Robust Optimization for the Resource Constrained Multi-Project Scheduling Problem with Uncertain Activity Durations

Elham Nabipoor Afruzi¹
Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran 1999143344, Iran,
Phone: +98 21 84063363, Cell Phone: +989112111960, Fax: +98 21 88674858,
E-mail: enabipoorafruzi@mail.kntu.ac.ir

Abdollah Aghaie
Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran 1999143344, Iran,
Phone: +98 21 84063363, Fax: +98 21 88674858, E-mail: aaghaie@kntu.ac.ir

Amir Abbas Najafi
Faculty of Industrial Engineering, K.N. Toosi University of Technology, Tehran 1999143344, Iran,
Phone: +98 21 84063378, Fax: +98 21 88674858, E-mail: aanajafi@kntu.ac.ir

Abstract

This paper studies the multi-project scheduling problem which involves multiple projects with different importance weight; with predefined assigned due dates; with activities that have uncertain durations; and with renewable resources that are constrained. The resource sharing policy is applied to share the resources among projects. Due to the environmental rapid changes and also the uniqueness of projects, the probability distribution function of uncertain durations cannot be estimated with confidence. Besides, the multi-project scheduling problem with its large scale investment dictates a conservative approach to deal with the existing uncertainty. Therefore, the Robust Resource-Constrained Multi-Project Scheduling Problem (RRCMPS) is studied in this paper while the maximum total weighted tardiness of the projects should be minimized. A scenario-relaxation algorithm is implemented which results in optimal solutions for the RRCMPS. The aim is to find an optimal structure containing all the projects in such a way that it transfers the resources between the activities based on the resource sharing policy while the maximum weighted differences between the projects finish times and their assigned due dates will be minimum.

Keyword: Multi-Project Scheduling Problem, Resource Sharing Policy, Robust Optimization, Resource Constraint, Uncertain Activity duration.

¹ Corresponding Author
1. Introduction

The resource constrained project scheduling problem (RCPSP) aims at minimization of the project makespan with consideration of precedence and resource constraints [1]. This problem is one of the most well-known problems which researchers have devoted considerable efforts for its studying over the past decade.

The RCPSP is applicable in many areas such as make to order industries, construction, software development, etc. In modern enterprises in which a large number of projects are set up to achieve the product innovation, the key resource is mostly manpower which belong to renewable resources. In contrary to the importance of renewable resources and its role in project management success, the renewable resources have not attained sufficient consideration in the literature [2]. As a brief definition, the renewable resources are the resources such as manpower, machines, etc. which are constrained and there is a certain available capacity of this kind of resources in each time period. By finishing one activity, its required renewable resources can be released and applied in other activities. In this paper, the project scheduling problem is investigated under renewable resource-constraint condition.

The resource constrained multi-project scheduling problem (RCMPSP) as an extension of the RCPSP is considered as the simultaneous scheduling of two or more projects which demand the same scarce sources [3]. Multi-project management is a major way of doing business both in manufacturing and services, and, being a large-scale complex problem, constitutes an important research area [4]. Payne [5] studied that up to 90% of all projects in the world are executed in a multi-project management environment. It is mentionable that the management of multiple projects presents challenges that are fundamentally different from single project management [6]. So, managing the multi-project problem is not simply an aggregate of single project efforts. In this paper, the multi-project management problem is investigated.

During project execution in an indeterminate environment, the projects are subject to considerable uncertainty. In other words, due to unavailable resources, delays in delivery of materials, absent employees, bad weather conditions, accident and many other uncontrollable factors, some project activities may last longer than expected, threatening the operational viability of the planned schedule [7]. Therefore, the obtained results of the project scheduling model with deterministic parameters are no longer valid. In other words, when the project parameters take realized values, the usability of any result from deterministic models is under question. Therefore, it is conceivable that as the data takes values different from the nominal ones, several constraints may be violated and the optimal solution found using the nominal data may be no longer optimal or even feasible [8]. In this paper, the uncertainty of the activities duration is under study.

There are several approaches for scheduling the projects under uncertainty. In order to select an appropriate approach for dealing with uncertainty in problems, first of all we should investigate the nature and characteristic of the studied problem. The fundamental approaches for scheduling projects under uncertainty are reactive scheduling, stochastic scheduling, scheduling under fuzziness, proactive (robust) scheduling, and sensitivity analysis [9].

Considering the uniqueness of each project in real world, it is not uncommon that its activities are seldom or even never have been executed before. Therefore, these indeterminacies cannot be treated as fuzziness, probability, roughness, ambiguity or entropy. Instead, uncertainty theory, can be a useful tool [1]. Robust optimization (RO) is an appropriate approach that is totally compatible with the nature of project scheduling problem and is applied in this paper.
Robust optimization belongs to an important methodology for dealing with optimization problems with data uncertainty. In this type of method, a deterministic data set is defined within the uncertain space, and the best solution which is feasible for any realization of the data uncertainty in the given set is computed through the solution of the robust counterpart optimization problem [10].

The major advantages of robust optimization compared to stochastic programming are that no assumptions are needed regarding the underlying probability distribution of the uncertain data [11]. It is also true when comparing the robust optimization approach with the fuzzy approach because there is no need for RO to define membership function for the uncertain parameter.

On the other hand, in this paper, the multi project scheduling problem is investigated which requires time, cost, resources, etc. in a large-scale quantity. So, it seems that a conservative approach is essential which can immune the project scheduling problem against data uncertainty. It is exactly the characteristics of the robust optimization approach which is applied for dealing with uncertainty in this paper.

In this paper, the robust optimization approach is applied for multi-project scheduling problem under resource constraint and uncertain activities duration to cover some shortcoming in the existing multi-project models. The problem is represented in a two-stage model in which, the objective function is to minimize the maximum total weighted tardiness of the projects.

The structure of the paper is as follows: section 2 describes the related literature review. The problem definitions are in detail in section 3. The proposed mathematical model and the two-stage approach is explained in detail in section 4. Section 5 describes one simple numerical example with its results to clarify the proposed model. Computational experiments are explained in section 6. Finally, the conclusion and further research come in section 7.

2. Related Works

The related works about the multi-project scheduling problem, the resource management policies and the project scheduling problem under uncertainty is mentioned in this section briefly.

