Multiple parallel machines scheduling with setup resources
by Shaowei Wang

A thesis submitted in partial fulfillment Of the requirements for the degree of Master of Science in Industrial and Management Engineering
Montana State University

Abstract:
In this paper a multiple parallel machines scheduling problem with setup resources (MPMSRP) is modeled and studied in which a set of independent tasks need to be processed on a set of renewable resources and nonrenewable resources in a single stage. .Each task can be carried out in several alternative modes; that is, with different resource sets and processing times. The objective is to assign a mode and a start time for each task so that the throughput is maximized. The problem instances are generated and solved by a LP-solver with LP relaxation and a proposed local search heuristic. The computational results for the heuristic are discussed.

MULTIPLE PARALLEL MACHINES SCHEDULING WITH SETUP RESOURCES

by

Shaowei Wang

A thesis submitted in partial fulfillment
Of the requirements for the degree

of

Master of Science

in

Industrial and Management Engineering

MONTANA STATE UNIVERSITY
Bozeman, Montana

May 2003

APPROVAL

of a thesis submitted by

Shaowei Wang

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

Edward L. Mooney, Ph.D. Edward L. Mooney 5/16/03
(Signature) Date

Approved for the Department of Mechanical & Industrial Engineering

Vic A. Cundy, Ph.D. Vic A. Cundy 5-15-03
(Signature) Date

Approved for the College of Graduate Studies

Bruce R. McLeod, Ph.D. Bruce R. McLeod 5-20-03
(Signature) Date

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Signature _____

Date 05/16/03

ACKNOWLEDGEMENTS

I would like to thank Dr. Ed Mooney for his assistance. He gave me this topic and encouraged me to go on thesis study. He did a lot of works in the model development, algorithm and data structure design and coding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In this paper a *multiple parallel machines scheduling problem with setup resources* (MPMSRP) is modeled and studied in which a set of independent tasks need to be processed on a set of renewable resources and nonrenewable resources in a single stage. Each task can be carried out in several alternative modes; that is, with different resource sets and processing times. The objective is to assign a mode and a start time for each task so that the throughput is maximized. The problem instances are generated and solved by a LP-solver with LP relaxation and a proposed local search heuristic. The computational results for the heuristic are discussed.

CHAPTER 1

INTRODUCTION

Scheduling concerns allocating limited resources to tasks over time. In this paper, we consider a multiple parallel machines scheduling problem in which three types of resources are available over time: *renewable parallel resources* (R_1), *renewable setup resources* (R_2) and *nonrenewable (consumable) setup resources* (R_3). Renewable (parallel and setup) resources are assumed available at every time period. Examples would be manpower and tools. Railcars may be modeled as parallel, renewable resources and setup tools and operators are examples of renewable setup resources.

In contrast to the renewable resources, nonrenewable resources are consumable over time periods with a limited total consumption. An example would be raw materials. The setup resources are consumed in 1 period.

The study was motivated by a railcar scheduling problem studied by Li [1]. In this model, the railcars were renewable parallel resources, the loading facilities were renewable, setup resources and the inventories of material were the consumable resources.

A set of tasks has to be carried out without preemption using a specified set of resources, or *mode*. Each task can be performed in one out of a set of alternative modes with a processing time specified for each mode. We assume that there is no precedence

constraints between tasks and the setup resources are requested for only one time period. The goal is to choose a mode and a start time for each task so that the throughput is maximized over a given planning horizon.

This paper defines a model for the *Multiple Parallel Machines with Setup Resources Problem* (*MPMSRP*) and explores the relationship of the problem instance characteristics to the solution methods. The problem instances are generated and solved by a LP-solver with LP relaxation and a proposed local search heuristic. The computational results are discussed.

The paper is organized as follows. In Chapter 2, we introduce several classical scheduling problems that are related to MPMSRP and address the differences. In Chapter 3, we present the notation and formulate three models. In Chapter 4, a test problem generator is given. In Chapter 5, we discuss some properties of the model and a local search heuristic is proposed in Chapter 6. Then the experimental design and the performance of proposed heuristic are covered in Chapter 7. Finally, Chapter 8 provides a brief summary and conclusions.

