizing the hierarchical nature of RPP, a two-phase GA is proposed, where the first phase in GA is the resource dedication (RD) phase and the second one is the resource portfolio (RP) phase. In the first phase of the GA, the RDS is explored using the GA proposed for RDP in Chapter 3. The search algorithms (GA operators and CA for RD) used in this phase operate only on the RD part of the individuals. In the second phase of the GA, individuals are subject to RCS search in addition to the RDS search. For RCS search new GA operators and an improvement heuristic, CA for RP are used. The rationale behind this two-phase approach is to facilitate RCS search with a widely explored RDS.

The details of the individual representation and fitness calculation, initial population generation, execution of the two-phase GA, a summary for RDS search and a detailed discussion of RCS search are given in the following Sub-sections.

4.2.1.1. Individual Representation and Fitness Calculation. The representation of an individual is shown in Table 4.5 for 4 projects (P1, P2, P3, P4) and 3 renewable (R1, R2, R3) and 2 nonrenewable (W1, W2) resources. The general resource capacities and resource dedications are combined into a single chromosome. The RD part of the individual is represented with the values under the project columns. These dedication values are feasible according to the general renewable and nonrenewable resource capacities which are presented under the resources column. On the other hand, the general resource capacities constituting the RC part are feasible for the total resource budget.

Table 4.5. Representation of an individual in the two-phase GA.

	Projects				
Resources	**Project1**	**Project2**	**Project3**	**Project4**	**Resource Capacities**
R1	30	40	25	60	155
R2	40	25	40	20	125
R3	35	50	25	30	160
W1	50	60	30	50	190
W2	15	40	50	40	145

The fitness value for an individual is the total weighted tardiness value for all projects. The weighted tardiness for each project is calculated by solving an MRCPSP for each project when a new resource dedication is selected. Although an individual has general resource capacities and resource dedications feasible for the total budget and general resource capacities, respectively, the individual projects can have infeasible schedules because of the resource dedication values of projects. This infeasibility is reflected by a penalty to the fitness calculation.

4.2.1.2. Initial Population Generation. To generate an individual, first of all, general resource capacities part of the individual must be determined in a way feasible to the general resource budget. After that, according to the general resource capacities, resource dedication values can be generated.

To generate the general resource capacities part of the individuals in the initial population according to the total budget, three different methods are used. The first method generates general resource capacities directly from the total budget proportional to the no-delay resource requirement totals of the projects. One instance is generated in this manner. The second approach starts by satisfying the no-delay resource requirement of a given resource from the total budget and generates the general resource capacities for the remaining resources from the remaining total budget, proportional to the no-delay resource requirement totals of the projects. Instances as many as the number of resources are generated in this manner. And finally, the last approach generates as many general resource capacities randomly according to the given total budget so as to complement the total number of general resource capacities instances generated to a given number Z.

For each general resource capacities instance generated as described above, an RD population is generated using three different methods to obtain the RD parts of the individuals. The first method dedicates the available general resource capacities proportional to the no-delay resource requirements of the projects. One individual is generated in this manner. In the second method, for each project, the no-delay resource requirement is satisfied, and the remaining general resource capacities are dedicated to the remaining projects proportional to the no-delay resource requirements of the corresponding projects. Individuals as many as the number of projects (V) can be generated in this manner. In the last approach, resource dedications are randomly generated for each project so as to end up with H number of RD parts in total for each general resource capacity instance under consideration. Thus, as a result, an initial population of ZxH individuals is created.

In Table 4.6 an example for the generation of an individual in the initial solution is given where general budget is 1000 and all resources have a price equal to one. The individual is obtained by employing the "proportional to no-delay resource requirements" approach for both general resource capacities and RD part.

Table 4.6. Sample individual generation (employing "proportional to no-delay resource requirements" for both the general resource capacities and the RD part).

Resources	No-delay Requirements	Resource Capacities	P1	P2	P3	P4
R1	P1:20 P2:15 P3:30 P4:35	67	14	10	20	23
R2	P1:50 P2:80 P3:30 P4:40	133	33	53	20	27
W1	P1:150 P2:80 P3:130 P4:240	400	100	53	87	160
W2	P1:200 P2:80 P3:100 P4:220	400	133	53	67	147

In Table 4.7 another example for the generation of an individual in the initial solution is given. This time no-delay requirements for R1 is satisfied in RP part of the individual whereas no-delay requirements for P1 is satisfied for RD part.

Table 4.7. Sample initial solution generation (satisfying no-delay requirement of R1 and P1).

Resources	No-delay Requirements	Resource Capacities	P1	P2	P3	P4
R1	P1:20 P2:15 P3:30 P4:35	100	20	15	30	35
R2	P1:50 P2:80 P3:30 P4:40	129	50	42	16	21
W1	P1:150 P2:80 P3:130 P4:240	386	150	42	68	126
W2	P1:200 P2:80 P3:100 P4:220	385	200	37	46	102

4.2.1.3. Resource Dedication Space Search. RDS search is executed on the RD part of the individuals with different crossover and mutation operators in addition to CA for RD and in fact this search space corresponds to RDP, which is extensively investigated in Chapter 3. Thus only a summary for CA for RD and the GA operators used will be given here (see Chapter 3 for a detailed explanation of GA operators and CA for RD procedure).

Genetic Algorithm Operators

There are three mutation operators defined for RDS search. The first mutation operator swaps RD values of two different projects of a randomly selected resource in an individual. The second mutation operator swaps two randomly selected RD values within a randomly selected project in an individual. Finally, in the last mutation operator two RD values are randomly swapped without any resource or project selection.

Three different crossover operators are defined for RD space search. The first crossover operator creates two new individuals from two randomly selected individuals by changing resource dedications of two randomly selected projects. The second crossover operator swaps RD values of selected resources of two different individuals. And finally, the last crossover operator swaps strings of resource dedications without any project or resource selection between two individuals.

CA for RDP is a local improvement heuristic which utilizes the preference concept. The preference of a project for a resource is a metric for the value of a resource for the project according to the current resource state of the project. The preference of a resource can be thought as the value of the resource for the project or a criterion for the amount of improvement that will be seen in the objective if an additional unit of that resource is obtained. The basic rationale behind CA for RDP procedure is moving the resource state of the project to a more preferable state which will result with an improved objective function value (at least as good as the previous one). There are two different methods proposed to obtain the preferences of projects for resources, based on MRCPSP formulation of Talbot (1982): the first one is based on linear relaxation of the model and the other one is based on a Lagrangian relaxation of the model.

As shown in Chapter 3 the Lagrangian relaxation based approach dominates the linear relaxation based approach. Thus Lagrangian relaxation based preference calculation will be used here.

The Lagrangian relaxation based preference calculation employs a modified Lagrangian relaxation formulation of MRCPSP where the renewable and nonrenewable resource constraints are relaxed. The values of the Lagrangian coefficients, after one step sub-gradient optimization, are taken as the preferences of the projects for the resources.

After the calculation of preferences, the remaining issue is moving the current resource dedication to a more preferable state. This is carried out by distributing the slack resources (the unused resources) according to the preferences of project for resources. A continuous knapsack model is used for this task where the preferences are profits, the slack resources are knapsack capacities and the objective is maximizing the profit by distributing the limited capacity to the projects. When the slack resources are distributed among the projects, a more preferable resource state is obtained and the new solution can be calculated by solving an MRCPSP for each project with the new resource dedications.

4.2.1.4. General Resource Capacities Space Search. The RCS search is carried out on the general resource capacities parts of the individuals by employing GA operators defined for RPP and a new improvement heuristic, CA for RP, which are explained below in detail.

Genetic Algorithm Operators

To search general resource capacities space a mutation and a crossover operator are defined. The mutation operation swaps two general resource capacity values for a given individual randomly. In crossover operation two different individuals are selected and two random general resource capacity values are swapped between these individuals. After the application of GA operators, an individual can become infeasible according to the general resource budget and the RD totals. The general resource budget infeasibility is corrected by changing the general resource capacities of the resources that are not swapped in the first place. If the general resource capacities infeasibility cannot be corrected by only decreasing the general resource capacities of the unchanged resources, then the resources affected by the GA operators are also decreased accordingly. Similarly, when there is an RD infeasibility, the RD values are adjusted in proportion to the general resource capacities.

Combinatorial Auction for Resource Portfolio Problem

CA for RP is an improvement heuristic based on preferences for the general resource capacities. The application of CA for RP is similar to CA for RD. With the preferences and slack budget at hand, the slack budget is distributed among general resource capacities according to the preferences. The amount of budget that will be used for different resources are determined using a bounded knapsack model similar to the one used in CA for RD where preferences are profits and slack budget is the knapsack capacity. The key point of the algorithm is the calculation of preferences for the general resource capacities.

To determine the preferences of the general resources, the preferences obtained from the CA for RD can be utilized. The preference of a general resource is calculated as the sum of the preferences of the individual projects for renewable (Equation 4.12) and nonrenewable (Equation 4.13) resources as follows.

$$g_k = \sum_{v=1}^{V} p_{vk} \tag{4.12}$$

$$g_i = \sum_{v=1}^{V} p_{vi} \tag{4.13}$$

where p_{vk} and p_{vi} are preferences of project v for renewable resource k and nonrenewable resource i, respectively.