2.1. Multi-Project Scheduling Problem

The RCMPSP comes from practical multi-project environments in which a number of projects concurrently share limited resources in precedence or other constraints [12]. In fact, the single project management rarely occurs today and the companies usually manage more than one project simultaneously titled “multi project management”. The importance of multi-project management has increased over the last decades and is still growing. In the middle of the last century, project and multi-project management gained momentum; the share of project work has increased since then and the penetration of firms by corresponding management methods has not stopped at the beginning of this century [13]. The researchers concur that the literature of project management problem is heavily biased towards the single project environment while the studies related to the multi-project problem is little [14].

The main reason of not much fruits on the topic of multi-project scheduling in compare with the single project one, comes from its high complexity, which is affected by many factors, such as the huge solution space, the intensely contending for resources, various and conflicting objectives, the inter-project dependence and priority, the high level of uncertainty and so on [12].
Therefore, many researchers have studied recently the multi-project problem to overcome this identified gap [15-18]. Also, some heuristic priority rules and metaheuristics has been studied to solve the Resource Constrained Multi-Project Scheduling Problem (RCMPS) [19-22].

2.2. Resource Sharing Policy

In the literature of multi project problem, the primary topic is the allocation of common resources between simultaneous projects since the resource-based relations define the multi project problem by joining the individual projects together. The characteristics of resource usage by individual project in the multi project environment is described as resource management policy [4]. In the multi project problem, there are several projects which are executed in parallel and they use the common resource pool at least for one resource type. There are several approaches to optimally allocating the resources to the activities of multi projects, such as the resource sharing policy, the resource dedication policy and etc. [16, 17, 19]. Regarding the different existing policy, in this paper, the most common one, i.e. resource sharing policy is applied to determine how to allocate the common resources among projects.

2.3. The Project Scheduling under Uncertainty

There are many studies in which deterministic environment is considered for the project scheduling problem [19, 15, 16, 18, 23]. But, in the real world, the uncertainty exists during the project execution. In order to consider the uncertainty in problem modeling, different assumption can be applied. In some researches, the costs of activities are considered uncertain [24, 25] while studying the project scheduling problem. However, the most often objective function in project scheduling problem is the optimization of the project duration [9]. So, the duration of activities which have direct influence on the makespan of project, is studied as uncertain parameter in the following studies.

2.3.1. Stochastic Project Scheduling Problem

The stochastic RCPSP or SRCPSP is the optimization problem that results when the deterministic durations in RCPSP are replaced by stochastic variables. While in the classic RCPSP the goal is to find a schedule with minimum schedule length or makespan, the goal in SRCPSP is to minimize the expected makespan [26]. For more information, please refer to many studies which apply stochastic approach to consider uncertainty in project scheduling problem [27- 29]. The serious challenging point for stochastic RCPSP is that in according to the main characteristic of the project; uniqueness, there are difficulties accessing to enough historical data to fit a probability distribution for an uncertain parameter. So, applying the stochastic approach to the project scheduling problem has limitation from the practical point of view.

2.3.2. Fuzzy Project Scheduling Problem

Fuzzy project scheduling approach is based on the concept of fuzzy activity duration, produces fuzzy schedules and requires the membership function of the uncertain activity duration [30]. In this approach, the duration of the activities is estimated by experts and the project manager deals with the imprecise and vague judgements. For more information about fuzzy RCPSP please refer to [31- 34]. So, similar to the determination of the distribution function for
activities duration in stochastic approach, there are some challenges for project managers to determine the membership function for fuzzy activity durations. Thus, fitting distribution function with its parameters or defining fuzzy membership function for the activities duration have challenges from a practical point of view. In other words, this can seriously limit the application of these two approaches for the project scheduling problem.

2.3.3. Robust Project Scheduling Problem

The robust optimization approach can immune the project scheduling problem against uncertainty. There are only three studies applying this approach for the RCPSP with uncertain duration in the single project problem which are mentioned in the following. Chakrabortty et al. [35] study the RCPSP in which the activity durations are represented by random variables with different probability distribution functions. They propose robust optimization-based approach which produces reasonably good solutions under any likely input data scenario. Their proposed approach guarantees the feasibility of solutions and produces high-quality solutions. Bruni et al. [7] propose an adaptive robust optimization model to derive the resource allocation decisions that minimize the worst-case makespan, under general polyhedral uncertainty sets assuming that the activity durations are subject to interval uncertainty. Also, a general decomposition approach is proposed by them to solve the robust counterpart of the RCPSP, further tailored to address the uncertainty set with the protection factor. Artigues et al. [36] propose models for project scheduling when there is considerable uncertainty in the activity durations. They develop and implement a scenario-relaxation algorithm and a scenario-relaxation-based heuristic. The first algorithm produces optimal solutions but requires excessive running times even for medium-sized instances; the second algorithm produces high-quality solutions for medium-sized instances and outperforms two benchmark heuristics.

The above mentioned studies have been done in the area of single project scheduling problem. According to the large scale of multi-project scheduling problem, the effect of uncertainty can be more destructive. In the multi-project scheduling problem, some projects are related to each other by the common resources and the investment of time, cost, resources and etc. are in large scale. So, applying the robust optimization approach, as a more conservative approach, which can immune the problem against uncertainty is totally necessary. To the best of our knowledge, there is no research for applying the robust optimization approach in the multi-project scheduling area. In the present paper, the robust optimization approach is applied for the multi-project scheduling problem under resource constraint and uncertain duration of activities. In this research, the resource sharing policy is considered. Each project has a determined due date. Also, the importance weight of the projects is different. The aim is to obtain an optimized structure for all of the projects in such a way that the maximum total weighted tardiness of the projects will be minimum. In this study, the development of the existing models can be demonstrated in two ways according to the Fig. 1.