CHAPTER 2

LITERATURE SURVEY

The multiple parallel machines scheduling with setup resources problem (MPMSRP) is a restriction of the general resource-constrained scheduling problem without precedence constraints and a generalization of several other problems. There are four types of classical scheduling problems that are most closely related with the MPMSRP: Parallel Machines, Flexible Job/Flow Shop Scheduling, Routing & Scheduling (Transportation, Logistics) and Resource Constrained Project Scheduling (RCPSP). See Figure 1 for detail.

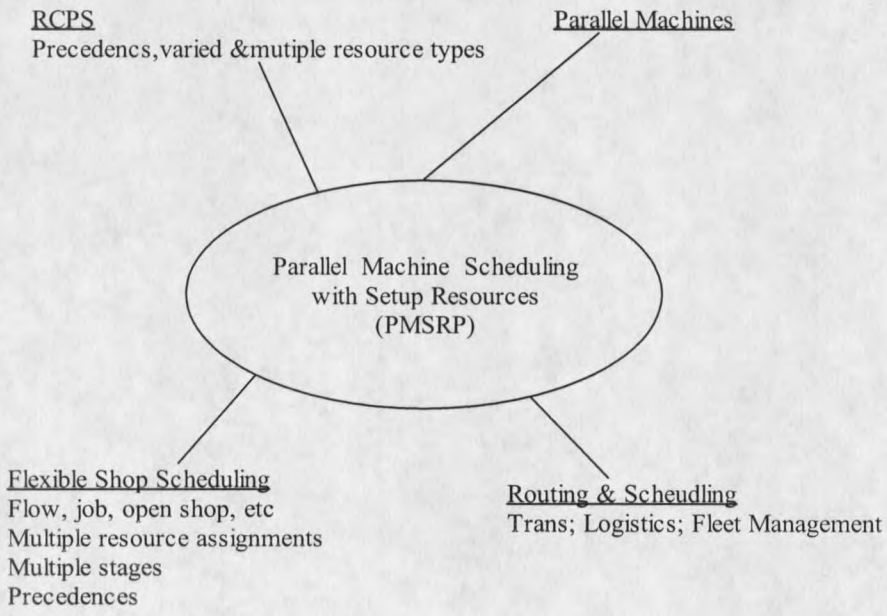

Figure 1. Classical Scheduling Problems Related with MPMSRP

In the classical parallel machine scheduling problem only parallel (renewable) machine capacity is considered; no additional resource types are considered. Daniels et al.[2] explore the impact of resource flexibility by developing and analyzing heuristics for the identical parallel-machine flexible-resource scheduling problem with unspecified job assignment (UPMFRS). No setup resources are considered here, a mixed-integer program is used for model formulation.

A few scheduling models have appeared in the literature which take additional resources into consideration. Ventura and Kim[3] identified a problem of scheduling jobs on parallel machines with an unrestricted due date and additional resources. They considered only a single type of additional resource and each job requires one machine and at most one unit of additional resource for it's processing. Bourland et al. [4] discussed the fractional setup operator requirements in processing a task. In their studies, the operator can handle many machines. They grouped the machines in different stages in order to find the optimum number of operators required to process all tasks. Since the workers movement is limited to one stage, grouping is confined to single stage. They concluded that this approach might not be feasible for realistically sized problems. There is a finite planning horizon of t periods. Setup times are assumed to be integer multiples of one period. The operator is treated as a setup resource type. Slowinski [5] considered a parallel machine scheduling problem where the jobs may require an additional renewable resource, but he assumed setup time is zero and allows pre-emption without penalty.

Herrmann and Lee [6] examined parallel machine scheduling with operator constrained setups. Only one setup resource is considered. Olafsson and Shi[7] address the Parallel-Machine Flexible-Resource Scheduling (PMFRS) problems of simultaneously allocating flexible resources and sequencing jobs. One renewable resource is considered.

