# SYSTEMATIC APPROACH FOR MODIFYING PROJECT SCHEDULES DUE TO UNEXPECTED CHANGES

S M Anamul Haque

A thesis

in

The Department

of

Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science (Quality System Engineering)

Concordia University

Montréal, Québec, Canada

November 2012

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **S M Anamul Haque**

Entitled: **Systematic Approach for Modifying Project Schedules due to Unexpected Changes**

and submitted in partial fulfillment of the requirements for the degree of

**Master of Applied Science (Quality System Engineering)**

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

______________________________ Chair
Dr. Z. Tian

______________________________ Examiner
Dr. A. Schiffauerova

______________________________ Examiner
Dr. I. Contreras

______________________________ Supervisor
Dr. S. Li

Approved by ______________________________
Chair of Department or Graduate Program Director

______________________________
Dean, Faculty of Engineering and Computer Science

Date November 2012

# Abstract

Systematic Approach for Modifying Project Schedules due to Unexpected Changes

S M Anamul Haque

Project schedules are subject to change due to uncertain aspects, such as failure of machines, worker Absenteeism and turnovers, changes of scope, and reworks, etc. These changes may often result in project delays, cost overruns, quality defects and other negative impacts. In response to changes, project managers need to revise the schedule to minimize the impact of the changes. They usually revise the schedule by modifying allocation of resources and arrangement of tasks to cope with the changes. In general, it is extremely difficult to modify a schedule due to limited resources, extensive interaction among activities and resources besides the typical constraints. The problem remains in how to control and minimize the overall impact of changes by taking necessary corrective actions.

In the above context, we first introduce a standard model for task-resource allocation schedule that incorporates necessary relationships among tasks and resources, and possible constraints of the project. We then propose a reactive scheduling approach to modify the baseline schedule to address changes due to the absence of workers during project execution. To modify the schedule, we define three change options based on preemptive and non-preemptive resource reassignment strategies. When a change occurs, the reactive scheduling framework selects the best change option using systematic decision process by capturing the change scenario and assessing the change impact. The change impact is measured in view of the importance of absent worker, length of absence, and criticality of affected tasks. The objective of this approach is to limit the increasing of the project duration from initial deadline (i.e., delay) without changing too many task-resource assignments. Finally, an example application related to software development project is presented to illustrate the implementation and features of our proposed approach.

**Keyword:** Project scheduling, resource unavailability, change impact, reactive scheduling, delay, and re-organization.

# Acknowledgments

I would like to express sincere appreciation and gratitude to my supervisor, Dr. Simon Li, for his invaluable guidance, support, patience, and encouragement. His technical and editorial advice was essential to the completion of this thesis. I have learned numerous lessons and insights from him on the work of academic research.

I would also like to express my appreciation to all the faculty and people at Concordia University who contributed to my success one way or the other.

Finally, I would also like to express my sincerest gratitude to my family, whose love and support have made this work possible.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Background and Motivation

Project management is usually centered around a *baseline* or *predictive schedule*, which is used to plan, monitor, measure and control the project. The project schedule represents the detailed execution plan, which specifies start date, duration, completion date of activities, and the allocations of resources such as machines, materials, and workers to activities with respect to time (Schwalbe, 2010).

During project execution, projects are subject to considerable uncertainty that may lead to numerous schedule disruptions. These uncertainties may originate from a number of possible different sources or events: resource failure, workers absenteeism, scope change, activity's duration change, rework, etc. These uncertainties may cause the initial schedule to be no longer optimal or feasible. When the deviations between actual and initial schedule become noticeable, we say that the project schedule is *disrupted* or *changed*. A disrupted schedule incurs higher costs due to missed due dates which causes resource idleness, higher work-in-process inventory and increased schedule perturbation due to frequent updating. For small deviations, the initial schedule may still be followed with little adjustment. In more serious cases, the initial schedule will no longer be feasible with respect to the original objectives of the project. Some recovery actions may be necessary to overcome the adverse effects. However, the problems remain in how to measure the impact, when to react to the unexpected events, and which recovery method will be efficient to obtain a high-quality solution.

In general, it is extremely difficult to manage project changes due to the extensive interaction between activities, resources, and project stakeholders as well as the time and budget constraints. For instance, a delay in one activity may affect the schedule of all subsequent activities further causing disruptions in material supply, human resources, and possibly other projects. Moreover, it becomes more challenging when a project is executing with limited proprietary resources, i.e., without sharing resources of other projects or third parties. In such cases, the unexpected absence or sudden turnover of a worker is insensitive to schedule change. The potential consequence of this change may include interruption of work accomplishment, reduced productivity and thus causing delays in deliverables and cost overrun. Change management related to worker unavailability is emerging field in the labor intensive industry, such as software development and construction industries.

In literature, two important research tracks, proactive scheduling and reactive scheduling, are identified in the field of project scheduling in response to such unexpected changes or disruptions (Herroelen and Leus, 2005). The proactive scheduling procedure is used to build a robust baseline schedule that incorporates some protections (e.g., through time buffering or resource buffering) against possible disruption during schedule execution. However, no matter how much we try to protect the initial schedule against possible disruptions, we can never totally eliminate their occurrences due to random nature of some changes. Thus, project scheduling will always be subject to ongoing reactive process where changing circumstances continually force adjustment and revision of baseline schedule. That's why, both predictive and proactive schedule will always require a *reactive* scheduling procedure or *rescheduling* method. A reactive scheduling deals with disruptions during project execution by rescheduling, modifying (reallocation of resources, activity crashing, etc.) or repairing the disrupted schedule (Artigues et al., 2010; Dablaere et al., 2011). Hence, in this research work, we focus exclusively on the reactive strategy. A rescheduling process is more challenging for complex and large project in many aspects. First, the rescheduling circumstances is less flexible as certain options may not be applicable those were feasible during initial schedule development. Moreover, it may need to consider additional constraints and satisfy new objectives.