Please insert Fig. 1 approximately here

3. Problem Statement

The resource constrained multi project scheduling problem with uncertain activity durations is studied in this paper. The considered multi project problem contains defined projects, $G = 1, 2, ..., q$. All of the projects are shown by activity-on-node (AON) network Graph $= (V, E)$ in
which the nodes demonstrate the activities of projects and the arcs represent the precedence
relations between activities, $E$. The set of activities for each project is indicated by
$V=\{0,1,...,n+1\}$. For each activity $i \in V$ of the project $g$, there is a set $P_{ig} \subset \mathbb{R}_+$ containing
the possible values for the duration of activity $i$ of project $g$ ($\mathbb{R}_+$ is the set of non-negative real
numbers). Therefore, in the discrete set of $P_{ig} = \{p_{ig1}, p_{ig2}, p_{ig3}, ..., p_{ig[p_i]}\}$, the minimum and
maximum durations for activity $i$ of project $g$ are $P_{ig}^{\min} = \min_{p_{ig}} p_{ig}$ and $P_{ig}^{\max} = \max_{p_{ig}} p_{ig}$
respectively. The durations of activities 0 and $n+1$ are considered zero: $P_{0g} = P_{n+1,g} = \{0\}$, $\forall g$. It
is noticeable that when $p_{ig} \in P_{ig}$, then $p_g = (p_{0g}, p_{1g}, ..., p_{n+1,g})$ shows one possible
scenario for the activities duration of project $g$. When $|P_{ig}| = 1$, $\forall i \in V$, $\forall g \in G$, the problem
converts to deterministic RCPSP.

As mentioned before, the precedence relationship between activities is shown by the binary
relation of $E \subset V \times V$. The activity $i$ of project $g$ can be started after all its predecessors are
finished. The projects apply resource sharing policy. It means they utilize common resources
from the resource pool. There are $b_{igk} \in \mathbb{N}$ units of resource $k$ required by activity $i \in V$ of
project $g$ during its execution. In each project, the required resources in any type for dummy
activities of 0 and $n+1$ are zero: $b_{0g} = b_{n+1,g,k} = 0$, $\forall g \in G$, $\forall k \in R$.

A set of activities $F \subset V$ is one “Forbidden Set” of a precedence relation $A$ if it is an anti-
chain of $A$ and at least for one type of resource $k \in R$ : $\sum_{i \in F} b_{ig} > b_k$. So, these sets can give rise to
resource conflicts during project execution. A subset-minimal forbidden set is called a “Minimal
Forbidden Set” or $mfs$s. The set of $mfs$s for precedence relation $A$ is written as $F(A)$ [36]. Any of
the resource conflicts can be removed by adding extra precedence relationships to the primary
precedence graph for postponing some activities in such a way that the makespan can be
determined by applying an early start policy (ES-Policy) on an extended graph. So, the extra
precedence relationships $X \subseteq (V \times V) \setminus E$ should be found in such a way that the extended graph
$Graph'(V, (E \cup X))$ is acyclic and $F(T(E \cup X)) = \emptyset$ [7]. According to Balas [37] the set $X$, containing
pairs of activities that leads to one feasible ES-policy, can be called a sufficient
selection. After defining one selection and adding the extra precedence relationship $X$ to the
primary precedence graph $E$, the resource constraints can be ignored according to the precedence
relation in the $EUX$ and the makespan can be obtained by calculating the critical path
problem on the extended graph $Graph'(V, (E \cup X))$ [7].

The binary decision variable $x_{igk'}$ is introduced in this paper to show the precedence
relationship between the activities. According to the characteristics of the multi project
scheduling problem, one activity and its predecessor activities are not essentially within the same
project and it can be possible that one activity becomes the predecessor of another activity from
different project. So, the precedence relationship between two projects are introduced in this
paper based on two reasons. The first reason is that, in many real world multi project scheduling
problem, the precedence relationship exists between the activities of two projects. For example,
consider two projects in an area with low population density: (1) construction of the residential
complex and (2) installation of the town gas station. In this example, the high pressure
equipment installation activity in the second project is the predecessor of the installation and

6
testing the town gas system of the residential complex in the first project. In the cases with no precedence relationship between two projects, the special case may happen with $g=g'$ in the $x_{igg'}$ notation.

The second reason is relevance to the applied calculation method. According to the Minimal Forbidden Set, any of the resource conflicts can be removed by adding extra precedence relationships $(X)$ to postpone some activities. Based on the resource sharing policy in the multi-project scheduling problem, the activities of different projects utilize the common resources from the resource pool. So, the extra precedence relationships $(X)$ can be also created between two activities from different projects. Therefore, the variable $x_{igg'}$ should presents both of the projects between which the extra precedence relationship $(X)$ exists.

In the $Graph'(V, (E \cup X))$, the start time of activity $i$ of project $g$; $s_{i,g}(X, p)$ is the longest path from the scheduling time horizon $0$ to activity $i$ of project $g$. So, one should check the paths originated from the start activities of all projects (not only the start activity of the project $g$ containing $i$) while calculating the $s_{i,g}(X, p)$.

The resource flows between the activities are demonstrated by the transshipment networks [36] which can be called as (resource) flow network. The number of resource type $k$ transferring from the end of activity $i$ of project $g$ to the start of activity $j$ of project $g'$ is represented by flow $f(i, g, j, g', k) = f_{igjg'k} \in \mathbb{N}$. It is noticeable that for each resource type, a separate flow network will be created. The resource flow should satisfy the conservation constraints and also the lower and upper bounds on the flow for intermediate (not start or end) nodes [36].

There can be several selections for the same schedule. Bruni et al. [7] presented a numerical example for two different selections in the project with 5 activities. They illustrated that when the activities durations are deterministic, the project makespan would be the same for two different selections. But according to the uncertainty condition, when the delays of the activities are also considered, the different selections cause different makespan. This example shows the importance of proper resource allocation policy under uncertainty. They also declare that in some cases especially in the multi-project scheduling problem the resources cannot easily transfer between the activities, so the decisions about resource transfers should be taken with more attention and sensitivity. As it mentioned before, to the best of our knowledge there is no research investigating this problem in the multi-project scheduling environment.