Flow shop, job shop and flexible shop scheduling problems are generalizations of our problem where jobs require more than one group of parallel machines. In flow shop problems, all the jobs follow the same sequence. In Job shop problems, all the jobs can have different sequences. Both require multiple resources but with only one request unit. In flexible flow shop problem, like Yang, S.K.a.M.P. [8] Chung and Shi [9], one or more resource units are required. Daniels and Mazzola [10] consider a scheduling problem in which the tasks follow the same sequence. The best sequence that minimizes the schedule makespan is found using the iterative procedure. Only one resource is studied here .The resource is allocated by assigning an integer number of workers to a job. They concluded that the complexity reduces the optimality of large practical problems. Our problem differs from Daniels and Mazzola's problem as each task in our problem requires multiple resources. Brah and Loo [11] studied the flow shop with multiple processors problem. It is a generalization of flow shop problem, where at least one stage the processor has more than one identical machine. The jobs are subject to precedence constraints. No setup resource constraint is considered.

Routing and Scheduling problems (Transportation, Logistics, etc) are related to our problem. Sherali et al. [12] mainly focused on the Kuwait Petroleum Corporation(KPC) Problem. Crude oil and a number of refined oil-related products are shipped to customer. Two classes of vessels are considered: the first is the fleet of vessels controlled by KPC, the second is spot-chartered. A mixed-integer programming model is constructed. No setup resource is considered. Powell et al. [13] develop a new method for solving Dynamic Resource Allocation Problems(DRAP). Xu and Kelly [14] use network flow-based tabu search to solve the problem. Such problems focus on the allocation of resources to perform tasks over a network, for example, routing and scheduling over 5,000 drivers to serve 30,000 loads over a four days horizon.

In the literature, scheduling problems where multiple resources are considered are known as resource-constrained project scheduling problems (RCPSP) and have received considerable attention during the past several decades. Tasks are subject to precedence relations, require units of multiple renewable, non-renewable constrained resources and can be performed in multiple modes. Like Salewski et al. [15], Shewchuk and Chang [16], Bert and Eilly [17] and Rainer and Andreas [18], etc. In our case, all the jobs have no precedence.

CHAPTER 3

NOTATION AND GENERAL PROBLEM FORMULATION

The notation used to model the parallel machine scheduling with setup resources (MPMSRP) is summarized in Table 1.

Table 1. Problem Parameters of the MPMSRP

Problem Notation	Definition
$T=\{t: t=1,2,\ldots n_t\}$	Set of time period(planning horizon) indexes
$I=\{i: i=1,2,\ldots n_i\}$	Set of task indexes
$M_i=\{m: =1,2,\ldots n_{m_i}\}$	Set of mode indexes for task i
$d(i,m)$	Duration of the i^{th} task in mode m
$R=\{r: r=1,2,\ldots n_r\}$	Set of resource indexes $r\in R=\{R_1\|R_2\|R_3\}$
R_1	parallel, renewable resources with resource number $NR_1=\|R_1\|$
R_2	setup, renewable resources with resource number $NR_2=\|R_2\|$
R_3	setup, consumable resources with resource number $NR_3=\|R_3\|$
A_r	Resource capacity for resource r
a_{imr}	Number of units of resource r required in mode m for task i

We define a model for the problem of simultaneously sequencing tasks on multistage parallel, identical resources and scheduling setup resources over a planning horizon consisting of n_t time periods

Sets of discrete renewable and consumable resources are available. Each renewable resource $r\in R_1 \cup R_2$ is always available. Each consumable resource $r\in R_3$ is only available over the whole horizon and is consumed over time. Each task i can be carried out in

several modes, and each mode requires a set of resources $R_m \subseteq R$ with a mode-dependent processing time. Here we assume $d(i,m)$ is the duration of task i performed in mode m. During this time, the renewable parallel resources R_1 are used, while for renewable (R_2) and consumable setup resources (R_3) only one time period is assumed. Each mode is treated as different from any other mode, even if two modes of different tasks are assigned with the same resource set and with the same duration.

Many linear objective functions can be selected, such as minimizing makespan, minimizing total tardiness, etc. In many manufacturing systems, system throughput is very important. In this paper, the objective is to schedule each task in one of its modes, subject to the resource constraints under the objective of maximizing the throughput during a fixed horizon.