In the literature, the rescheduling problem has been extensively studied in a machine scheduling environment. In contrast to machine scheduling environment, few attempts are concentrated on the project reschedule problem. Most of the efforts tried to develop a basic mathematical model based on the initial baseline schedule model rather than emphasis on the real life practice. Moreover, the consequence of the change impact in rescheduling decision making is totally void in their methods. In practice, the project managers take rational measures based on the sensitivity level of the schedule changes. For instance, the project manager must take prompt actions to handle high impact changes compare to less impact changes, when the manager has the option to be flexible. In resource limited project schedule, the sensitivity of any change must depend on the importance of workers, duration of the absence and the criticality of the task. For example, a worker with higher skill working on critical task must have high change impact than the general skill worker and vice versa. Besides, a worker reported to be absent for one week must have less impact in comparison to a worker who is going to leave the project permanently. Thus, incorporating the significance of change impact in rescheduling method is an important issue to be considered in change management. This is another main concern of the present research trends.

Traditionally, the project managers update the baseline schedule considering the unexpected events or changes based on their experience and skill, but often at a great cost. Most of the time, the project managers will face difficulty to make proper decisions for change management especially when the project has new contents and contexts. With few exceptions, the initiatives taken by the project manager become undesirable because of the poor evaluation of the change impact and proper corrective action. Finally, this may lead to project delay, cost overrun, employee malcontent, etc. Indeed, they are in need of some supporting tools to assists them to take proper decision and suitable actions to face such kinds of unexpected changes.

The aforementioned facts has motivated us to investigate and develop a systematic rescheduling (e.g., rational schedule modification) approach for coping with changes in project management.

## 1.2 Research Objective

In this research, we propose a systematic approach to repair or modify the baseline schedule to address changes due to the absence of workers during project execution time. We assume that the baseline schedule was already developed. So, we will not focus on the construction of the baseline schedule, rather, we will use the terminology and constraints of the baseline schedule to model the rescheduling framework. The baseline schedule is used for the identification of change, evaluation of change impact and finally implementation of rescheduling procedure accordingly. The specific research topic focuses on the early-phase event driven rational change decisions to revise the baseline schedule. For example, when a worker reports for his absence for a specific period of time during project's period, the project managers are required to decide whether they should revise the schedule radically or just adapt the changes in convenient time by the existing available workers. The decision is made on the basis of the importance of the absent worker, the length of the absence and the criticality of the affected task(s) due to his/her absence. The objective of the proposed method is to revise the schedule to limit the increasing of project duration from its deadline without changing too many task-worker assignments. To address the objective, we define three different change option procedures to revise the schedule for the different levels of change impacts.

## 1.3 Organization of the Thesis

A comprehensive literature review related to our work is described in Chapter 2. We focus on the literature on three aspects: project scheduling problem, schedule change management, and project rescheduling problem. Finally the existing rescheduling approaches are summarized at the end of this chapter.

In Chapter 3, we define the basic elements, constraints and model of the project scheduling problem that we are going to consider. A rescheduling model with change impact factors, schedule modification strategies and evaluation criteria of the revised schedule is presented at the end of this chapter.

In Chapter 4, we present the overall rescheduling procedure based on the model in Chapter 3. We highlight the recognition of changes, selection of change option, and implementation of change option modules of the proposed rescheduling procedure with detailed examples.

In Chapter 5, an example application related to software development project is presented to illustrate the implementation and features of the proposed approach.

We conclude our work with a summary and possible future works in Chapter 6.

# Chapter 2

# Literature Review

A project is a temporary endeavor undertaken to create a unique product or service and a schedule is the basis for planning and management of this project (PMI, 2008). The two key elements in scheduling are the development of baseline schedule and the revisions of schedule during project execution to cope with unexpected changes (Li and Ierapetritou, 2008). The scheduling development establishes the start date, duration, completion date, and resource assignment for each activity based on given requirements and constraints prior to the project unfolds. On the other hand, schedule revision is a reactive part, which monitors execution of schedule and deals with unexpected events. In this chapter, we first review the literature related to project scheduling problem in section-2.1. The existing practices in schedule change management are introduced in section-2.2. The rescheduling environment, strategy, and method are discussed in section-2.3. Finally, we summarize some existing rescheduling approaches in section-2.4.

## 2.1 Project Scheduling Problem

The development of project schedule includes activity precedence relations, duration and resource assignments for each activity in the network. A typical real life project has limited capacities or availabilities of resources that limits the schedulers to develop schedule in a simple and flexible way. To address this issue, various analytical and heuristic techniques have been developed to apply resource availability into the scheduling process. A Resource Constrained project Scheduling Problem (RCPSP) considers resources of limited availability and activities of known duration and resource requests, linked by precedence relations. The objective is to schedule all project activities over time such that scarce resource capacities and precedence constraints are respected and a certain objective function is optimized. The most traditional objective function is to minimize the project makespan but other possibilities include the minimization of cost, the maximization of some quality measures, leveling the resource usage over time, or a combination thereof (Brucker et al., 2012). After the development of the basic mathematical model for the RCPSP by Pritsker et al. (1969), various extensions of the basic RCPSP related to activity concept, temporal and resource constraints, objective function have been developed. Recently, Hartmann and Briskorn (2010) reviewed the various extensions of the basic RCPSP problem and gives an overview of these extensions. The review paper summarized and classified the recent research works on various variants and extensions.