In this paper, the Robust Resource-Constrained Multi-Project Scheduling Problem (RRCMPSP) is studied as a two-stage robust optimization model. In this study, some projects are considered as a multi-project problem and should be scheduled while the duration of activities are not certain. For each project, a due date $DD_g$ is determined by the global project manager which is defined as a deadline for finishing each of the projects and is notified to the local project managers. The aim is to minimize the deviation of each project makespan from its due date while the required resources are in common and the activities duration are uncertain. It is noticeable that in the multi-project problem, the cost of deviating from due date is not equal for different projects. So, the degree of priority and importance of project $g$ which is demonstrated by $w_g$ as its weight, should be considered in the calculation in such a way that: $\sum_g w_g = 1$. The question is how to allocate and share the common resources between different activities in such a way that while the activities durations are uncertain, the maximum weighted tardiness for all projects; shaping multi-project, will be minimized. Therefore, we search for one sufficient selection
containing all projects in which the maximum weighted difference between the projects makespans and their due dates is minimized.

4. Mathematical Modeling of the Problem

The Resource-Constrained Multi-Project Scheduling Problem under uncertain duration of activities is formulated as a two-stage robust optimization model. In the following, the notation of indices, parameters and variables used in the proposed models is represented.

4.1. The Notations

The list of notations applied in the proposed models are as follows.

- **Indices**
  
  \( G \)  The set of projects in the multi-project problem
  
  \( V \)  The set of activity nodes
  
  \( R \)  The set of renewable resources
  
  \( E \)  The set of precedence relations between activities
  
  \( P \)  The set of scenarios belonging to activities duration

- **Parameters**
  
  \( w_g \)  The weight (priority degree) of project \( g \)
  
  \( DD_g \)  The due date of project \( g \)
  
  \( P^h_{ig} \)  The duration of activity \( i \) in project \( g \) under scenario \( h \)
  
  \( b_{igk} \)  The required resource type \( k \) for performing activity \( i \) of project \( g \)
  
  \( b_k \)  The capacity of resource type \( k \)
  
  \( P^\min_{ig} \)  The minimum scenario value for duration of activity \( i \) in project \( g \)
  
  \( P^\max_{ig} \)  The maximum scenario value for duration of activity \( i \) in project \( g \)

- **Variables**
  
  \( TTa^* \)  The total weighted tardiness of projects
  
  \( Ta_g \)  The tardiness of project \( g \)
  
  \( S^h_{ig} \)  The start time of activity \( i \) of project \( g \) under scenario \( h \)
  
  \( x_{ijg'} \)  The decision variable with value one when activity \( i \) of project \( g \) is the predecessor of activity \( j \) of project \( g' \) and otherwise it takes the value zero.
  
  \( f_{ijgk} \)  The number of resource units of type \( k \) that are transferred from the end of activity \( i \) of project \( g \) to the start of activity \( j \) of project \( g' \).
  
  \( a_{ig} \)  The decision variable with value one if the duration of activity \( i \) of project \( g \) takes the maximum value and it takes the value zero if the duration of activity \( i \) of project \( g \) takes the minimum value.
The longest path of project $g$ in the multi-project network

$\varphi_{g,igjg}^{\text{min}}$ The minimum and maximum flow belonging to project $g$ transferred from activity $i$ of project $g$ to activity $j$ of project $g'$ respectively.

$S_{i,g}$ The start time of activity $i$ belonging to project $g$.

### 4.2. The First Stage Model

\[
\min TTa^* = \sum_{g=1}^{G} w_g T_a_g
\]  

\[s.t.
\]

\[T_a_g \geq S_{h+1}^g - DD_g \quad , \forall g \in G, h = 1,...,|P| \]  

\[S_{h,jg}^g \geq S_{i,g}^h + P_{i,g}^h - M(1 - x_{igjg}) \quad , \forall (i,j) \in V \times V , \forall g, g' \in G \times G , i \neq j \text{ or } g \neq g', h = 1,...,|P| \]  

\[
\sum_{g'} \sum_{i \in V \setminus g} \sum_{g} f_{0igk} = b_k \quad , \forall k \in R
\]  

\[
\sum_{g} \sum_{j \in V \setminus g} \sum_{i \in 0} f_{jg+igk} = b_k \quad , \forall k \in R
\]  

\[
\sum_{g' \in G} \sum_{j \in V, j \neq i} \sum_{g \in G} f_{jg'igk} = b_{igk} \quad , \forall i \in V \setminus \{0, n+1\} , \forall k \in R , \forall g \in G
\]  

\[
\sum_{g' \in G} \sum_{j \in V, j \neq 0} \sum_{g \in G} f_{jg'igk} = b_{igk} \quad , \forall i \in V \setminus \{0, n+1\} , \forall k \in R , \forall g \in G
\]  

\[
f_{igjgk} \leq \min\{b_{igk}, b_{jigk}\}, x_{igjg'} \quad , \forall (i,j) \in V \times V , \forall (g,g') \in G \times G , \forall k \in R \quad , i,j \neq 0, n+1
\]  

\[x_{igjg'} = 1 \quad , \forall (i,g,j,g') \in E
\]  

\[S_{0g} = 0 \quad , \forall g \in G
\]  

\[T_a_g = 0 \quad , \forall g \in G
\]  

\[S_{h}^g = 0 \quad , \forall i \in V , \forall g \in G , \ h = 1,...,|P|
\]  

\[f_{igjgk} \geq 0 \quad , \forall (i,j) \in V \times V , \forall (g,g') \in G \times G , \forall k \in R
\]  

\[x_{igjg'} \in \{0,1\} \quad , \forall (i,j) \in V \times V , \forall (g,g') \in G \times G
\]  

The minimization of the total weighted tardiness of the projects is displayed in Eq. (1) as the objective function. The tardiness of each project is the difference between the project makespan and its determined due date and obtained by constraint (2). Constraint (3) demonstrates the precedence relationships between the activities where $M$ is a big number. So based on this constraint, the successor activity $j$ cannot start earlier than the finish time of its predecessors under each scenario. The sum of resource flows type $k$ sending from dummy start nodes 0 is equal to the available capacity of resource type $k$ which is mentioned in Eq. (4). In addition,
based on Eq. (5), the sum of resource flows type \( k \) sending from the activities of all projects to the dummy finish nodes \( n+1 \) of projects is equal to the available capacity of resource type \( k \).