The slack for general resource capacities can easily be calculated from the solution of individual MRCPSP for each project. The difference between the general budget and the total resource used by all the projects (corrected with resource prices) will give the slack budget. Combining all these information will give the following knapsack model.

$$max\ z. = \sum_{i=1}^{I} y_i g_i + \sum_{k=1}^{K} y_k g_k \tag{4.14}$$

Subject to

$$\sum_{i=1}^{I} cw_i y_i + \sum_{k=1}^{K} cr_k y_k \leq b \tag{4.15}$$

$$y_i \leq a_i \tag{4.16}$$

$$y_k \leq a_k \tag{4.17}$$

$$y_i \text{ and } y_k \in Z^+ \tag{4.18}$$

where y_i (y_k) is the positive decision variable for the amount of spare budget given to nonrenewable resource i (renewable resource k), a_i and a_k are the upper limits for the transferred nonrenewable and renewable resources respectively, calculated from the current general resource capacities and no-delay general resource requirements and b is the slack budget.

With the results of the above knapsack model the general resource capacities are updated such that the unused general resource budget is transferred to the resources with high preference values. The resource dedication values for each project are also updated according to the new general resource capacities with respect to preferences of projects for resources. A new solution is obtained by solving MRCPSP for each project with the new resource dedication values.

4.2.1.5. Execution of the Two-Phase GA. The GA initially starts with *ZxH* individuals generated as described in Section 4.2.1.2. RD sub-populations generated from a general resource capacities instance are subject to only RD space search until convergence. In other words, only RD parts of individuals in the sub-populations are changed. Here, convergence is defined as the stability in the best fitness value of an RD sub-population for a general resource capacities instance. In other words, if no improvement is observed for a specific number of generations for an RD sub-population, this specific RD sub-population is said to have converged. Whenever an RD sub-population converges, the distinct individuals in it are migrated into the RC population for phase two. The rationale behind applying RCS search after the convergence of an RD population is the possibility to obtain better individuals with RCS search operators applied on an evolved RD part. For CA for RP heuristic, it is very important to have "good" preferences of projects for resources since general resource preferences are calculated from those values. With a converged RD part, the preferences of general resource capacities will reflect better the value of a general resource with respect to the needs of the projects. Thus, especially for CA for RP, it is important to have a converged RD part which would have useful and evolved information that heuristic can use. Note that until all RD sub-populations converge, RD sub-populations and RP population run in parallel.

While RD space is searched in the first phase, the evolution in the RP population involves RCS search as well as RDS search. In the second phase, along with GA operators for both RD and RP, CA for RP is employed every time CA for RD is employed.

An elitist strategy is used to select the individuals for the next generation. In other words, best individuals among the current population and the newly generated individuals according to their fitness values are included in the next generation. The individuals for crossover, mutation and CA operators are selected randomly as a diversification strategy to compensate for the intensifying effect of the elitist selection strategy.

The GA parameters used in the test runs are as follows. As defined above, the size of the population is ZxH. In the preliminary runs it is observed that when Z is lower than 8 and H is lower than ten, GA cannot give good results overall and when corresponding parameters are greater than eight and ten, respectively, the algorithm does not improve. Thus the population size is taken as 80. Based on preliminary test runs, the probabilities for each crossover and mutation operator and as well as CA operators are taken as 0.1. GA approaches are terminated when the best solution in the population does not change for ten iterations or within 180 minutes, whichever is reached first.

The execution of the two-phase GA is summarized below.

Initialization: Initialize general resource capacities instances according to the general resource budget and resource dedication instances according to the general resource capacities instances.

Step 1: Generate general resource capacities proportional to no-delay resource requirement totals of the projects.

Step 2: For each resource, determine the capacity equal to the no-delay requirement totals of all projects, generate capacities for remaining resources proportional to no-delay resource requirements of the projects

Step 3: Generate random general resource capacities

Step 4: Apply steps *4.1-4.3* for each general resource capacities instance generated to obtain RD population pools.

Step 4.1: Generate resource dedications proportional to no-delay resource requirements of the projects

Step 4.2: For each project set the project as the selected project, dedicate nodelay resource requirements to the selected project. For the remaining projects generate resource dedications proportional to no-delay resource requirements of the projects

Step 4.3: Generate random resource dedications

Run: Run until allowed execution time of 180 minutes is reached or the best solution in the population does not change for ten iterations

Step 1 Resource Dedication Space Search: Apply the resource dedication space search operators in RD populations and RP population

Step 1.1: Apply crossover operations for resource dedication with corresponding probabilities

Step 1.2: Apply mutation operations for resource dedication with corresponding probabilities

Step 1.3: Apply CA for RD with corresponding probabilities

Step 2 Resource Dedication Space Convergence Check: For each RD population check convergence. If an RD population has converged, move distinct instances to RP population after applying CA for RP to each individual.

Step 3 General Research Capacity Space Search: For each individual in RP population:

Step 3.1: Apply crossover operation for general resource capacities with the corresponding probabilities

Step 3.2: Apply mutation operation for general resource capacities with the corresponding probabilities

Step 3.3: Apply CA for RP, if CA for RD has been applied to the individual with the corresponding probabilities

Report: Report the resulting schedules when algorithm terminates.

4.2.2. A Genetic Algorithm with Simultaneous Resource Dedication and Resource Ccapacity Space Search

Adopting a monolithic view to RPP, a simultaneous execution mode for the proposed GA operators and the CA for RP are introduced. In this execution mode the individuals in the population are evolved applying general resource capacities space search (mutation, crossover and CA for RPP) simultaneously with RDS search. In this approach, CA for RP is applied to individuals, which have completed a CA for RD iteration. But note that in this case, the RDS may not be searched extensively with the GA operators; thus the resulting individuals may not have evolved preferences of projects for resources which in turn may have a negative effect on RCS search.

4.3. Experimental Results

The solution approaches for RPP is tested using a series of test problems. Two different measures, proposed by Kurtuluş and Narula (1985), are used to characterize and group test problems: network complexity (NC) and maximum utilization factor (MUF).

When an activity on node representation is used, NC is defined as the number of arcs divided by total number of nodes. MUF is calculated as the ratio of the no-delay schedule resource requirement and the available resource. If MUF value is less than or equal to one, then the resource capacity is sufficient to obtain a no-delay schedule. In the proposed multi-project environment the resources cannot be shared between projects. Thus, to have a meaningful MUF measure for the multi-project environment, the calculation of no-delay resource requirement of the multi-project problem is calculated as the sum of no-delay resource requirements of the individual projects. This approach reflects the RD policy to the MUF calculation. In addition to this, in the proposed multi-project scheduling environment the resource capacities are not readily present but are determined from a given resource budget. Thus, MUF calculation is modified by calculating the budget value of no-delay resource requirement divided by the general resource budget. If MUF value is equal to one, then the general resource budget is enough to construct a general resource capacities instance that will allow a no-delay schedule for all projects. Similarly when MUF value is increased, then the general resource budget becomes tight.

Multi-project problems are generated by combining 6 projects either from *j20* or *j30* sets of PSPLIB (http://129 .187.106.231/psplib/), developed by Kolisch and Sprecher (1996). The modes of the activities of the projects are modified to obtain a mode set that will allow to investigate the behavior of the algorithm for resources with different prices and contribution to the process of the activities but the general network structure is kept as it is. The problem sets have four resources, two being renewable and two nonrenewable. From this resource set, one of each resource type is designated to lead to faster processing times but to have a higher cost. In addition to this, since renewable resources have an important impact on decisions in multi-project scheduling environments, the prices of renewable resources are set slightly higher than nonrenewable resources in the problem set.

Four different modes are generated with different durations and resource usages as follows. The fastest mode has the highest cost and it has resource consumption from all of the resources. The second fastest mode has resource consumption of only the costly resources with a major amount. The third fastest mode uses only the costly modes in a moderate amount. And finally, the slowest mode uses only the cheap resources. The mode generation is depicted Table 4.8. The values are not the exact coefficients for resource usages or durations but only show the general process. The durations and resource usages are corrected according to total resource costs of modes but have random components e and f with a relatively small magnitude. Bold letters indicate the relatively costly resources.

The test problems are grouped according to different levels of number of activities, NC and MUF values. Two levels of number of activities (22 and 32) and NC (1.4 and 1.8) and four different levels of MUF (1.4, 1.5, 1.6 and 1.7) are defined and a full

Table 4.8. General resource usage and duration characteristics of the modes.

Modes	**R1**	R2	**N1**	N2	Duration
M1	2a	b	2c	d	x
M2	2a±e	0	2c±e	0	x+f
M3	a±e	0	c±e	0	2x-f
M4	0	b±e	0	c±e	3x

factorial design with ten problems in each combination is generated. In test runs it has been observed when MUF values are lower than 1.4 the problem approaches the unconstrained case which is of no interest here. MUF values higher than 1.7 lead to infeasibility for most of the cases for exact solution approaches. To be able to compare different solution approaches for different problem characteristics, a base problem group with ten problems is generated with NC value of 1.8 and MUF value of one for multi-project problems with 22 and 32 activities. From this base set, problems with different NC values are generated by randomly deleting precedence relations between different activities. Similarly problems with different MUF values are generated by adjusting the general resource budget accordingly. For example, the problem sets with activity count 22-NC 1.4-MUF 1.5 and activity count 22-NC 1.8-MUF 1.5 have the same mode structure and general resource budget but the previous combination has some of its precedence relations deleted to achieve an NC value of 1.4.