One of the most popular extensions of the classical RCPSP is multi-mode RCPSP which often referred as MRCPSP in the literature. In standard RCPSP, it is assumed that an activity can only be executed in a single mode which is determined by a fixed duration and fixed resource requirement. In MRCPSP, the activity concept of standard RCPSP has been extended by allowing several alternatives or execution modes in which an activity can be performed (Alcaraz et al., 2003; Zhu et al., 2006). Each mode may have different resource requirements and the duration of the tasks also depend on the selection of mode, particularly, the number of resources. For example, a contractor can build a wall by hiring five workers in six days or with ten workers possibly in three or four days. So, there is no need of dedicated amount of resources for the execution of the activities. These execution modes allow schedulers to consider different combinations of duration and resource requirements in the process of scheduling optimization. This feature provides more flexibility to the project managers to deal with different disruptions during project execution. More specifically, project manager can select the mode of an activity with lower resource requirements to handle the resource shortage or unavailability. On the other hand, s/he can also assign a mode with more resource capacities for an activity running behind the schedule.

Poder et al. (2004) proposed a specific RCPSP where the resource consumption of each task is continuously varying over time and the duration and starting time of each activity can vary within real intervals. Drezet and Billaut (2007) discussed a project scheduling problem where resources are employees and activity requirements are time-dependent. Most recent, Hartmann (2012) proposed an extension of the classical RCPSP with both time-dependent resource capacities and activity resource requirements and they referred the extended model as RCPSP\t. The other most popular extension is Multi-skill Project Scheduling Problem (MSPSP). In MSPSP, the resources are usually staff that master several skills. This type of problem has been addressed by Bellenguez and Nron (2005); Bellenguez (2008).

In this thesis, we follow the multi-mode RCPSP (MRCPSP) as the baseline schedule considering the resource as worker having multiple skills for our *reactive* schedule model.

## 2.2 Schedule Change Management

During project execution, projects are subject to considerable uncertainty in real environment. The changes may originate from a number of possible different sources or events: activities may take more or less time than originally estimated because of over or underestimate of processing time, resources may become unavailable (e.g., worker absenteeism or machine failure), material may arrive behind schedule, shortage of materials, activities priority may be changed because of urgent or rush jobs arrival, start times and duration may have to be changed due to productivity variation, new activities may have to be incorporated or activities may have to be dropped due to changes in the project scope, weather conditions may cause severe delays, rework may be required, etc (Herroelen and

Leus, 2005). Besides, Xiao et al. (2010) identify requirement changes, urgent bug fixing, incorrect or unexpected process execution, and staff turnover as the possible disruption events in software development environment. These uncertainties may cause the initial schedule no longer optimal or feasible. When the deviations between actual and initial schedule become noticeable, we say that the project schedule is *disrupted* or *changed*. A disrupted schedule incurs higher costs due to missed due dates which causes resource idleness, higher work-in-process inventory and increased schedule perturbation due to frequent updating (Herroelen and Leus, 2005). Thus, change management is an emerging field in project management to deal with unexpected events.

In response to such unexpected changes in project schedule, researchers have studied different approaches. Among them robust scheduling, dynamic scheduling, and reactive scheduling are most common in practice. Herroelen and Leus (2005) survey the fundamental approaches in project scheduling under uncertainty. They discussed several approaches for obtaining schedules with uncertain information: reactive scheduling, stochastic project scheduling, fuzzy scheduling, robust scheduling, and sensitivity analysis. Some future directions are also discussed.

Robust or proactive scheduling approaches try to accommodate uncertainties in advance. The aim of proactive schedule is to build a robust baseline schedule that is protected as much as possible against disruption during project execution. Robust scheduling tries to anticipate the effects of possible disruptions using statistical knowledge of uncertainty. In the field of proactive scheduling, the problem of coping with activity duration variability has been covered by Van de Vonder et al. (2008), Dablaere et al. (2011). On the other hand, the problem of uncertainty with respect to resource availability has been addressed by Lambrechts et al. (2008a,b), Deblaere et al. (2011), and Xiong et al. (2012). Most of the approaches follow the redundancy policy to build the robust schedule. This implies the reservation of extra time (e.g., time buffer) and/or resource capacities (e.g., resource buffer) to absorb the unexpected events during execution (Lambrechts et al., 2008b). In practice, the allocation of extra time and resource buffer is not achievable due to the time and budget limits. Moreover, robust scheduling is most effective when there are limited and predicable disruption in the project. Unfortunately, no matter how much we try to protect the initial schedule against possible disruptions, we can never totally eliminate their occurrences. So, a proactive schedule will always require a reactive scheduling procedure to deal with schedule disruption that can not be absorbed by the baseline schedule (Van de Vonder et al., 2007). A reactive scheduling deals with changes during project execution by fully rescheduling or partially modifying the initial schedule. In this research work, we focus exclusively on the *reactive scheduling* or *rescheduling*. In literature, rescheduling is the process of modifying an existing project schedule in response to disruptions or changes (Vieira et al., 2003).

## 2.3 Project Rescheduling Problem

The literature on project scheduling shows that vast efforts were spent on the development of exact and heuristics methods for generation of initial project schedule (i.e., baseline or predictive schedule). Besides, rescheduling has been widely discussed in the manufacturing industry. On the contrary, very few papers deal with rescheduling problems for project scheduling. Vieira et al. (2003) presented a framework for rescheduling in manufacturing systems. This framework defines four important dimensions: *rescheduling environment, rescheduling strategy, rescheduling policy*, and *rescheduling method*. Fig. 2.1 shows the complete rescheduling framework as they proposed.

The rescheduling environment identifies the set of jobs or activities to be taken into consideration in the schedule. Two different types of environment are defined: *static* and *dynamic*. In static scheduling environment, there are a finite number of jobs to be scheduled, whereas in a dynamic environment infinite set of jobs arrive on a continuous basis. In this work, we consider a *static environment* and the set of activities are fixed and known in advance (i.e., deterministic).