Three different MPMSRP models are given below ($MPMSRP_1$, $MPMSRP_2$, and $MPMSRP_3$). $MPMSRP_1$ is a 3-way assignment model. It uses 0-1 variables to formulate the model using the general idea given in Pritsker, Watters, and Wolfe [19]. Using this model, the process of finding an optimal solution is represented as a 3-way assignment: a mode and a start time are assigned to tasks. In mode assignment, a specific resource set (mode), is assigned to each task. After that tasks with fixed mode are assigned to different start time to maximize the throughput. An example is showed in Figure 2.

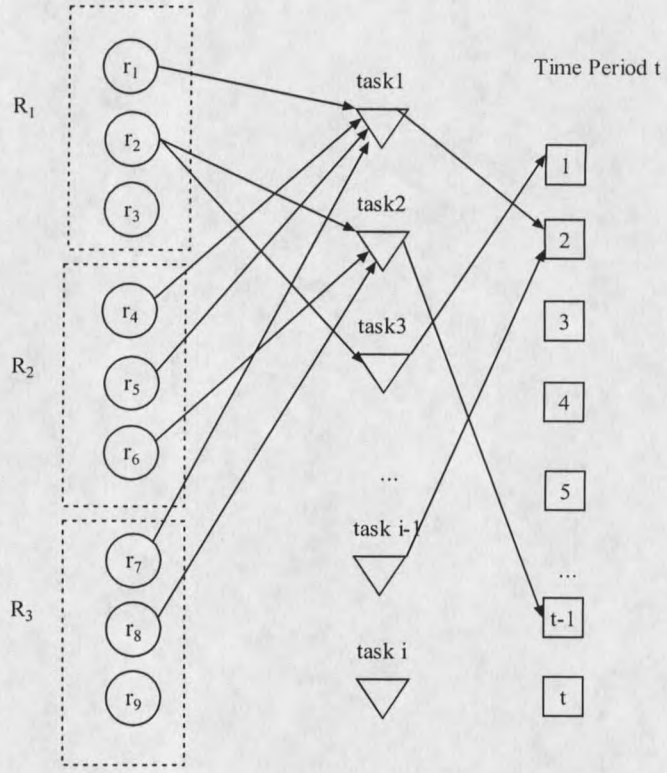

Figure 2. Example of 3-way assignment process

We wish to choose "optimal" values of decision variables x_{imt}, where $m \in M_i$, and

$$x_{imt} = \begin{cases} 1 & \text{if task } i \text{ is assigned to mode } m \text{ at start of period } t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The problem can be formulated as follows:

$$Max \sum_{i=1}^{I} \sum_{m=1}^{N_{mj}} \sum_{q=1}^{T} x_{imq} \quad (2)$$

Subject to:

$$\sum_{m=1}^{n_{mj}} \sum_{q=1}^{t} x_{imq} \leq 1 \qquad \text{for all } i \in I \quad (3)$$

$$(\text{MPMSRP}_1)\ \sum_i \sum_{\substack{m\in M_{ir},and \\ t-d(im)<q\le t}} a_{imr}x_{imq} \le A_r \quad \text{for all } r\in R_1, t\in T \tag{4}$$

$$\sum_i \sum_{m\in M_{ir}} a_{imr}x_{imt} \le A_r \quad \text{for all } r\in R_2, t\in T \tag{5}$$

$$\sum_i \sum_{m\in M_{ir}} \sum_t a_{imr}x_{imt} \le A_r \quad \text{for all } r\in R_3 \tag{6}$$

The objective function (2) maximizes the throughput. Constraint (3) ensures that at most one mode is assigned to any task. This also makes sure that each task can be assigned at most once. Constraint (4) guarantees that parallel, renewable resources assigned at each time period do not exceed their capacity. Constraint (4) also ensures that each task must be continuously processed to its completion without preemption. We assume setup, renewable resource and setup consumable resources only need one time period for processing. So constraint (5) ensures that at any time period the assignment of setup, renewable resources cannot exceed its capacity. Constraint (6) ensures that the total consumption of setup, consumable resources cannot exceed its total capacity for the whole horizon.