The sum of incoming resource flows type \( k \) from other activities to activity \( i \) of project \( g \) is equal to the required resource type \( k \) for performing the activity \( i \) of project \( g \) which is described in Eq. (6). Similarly, Eq. (7) ensures that the sum of resource flows type \( k \) exiting from activity \( i \) of project \( g \) to other activities is equal to the required resource type \( k \) for executing the activity \( i \) of project \( g \). Constraint (8) ensures that the resource flow type \( k \) transferring from activity \( i \) of project \( g \) to the activity \( j \) of project \( g' \) is utmost equal to the minimum value of \( \{b_{igk}, b_{jgk}\} \). In addition, this equation prevents resource transferring between two activities that there is no precedence relationship between them.

According to Eq. (9), the binary variable \( x \) is equal to 1 for the two activities with precedence relationship between them. The start time of (dummy) activities 0 for all projects is zero (the start point of scheduling horizon) and demonstrated in Eq. (10). Constraint (11) mentions that the tardiness of projects cannot be negative. Constraints (12) and (13) introduce the nonnegative decision variables of the start time of activities and the resource flow between the activities, respectively. At last, the binary variable \( x \) is presented in Eq. (14).

In this stage, the best structure \( EUX \) is obtained for the existing scenarios regarding the precedence relationships and resource requirements. This structure is the output of the first stage model and is needed as an input for the second stage model. In fact, this structure is achieved while the total weighted tardiness of projects as an objective function is minimized.

### 4.3. The Second Stage Model

\[
\max TT^* = \sum_{g=1}^{G} w_g T_a_g
\]

\( s.t. \)

\[
T_a_g \leq (LP_g - DD_g)
\]

\[
LP_g = \sum_{(i,g,j,g') \in EUX} (p_{ig} \cdot \phi^m_{g' \to ig} + p_{ig} \cdot \phi^m_{g' \to ig})
\]

\[
\sum_{(i,g,j,g') \in EUX} \phi^m_{g' \to ig} \leq a_{ig}
\]

\[
\sum_{(i,g,j,g') \in EUX} \phi^m_{g' \to ig} \leq 1 - a_{ig}
\]

\[
\sum_{(i,g,j,g') \in EUX} (\phi^m_{g' \to ig} + \phi^m_{g' \to ig}) = 1
\]

\[
\sum_{(i,g,j,g') \in EUX} (\phi^m_{g' \to ig} + \phi^m_{g' \to ig}) = 1
\]

\[
\phi^m_{g' \to ig} = 0
\]

\[
\phi^m_{g' \to ig} = 0
\]

\[
T_a_g \geq 0
\]
\[ \varphi_{g_{ijg'}}^{\min} \geq 0, \forall (i, g, j, g') \in EUX, \forall g'' \in G \]  
\[ \varphi_{g_{ijg'}}^{\max} \geq 0, \forall (i, g, j, g') \in EUX, \forall g'' \in G \]  
\[ x_{ijg'} \in \{0, 1\} \]  
\[ x_{ijg'} = 1, \forall (i, j) \in V \times V, \forall g, g' \in G \times G \]  
\[ a_{ig} \in \{0, 1\} \]  
\[ a_{0g} = a_{n+1g} = 0 \]  
\[ S_{0g} = 0 \]

In the second stage model, the worst scenario should be found in such a way that the total weighted tardiness of the projects become maximized as represented in Eq. (15). Eq. (16) shows how to obtain the projects tardiness. In this equation, the finish time of each project is obtained by the longest path (LP) method in the overall network of the projects which is demonstrated in Eq. (17). In the single project problem, the longest path can be obtained by \( \sum_{(i,j) \in UX} (p_i, \varphi_{ij}) \) where \( p_i \) is the duration of activity \( i \) and \( \varphi_{ij} \) is the transferring flow from activity \( i \) to activity \( j \). The multiplication of \( p_i \) and \( \varphi_{ij} \) leads to the nonlinearity of this formula. The binary variable \( a_i \) is introduced to linearize the formula and converted it to \( \sum_{(i,j) \in UX} (p_i^{\min} \cdot \varphi_{ij}^{\min} + p_i^{\max} \cdot \varphi_{ij}^{\max}) \) in which the \( p_i^{\min} \) and \( p_i^{\max} \) are the minimum and maximum values of the duration belonging to activity \( i \), respectively. For detailed information about calculating the longest path of “single project” and how to linearize it, please refer to Artigues et al. [36].

In the multi-project scheduling problem, the \( EUX \) is an overall structure for all projects including the primary precedence relationships between activities (\( E \)) and the extra precedence relationships caused by resource constraint (\( X \)). So, in the studied problem, the projects are interrelated to each other in this structure. Thus, for obtaining the longest paths of the projects, a flow per project should be sent from the 0 activities to other activities in the overall structure which is demonstrated by \( \varphi_{g\gamma_{ijg'}} \). Worth to mention that the first index \( (g'') \) in the decision variable \( \varphi_{g\gamma_{ijg'}} \), shows the project for which we want to calculate the longest path.

In order to linearize the longest path formula, constraints (18) and (19) are created in which the binary variable \( a_{ig} \) takes the value 0 when the duration value of activity \( i \) of project \( g \) is minimum and so \( \varphi_{g_{ijg'}}^{\max} = 0 \). On the other hand, the \( a_{ig} \) takes the value 1 showing that the duration value of activity \( i \) of project \( g \) is maximum and so \( \varphi_{g_{ijg'}}^{\min} = 0 \).

As mentioned before, the predecessor of one activity can be the activity within the same project or from the other projects. So, the longest path of one project is not necessarily originated from the activity 0 of that project and it can be also started from the 0 activity of other projects. According to Eq. (20), the summation of flows calculating the longest path of project \( g'' \) originated from start nodes 0 of all projects to the overall structure should equal to 1. Besides,
Eq. (21) implies that the flow calculating the longest path of project \( g \) should end in the node \( n+1 \) of project \( g \). Eq. (22) and Eq. (23) ensure that the flow calculating the longest path of one project cannot enter to the end node \( n+1 \) of other projects. For each flow, the conservation law should be satisfied; i.e. the sum of flows entering to the activity \( i \) of project \( g \) should be equal to the sum of flows exiting from the activity \( i \) of project \( g \). This law is presented in Eq. (24).