The second dimension is the rescheduling strategy. It determines whether or not a

| Rescheduling environments | | | | |
|---|---|---|---|---|
| Static (finite set of jobs) | | Dynamic (infinite set of jobs) | | |
| Deterministic (all information given) | Stochastic (Some information uncertain) | No arrival variability (cyclic production) | Arrival variability (flow shop) | Process flow variability (Job shop) |

| Rescheduling strategy | | | | |
|---|---|---|---|---|
| Dynamic (no schedule) | | Predictive-reactive (generate and update) | | |
| Dispatching rules | Control-theoretic | Rescheduling Policies | | |
| | | Periodic | Even-driven | Hybrid |

| Rescheduling methods | | | | |
|---|---|---|---|---|
| Schedule generation | | Schedule repair | | |
| Nominal schedule | Robust schedule | Right-shift rescheduling | Partial rescheduling | Compete regeneration |

Figure 2.1: Rescheduling framework Vieira et al. (2003)

baseline schedule is generated. The *predictive-reactive strategy* is considered in this thesis to generate and update an initial schedule in response to unexpected changes. To implement a predictive-reactive rescheduling strategy, a rescheduling policy is needed. The rescheduling policy specifies the time and events that will trigger the rescheduling in response to disruption. In other words, rescheduling policy specifies when and how rescheduling will be performed. Two types of rescheduling policies are commonly used: *periodic* and *event-driven*. A periodic policy updates the schedule periodically at the beginning of the predefined rescheduling interval. The event-driven rescheduling actions are taken every time the system is affected by unexpected events such as machine failure. A third possibility is to use a mixed policy, called *hybrid*. Our problem forces us to reschedule whenever an unexpected event occurs (i.e., when a worker reports to be absent). However, a hybrid approach can also be introduced in case of multiple events.

Finally, rescheduling method generates and updates the schedule using some predefined methods or procedures. Three methods are used to update or repair infeasible schedules: *right shift scheduling*, *partial rescheduling*, and *complete generation* or *regeneration*. Right shift rescheduling is the easiest method to repair schedule. In project schedule, it maintains original activities sequences and task assignment by shifting the tasks to the

Table 2.1: Rescheduling framework for this thesis.

| **Dimension** | **Classification** |
|---|---|
| Environment | Static and deterministic |
| Strategy | Predictive and reactive |
| Policy | Event driven |
| Method | Right-shift rescheduling and partial rescheduling |

right by the amount of time needed to make the schedule stable. Partial rescheduling methods reschedule only those operations or activities affected directly or indirectly by the disruption. This method maintains scheduling stability with little nervousness by preserving the initial schedule as much as possible. Finally, complete rescheduling develops a totally new schedule. The model presented in this thesis considers both right shift scheduling and partial rescheduling.

Table-2.1 summarizes the project rescheduling problems treated in this thesis based on the framework of Vieira et al. (2003). In the next section, we are going to give an overview of research works covering rescheduling in project scheduling environments.

## 2.4 Existing Rescheduling Approach

The rescheduling problems have been extensively studied in production scheduling environment for different types of disruptions. For review papers on machine rescheduling, we would like to refer Vieira et al. (2003), Aytuga et al. (2005) and Katragjini et al. (2012). On the other hand, this subject is almost new in the field of project scheduling. In this section, we summarize some existing works in project scheduling environments.

The literature on reactive scheduling methods for single mode RCPSP has been recently developed by considering different sources of disruptions. The problem of coping *activities duration variability* has been addressed by Vonder et al. (2007). They introduce several heuristic approaches for fully rescheduling a project subject to activity duration disruptions. Four reactive approaches are introduced. First of all, simple priority rules are used in conjunction with a schedule generation scheme. The second approach is to fix resource allocations by right-shifting the affected activities such that a feasible schedule is generated. A third procedure is a sampling approach that considers several alternative solutions by combining various priority lists with various schedule generation schemes and finally selects the best amongst those. Time window sampling is a modification of the sampling approach that focuses on the activities planned to start within a certain time window from the rescheduling point.

Bagheri et al. (2012) addressed reactive scheduling in resource leveling problem (RPL) rather than RCPSP to handle the disruption related to activity duration variation (i.e., activity may take longer or shorter than expected). Four reactive procedures: simple shifts, simple shifts with railway scheduling, shift vector, shift vector with railway scheduling are proposed to react against the disruption in the execution phase of the project.

The problem of schedule disruption with respect to *resource availability* in RCPSP has been addressed by Lambrechts et al. (2008a). They proposed reactive policies based on list scheduling to restore the schedule in feasible condition after the occurrence of resource breakdown. More precisely, when a disruption occurs, a schedule order list is created with the in-progress and not yet completed tasks and sorted them based on their baseline starting time. Finally, the schedule order list is decoded in to feasible solution using modified serial schedule generation scheme and taking into account the new resource availabilities. They also introduced a tubu search technique to obtain feasible schedule.

Liu and Shih (2009) considered the problem of rescheduling in construction project in the face of project changes due to *unexpected progress of the activities* that is when a activity is behind the schedule because of productivity disruption. They proposed a optimization model using constraint programming technique to reschedule the project. The model implements two rescheduling methods, Complete Rescheduling (CR) and Partial Rescheduling (PR). The complete rescheduling method produces a totally new schedule irrespective of whether the new schedule is feasible or not. On the other hand, the partial rescheduling method rearrange only the affected activities while keeping allocation of being-scheduled activities (i.e., not yet started) same as baseline schedule.

The works discussed above are relevant to the single mode project scheduling problem where each activity needs fixed amount of resources for execution. Recently, the project schedules prefer multi-mode project scheduling strategy because of its higher flexibility in resource allocation as discussed in section-2.1. The literature on reactive scheduling in multi-mode project scheduling environment is almost new. To the best of our knowledge, the works done by Zhu et al. (2005) and Deblaere et al. (2011) supports multi-mode project scheduling problems.