The due dates for projects are calculated as in Chapter 3 to achieve a positive weighted tardiness value. The project with the highest weight has its due date as the calculated makespan of the unconstrained case using CPM which is named as no-delay due date. The due dates become less than the no-delay due date as the weight of the projects decrease. The total weighted tardiness value calculated with no-delay due dates and assigned due dates gives a lower bound for the problem.

Results are presented in Tables 4.9 and 4.10 for projects with 22 and 32 activities respectively, where Two-Phase GA group refers to GA that employs a two-phase search for RD and RC spaces and Monolithic GA group refers to the monolithic approach. Exact group is for the exact solution approach for the given mathematical formulation employing CPLEX 11.2. Every row in the tables represents a problem group and is identified with the corresponding entry in the NC-MUF column which shows the corresponding network complexity and maximum utilization factors used. There are ten problems instances in a problem group. The following notation is employed in Tables 4.9 and 4.10. AWT stands for the average weighted tardiness for a problem group. AWTD is the average deviation for the weighted tardiness values for a problem group from the corresponding AWT values obtained by the Two-Phase GA approach for the cases where exact solution approach can find a feasible (or optimal) solution. NA indicates the case where no feasible solution can be reached when employing the exact solution approach. ART column reports the average execution time of the solution approaches in minutes for a problem group. All of the solution approaches have a run time limit of 180 minutes, in addition to this the GA approaches terminates if the best solution is not improved for ten iterations. OS column shows the number of instances that the optimal solution is found in a problem group whereas NS column shows the number of instances that no solution is found in a problem group. For the exact solution approach, if $OS + NS \neq 10$, then the difference is the number of incumbent solutions (a feasible solution that is not proven to be optimal).

Table 4.9. The results for problem groups with 22 activities.

	Two-Phase GA				Monolithic GA				Exact			
NC-MUF	AWT	ART	OS	NS	AWTD	ART	OS	NS	AWTD	ART	OS	NS
1.4-1.4	36.00	36.11	8	0	6.60	49.18	7	0	0.00	3.31	8	0
1.4-1.5	47.70	84.58	8	0	13.90	96.14	2	0	1.11	108.48	8	1
1.4-1.6	67.00	127.70	1	0	31.80	157.30	0	0	22.75	226.50	2	6
1.4-1.7	104.00	135.50	1	0	34.50	162.40	0	0	5.00	240.00	1	8
1.8-1.4	39.40	32.31	9	0	5.10	38.51	7	0	0.00	21.12	9	1
1.8-1.5	51.80	76.96	7	0	19.60	91.39	0	0	0.00	105.55	7	3
1.8-1.6	72.90	144.90	1	0	32.60	171.40	0	0	19.30	240.00	1	7
1.8-1.7	99.70	144.40	1	0	31.10	167.60	0	0	10.00	240.00	0	9

Table 4.10. The results for problem groups with 32 activities.

	Two-Phase GA				Monolithic GA				Exact			
NC-MUF	AWT	ART	OS	NS	AWTD	ART	OS	NS	AWTD	ART	OS	NS
1.4-1.4	35.00	18.56	10	0	0.00	19.55	10	0	0.00	14.80	10	0
1.4-1.5	35.60	98.42	7	0	17.70	111.80	0	0	3.83	114.73	4	4
1.4-1.6	62.40	144.80	0	0	30.60	170.60	0	0	NA	240.00	0	10
1.4-1.7	73.80	133.80	0	0	32.40	163.80	0	0	NA	240.00	0	10
1.8-1.4	35.00	49.65	10	0	1.90	47.33	8	0	0.00	62.13	10	0
1.8-1.5	39.20	73.17	7	0	9.00	85.69	4	0	3.83	118.09	4	4
1.8-1.6	63.30	135.70	0	0	37.00	173.30	0	0	NA	240.00	0	10
1.8-1.7	82.60	146.90	0	0	32.10	166.00	0	0	NA	240.00	0	10

The results can be examined according to the solution quality and solution time and compared using *paired t-test*. It is intuitive and obvious that MUF is the most significant factor for the difficulty of the problem instances. For the relatively easy problems (with MUF values 1.4), all of the solution approaches can find the optimal solution in a reasonable time. With moderate MUF values (1.5), the monolithic GA with simultaneous RD and RP search and the exact solution approach start to fall back compared to the two-phase GA. For the hardest problems of the set (with MUF values 1.6 and 1.7), the gap between the two-phase GA and the other solution approaches widens. Note that the exact solution approach cannot find results in the given time limit for a significant number of problems for projects with 22 activities and for any of the problem instances for projects with 32 activities. To summarize the results for solution quality, one can say that two-phase GA gives significantly better results than the monolithic GA, which shows the benefit of exploring the problem space in different phases. Furthermore, for relatively easy problems, the two-phase GA gives competitive results compared with the exact solution approach. For problem instances with higher MUF values, the two-phase GA has a clear solution quality advantage over the exact solution approach, which even fails to find feasible solutions for most of the cases.

When the results are compared according to the solution times, it can be seen that the exact solution approach is significantly better than the GA approaches when problems are relatively easy (with MUF values 1.4). This can be explained by the heavy initialization cost for the GA approaches, which does not pay off for the easy cases. For MUF values 1.5, 1.6 and 1.7, the two-phase GA is significantly better than the remaining approaches with respect to solution times. This shows that the two-phase GA approach explores the solution spaces more effectively than the monolithic GA with simultaneous space search. In other words, searching RCS after the RD space search has converged, improves the execution time of GA.

As we have mentioned in Chapter 3, a strong lower bound for the test cases can help us to judge the quality of the solution approaches. To obtain a strong lower bound for the problems, similarly, we have added an initial solution, which is the best solution obtained by Two-Phase GA procedure, to the *CPLEX* and used the reported lower bound. Average Gap column shows the average distance of the weighted tardiness values obtained by GA procedures to these lower bounds. WT and WTD columns show the the total weighted tardiness and deviation between GA procedures and *CPLEX* feed by initial solution, respectively. The average row show the average values for the respective columns. We have tested 12 cases from the problem set with different NC and MUF values and obtained the following average gaps for Two-Phase and Monolithic GA in Table 4.11.

Table 4.11. The results for the gaps for GA procedures for selected cases.

	Exact		Two-Phase			Monolithic		
Problem	LB	UB	WT	WTD	Gap	WT	WTD	Gap
Problem1	35	35	41	6	0.17	48	13	0.37
Problem2	44	44	53	9	0.20	82	38	0.86
Problem3	67	67	69	2	0.03	105	38	0.57
Problem4	39	39	41	2	0.05	65	26	0.67
Problem5	46	51	69	18	0.5	104	53	1.26
Problem6	67	71	82	11	0.22	100	29	0.49
Problem7	35	35	35	0	0	48	13	0.37
Problem8	49	52	82	30	0.67	118	66	1.41
Problem9	70	70	76	6	0.09	112	42	0.60
Problem10	35	35	36	1	0.03	55	20	0.57
Problem11	49	57	77	20	0.57	99	42	1.02
Problem12	70	70	77	7	0.1	123	53	0.76
Averages	50.50	52.17	61.50	9.33	0.22	88.25	36.08	0.75

As it can be seen from the results, the reported average weighted tardiness deviation and average gap value are tolerable for Two-Phase GA.

5. RESOURCE PORTFOLIO PROBLEM UNDER THE RELAXED RESOURCE DEDICATION POLICY

5.1. Problem Definition and Mathematical Model for Resource Portfolio Problem under the Resource Dedication Policy

Resource Dedication policy addresses an important aspect in multi-project environments and together with RS policy define two extreme resource management policies in multi-project scheduling problem environments. Even though RD and RS policies cover an important part of the multi-project environments, there can be different extensions to aforementioned resource management policies because of the different characteristics of the multi-project environment. In this Chapter we will investigate an extension for the RD policy where transferring renewable resources of a project is allowed after the project is completed but only to the projects that start after the completion time of the transferring project. This policy is called Relaxed Resource Dedication (RRD) policy. This concept is enabled with possible sequence relations among projects with start time decision variables.

RPP in a multi-project environment is the determination of general resource capacities for a given total resource budget, dedication of a set of resources to a set of projects with assigned due dates according to the determined general resource capacities, and determining the project sequence relations and resource transfers between feasible projects (according to the project sequence relations) in such a way that individual project schedules would result in an optimal solution for a predetermined regular objective function. The objective for the problem environment is taken here as the minimization of the total weighted tardiness of the projects. Mathematical programming formulation for the RPP under the RRD and the corresponding sets (Table 5.1), parameters (Table 5.2) and decision variables (Table 5.3) are given below.

Table 5.1. The sets for RPP under RRD policy.

V	Set of projects, $v \in V$
J_v	Set of activities of project v, $j \in J_v$
P_v	Set of all precedence relationships of project v
M_{vj}	Set of modes for activity j of project v, $m \in M_{vj}$
K	Set of renewable resources, $k \in K$
I	Set of nonrenewable resources, $i \in I$
T_v	Set of time periods

Table 5.2. The parameters for RPP under RRD policy.