We also define a two-way assignment model. MPMSRP_2 represents the problem as the assignment of slots to task where slots represent feasible start time-mode pairings.

$$x_{is} = \begin{cases} 1 & \text{if task } i \text{ is assigned to slot } s(m,t) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$Max \sum_{i=1}^{I} \sum_{s=1}^{S_i} x_{is} \tag{8}$$

Subject to:

$$\sum_{s=1}^{S_i} x_{is} \le 1 \text{ for all } i \in I \tag{9}$$

(PSMRP$_2$)
$$\sum_{i} \sum_{\substack{s \in S_{ir}, \\ \text{and } t-d(im(s))<q\le t}} a_{im(s)r} x_{is} \le A_r \quad \text{for all } r \in R_1, t \in T \tag{10}$$

$$\sum_{i} \sum_{s \in S_{ir}} a_{im(s)r} x_{is} \le A_r \quad \text{for all } r \in R_2, t \in T \tag{11}$$

$$\sum_{i} \sum_{s \in S_{ir}} \sum_{t} a_{im(s)r} x_{is} \le A_r \quad \text{for all } r \in R_3 \tag{12}$$

We also proposed a model MPMSRP$_3$ as follows. Unlike the MPMSRP$_1$, by using this model, we can easily code the heuristics and simplify the search base on simple complement moves.

$$x_j = \begin{cases} 1 & \text{if task } i(j) \text{ is assigned to choice } j \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where each j indexes a choice triple (i (j), m (j), t (j))

$$Max \sum_{j=1}^{J} x_j \tag{14}$$

Subject to:

$$\sum_{j(i(j))} x_j \le 1 \quad \text{for all } i \in I \tag{15}$$

(MPMSRP$_3$)
$$\sum_{j \in J_r} \sum_{t-d(i(j)m(j))<q\le t} a_{r(m(j))} x_j \le A_r \quad \text{for all } r \in R_1, t \in T \tag{16}$$

$$\sum_{j \in J_r} a_{r(m(j))} x_j \le A_r \quad \text{for all } r \in R_2, t \in T \tag{17}$$

$$\sum_{j \in J_r} \sum_{t} a_{r(m(j))} x_j \le A_r \quad \text{for all } r \in R_3 \tag{18}$$

CHAPTER 4

TEST PROBLEMS

We need some test instances to investigate the properties of our proposed models and testing the proposed algorithm. Normally, two possible approaches can be used to come up with test instances: First, we could use data from real-world cases. However, even if an algorithm performs well on some instances, it does not guarantee that it will perform well on other instances. A second approach is to generate artificial instances. If a tough test instance can be solved well by an algorithm, it is likely that a real-world instance could be solved as well.

Therefore, we decided to test our algorithm with randomly generated test instances and a set of factors were proposed as the parameters to control the experiments. Here we assume that an instance is determined by the length of the planning horizon, the number of tasks, the number of modes, *resource strength* (*RS*) and *resource factor* (*RF*).

All the values and parameters used in our experiment are described in Table 2. RS_k, the resource strength of resources of type k=1, 2, or 3, is defined by Kolisch et al.(1996)[20].

$$RS_k = (A_r - a_r^{\min})/(a_r^{\max} - a_r^{\min}) \qquad r \in R,\ k = 1,2,3 \tag{19}$$

RS_k is used to determine resource capacity, A_r, for each resource.

$$A_r = a_r^{\min} + RS_k \times (a_r^{\max} - a_r^{\min}) \qquad r \in R,\ k = 1,2,3 \tag{20}$$

Table 2. Generator parameter settings

Parameter	Value
T (Horizon)	10;20
n_i (Number of tasks)	10;20;30;40;50
n_{m_i} (Number of modes per task)	1;2;3
$d(i,m)$ Duration of the ith task in mode m	[1,10]
r Number of parallel, renewable resource (R_1)	[3,3]
$\neq$ Number of Setup, renewable resource (R_2)	[3,3]
$\neq$ Number of Setup, consumable resource (R_3)	[3,3]
a_{imr} (parallel, renewable resource demand)	[1,10]
$a_{im\neq}$ (setup, renewable resource demand)	[1,10]
$a_{im\neq}$ (Setup, consumable resource demand)	[1,10]
RF_1 (resource factor for R_1 type)	1
RS_1 (resource strength for R_1 type)	0.1;0.25
RF_2 (resource factor for R_2 type)	1
RS_2 (resource strength for R_2 type)	0.1;0.25
RF_3 (resource factor for R_3 type)	1
RS_3 (resource strength for R_3 type)	0.1;0.25