The constraint (25) introduces the nonnegative variable of the projects tardiness. The flows related to the longest path calculations are represented in Eqs. (26) and (27). The binary variable \( x \) is defined in Eq. (28) while it should take the value 1 for the activities with precedence relationship between them, stated in Eq. (29). The binary variable \( a \) is described in Eq. (30). Equation (31) represents that for all start nodes 0, the variable \( a \) takes the value 0. Finally, the start time of the projects are set at time 0 as shown in Eq. (32).

4.4. The Two-Stage Exact Approach

The scenario relaxation algorithm is an iterative optimization algorithm which generates optimal robust decisions for the deviation and relative robust objectives. The key insight of the scenario relaxation algorithm is that in a problem with a large number of possible scenarios, only a small subset of scenarios actually has to be explicitly examined when searching for the optimal deviation (or relative) robust solution. For more information about the scenario relaxation algorithm, please refer to [38].

In this paper, the objective function is to minimize the maximum total weighted tardiness of the multi project problem under uncertain activities duration. A two-stage model is presented for the RRCMPSP in parts 4.1 and 4.2. Based on the mentioned modeling, the set of decision variables can be divided into two groups. The first one is the variables related to the sufficient selection decisions \( X \in \mathcal{X} \). The second group variables are related to calculation of the longest paths in the structure obtained by the first stage model. According to the Benders alphabet, the first stage model is corresponded to “Master problem” and the second stage model is similar in spirit to the “Sub problem”.

According to Artigues et al. [36] a duration scenario \( p \) is extreme if: \( p_i = p_{i}^{\text{min}} \) or \( p_i = p_{i}^{\text{max}} \) for all \( i \in \mathcal{V} \). They also proved that there is always an extreme duration scenario for which the maximum absolute regret of an ES-policy \( X \) is reached. Therefore, in the worst case for the studied problem in this paper, the number of algorithm iteration can be \( |P| = 2^\nu \) where \( \nu \) is the number of activities belonging to all projects.

In this approach, the scenarios are gradually added to the problem structure in the sequential iterations. First, one scenario of activity durations is considered (any arbitrary number of scenarios can be considered) and the first stage model is solved. The aim is to obtain the structure \( E \cup X \) for which the total weighted differences between the projects makespan and their due dates is minimized. In other words, considering the existing scenario, we search for an optimized \( E \cup X \) with minimum total weighted tardiness of the projects. In the next step, the second stage model results the worst scenario for the obtained structure of the first stage model in such a way that the objective (total weighted tardiness of the projects) will be maximized. Then the mentioned scenario should be added to the scenario set of the first stage model. This algorithm continues until the objective functions of the both stages become equal. Other words, the algorithm terminates when the minimum weighted tardiness of the optimized structure for the existing scenarios is equal to the maximum weighted tardiness of the worst scenario for the assigned structure.

12
The steps of the applied approach are described in the following where \( \text{iter} \) is the counter of algorithm iterations.

**Step 1 (Preliminary)** the set \( \hat{P}_1 \) containing only one scenario \( p^1 \) for the duration of all activities of the projects is considered. Also \( \text{iter} = 1, \; LB = 0 \) and \( UB = +\infty \) are assumed.

**Step 2 (First stage Model)** the model (1) - (14) is solved in order to obtain \( LB = TTa^* (\hat{P}_{\text{iter}}) \). Also, the corresponding \( ES\text{-policy}; \; X_{\text{iter}} \) is resulted.

**Step 3 (Second stage Model)** the model (15) - (32) is solved and the maximum \( TTa^{\max} (X_{\text{iter}}) \) for \( X_{\text{iter}} \) is obtained. The corresponding worst scenario; \( p^{{iter}+1} \) is resulted. In addition, the \( UB = TTa^{\max} (X_{\text{iter}}) \) is considered.

**Step 4 (Optimality investigation)** when \( LB = UB \) , then stop the algorithm. If \( LB \neq UB \), then \( \text{iter} = \text{iter} + 1, \; \hat{P}_{\text{iter}} = \hat{P}_{\text{iter} - 1} \cup \{ p^\text{iter} \} \) and the algorithm should continue from step 2.

5. **Numerical Example**

In this section, one simple example is presented to illustrate the application of the mentioned approach for multi-project problem. Consider a multi-project problem which consists of three projects. Each project has only four activities (the start activities and end activities are dummy) as shown in Fig. 2. There is only one renewable resource with \( (b_1 = 7) \). The required resource for performing each activity, the possible durations of activities, the determined due date of projects and the importance weight of the projects are all represented in Table 1. Both of the first stage and the second stage models are coded in GAMS v24.1.2 and solved by the “CPLEX” solver.

The \( EUX1 \) is obtained after solving the first stage model in the first iteration. According to this structure, the total weighted tardiness of the projects according to the first scenario will be minimized. In the first scenario, all the activities durations are considered at their minimum values (Table 2). To avoid untidiness caused by too many arcs, the representation of the whole \( EUX \) s are neglected in each iteration. The longest paths of the projects according to the first scenario are calculated and shown in Fig. 3.a.

For the given \( EUX1 \) from the first stage model, the second stage model should be solved. The maximum total weighted tardiness for \( EUX1 \) are resulted by finding the worst scenario which is presented in Table 3.
Fig. 3.b shows the longest paths of the projects according to the worst scenario (demonstration of the longest paths based on the first scenario is ignored) resulted from second stage model in the first iteration.

The first stage model should be solved regarding two scenarios for the activities durations in the second iteration. The total weighted tardiness of the projects should be minimized regarding these two scenarios. So, the optimized EUX2 structure is obtained which will be the input for the second stage model. The longest paths of the projects only for the second scenario are depicted in Fig. 4.a.

After that, the second stage model is solved while the objective is to maximize the total weighted tardiness of projects. In fact, for the given EUX2, the worst scenario should be achieved which is demonstrated in Table 4. The longest paths of the projects only based on the worst scenario is shown in Fig. 4.b.

In the third iteration, the first stage model is solved regarding three scenarios. The longest paths of projects only for the third scenario are shown in Fig. 5.a.

Obtaining the EUX3, the second stage model can be solved. The resulted worst scenario and the longest paths of the projects are depicted in Table 5 and Fig. 5.b. respectively.