E_{vj}	Earliest finish time of activity j of project v
L_{vj}	Latest finish time of activity j of project v
d_{vjm}	Duration of activity j, operating on mode m
r_{vjkm}	Renewable resource k usage of activity j of project v, operating on mode m
w_{vjim}	Nonrenewable resource i usage of activity j of project v, operating on mode m
dd_v	Assigned due date for project v
c_v	Relative weight of project v
cr_k	Unit cost of renewable resource k
cw_i	Unit cost of nonrenewable resource i
tb	Total resource budget
Ω	A big number calculated as the sum of time periods of all projects

Table 5.3. The decision variables for RPP under RRD policy.

X_{vjmt}	$\begin{cases} 1 & \text{if activity } j \text{ of project } v\text{, operating on mode } m \text{ is finished at period } t \\ 0 & \text{otherwise} \end{cases}$
BR_{vk}	Amount of renewable resource k dedicated to project v
BW_{vi}	Amount of nonrenewable resource i dedicated to project v
TC_v	Weighted tardiness cost of project v
R_k	Total amount of required renewable resource k
W_i	Total amount of required nonrenewable resource i
F_v	Release time of project v
$SR_{vv'k}$	Amount of renewable resource k transferred to project v' from project v
$Y_{vv'}$	$\begin{cases} 1 & \text{if project } v' \text{ is released after project } v \text{ is finished} \\ 0 & \text{otherwise} \end{cases}$

Mathematical Model RPP-RRD

$$min.\ z = \sum_{v \in V} TC_v \tag{5.1}$$

Subject to

$$\sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{jv}} X_{vjmt} = 1 \quad \forall\ j \in N_v \text{ and } \forall\ v \in V \tag{5.2}$$

$$\sum_{m \in M_{vj}} \sum_{t=E_{vb}}^{L_{vb}} (t - d_{vbm}) X_{vbmt} \geq \sum_{m \in M_{vj}} \sum_{t=E_{va}}^{L_{va}} t X_{vamt}$$

$$\forall\ (a, b) \in P \text{ and } \forall\ v \in V \tag{5.3}$$

$$F_{v'} - F_v - \sum_{t=E_{vN}}^{L_{vN}} \sum_{m \in M_{vN}} t X_{vNmt} \leq \Omega(Y_{vv'}) \quad \forall\ v, v' \in V \tag{5.4}$$

$$F_v + \sum_{t=E_{vN}}^{L_{vN}} \sum_{m \in M_{vN}} t X_{vNmt} - F_{v'} \leq \Omega(1 - Y_{vv'}) \quad \forall\ v, v' \in V \tag{5.5}$$

$$SR_{vv'k} \leq \Omega(Y_{vv'}) \quad \forall\ v, v' \in V \text{ and } \forall\ k \in K \tag{5.6}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{q=t}^{t+d_{vjm}-1} r_{vjkm} X_{vjmq} \leq BR_{vk} + \sum_{v' \in V} SR_{v'vk}$$

$$\forall\ k \in K\ \forall\ t \in T\ \forall\ v \in V \tag{5.7}$$

$$\sum_{j \in N_v} \sum_{m \in M_{vj}} \sum_{t=E_{vj}}^{L_{vj}} w_{vjim} X_{vjmt} \leq BW_{vi} \qquad \forall\ i \in I \text{ and } \forall\ v \in V \tag{5.8}$$

$$BR_{vk} + \sum_{v' \in V} SR_{v'vk} \geq \sum_{v' \in V} SR_{vv'k} \quad \forall\ k \in K \text{ and } \forall\ v \in V \tag{5.9}$$

$$\sum_{v \in V} BR_{vk} \leq R_k \quad \forall\ k \in K \tag{5.10}$$

$$\sum_{v \in V} BW_{vi} \leq W_i \quad \forall\ i \in I \tag{5.11}$$

$$\sum_{i \in I} cw_i W_i + \sum_{k \in K} cr_k R_k \leq tb \tag{5.12}$$

$$TC_v \geq C_v(F_v + \sum_{t=E_{vN}}^{L_{vN}} \sum_{m \in M_{vN}} tX_{vNmt} - dd_v) \quad \forall\ v \in V \tag{5.13}$$

$$X_{vjmt} \in \{0,1\} \quad \forall\ j \in J, \forall\ t \in T, \forall\ m \in M_{vj} \text{ and } \forall\ v \in V \tag{5.14}$$

$$BR_{vk},\ BW_{vi},\ R_k,\ W_i,\ TC_v \in Z^+ \quad \forall\ v \in V, \forall\ k \in K \text{ and } \forall\ i \in I \tag{5.15}$$

$$Y_{vv'} \in \{0,1\},\ F_v \text{ and } SR_{vv'k} \in Z^+ \quad \forall\ v, v' \in V \text{ and } \forall\ k \in K \tag{5.16}$$

The objective function (Equation 5.1) is the minimization of the total weighted tardiness cost for all projects. Constraint sets (Equation 5.2) and (Equation 5.3) impose activity finish and precedence relations. Constraint sets (Equation 5.4) and (Equation 5.5) set the decision variable $Y_{vv'}$ to one, if project v is finished before project v' is released, and to 0 otherwise. Thus, the $SR_{vv'k}$ values will only have positive values, if project v is finished before project v' is released (constraint set (Equation 5.6)). Constraint set (Equation 5.7) limits the renewable resources employed for each project with the dedicated renewable resources and the transferred renewable resources from the other projects. Constraint set (Equation 5.8) calculates the nonrenewable resource dedication values available for each project. In constraint set (Equation 5.9), the total resource that can be transferred by a project is limited with the total resource dedicated to this project and the total resource it gained from transfers. Constraint sets (Equation 5.10) and (Equation 5.11) calculate the total renewable and nonrenewable resource requirements, respectively. Constraint (Equation 5.12) limits the sum of the total renewable and nonrenewable resource costs with the general resource budget. And finally, constraint set (Equation 5.13) calculates the weighted tardiness value for each project. Constraint sets (Equation 5.14)-(Equation 5.16) define the feasible ranges for the decision variables.

Note that in the above formulation every project has an independent time index. The following constraints enables an independent time index for each project. First of all, the coupled resource constraints for projects (constraint sets (Equation 5.4) and (Equation 5.5)) do not include a time period index. In addition to this, to calculate start and finish times of projects while including project sequence relations, constraint sets (Equation 5.10) and (Equation 5.11) are used. In this constraint, if there is a sequence relation between two projects, then the successor projects start time decision variable ($F_{v'}$) is given a lower bound of finish time of the predecessor project (the finish time of the predecessor project is also corrected with the corresponding start time decision variable, F_v). Note that the separation in time indices for each project decreases the number of activity finish decision variables (X_{vjmt}) since every project will have its individual upper bound for time periods.

5.2. Solution Methodologies

5.2.1. A Modified B&C Approach for RPP under the RRD

The given mathematical model for RPP under the RRD policy is a complex scheduling problem. It is even very difficult to reflect all the aspects of the problem to a heuristic approach. Thus to solve RPP-RRD, B&C procedure of *ILOG CPLEX* version 11.2, which will be referred as *CPLEX* from now on, is modified with different branching strategies, feasible solution approaches and valid inequalities. First of all, the general procedure of *CPLEX* will be briefly discussed and then modifications will be explained in detail.

5.2.1.1. General Branch and Cut Procedure of ILOG CPLEX. *CPLEX* employs the well known B&C procedure to solve integer problems. The general B&C procedure starts with preprocessing of the given problem and proceeds with the given steps below as given in ILOG (2009):

(i) A node from the tree is selected.
(ii) The linear programming (LP) relaxation of the problem at the selected node is solved and a lower bound is generated.
(iii) Cutting planes are generated to cut off the current solution.
(iv) An heuristic procedure is invoked to generate an integer feasible solution close to the current relaxation solution and an upper bound is generated.
(v) A branching variable is chosen
(vi) Two nodes are generated and included to the tree.

All these steps can be modified by the user by employing special purpose algorithms that are developed for the problem at hand. The control of the above procedures is enabled with callback functions in the *CPLEX*. At every node, when a feasible LP relaxation solution is found for the problem, the associated callback functions for each of the above procedures can be used to control the details of the B&C procedure. Three basic steps will be modified to control the details of the B&C procedure in this study; branching, feasible solution generation and local cut generation. Below explanations of these modifications are given in detail.

5.2.1.2. Branching Strategy for the RPP under the RRD Policy. The selection of branching variables can greatly improve the execution time of the B&C procedure. Model RPP-RRD has two sets of binary and seven sets of integer decision variables which can be used to generate branches. In *CPLEX*, the branching can be modified by giving priorities to decision variables or directly selecting a decision variable for branching on a viable node. In this study, we select the branching variable explicitly or leave it to *CPLEX*.

The proposed branching strategy is based on the structure of the mathematical model RPP-RRD. Note that when $Y_{vv'}$ variables are known, the problem reduces to RPP with available resource transfer options according to the values of $Y_{vv'}$ decision variables. In other words, when $Y_{vv'}$ variables are set, the remaining decisions are general resource capacities, resource dedications, resource transfers between feasible projects (for projects finish before the start of another one) and finally scheduling of individual projects. This structure of the problem not only facilitates the search but also allows for different feasible solution generation approaches. To branch on $Y_{vv'}$ decision variables, the branch callback function of *CPLEX* is used and branches are generated from integer infeasible variables from the linear relaxation solution on the node explicitly and fed into B&C procedure of *CPLEX*.