Note: the indication $[x,y]$ means uniform distribution between x,y

Setting RS_k=0 will let $A_r = a_r^{\min}$ and give the smallest feasible resources for that resource type $r \in R$, whereas, setting RS_k=1 gives the largest feasible resources of $a_r^{\max}$ with no resource constraint. Experience showed that, RS_k>0.5 makes test instances very easy, so here we only choose RS_k=0.1 and 0.25.

For the renewable resource type $r \in R_1 \cup R_2$, the lowest availability $a_r^{\min}$ is obtained by setting all the tasks a mode with the lowest demand and selecting the largest one, that is,

$$a_r^{\min} = \max_{i=1}^{n_i} \min_{m=1}^{n_{m_i}} \{a_{imr}\} \quad , r \in R_1, R_2 \tag{21}$$

The maximum level of $a_r^{\max}$ is the peak demand for renewable resource type $r \in R_1 \cup R_2$. Since there is no precedence between tasks, $a_r^{\max}$ is obtained by assigning all the tasks at the same time in a mode with the largest per-period demand for the resource type r and calculate the peak demand, that is,

$$a_r^{\max} = \sum_{i=1}^{n_i} \max_{m=1}^{n_{m_i}} \{a_{imr}\} \quad , r \in R_1 \cup R_2 \tag{22}$$

For a nonrenewable setup resource r, $r \in R_3$,

$$a_r^{\max} = \sum_{i=1}^{n_i} \max_{m=1}^{n_{m_i}} \{a_{imr}\} \quad , r \in R_3 \tag{23}$$

$$a_r^{\min} = \sum_{i=1}^{n_i} \min_{m=1}^{n_{m_i}} \{a_{imr}\} \quad , r \in R_3 \tag{24}$$

The resource factor, *RF*, (Pascoe,1966[21]) reflects the average portion of resources used or consumed. *RF*=1 means each task demands every resource, whereas *RF*=0 indicates a problem without resource demands. Equation (25) defines RF_k for resource type k=1, 2, 3.

$$RF_k = \frac{1}{n_i} \sum_{i=1}^{n_i} \frac{1}{M_i} \frac{1}{|R|} \sum_{m=1}^{n_{m_i}} \sum_{r=1}^{|R|} \begin{cases} 1 & if \quad a_{imr} > 0, \\ 0 & otherwise. \end{cases} \tag{25}$$

For our experiments, test instances were generated using the following procedure with the parameters above.

First, given the upper and lower levels shown in Table 2, the number of resources for each resource type was randomly chosen from Uniform [Lower, Upper]. With the number of tasks n_i and upper and lower level of the number of mode, for each task, the number of modes was randomly selected from Uniform [Lower, Upper]. For each mode, its duration was randomly chosen from uniform [Lower, Upper]. Then using the resource factor, RF_k, for each resource type and equation (25), a resource usage matrix was generated and resource units were randomly assigned from uniform distribution between the lower and upper. Finally, given the resource strength RS for each resource type, the capacity of each resource was calculated by equations (20) through (24).

CHAPTER 5

PRELIMINARY MPMSRP PROBLEM ANALYSIS

The 0-1 ILP formulation in Chapter 3 precisely models the problem and is equivalent to the problem:

$$\max\{cx \mid Ax \le b, x_j = 0 \text{ or } 1, j \in N\}, \tag{26}$$

Where A is $m \times n$, b is $m \times 1$, and $N=\{1,\ldots n\}, M=\{1,\ldots,m\}$. The LP relaxation of this problem can be solved with the simplex method. We found that the optimal relaxation solution is often integer when a_{imr} is 0 or 1 and A_r is integer. We began by exploring the relationship between resource constraints and integer feasible LP relaxations.