By comparing the objective functions of the first stage model and the second stage model in the third iteration, it is realized that the algorithm should be stopped while both objective functions attain to the same value, 5.9 in this simple example. The results of each iteration are represented in Table 6 in summary.

After three iterations, the optimized value for the objective function of this example has obtained. This value is resulted by the best structure according to the resource-constraint and precedence relationships in which the total weighted tardiness of the projects is minimum. With regard to existing uncertainty in the activities durations, the obtained result is robust. In fact, it ensures that if any scenarios happens for the activities durations (in this example we have $3^4 \times 2^2 = 324$ possible scenarios occurrence), the total weighted tardiness of the projects will not be greater than 5.9. This is exactly the characteristic of the robust optimization method which immunes the problem from uncertainty and keeps the result feasible and near optimal.

6. Computational Experiments
Both of the first stage and second stage models are coded in GAMS v24.1.2 and solved with the CPLEX solver. The experiments were run on a personal computer with an Intel(R) Xeon(R) CPU E7-8890 v4 @ 2.20 GHz 2.19 GHz (2 processors) and 42 GB RAM under Windows 10 operation system.

6.1. The Test Problems

In this paper in order to generate the test problems, the software RanGen [39] is applied for deterministic RCPSP. For adapting the test problems to RRCMPSP, the required additional data are considered. Also, the number of activities can be chosen. In this research the number of activities, n=30 is considered for each project in the multi-project problem. Applying this software provides us instances with different values of the parameters related to the structure of the projects. The considered parameters are: order strength, resource factor and resource constrainedness which are explained briefly in the following.

**Order Strength (OS):** The number of precedence relations divided by the theoretical maximum number of precedence relations in the network. The minimum value for OS is 0 (in the parallel network) and the maximum value for OS is 1 (in the serial network case). So, it can take values from 0 to 1. In this research, OS can be chosen from two values {0.4, 0.7}.

**Resource Factor (RF):** How many different resources used on average by the activities are determined by this factor. The minimum value for RF is 0 (no resource requirements for executing the activities) and the maximum value for RF is 1 (when all the activities require all kinds of resources). So, it can take values from 0 to 1. In this research, RF chooses value from the set {0.25, 0.5, 0.75}.

**Resource Constrainedness (RC):** (per resource type) This factor can be obtained by the Eq. (33) [40]:

\[
RC_k = \frac{DMND_k}{R_k}, \text{ for all } k \in R
\]  

(33)

Where: \( R_k \) is the capacity of resource type \( k \) and \( DMND_k \) is the average quantity of resource type \( k \) demanded when required by an activity and can be calculated by Eq. (34):

\[
DMND_k = \sum_{N} \left\{ \begin{array}{ll} 1 & \text{if } r_{ijk} > 0 \\ 0 & \text{if } r_{ijk} = 0 \end{array} \right\}, \text{ for all } k \in R
\]  

(34)

In which \( r_{ijk} \) is per-period requirement of resource type \( k \) by activity \( j \) of project \( i \) and \( N \) is the set of all activities to be scheduled. In this research, RC chooses value from the set \{0.3, 0.6\}. For each combination of OS, RF and RC, five instances of the RRCMPSP is considered. Each multi project problem is assumed containing three projects. So,

\[
2(OS) \times 3(RF) \times 2(RC) \times 5(\text{per combination examples}) \times 3(\text{number of project in each multi project problem}) = 180
\]

instances are needed to be randomly generated by RanGen.

6.2. Computational Experiments
There are 12 classes of problems with respect to the different values of factors; $OS$, $RF$ and $RC$. The average execution time per iteration for both of the first stage and the second stage models are calculated. In addition, the average number of iterations are recorded. Table 7 represents the computational results for the test problems.

Please insert Table. 7 approximately here

Figs. 6 to 8 illustrate the sensitivity analysis of the obtained results for different levels of the factors $RC$, $RF$ and $OS$ respectively. As it is shown by these figures, the behavior of the solution approach is strongly related to the instances and their characteristics.

Please insert Fig. 6 approximately here

There are 6 classes of problems based on the different values of $OS$ and $RF$, in which the effect of $RC$ factor should be examined. As shown in Fig. 6, the most effective factor is $RC$ which impacts strongly on the performance of the applied approach. It is mentionable that the linear histogram is fitted just for showing the effect of the factors value on the results schematically. According to these 6 experiments, the computational time grows rapidly according to the higher value of $RC$. In other words, when the value of $RC$ increases, the instances become harder to solve and the approach needs more time for execution. Worth to mention that this impact is mainly on the performance of the first stage model in which the extra precedence relationships $(X)$ should be resulted (according to the resource constraint). So, $RC$ which is mainly related to the resource constraint has strong influence on the performance of the first stage model and the second stage model does not take influence significantly by this factor.

Please insert Fig. 7 approximately here

The next effective factor is $RF$ which is also related to the resource constraint and has an observable influence on the first stage model. When $RF$ increases, the first stage model becomes harder to solve and consequently consumes more time. So, $RF$ is a second effective factor which influences on the obtained results. As demonstrated by Fig. 7, a sensible increase in the computational time happens by increasing the $RF$.

Please insert Fig. 8 approximately here

The $OS$ factor has the least influence on the computational time of the obtained results as depicted in Fig. 8. In most cases, there are a decrease for the computational time when the value of factor $OS$ changes (in some cases there is not significant changes). With respect to the definition of $OS$ factor, when $OS$ increases, the structure of the projects moves from the parallel structure to the serial structure. So, the problem becomes easier according to the resource constraints and it is expected that the running time of the algorithm decreases significantly. But, why does it not happen? The reason is related to the extra precedence relationship $X$ which is added to the $E$ set to remove the resource conflict. Other words; the $E$ set (based on the $OS$ factor) is not the only parameter which has influence on the physical structure of the project network and $EUX$ is the final structure of the network.

7. **Conclusion and Further Research**
The robust resource-constrained multi-project scheduling problem (RRCMPSP) was studied in this paper in which the objective function was to minimize the maximum total weighted tardiness of the projects. The duration of the activities belonging to the projects were uncertain and defined with discrete values called scenarios. The resource sharing policy was applied in this study for resource allocation in the multi project problem. Also, there was a deadline for each project determined by the global project manager and each project had its own weight of importance which dictates to which project there should be more consideration. For obtaining exact results, a scenario-relaxation algorithm was applied and implemented for the proposed robust multi-project scheduling problem. Then, the computational results were discussed. It was resulted obviously that the factor $RC$ had more influence on the behavior of the solution approach which is an important factor especially in the multi-project problems.