The selection rule for $Y_{vv'}$ variable among the integer infeasible ones is given below:

Initialization: Select all the projects that have integer infeasible $Y_{vv'}$ decision variables in a node and add selected projects to set *InP*.

Step 1: Select the projects from *InP* that do not have any sequence relations (i.e., all $Y_{vv'}$ and $Y_{v'v}$ variables are integer infeasible for project v) and add selected projects to set *SP*. If set *SP* is empty, then go to *step 2*; else go to *step 4*.

Step-2: Select the projects from *InP* that are the predecessor projects in their sequence relations (i.e. all $Y_{v'v}$ variables are integer feasible and 0 for project v) and add selected projects to set *SP*. If set *SP* is empty, then go to *step 3*; else go to *step 4*.

Step-3: Select the projects from *InP* that have integer infeasible $Y_{vv'}$ decision variables and add selected projects to set *SP*, go to *step-4*.

Step-4: Select the project v from *SP* with maximum weight and select the project v' with minimum weight among the projects that have integer infeasible $Y_{vv'}$

values. Generate two branches with the selected $Y_{vv'}$ decision variable ($Y_{vv'} \leq 0$ and $Y_{vv'} \geq 1$).

This selection procedure gives the priority to projects without any sequence relations, and then to the projects that are predecessor in their sequence relations. For tie breaking projects weights are used. When all $Y_{vv'}$ decision variables are integer feasible in a node, then BR_{vk} decision variables are selected as the next variables for branching.

Another selection procedure based on ratio of due date and weight of the projects. This ratio prioritize projects with earlier due dates and higher weight values. In the preliminary test runs it has been observed that there is not a significant difference between two approaches.

The selection rule for BR_{vk} variable among the integer infeasible ones is given below:

Initialization: Select all the projects that have integer infeasible BR_{vk} decision variables in a node and add selected projects to set *InP*.

Step-1: Select the projects from *InP* that do not have any predecessors (i.e., all $Y_{v'v}$ variables are 0 for project v) and add selected projects to set *SP*. If set *SP* is empty then go to *step 2* else go to *step-4*.

Step:2 Select the projects from *InP* that have at least one successor (i.e., at least one $Y_{vv'}$ variable is equal to one for project v) and add selected projects to set *SP*. If set *SP* is empty, then go to *step 3* else go to *step-4*.

Step-3: Select the projects from *InP* that have integer infeasible BR_{vk} decision variables and add selected projects to set *SP*, go to *step-4*.

Step:4 Select the project v from *SP* with maximum weight and select correspond-

ing renewable resource k with integer infeasible BR_{vk} decision variable and maximum resource cost. Generate two branches with the selected BR_{vk} decision variable.($BR_{vk} \leq \lfloor LinearRelaxationSolutionValue \rfloor$ and
$BR_{vk} \geq \lceil LinearRelaxationSolutionValue \rceil$).

This selection procedure similarly favors predecessor projects in their sequence relations. Note that, when all BR_{vk} variables are integer feasible the branching procedure is left to *CPLEX*.

5.2.1.3. Feasible Solution Generation Procedure. At each viable node, *CPLEX* attempts to generate a feasible solution close to the solution of the LP relaxation. If good (in some cases any) feasible solutions can be generated, *CPLEX* can use these solutions to facilitate the execution of the B&C algorithm. The proposed branching strategies prioritize branching on $Y_{vv'}$ variables which results in integer feasible values for these variables in the early stages of B&C procedure. This structure can be used to generate feasible solutions, since when $Y_{vv'}$ variables are known, the remaining problem reduces to RPP with possible resource transfers according to the values of $Y_{vv'}$ decision variables. The solution approaches developed for RDP in Chapter 3 and RPP in Chapter 4 can be modified and used to generate feasible solutions at these stages of the B&C procedure. In Chapter 3, an improvement heuristic based on preferences of projects for resources, CA for RD, is proposed which moves the current resource dedication state of the problem to a more "preferable" resource state within the given general resource capacities, such that the objective function would be improved (the objective function value will at least remain the same). Similarly, in Chapter 4, an improvement heuristic based on general resource preferences, CA for RP, is proposed which moves the current general resource capacity state to a more "preferable" resource state within the given resource budget.

The proposed feasible solution generation procedure basically determines the general resource capacities, resource dedication values and resource transfers (according to the values of $Y_{vv'}$ variables) and generates an initial solution (not necessarily feasible). And then, this initial solution is improved with CA for RD and CA for RP improvement heuristics. The procedure for feasible solution generation is given below:

Initial Solution Generation

Step-1: Calculate no-delay renewable and nonrenewable resource requirements of all projects and determine total required budget for these values. Dedicate nonrenewable resource values proportional to the no-delay requirements of the projects and total required budget as follows:

$$BW'_{vi} = \frac{NoDelayRequirement_{vi} \times GeneralBudget}{TotalRequiredBudget} \tag{5.17}$$

Calculate remaining resource budget after nonrenewable resource dedication.

Step-2: Select all projects that does not have any predecessor projects in their sequence relations, such that for project v, for all v', $Y_{v'v}$ values are 0. Add these projects to *NP* set. Add projects v' to set SP_v for each project v in *NP*, such that for project v', $Y_{vv'} = 1$.

Step-3: Calculate no-delay renewable resource requirements of projects in *NP* set. Check if these resource amounts are sufficient for no-delay resource requirements of successor projects of v in set SP_v, if not, calculate necessary amounts as additional renewable resources. Determine desired general renewable resource capacities with these calculated values and then determine total required budget for these desired values. Calculate realized general renewable resource capacities from remaining budget, proportional to desired values and remaining resource budget as follows:

$$R'_k = \frac{DesiredResourceCapacity_k \times RemainingBudget}{TotalRequiredBudget} \tag{5.18}$$

Step-4: Calculate total no-delay resource requirement of a project as the sum of no-delay resource requirement of corresponding project and additions required for successor projects. Calculate renewable resource dedication values for projects in *NP* proportional to their total no-delay resource requirements using general renewable resource capacities calculated in *step-3* as follows:

$$BR'_{vk} = \frac{TotalNoDelayRequirement_{vk} \times R'_k}{TotalNoDelayRequirement_k} \tag{5.19}$$

Calculate the renewable resource transfer values for projects in *SP* with the following mathematical model:

$$min.z = \sum_{v}^{V} \sum_{k}^{K} (ND_{vk} - \sum_{v'}^{V} SR_{v'vk}) \tag{5.20}$$

Subject to

$$SR_{vv'k} \leq BR'_{vk} Y_{vv'} \quad \forall\, v, v' \in V \text{ and } \forall\, k \in K \tag{5.21}$$

$$\sum_{v'}^{V} SR_{vv'k} \leq BR'vk \quad v \in V \text{ and } \forall\, k \in K \tag{5.22}$$

$$\sum_{v'}^{V} SR_{v'vk} \leq ND_{vk} \quad v \in V \text{ and } \forall\, k \in K \tag{5.23}$$

where BR'_{vk} is the renewable resource dedication values for projects in *NP* set calculated in previous *step-4* and ND_{vk} is the no-delay renewable resource requirement for projects in *SP* set. This mathematical model tries to determine resource transfers among available projects such that the deviation between total resource transfers to a project and its no-delay resource requirement is minimized over all projects.

Step-5: With renewable and nonrenewable resource dedication values, resource transfer values and precedence relations of projects, at hand, problem reduces to MRCPSP for each individual project. Solve single project scheduling problems for each project and obtain an initial solution by calculating start times and finish times of the projects considering sequence relations and single project scheduling results.

Note that the special structure of the mathematical model RPP-RRD allows the calculation in *step-5*. The individual projects have their time periods independent from the other projects in the multi-project environment. Thus when their individual schedules and sequence relations are given the resulting schedule for multi-project problem can be easily calculated.

Solution Improvement with CA for RP

Step-1: Apply CA for RD to the initial solution till no improvement is seen.

Step 1.1 Determine preferences of projects for resources for each project

Step 1.2 Determine total slack resources for each project (total difference between allocated resources and given resource capacities for each project)

Step 1.3 Employ a continuous knapsack model to maximize the total preference value gained by dedicating the spare resources to projects.

Step 1.4 Calculate schedules for each project with the given resource state.

Step-2: Apply CA for RP to the initial solution till no improvement is seen.

Step 2.1 Calculate preferences for general resources summing up preferences of projects for resources over all projects.

Step 2.2 Determine the slack resource budget for each resource (total difference between used budget and total budget)

Step 2.3 Employ a continuous knapsack model to maximize the total preference values gained by transferring slack budget to general resource capacities.

Step 2.4 Apply CA for RD to the new general resource capacity state

The preferences of projects for resources are calculated with the Lagrangian relaxation of the project scheduling formulation of Talbot (1982). The MRCPSP formulation of Talbot (1982) is modified with a constraint which limits the completion time of the project with its no-delay completion time. The basic rationale behind this approach is to capture the infeasibility tolerated for the corresponding resource to reach the no-delay completion time with corresponding Lagrangian coefficients. The values of Lagrangian coefficients are taken as preferences of projects for resources. Then using these preferences, the resource dedication state of the solution is moved to a more preferable state and a new schedule for each project is calculated. Detailed information related with CA for RD can be found in Chapter 3.