We already knew the problem was NP-hard. Bianco et al. [22] proved that even in the simplest case, if a single mode is given for each task, the multiple modes scheduling problem (MMSP) without precedence is NP-hard. MMSP is a restriction of our problem, so MPMSRP is NP-hard. No efficient solution with an exact algorithm would likely give optimal solutions in a reasonable time. We confirm these properties with experiments involving LP relaxation. Since no benchmark problems are available, we use randomly generated problem instances to test the integer property.

As discussed in Chapter 4, the resource capacity (A_r) relative to demand (a_{ir}) is controlled by *RS*. We chose two levels for *RS*, loose capacity and tight capacity, corresponding to *RS*=0.5 and 0.1 respectively for this study.

To illustrate, consider the simplest case: single mode (n_{m_i}=1), one type of parallel, renewable resource ($r_1 \in R_1$) and one type of renewable, setup resource ($r_2 \in R_2$). Table 3 shows the parameters used to generate test problems. Different parameter combinations are used to randomly generate 5 test instances.

Table 3. Parameters used to generate test problems

Problem Parameter	Values considered	Total
Planning horizon ($\|T\|$)	10	1
Number of tasks ($\| n_i \|$)	10,20,30	3
Duration of task i, $d(i)$	uniform[1-10]	

To test the integer property, here we use an *Integrality Index* obtained by solving the LP relaxation with GLPK (GNU Linear Programming Kit) 3.23.

$$Integrality\ Index = \frac{Total\ Number\ of\ Integer\ results}{Total\ Number\ of\ Variables} \times 100$$

Table 4 shows results for one parallel renewable resource with NR_1=1, T=10, ar_1=1, $d(i)$=[1,10] and different number of tasks (problem size). The first column is the number of tasks. The second column is the resource strength RS and total resource capacity. There are five replications for each case. In some cases, we get integer LP relaxation. In general, the tighter the constraints are, the lower the percentage of integer results is.

Table 4. LP-solver Results for T=10, NR_1=1, ar_1=1, $d(i)$=[1,10]

n_i	RS/A_{r1}	Cons.	Var.	Integrality Index(Rep.)					LP relaxation value(Rep.)				
				1	2	3	4	5	1	2	3	4	5
10	0.5/6	20	100	100	92	94	100	100	10	10	10	10	10
	0.1/2	20	100	86	89	88	86	91	7.5	8.8	8.3	6	7
20	0.5/11	30	200	100	100	100	100	100	10	10	10	10	10
	0.1/3	30	200	94	90	92	97	93	14	11	12.7	11	11
30	0.5/16	40	300	97	97	96	97	97	30	30	30	30	30
	0.1/4	40	300	92	96	93	95	97	20.4	15	16.6	18.8	19

It is not always the case that a tighter constraint gives a lower percentage of integer results. As seen from the results in Table 5, instances with one parallel, renewable resource and one additional renewable setup resource (NR_1= NR_2=1) and one required unit for each type of resource (ar_1= ar_2=1), as setup resource constraints become tighter, the relaxation solution becomes "more integral." If we increase the required units for each type of resource to uniform [1, 2], we get the same results, as shown in Table 6.

Table 5. LP-solver Results for T=10, NR_1= NR_2=1, ar_1= ar_2=1, $d(i)$=[1,10]

n_i	RS_1 /A_{r1}	RS_2 /A_{r2}	Const.	Var	Integrality Index (Rep.)					LP relaxation value (Rep.)				
					1	2	3	4	5	1	2	3	4	5
10	0.5/6	0.5/6	30	100	100	100	94	92	100	10	10	10	10	10
	0.1/2	0.1/2	30	100	89	89	89	82	91	7.5	8.8	8.3	6	7
	0.5/6	0.1/2	30	100	100	100	100	89	100	10	10	10	10	10
20	0.5/11	0.5/11	40	200	100	100	100	100	100	20	20	20	20	20
	0.1/3	0.1/3	40	200	94	90	92	97	93	14	11	12.7	11	11
	0.5/11	0.1/3	40	200	100	100	100	100	100	20	20	20	20	20
30	0.5/16	0.5/16	50	300	97	97	96	97	98	30	30	30	30	30
	0.1/4	0.1/4	50	300	92	96	93	95	97	20.4	15	16.6	18.8	19
	0.5/16	0.1/4	50	300	98	100	96	100	100	30	30	30	30	30