Zhu et al. (2005) presented a hybrid mixed integer programming/constraint propagation approach to address wide range of changes in RCPSP. The model includes project *network disruption*, *activity disruption*, *resource disruption* and *milestone disruption*. Three recovery options are included in their model. The *rescheduling* options assigns finish time to activities that deviates from the original schedule. the *mode alternative* option uses a different resource duration for an activity. The third, *resource alternative*, increases the resource capacities in the project.

Deblaere et al. (2011) also formulated a reactive scheduling problem for the multi-mode RCPSP. The problem considers both *resource disruption* and *activity duration disruption*. They proposed a number of dedicated exact reactive scheduling procedures and tabu search heuristics for repairing the disrupted schedule.

Both Zhu et al. (2005) and Deblaere et al. (2011) reactive scheduling approaches are applicable only for the project schedules dealing with single skill resources. But it has to be stated that the assumption of single skill resources is quite limited in real-world project schedules. Thus, we are interested to integrate the multi-skill resources in our reschedule framework.

Except for the Liu and Shih (2009) works, most of the aforementioned approaches follow *full rescheduling* or *complete regeneration* approaches. The major drawback of fully rescheduling approach is that it always generates a new schedule irrespective of whether the initial schedule is feasible in some respect. Moreover, fully rescheduling approaches in some complex and large projects are impractical as the original schedule is a result of deliberate efforts from many experts, including engineers, financial controllers and administrators.. This is basically motivated the emergence of partial rescheduling approach which tries to reschedule only a subset of activities, especially those are affected by the disruptions.

The existing partial rescheduling approaches in the literature focused on production-specific scheduling. To the best of our knowledge, the literature on partial rescheduling considering the significance of change impact is virtually void. We are only aware of a Local Rescheduling (LRS) approach in complex project scheduling problems by Kuster et al. (2010). The approach performs partial rescheduling within a *time window* using local search technique. On the other hand, Chen (2010) has preliminarily investigated the basic strategies to control the change propagation in construction projects taking the change impact into account. The proposed work of this thesis is a continuation of his work.

## 2.5 Conclusion

In this chapter, we have described topics related to our research work and a selection of works from literature that addressed project reactive scheduling problems. The scope of those existing reactive approaches reveals some shortcomings with respect to real life practice or scenario. The works done for single mode RCPSP are in conflicts with some real project schedules where an activity can be executed in a number of different modes (i.e., different activity durations and different resource requirements). We are aware of only two works done by Zhu et al. (2005) and Deblaere et al. (2011) support multi-mode project scheduling problems. But the scope of their approaches are limited for single skill resources whereas most of the real project schedules have multi-skills resources. To the best of our knowledge, the literature on reactive scheduling policies supporting both multi-mode activity execution and multi-skills resources is virtually void. On the other hand, the consideration of change impact in the existing rescheduling solution approaches is also void. In this context, we propose a new reactive scheduling approach to overcome some shortcomings of the existing approaches in our research work. Our reactive scheduling approach is capable to modify a project schedule having multi-mode activities and multi-skill workers. Moreover, we also incorporate the consideration of change impact by defining a number of change options to revise the schedule for different levels of changes (i.e., low, medium, and high). The next chapter presents our proposed change management approach in details.

# Chapter 3

# Schedule Change Management Approach

## 3.1 Introduction

The project baseline schedule is subject to ongoing reactive process as changing circumstances continually force adjustment and revision of initial plans. We need to revise a schedule by considering the unexpected events or changes. The modified schedule may be quite different from the initial schedule and sometime becomes undesirable because of the poor evaluation of the change impact and the lack of proper corrective action. In this research, we propose a repair based scheduling approach for Multi-mode Resource-Constrained Project Scheduling (MRCPSP) problem to address change due to the absence of workers. In this context, we focus on the early-phase event driven reactive scheduling to revise the baseline schedule. For example, during the project's period, when a worker reports for his absence for a specific period of time, the project managers are required to decide whether they should revise the schedule radically or just adapt the changes in convenient time by the remaining workers in order to minimize the change impact. The objective of our proposed reactive scheduling system is to minimize the difference between the new and the initial schedule in terms of delay and perturbations (e.g., without changing too many assignments). To address this, we define different change options to revise the schedule according to the various levels of change impact.

In this chapter, we first introduce the details of the baseline schedule that can be used during project execution to identify changes, evaluate change impacts, and finally, modify the schedule. The overall reactive scheduling model and properties of the major components are defined in subsequent sections. Finally, the benchmarking criteria for the revised schedule is described at the end of this chapter.

## 3.2 Project Schedule Model

The project schedule would guide the project management team to execute the project plan, reflect change, and monitor progress throughout the life of the project. The schedule is also one of the most important tool in managing and controlling changes. It helps to project management team to identify the changes from the initial plan and evaluate the resulting impacts on project throughout the entire project life cycle. In this context, at the beginning of this research we try to establish a standard model of the project schedule that can be used in future for the identification of change, evaluation of change impact and finally implementation of reschedule. The framework of our baseline schedule is based on the concept of multi-mode resource-constrained project scheduling problem (MRCPSP) (Drezet and Billaut, 2007) and multi-skill project scheduling problem (MSPSP)(Bellenguez and Neron, 2004; Drezet and Billaut, 2007). To limit the scope of this research work, we will not focus on the construction of the baseline schedule and assume this is given. Here, we will only focus on the characteristics of baseline schedule that will be followed in our rescheduling model.