Similarly CA for RP uses preferences of projects for resources to calculate general resource preferences simply summing up project preferences for each resource. Then the general resource capacity state is moved to a more preferable state and CA for RD is applied for this new solution to modify resource dedication values accordingly and to obtain a new solution as described in Chapter 4.

5.2.1.4. Valid Inequality Generation. Valid inequalities can improve the B&C procedure and can have an important impact on the efficiency of the procedure. *CPLEX* generates various valid inequalities based on polyhedral theory. One additional valid inequality, namely sequence valid inequality, will be generated at every viable node whenever a $Y_{vv'}$ decision variable is set to one:

$$F_{v'} \geq F_v + EP_v Y_{vv'} \qquad \forall (v, v') \in V : Y_{vv'} = 1 \tag{5.24}$$

where EP_v is the possible earliest completion time of project v calculated from CPM.

Note that these valid inequalities can be applied a priori in the problem formulation with the following constraint set:

$$F_{v'} \geq F_v + PE_v + \Omega(y_{vv'} - 1) \qquad \forall\, v, v' \in V \tag{5.25}$$

5.3. Experimental Results

To test the efficiency of the proposed modification for *CPLEX* two different problem sets are used. The first problem set is a control set, which is used to roughly identify the effects of different modifications and rapidly observe the project sequence relations among the projects. The second problem set is the instances used in Chapter 4.

The problems in the first set are taken from the PSPLIB (http://129 .187.106.231/ psplib/), developed in Kolisch and Sprecher (1996). Two projects from each *j16*, *j20* and *j30* sets are taken and one of the projects in each problem set have a weight value significantly higher than the other one. This modification enabled a rapid observation of project sequence relations when combined with a tight resource budget. The modes of the activities are modified such that the total resource cost of modes and their corresponding durations are proportional with diminishing returns for mode durations. The due dates of projects are set to their calculated no-delay due dates. A total number of 20 problems are generated in this way. The results are given in Table 5.4 below. The first row shows the best combination of the proposed modifications according to absolute gap and objective value of the best feasible solution which turns out to be $Y_{vv'}$ *Branching - Heuristic Solution - Sequence Valid Inequality.* AWT and AG columns show the average weighted tardiness and average gap respectively. Note that for both of the columns, only the common problems that have given feasible solution are used. Finally ART and NS columns show the average run time for the procedure and the number of problems that no feasible solutions can be found with the given combination, respectively.

Table 5.4. The results for different combinations of the proposed modifications.

Combination	**AWT**	**AG**	**ART**	**NS**
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**33.00**	**1.83**	**240**	**0**
Unmodified	39.50	3.81	240	18
$Y_{vv'}$ Branching	74.71	6.65	240	3
Sequence Valid Inequality	39.5	3.43	240	18
A priori Valid Inequality	39.5	3.47	240	18
$Y_{vv'}$ Branching - Heuristic	60.30	4.75	240	0
$Y_{vv'}$ Branching - Heuristic Solution- A priori Valid Inequality	34.95	1.85	240	0
BR_{vk} Branching	51.50	4.51	240	18
BR_{vk} Branching - Heuristic	203.62	25.56	240	4
$Y_{vv'}$ and BR_{vk} Branching - Heuristic - Sequence Valid Inequality	89.00	34.09	240	2

As it can be seen from the Table 5.4 two of the approaches cannot find feasible solution for most of the problems: (i) *Unmodified CPLEX* and (ii) BR_{vk} *Branching* approaches could only find two feasible solutions out of 20 problems. Thus for these cases a comparison cannot be made with the other solution approaches and they will be excluded from the analysis. When branching strategy of *CPLEX* is modified with $Y_{vv'}$ branching, an important improvement can be obtained such that only three out of 20 problems are infeasible. In addition to this, when heuristic solution approach is included to the B&C procedure of *CPLEX*, the remaining three problems are also solved. The same improvements cannot be seen when branching strategy is changed with BR_{vk}. Even though the B&C procedure is improved with addition of heuristic solution approach, the results are inferior to $Y_{vv'}$ branching when they are compared according to solution quality.

Note that, the valid inequality additions do not have a value when solely applied (could not find feasible solution for 18 problems out of 20). This shows that the valid inequality modification only improves the procedure when there is an upper bound provided with $Y_{vv'}$ branching and heuristic solution approach. In addition to this, there is not a significant difference between a priori applied and locally applied sequence valid inequalities. Thus, we will employ locally applied valid inequalities which is referred as sequence valid inequalities.

When modifications are compared according to the solution quality it is observed that $Y_{vv'}$ *Branching - Heuristic Solution - Sequence Valid Inequality* combination gives the best results overall. We can observe that all $Y_{vv'}$ branching, heuristic solution approach and sequence valid inequality modifications have a cascading effect, such that the solution quality always increases significantly when modifications are used on top of each other. In Table 5.5, the AWT for problems that all of these three different levels of modifications have found feasible solutions are given.

Table 5.5. The results for cases where all approaches found feasible solutions.

Combination	**AWT**
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**32.06**
$Y_{vv'}$ Branching - Heuristic	62.88
$Y_{vv'}$ Branching	74.71

Even though the modifications improve the UB values, the gap between LB and UB values are very high. The $Y_{vv'}$ branching and heuristic solution approach modifications does not improve the average gap but sequence valid inequality has a significant effect on gap values when used with heuristic solution approach and $Y_{vv'}$ branching strategy. Not that, despite of the improvement in UB values, the AG is still very high such that a termination by optimality cannot be obtained.

The second problem set consists of the problems generated in Chapter 4. The results in Chapter 4 showed that for cases with MUF values 1.4 and 1.5 the problems are relatively easy such that all of the solution approaches can find optimal solutions for most of the cases. Thus for this experimental study, problem sets only with MUF values 1.6 and higher will be used. In addition to this, it has been observed that NC does not have a significant effect on solution quality, thus only cases with NC value of 1.8 will be used. The results for problem sets for projects with 22 activities are given in Tables 5.6, 5.7, 5.8; and for 32 activities in Tables 5.9, 5.10, 5.11, respectively. Note that when a solution procedure cannot find a reasonable number of feasible and/or optimal solution, it is not included in the following tables with higher MUF values since the problems become tighter with respect to resource constraints.

Table 5.6. The results for 22 activities, 1.8 NC and 1.5 MUF.

Combination	AWT	AG	ART	NS
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**41.00**	**0.04**	**158.34**	**0**
$Y_{vv'}$ Branching - Heuristic	42.50	0.12	190.62	0
$Y_{vv'}$ Branching	44.40	0.17	195.97	0
Unmodified	44.75	0.23	216.70	6

Table 5.7. The results for 22 activities, 1.8 NC and 1.6 MUF.

Combination	AWT	AG	ART	NS
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**44.33**	**0.17**	**200.50**	**0**
$Y_{vv'}$ Branching - Heuristic	51.17	0.33	240	0
$Y_{vv'}$ Branching	44.40	0.28	240	4

Table 5.8. The results for 22 activities, 1.8 NC and 1.7 MUF.

Combination	AWT	AG	ART	NS
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**79.50**	**0.39**	**200.50**	0
$Y_{vv'}$ Branching - Heuristic	82.30	0.47	240	0

Table 5.9. The results for 32 activities, 1.8 NC and 1.5 MUF.

Combination	AWT	AG	ART	NS
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**35.30**	**0.01**	**49.34**	0
$Y_{vv'}$ Branching - Heuristic	37.10	0.06	148.72	0
$Y_{vv'}$ Branching	35.20	0.01	49.34	0
Unmodified	237.83	5.79	219.06	4

Table 5.10. The results for 32 activities, 1.8 NC and 1.6 MUF.

Combination	AWT	AG	ART	NS
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**47.33**	**0.25**	**240**	**1**
$Y_{vv'}$ Branching - Heuristic	50.78	0.35	240	1
$Y_{vv'}$ Branching	94.75	1.58	240	6

Table 5.11. The results for 32 activities, 1.8 NC and 1.7 MUF.

Combination	AWT	AG	ART	NS
$\mathbf{Y_{vv'}}$ Branching - Heuristic Solution - Sequence Valid Inequality	**96.62**	**0.48**	**240**	**2**
$Y_{vv'}$ Branching - Heuristic	96.62	1.06	240	2

When results for cases with 22 activities with 1.5 MUF values it can be seen the AWT for each modification combination and unmodified *CPLEX* are not significantly different but *unmodified CPLEX* cannot find feasible solution for most of the cases (six out of ten problems). Even though the AWT values do not differ significantly for solution approaches, the AG for Y_{vv} *Branching - Heuristic Solution - Sequence Valid Inequality* modification combination is significantly lower than the other cases. This shows that when MUF values are not relatively tight project sequence valid inequalities can improve the lower bound for the B&C procedure. For the cases with MUF values 1.6 Y_{vv} *Branching - Heuristic Solution - Sequence Valid Inequality* modification combination has a significant AWT and AG advantage when it is compared to the remaining modification combinations. Note that $Y_{vv'}$ *branching* modification starts to fall behind and cannot find feasible solutions for four cases out of ten. The lower AWT values for $Y_{vv'}$ *branching* modification is the result of this limited feasible result base where it can find competitive results with the other approaches. The final test cases with MUF values 1.7 has some interesting implications. The two remaining modification combinations, namely Y_{vv} *Branching - Heuristic Solution - Sequence Valid Inequality* and Y_{vv} *Branching - Heuristic Solution*, do not have a significant AWT difference and the difference between AG values further diminishes with respect to cases with lower MUF values. This points out that the effectiveness of sequence valid inequality is highly correlated with MUF values and the improvement decreases when resource budget is tight.