Some extensions of this research as a future study might be of interest. While the limitation of this study is that the presented exact solution method is not able to solve large-size problems at reasonable time, developing the heuristic and metaheuristic algorithms is suggested to solve the large size RRCMPSP. As another extension, considering uncertainty in the resources availability and its effect in managing the multi-project problems would be of interest. Also, some constraints can be added to this model like the multi-mode activities, nonrenewable resources, multi-skill resources, etc. while the other objective functions like minimum cost or maximum quality is considered.

References


**Elham Nabipoor Afruzi** is currently a PhD candidate in Industrial Engineering in K. N. Toosi University of Technology (KNTU), Tehran, Iran. She obtained B.S. and M.S. degrees in Industrial Engineering both from KNTU in 2010 and 2012, respectively. Her research interests include: Project Scheduling and Management, Uncertainty Modeling, Exact and Meta-Heuristic Algorithms.

**Abdollah Aghaie** is a professor of Industrial Engineering at K. N. Toosi University of Technology in Tehran, Iran. He received his BSc from Sharif University of Technology in Tehran, MSc from New South Wales University in Sydney in Australia and PhD from Loughborough University in the U.K. His main research interests are in Modeling and Simulation, Queuing System, Quality Management and Control, Knowledge Management and Stochastic Process.

**Amir Abbas Najafi** is currently an Associate Professor of Industrial Engineering at K. N. Toosi University of Technology. He received his B.S. degree in Industrial Engineering from Isfahan University of Technology in 1996, and his M.S. and Ph.D. degrees in Industrial Engineering from Sharif University of Technology in 1998 and 2005, respectively. His research interests include Project Scheduling and Management, Portfolio Selection Models and Applied Operations Research.

**List of Figures Captions**

Fig. 1. The Model Development
Fig. 2. Multi-Project Network

Fig. 3.a. First-Iteration/Level 1. Obj: 1.3, Ta₁:2, Ta₂:1, Ta₃:1

Fig. 3.b. First-Iteration/Level 2. Obj: 5.9, Ta₁:8, Ta₂:5, Ta₃:5, aᵢⱼ=1 for {2.1, 3.2, 3.3}

Fig. 4.a. Second-Iteration/Level 1 (Second Scenario). Obj: 5, Ta₁:8, Ta₂:5, Ta₃:2

Fig. 4.b. Second-Iteration/Level 2: Obj: 7.1, Ta₁:11, Ta₂:8, Ta₃:2, aᵢⱼ=1 for {2.1, 2.2, 2.3, 3.2, 3.3}

Fig. 5.a. Third-Iteration/Level 1 (Third Scenario). Obj: 5.9, Ta₁:8, Ta₂:5, Ta₃:5

Fig. 5.b. Third-Iteration/Level 2: Obj: 5.9, Ta₁:8, Ta₂:5, Ta₃:5, aᵢⱼ=1 for {2.1, 3.2, 3.3}

Fig. 6. The Sensitivity Analysis of Factor RC

Fig. 7. The Sensitivity Analysis of Factor RF

Fig. 8. The Sensitivity Analysis of Factor OS

**List of Tables Captions**

Table 1. The Multi-Project Required Data

Table 2. First Scenario Values for Activities Duration

Table 3. The Worst Scenario obtained by Second Stage Model in the First Iteration

Table 4. The Worst Scenario obtained by Second Stage Model in the Second Iteration

Table 5. The Worst Scenario obtained by Second Stage Model in the Third Iteration

Table 6. The Summary of Results obtained from Iterations

Table 7. The Computational Results
Figures

**RCPSP**
(Resource-Constrained Project Scheduling Problem)

**RCMPSP**
(Resource-Constrained Multi-Project Scheduling Problem)

**RRCPSP**
(Robust Resource-Constrained Project Scheduling Problem)

**RRCMPSP**
(Robust Resource-Constrained Multi-Project Scheduling Problem)

---

**Fig. 1**

---

**Fig. 2**
Fig. 5.a

Fig. 5.b
Fig. 8

Average Total Execution Time vs. OS for different RF and RC values:
- RF=0.25, RC=0.3
- RF=0.25, RC=0.6
- RF=0.5, RC=0.3
- RF=0.5, RC=0.6
- RF=0.75, RC=0.3
- RF=0.75, RC=0.6
### Tables

#### Table. 1

<table>
<thead>
<tr>
<th>$i$</th>
<th>$g$</th>
<th>$b_{ig}$</th>
<th>$p_{ig}$</th>
<th>Due Date</th>
<th>Project Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>{0}</td>
<td>7</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>{4,5,6}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>{2,4}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>{0}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0</td>
<td>{0}</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>{3,4}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>5</td>
<td>{1,2,3}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>{0}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0</td>
<td>{0}</td>
<td>4</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2</td>
<td>{2,3,5}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>{4,5,6}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>{0}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Table. 2

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### Table. 3

<table>
<thead>
<tr>
<th>Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Table 4

<table>
<thead>
<tr>
<th>Project Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 5

<table>
<thead>
<tr>
<th>Project Activity</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 6

<table>
<thead>
<tr>
<th></th>
<th>Iter1</th>
<th>Iter2</th>
<th>Iter3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Projects</td>
<td>Projects</td>
<td>Projects</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td><strong>First Stage Model</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projects Tardiness</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>The total weighted tardiness</td>
<td>1.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Second Stage Model</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Projects Tardiness</td>
<td>8</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>The total weighted tardiness</td>
<td>5.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Optimality Check</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 7

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Time</th>
<th>Average Total Computational Time per Iteration</th>
<th>Average Number of Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OS</td>
<td>RF</td>
<td>RC</td>
</tr>
<tr>
<td>1</td>
<td>0.4</td>
<td>0.25</td>
<td>0.3</td>
</tr>
<tr>
<td>2</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.5</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.75</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>0.25</td>
<td>0.3</td>
</tr>
<tr>
<td>8</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.5</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.75</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>