### 3.2.1 Basic Schedule Elements

Our main focus is on the project scheduling problem for labor intensive industries, such as software development and construction industries. We assume that a project development life cycle has different phases, each phase has different tasks that is possible to accomplish by human resources (e.g., workers). The basic model considers that a worker can have several skills and capable to perform more than one tasks. For example in software companies, resource can correspond to analysts, designers, developers, testers, etc. On the other hand, the tasks of the project that need to be scheduled according to some precedence constraints require some specific skills. Thus scheduling a task at any time, needs to match its skills requirements with the skills of the workers who are available on that time. In short, we can say that the schedule consists of a set of tasks linked together by precedence constraints and a set of human resources linked with tasks through skill constraints.

We assume that the project consists of a set of tasks $T = \{t_1, t_2, ..., t_n\}$, where $n$ being the total number of tasks. Each task $t_i$ is associated with some skills $S = \{s_1, s_2, ..., s_p\}$ and a total effort $w_i$ expressed in man-day (MD) that needs to be done to accomplish the task. In order to perform these tasks, $m$ workers $P = \{p_1, p_2, ..., p_m\}$ with different skills $S = \{s_1, s_2, ..., s_p\}$ are available.

### 3.2.2 Baseline Schedule

The baseline schedule is a *task-resource allocation schedule* that shows the assignment of the workloads of $n$ tasks of a project among the available $m$ workers on each time period. According to the general situation of project management, we assume that the project is scheduled according to the standard calendar *five* working days per week and the maximum working effort for each staff is *eight hours* per day (e.g., man-day (MD)). For simplicity, we represent the time periods of the project as a set of days $D = \{d_1, d_2, ..., d_t\}$, where $t$ represents the project duration. The task-resource allocation schedule $SH$ is defined in Eq.(3.1).

$$SH = \begin{array}{c} \\ p_1 \\ p_2 \\ \vdots \\ p_m \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & d_3 & \dots & d_t \end{array} \\ \begin{pmatrix} sh_{11} & sh_{12} & sh_{13} & \dots & sh_{1t} \\ sh_{21} & sh_{22} & sh_{23} & \dots & sh_{2t} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ sh_{m1} & sh_{m2} & sh_{m3} & \dots & sh_{mt} \end{pmatrix} \end{array} \tag{3.1}$$

where, $i \in \{1, 2, ..., m\}, j \in \{1, 2, ..., t\}$ and $k \in \{1, 2, ..., n\}$

$$sh_{ij} = \begin{cases} t_k & \text{if employee } p_i \text{ is assigned to activity } t_k \text{ during time period } d_j, \\ Null & \text{otherwise.} \end{cases}$$

We assume that the duration of each task is only related to the completion of the total effort by using different combination of its modes and independent of material and other resources. More precisely, if task $t_1$ having total workload $w_1 = 12$ needs two or three workers for its two possible modes respectively, then the duration of this task depends on which modes are selected to complete the total effort. Fig. 3.1 shows two possible schedules with different time durations for the task $t_1$. Moreover, we also assume that the workloads of the tasks are distributed among the available workers by using existing methodologies that satisfies the following two constraints:

- Precedence constraint: A task is ready to be processed only when all its predecessor activities are completed.
- Skill requirement constraint: A worker performing a task must have all the skills required by that task.

For the ease of the change impact analysis in reactive schedule method, a *Gantt chart* is developed using the information of the task-resource allocation schedule ($SH$) and critical path method. The purpose of the Gantt chart is to illustrate the start date, duration,

| | D1 | D2 | D3 | D4 | D5 | D6 |
|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T1 | T1 | T1 |
| P2 | T1 | T1 | T1 | T1 | T1 | T1 |
| P3 | | | | | | |

(a) Schedule-1 with 6 days duration

| | D1 | D2 | D3 | D4 | D5 | D6 |
|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T1 | | |
| P2 | T1 | T1 | T1 | T1 | | |
| P3 | T1 | T1 | T1 | T1 | | |

(b) Schedule-2 with 4 days duration

Figure 3.1: Possible schedules of task $t_1$ having workload $w_1 = 12$.

slack time and finish date of the tasks more precisely. Besides showing the critical and non-critical tasks, the Gantt chart also includes the per day workload distribution of each task in the task-resource allocation schedule. The Gantt chart information is amended or updated according to the changes in the task-resource schedule during reactive scheduling.

To illustrate the concept of the above schedule model, we introduce a small project having four tasks $t_1$, $t_2$, $t_3$ and $t_4$ with estimated efforts 6, 8, 4, and 8 MD respectively. Fig. 3.2 presents the task precedence diagram for the project. Let four workers $p_1$, $p_2$, $p_3$, and $p_4$ are available to accomplish the tasks of this project. Fig. 3.3(a) shows the task-resource allocation schedule for this project and the corresponding Gantt chart with necessary information is illustrated in Fig. 3.3(b).

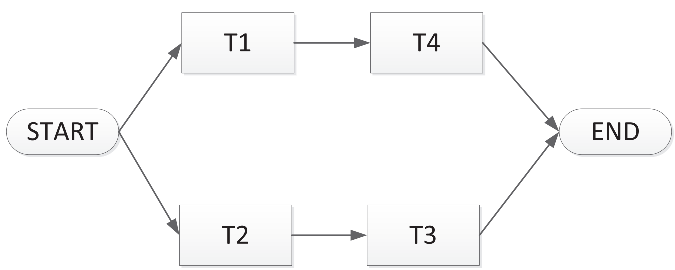
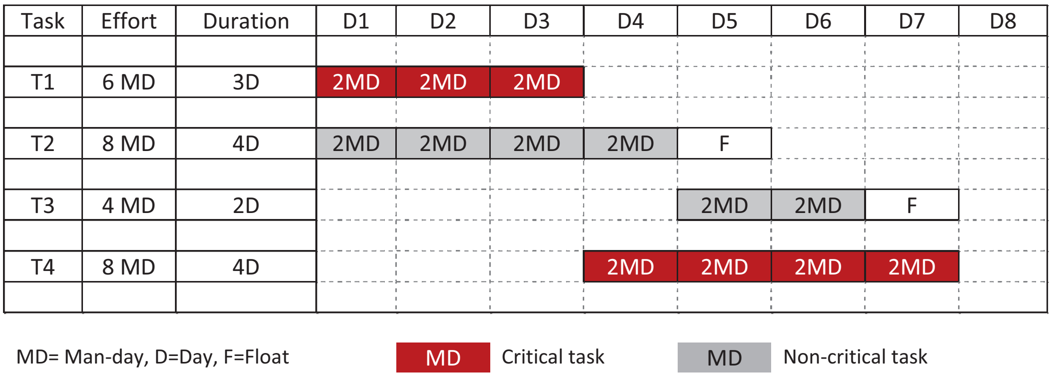