The results for cases with 32 activities are very parallel to results for the cases with 22 activities. $Y_{vv'}$ *Branching - Heuristic Solution - Sequence Valid Inequality* modification combination has a significant advantage for AWT and AG performance measures for all cases with diminishing returns respect to MUF values.

6. CONCLUSIONS AND FURTHER RESEARCH TOPICS

Multi-project management is a major way of doing business both in manufacturing and services and, being a large-scale complex problem, it constitutes an important research area. Available approaches to this problem in literature generally address a multi-project management environment where the individual projects can share the available resources from a common pool. In this dissertation, we have proposed different resource management policies that can define multi-project environments with different characteristics. The first resource management policy proposed is RD Policy where resources are dedicated to projects during their execution and cannot be shared. The second policy is an extended version of RD policy where resource transfers are allowed among projects when a project finishes before the start of another one. This policy is called the RRD policy. With RD concept we have defined a project management environment where resources are dedicated to the different projects and managed individually. On the other hand RRD extension introduces an internal sequence relation decision among projects which can be seen as another important managerial concept in multi-project scheduling environments.

The proposed resource management policies are investigated in a multi-project environment including general resource capacity decisions and a general budget. With this approach, general resource capacity considerations are incorporated into the problem environment, which are an important part of the problem environment since the coupling of projects are enabled through resource related constraints. This problem environment is called the Resource Portfolio Problem. We have introduced different solution procedures for RPP under proposed resource management policies and employed the insights we have gained from focused problems to the more complex ones.

6.1. Basic Contributions of the Dissertation

The contributions of the dissertation can be summarized under two main groups. The first group is related with the conceptual and mathematical modeling contributions to the multi-project scheduling literature. We have expanded on the resource management concept and proposed different resource management policies that cover different aspects of the multi-project scheduling problems. These resource management policies do not cover all the aspects of the multi-project scheduling environments but define some important characteristics of the different problem environments.

RD Policy is the first policy investigated in this dissertation. RD and RS policies define the two extremes of resource management where in RD policy, resources cannot be shared but, on the other hand, in RS policy, there is not a restriction on sharing the resources. Thus, with RD policy, an important inconsistency between certain multi-project scheduling environments (with geographical limitations and/or some certain characteristics of projects and/or resources that hinder RS) and general assumptions of multi-project scheduling literature is removed. In order to address a relatively larger set of cases from the multi-project environments met in practice, RD policy is extended introducing the so-called RRD policy. With RRD policy resources can be transferred among projects when a project starts after the finish of another one. This concept also introduces an internal sequence relation decision into the problem environment.

Apart from a couple of examples in literature, the general problem settings for multi-project scheduling do not include general resource capacity decisions into the model which are in fact conceptually one of the most important parts of the multi-project scheduling problems since they are the key aspects that couple different projects. Thus, to investigate different resource management policies with a complete multi-project problem environment general resource capacity decisions are added to the problem environment together with a limiting general budget. We call this problem as the Resource Portfolio Problem and we propose different mathematical formulations for RPP under different resource management policies. All these concepts and mathematical models developed for these issues constitute the first group of contributions of this dissertation.

The second group of contributions of this dissertation is basically the solution approaches developed for RPP under RD and RRD policies. First of all, to understand the basic characteristics of RD policy, we have used a narrowed problem environment without general resource capacity decisions. This problem is called the Resource Dedication Problem. A genetic algorithm is proposed for RDP with a new improvement heuristic, namely Combinatorial Auction for RD. This improvement heuristic is based on preference concept. The preference of a project for a resource can be defined as the value of the resource for the project with respect to the current resource state of the project. We have developed two different approaches for preference calculation: linear relaxation based and Lagrangian relaxations based. The latter one turned to be more efficient.

The insights gained from RDP are used to develop solution approaches for RPP under RD policy and resulted with a two-phased GA. In the first phase, the RD space is searched whereas in the second phase, resource portfolio space search is included to the GA. RP search is facilitated with a new improvement heuristic, namely CA for RP which is based on general resource preferences. General resource preferences can be defined as the value the multi-project environment gives to the corresponding general resource. The coupling of these two different improvement heuristics is achieved with preference calculation approaches. Finally for RPP under RRD policy the B&C procedure of CPLEX is modified with different branching strategies, valid inequalities and feasible solution approaches. We modified our improvement heuristics and employed them to generate feasible solutions for the B&C procedure. All the proposed solution approaches for the corresponding problems are found to be efficient.

6.2. Possible Future Research Directions

The first future research direction would be defining different resource management policies that describe different characteristics of multi-project environments that are not investigated in this dissertation. One case can be considering resource transfers among projects with a time and/or value cost. This case is investigated by Krüger and Scholl (2008) and Krüger and Scholl (2009) where the individual projects have characteristics of RCPSP and general resource capacities are given as parameters. Another research direction can be combining shared resources, dedicated resources and transferable resources and generating a general problem setting.

Changing the objective criterion would open other research directions. Even though makespan based objective functions are common in multi-project scheduling, cost/revenue based objective functions also have an important place. A cost/revenue based objective function with makespan based additional constraints can be an interesting research direction.

Another important future research direction can be changing the static nature of the problem environment. In a dynamic multi-project scheduling problem environment, resource management concept and RPP can be incorporated with project portfolio selection and/or with decisions for new project arrival.

Furthermore, the proposed solution approaches can be improved in different ways. First of all, the proposed solution approaches include solving MRCPSP during their execution. This task is still time consuming despite the improvements we have with solution strategies. A fast heuristic approach combined with exact solution approaches can improve this time consuming task. In addition to this, different parts of the modified B&C procedure can be improved with new valid inequalities and lower bound calculation approaches.

REFERENCES

Alcaraz J., C. Marato and R. Ruiz, 2003, "Solving the Multi-Mode Resource-Constrained Project Scheduling Problem with Genetic Algorithms", *Journal of Operational Research Society*, Vol. 54, pp. 614-626.

Bianco L., P. Dell'Olmo and M. G. Speranza, 1998, "Heuristics for Multimode Scheduling Problems with Dedicated Resources", *European Journal of Operational Research*, Vol. 107, pp. 260-271.

Boctor F. F., 1993, "Heuristics for Scheduling Projects with Resource Restrictions and Several Resource-Duration Modes", *International Journal of Production Research*, Vol. 31 No. 11, pp. 2547-2558.

Bouleimen K. and H. Lecocq, 2003, "A New Efficient Simulated Annealing Algorithm for the Resource Constrained Project Scheduling Problem and Its Multiple Mode Version", *European Journal of Operational Research*, Vol. 49, pp. 268-281.

Can A. and G. Ulusoy, 2011, "Multiproject Scheduling with 2-Stage Decomposition", *submitted to Annals of Operations Research*.

Demeulemeester E., 1995, "Minimizing Resource Availability Costs in Time-Limited Project Networks", *Management Science*, Vol. 41, No. 10, pp. 1590-1598.

Demeulemeester E. and W. S. Herroelen, 2002, *Project Scheduling: a Research Handbook*, Kluwer Academic Publishers, New York.

Gademann No. and M. Schutten, 2004, "Linear Programming Based Heuristics for Project Capacity Planning", *IIE Transactions*, Vol. 37, pp. 153-165.

Gonçalves J. F., J. J. M. Mendes and M. G. C. Resende, 2008, "A Genetic Algorithm for Resource Constrained Multi-Project Scheduling Problem", *European Journal* *of Operational Research*, Vol. 189, pp. 1171-1190.

Hans E. W., W. Herroelen, R. Leus and G. Wullink, 2007, "A Hierarchical Approach to Multi-Project Planning under Uncertainty", *Omega*, Vol. 35, pp. 563-577.

Hartmann S. and A. Sprecher, 1996, "A Note on Hierarchical Models for Multiproject Planning and Scheduling", *European Journal of Operational Research*, Vol. 94, pp. 377-383.

Hartmann S., 2001, "Project Scheduling with Multiple Modes: A Genetic Algorithm", *Annals of Operations Research*, Vol. 102, pp. 111-135.

ILOG, 2007, "Users Manual for CPLEX", 2009, ftp://public.dhe.ibm.com/software /websphere/ilog/docs/optimization/cplex/ps_usrmancplex.pdf, May 2012.

Jozefowska J., M. Mika, R. Rozycki, G. Waligora and J. Weglarz, 2001, "Simulated Annealing for Multi-Mode Resource-Constrained Project Scheduling Problem", *Annals of Operations Research*, Vol. 102, pp. 137-155.

Kim S. Y. and R. C. Leachman, 1993, " Multi-Project Scheduling with Explicit Lateness Costs", *IIE Transacitons*, Vol. 25, No. 2, pp. 34-44.