Figure 3.2: Tasks precedence diagram

|  | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T4 | T4 | T4 | T4 |  |
| P2 | T2 | T2 | T2 | T2 | T3 | T3 |  |  |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 |  |  |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 |  |

(a) Task-resource allocation schedule




(b) Project Gantt chart

Figure 3.3: Representation of baseline schedule.

### 3.2.3 Schedule Dependency Relations

As mentioned earlier, the task-worker allocation schedule consists of a set of tasks linked together by precedence constraints and a set of human resources linked with tasks through skill constraints. The relationships among the tasks, workers and skills are mathematically described below.

The tasks of the project are subject to precedence constraints, i.e, a task cannot start before all its predecessors are completed. This relationship regulates the priority order of tasks' execution. In our research work, the precedence relations of finish-to-start with a zero parameter value (i.e., $FS = 0$) is considered between the tasks. Suppose the matrix of precedence relationship of tasks is $TP$, then it can be represented by Eq. (3.2):

$$TP = \begin{bmatrix} tp_{11} & tp_{12} & tp_{13} & \dots & tp_{1n} \\ tp_{21} & tp_{22} & tp_{23} & \dots & tp_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ tp_{n1} & tp_{n2} & tp_{n3} & \dots & tp_{nn} \end{bmatrix} \tag{3.2}$$

where, $i, j \in \{1, 2, ..., n\}$

$$tp_{ij} = \begin{cases} 1 & \text{the task } t_j \text{ is the successor of task } t_i, \\ 0 & \text{the task } t_j \text{ is not the successor of task } t_i. \end{cases}$$

A task requires specific set of skills to be processed, that cannot be performed by all workers. If $S = \{s_1, s_2, ..., s_p\}$ be the set of required skills associated with different tasks, we can define the task-skill requirement matrix $TS$ by Eq. (3.3) as:

$$TS = \begin{bmatrix} ts_{11} & ts_{12} & ts_{13} & \dots & ts_{1p} \\ ts_{21} & ts_{22} & ts_{23} & \dots & ts_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ ts_{n1} & ts_{n2} & ts_{n3} & \dots & ts_{np} \end{bmatrix} \tag{3.3}$$

where, $i \in \{1, 2, ..., n\}$ and $j \in \{1, 2, ..., p\}$

$$ts_{ij} = \begin{cases} 1 & \text{if task } t_i \text{ is required the skill } s_j, \\ 0 & \text{otherwise.} \end{cases}$$

A worker cannot possibly execute all the tasks because of the skill requirement of the different tasks. For instance, a technical writer in a software industry would not be able to perform coding or testing tasks perfectly. When we have the details information about the capabilities and skills set of the workers and the skill requirement of the task, it will be more convenient to find the potential replacement one in place of absent worker. If $m$ workers with $p$ different skills are available for the execution of the project tasks, we define the resource-skill relationship matrix $RS$ in Eq.(3.4).

$$RS = \begin{bmatrix} rs_{11} & rs_{12} & rs_{13} & \dots & rs_{1p} \\ rs_{21} & rs_{22} & rs_{23} & \dots & rs_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ rs_{m1} & rs_{m2} & rs_{m3} & \dots & rs_{mp} \end{bmatrix} \tag{3.4}$$

where, $\quad i \in \{1, 2, ..., m\}$ and $j \in \{1, 2, ..., p\}$

$$rs_{ij} = \begin{cases} 1 & \text{if worker } p_i \text{ masters the skill } s_j, \\ 0 & \text{otherwise.} \end{cases}$$

## 3.3 Project Rescheduling Approach

Project managers facing a change scenario (i.e., knowing a worker will be absent for a specific time period) is required to assess the change impact and take corresponding corrective actions. Accordingly, the project managers need to decide whether they should revise the schedule radically or moderately by reallocating tasks among the the existing workers to cover the affected workload. The assessment of the change impact is not a trivial task for large complex projects due to higher inter tasks dependencies with limited number of skilled workforce. So, proper decision is required to revise the baseline schedule otherwise it can lead to unexpected outcomes such as project delay or poor output of the project. In this context, the purpose of this research work is to develop a decision based reactive scheduling approach to handle different change scenario in a systematic way.

### 3.3.1 Definition of Change Scenario

A change scenario indicates a worker to be absent for a specified period of time during project execution. When a change scenario is known, we can check which tasks are supposed to be accomplished by the absent worker via the task-resource allocation schedule (i.e., $SH$). In our research, we term these tasks as *outstanding* task and symbolically denote by $t_x^a$. The workload of the outstanding task is termed as *affected* workload and symbolically denoted by $w_x^a$. In short, the change scenario specifies the outstanding tasks and the corresponding amount of affected workload due to the absence of worker.