Kolisch R., A. Sprecher and A. Drexl, 1995, "Characterization and Generation of a General Class of Resource-Constrained Project Scheduling Problems", *Management Science*, Vol. 41, No. 10, pp. 1693-1703.

Kolisch R. and A. Sprecher, 1996, "PSPLIB - A Project Scheduling Problem Library", *European Journal of Operational Research*, Vol. 96, pp. 205-216.

Krüger F. and A. Scholl, 2008, "Managing and Modeling General Resource Transfers in (Multi-)Project Scheduling", *OR Spectrum*, Vol. 32, pp. 369-394.

Krüger F. and A. Scholl, 2009, "A Heuristic Solution Framework for the Resource Constrained (Multi-)Project Scheduling Problem with Sequence Dependent Transfer Times", *European Journal of Operations Research*, Vol. 197, pp. 492-508.

Kurtuluş I. S. and S. C. Narula, 1985, "Multi-Project Scheduling: Analysis of Project Performance", *IIE Transactions*, Vol. 17, No. 1, pp. 58-66.

Lawrance S. R. and T. E. Morton, 1993, "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.

Lova A., C. Maroto and P. Tormos, 2000, "A Multicriteria Heuristic Method to Improve Resource Allocation in Multiproject Environment", *European Journal of Operational Research*, Vol. 127, pp. 408-424.

Lova A., P. Tormos and F. Barber, 2006, "Multi-Mode Resource Constrained Project Scheduling Schemes, Priority Rules and Mode Selection Rules", *Inteligencia Artificial*, No. 39, pp. 69-86.

Mika M., G. Waligora and J. Weglarz, 2005, "Simulated Annealing and Tabu Search for Multi-Mode Resource-Constrained Project Scheduling with Positive Discounted Cash Flows and Different Payment Models", *European Journal of Operational Research*, Vol. 164, pp. 639-668.

Mika M., G. Waligora and J. Weglarz, 2008, "Tabu Search for Multi-Mode Resource-Constrained Project Scheduling with Schedule Dependent Set-up Times", *European Journal of Operational Research*, Vol. 187, pp. 1238-1250.

Mittal M. L. and A. Kanda, 2008, "Two Phase Heuristics for Scheduling of Multiple Projects", *International Journal of Operational Research*, Vol. 4 No. 2, pp. 159-177.

Montano B. R. and R. A. Malaga, 2003, "A Weighted Sum Genetic Algorithm to Support Multiple-Party Multiple Objective Negotiations", *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 4, pp. 366-377.

Mori M. and C. C. Tseng, 1997, "A Genetic Algorithm for Multi-Mode Resource Constrained Project Scheduling Problem", *European Journal of Operational Research*, Vol. 100, pp. 134-141.

Morton T. E. and M. R. Singh, 1988 "Implicit Costs and Prices for Resources with Busy Periods", *Journal of Manufacturing and Operations Management*, Vol. 1, Iss. 3, pp. 305-322

Möhring R. H., 1984, "Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion Time", *Operations Research*, Vol. 32, No. 1, pp. 89-120.

Pritsker A. A. B., J. W. Lawrance and P. M. Wolfe, 1988, "Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach", *Management Sciences*, Vol. 16, No. 1, pp. 93-108.

Smith R. G. and R. Davis, 1983, "Negotiation as a Metaphor for Distributed Problem Solving", *Artificial Intelligence*, Vol. 20, pp. 63-109.

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

Sprecher A., S. Hartmann and A. Drexl, 1997 "An Exact Algorithm for Project Scheduling with Multiple Modes", *OR Spektrum*, Vol. 19, pp 195-203.

Sprecher A. and A. Drexl, 1998, "Multi-Mode Resource-Constrained Project Scheduling by a Simple and Powerful Sequencing Algorithm", *European Journal of Operational Research*, Vol. 107, pp. 431-450.

Talbot F. B., 1982, "Resource-Constrained Project Scheduling with Time-Resource Tradeoffs: The Nonpreemptive Case", *Management Science*, Vol. 28, No. 10, pp. 1199-1210.

Tsubakitani S. and R. F. Deckro, 1990, "A Heuristic Approach for Multi-Project Scheduling with Limited Resources in the Housing Industry", *European Journal of Operational Research*, Vol. 49, pp. 80-91.

Vanhoucke M., 2002, "Optimal Due Date Assignments in Project Scheduling", *Working Paper, Ghent University*, 2002/159.

Vries S. D. and R. Vohra, 2000 "Combinatorial Auctions: A Survey", *Draft Paper, Department of Managerial Economics and Decision Sciences, Kellogg School of Management, Northwestern University*, 2000

Yang K. K. and C. C. Sum, "An Evolution of Due Date, Resource Allocation, Project Release and Activity Scheduling Rules in a Multiproject Environment", *European Journal of Operation Research*, Vol. 103, pp. 139-154.

APPENDIX A: NUMERICAL EXAMPLES FOR COMBINATORIAL AUCTION FOR RESOURCE DEDICATION PROBLEM

In this Chapter numerical examples for linear relaxation and Lagrangian relaxation based Combinatorial Auction for Resource Dedication Problem are given.

A.1. Linear Relaxation Based Preference Calculation

LR BASED COMBINATORIAL AUCTION FOR RDP

		Resource Dedications				Slack Values			
Projects	Obj. Value	R1	R2	N1	N2	R1	R2	N1	N2
P1	12	20	28	103	105	0	0	0	1
P2	5	21	26	98	97	2	0	36	32
P3	16	19	26	98	96	0	0	0	27
P4	18	19	26	98	97	0	0	16	0
P5	8	19	31	98	97	0	5	1	37
P6	9	19	26	98	96	0	0	0	0

OBTAIN ALLOWABLE UPPER BOUNDS FROM LR AND OBTAIN A_{KV} AND A_{IV} VALUES WITH (3.11) AND (3.13)

	R1		R2		N1		N2	
Projects	AUB	a_{kv}	AUB	a_{kv}	AUB	a_{iv}	AUB	a_{iv}
P1	20.01	110.70	28.01	72.87	107.24	0.24	106.79	0.56
P2	NA	0.00	NA	0.00	NA	0.00	NA	0.00
P3	19.06	17.71	26.08	11.98	98.22	4.49	NA	0.00
P4	20.67	0.60	27.01	0.99	NA	0.00	97.50	2.00
P5	19.11	9.20	26.20	5.05	NA	0.00	NA	0.00
P6	19.30	3.38	27.28	0.78	99.40	0.71	96.99	1.01

CALCULATE PREFERENCES WITH (3.12) AND (3.14)

	Preferences			
Projects	R1	R2	N1	N2
P1	0.7818	0.7949	0.0434	0.1569
P2	0.0000	0.0000	0.0000	0.0000
P3	0.1251	0.1307	0.8253	0.0000
P4	0.0042	0.0108	0.0000	0.5606
P5	0.0650	0.0551	0.0000	0.0000
P6	0.0239	0.0085	0.1314	0.2824

DETERMINE NEW RESOURCE DEDICATION VALUES WITH KNAPSACK MODELS AND OBTAIN SCHEDULES FOR EACH PROJECT WITH NEW RESOURCE DEDICATIONS

		New Resource Dedication Values			
Projects	Obj. Value	R1	R2	N1	N2
P1	12	22	33	103	104
P2	5	19	26	62	65
P3	8	19	26	151	69
P4	9	19	26	82	194
P5	8	19	26	97	60
P6	9	19	26	98	96

Figure A.1. General procedure for CA for RDP employing linear relaxation based preference calculation.

A.2. Lagrangian Relaxation Based Preference Calculation

LA BASED COMBINATORIAL AUCTION FOR RDP

		Resource Dedications				Slack Values			
Projects	Obj. Value	R1	R2	N1	N2	R1	R2	N1	N2
P1	16	25	25	101	103	0	0	1	0
P2	0	23	24	90	93	0	12	11	35
P3	165	20	24	88	92	0	0	1	0
P4	18	23	24	90	93	0	0	5	0
P5	12	23	24	90	92	0	8	17	0
P6	5	23	23	89	92	2	0	3	23

SOLVE MODEL LA-SP FOR EACH PROJECT AND OBTAIN PREFERENCES WITH (3.18) AND (3.19)

	Preferences			
Projects	R1	R2	N1	N2
P1	0.0100	0.0100	0.1904	0.0100
P2	0.0100	0.0100	0.0100	0.0100
P3	0.4383	0.6525	0.0470	0.0527
P4	0.0100	0.0100	0.0100	0.1599
P5	0.0100	0.0100	0.0100	0.0517
P6	0.1540	0.1027	0.0100	0.0100

DETERMINE NEW RESOURCE DEDICATION VALUES WITH KNAPSACK MODELS AND OBTAIN SCHEDULES FOR EACH PROJECT WITH NEW RESOURCE DEDICATIONS

		New Resource Dedication Values			
Projects	Obj. Value	R1	R2	N1	N2
P1	8	25	25	138	103
P2	0	23	12	79	58
P3	75	22	44	87	92
P4	9	23	24	85	151
P5	12	23	16	73	92
P6	5	21	23	86	69

Figure A.2. General procedure for CA for RDP employing Lagrangian relaxation based preference calculation.