If $t_i^a$ ; $i \in \{1, 2, ..., n\}$ be the affected tasks due to the absence of worker $p_j$; $j \in \{1, 2, ..., m\}$ for duration $d_k$ to $d_{k+n}$; where $k \in \{1, 2, ..., t\}$ and $\{n \in R : n > 0\}$, then the change scenario in terms of outstanding tasks $t_i^a$ and corresponding affected workload $w_i^a$ can be represented by the set $T_a$ as:

$$T_a = \{(t_i^a, w_i^a)\}; \qquad \text{where } i \in \{1, 2, ..., n\}$$

### 3.3.2 Change Impact Analysis

When a project is executing with limited number of proprietary resources (e.g., the allocated workers is fixed) without sharing resources of other projects or third parties, the project managers facing a change scenario (e.g., knowing a worker to be absence for a specific period of time) need to assess the impact of changes in view of the importance of the absent worker, the length of the absence, and the criticality of the tasks. In this research, the impact of the change scenario for a task is assessed based on the following three parameters:

- Task sensitivity
- Workload sensitivity
- Workforce sensitivity

**Task Sensitivity:** The first factor is about task sensitivity that is measured in terms of *sack* (float) time associated with each task. The slack time defines the maximum delay that can be tolerated in the execution time of a task without affecting the overall project duration. When a change scenario is known, we can find the slack time from Gantt chart

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | **(T1)** | **(T1)** | T4 | T4 | T4 | T4 | |
| P2 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P3 | T2 | T2 | **(T2)** | **(T2)** | T3 | T3 | | |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |

Figure 3.4: Task sensitivity

(Fig. 3.3(b)). The task with zero slack is considered as critical task and any delay on the execution time of this task will affect the overall project duration. On the other hand, the non-critical tasks may have different amount of slack times and a task having lowest amount of slack time is more sensitive than the others. For example, if $t_1$ and $t_2$ are two non-critical tasks with slack times $p$ and $q$ ( where, $p < q$) respectively, then task $t_1$ causes higher change impact than $t_2$ for the same change scenario. In general, the critical tasks cause a high change impact than the non-critical tasks for the same change scenario.

For instance, suppose that worker $p_1$ will remain absent for days $d_2$ and $d_3$ and worker $p_3$ on $d_3$ and $d_4$ days as shown in the task-resource allocation schedule Fig. 3.4. The change scenarios due to the absence of workers $p_1$ and $p_2$ are obtained by checking the task-resource allocation schedule in Fig. 3.3(a) as $(t_1, 2MD)$ and $(t_2, 2MD)$ respectively. The sensitivity of these outstanding tasks is determined using slack information in the Gantt chart Fig. 3.3(b). The change scenarios due to absent workers $p1$ and $p_3$ cause the impact on tasks on the critical and non-critical path respectively. In the above context, we may conclude that the change scenario corresponding to worker $p_1$ would have higher impact compare to worker $p_3$.

**Workload Sensitivity:** The second consideration for assessing change impact is the workload sensitivity which means the amount of affected workloads of each task that needs to reassign among the existing available workers (e.g., without hiring new staffs or considering overtime work). According to the definition of project schedule model, workload is referred to any non-zero entry on the task-resource allocation schedule, and each entry represents 8 hours working time per day (i.e., man-day (MD)). So if the

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |
| P2 | T2 | **(T2)** | **(T2)** | **(T2)** | **(T3)** | T3 | | |
| P3 | T2 | T2 | T2 | **(T2)** | **(T3)** | T3 | | |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |

Figure 3.5: Workload sensitivity

amount of affected workloads is high, we consider that the change scenario has a high change impact.

For example, if worker $p_2$ reports to be absent from $d_2$ to $d_5$ and worker $p_3$ for $d_4$ and $d_5$ days as shown in the task-resource allocation schedule in Fig. 3.5. The affected tasks are $t_2$ and $t_3$ and the change scenario is $T_a = \{(t_2, 4MD), (t_3, 2MD)\}$. We may say that the change scenario corresponding to task $t_2$ would have higher impact than task $t_3$ as $t_2$ has more workload than $t_3$.

**Workforce Sensitivity:** The third factor is about workforce sensitivity or workers' importance. In general perception, if a worker possesses some high skills that other workers do not have, this worker is more significant due to the difficulty of finding a replacement worker. To confine the scope of the research, it is considered that the sensitivity of a worker is associated with the number of replacement workers for a specific task.

For example, the task-resource allocation schedule as shown in Fig.3.3(a), let only workers $p_1$ and $p_4$ have the required skills for doing the task $t_1$. Besides, task $t_2$ can be performed by all the four workers (e.g., $p_1$, $p_2$, $p_3$, and $p_4$). If worker $p_1$ will remain absent for 1 day while working on task $t_1$, only worker $p_4$ can replace him. On the other hand, if worker $p_2$ will be absent for 1 day while working on task $t_2$, the potential replacement workers $p_1$, $p_3$, and $p_4$ can be found. It means that absence of worker $p_1$ is easier to delay duration of the outstanding tasks $t_1$ than worker $p_2$ would do because less workers can replace worker $p_1$. Therefore, worker $p_1$ is considered as more sensitive compare to worker $p_2$.

By considering the above three parameters, the overall impact for a change scenario is classified as *low*, *medium* and *high* based on the correlation between the number of potential replacement workers, and how much outstanding workload can be reassigned them without interrupting their present work or by interrupting present work in terms of delay. We termed the interrupting and non-interrupting workload distribution strategies as *preemptive* and *non-preemptive* policy respectively. We assume that if it is possible to distribute the affected workload among the potential replacement workers without interrupting their present work and delaying the task duration, the scenario has *low* impact, otherwise it has *medium* or *high* impact. The procedure for calculating the change impact for a change scenario based on the above principles is described in Chapter 4.

### 3.3.3 Strategies for Schedule Change

After properly assessing the change scenario (e.g., low, medium or high), the project managers need to decide whether they should take prompt or flexible actions based on the significance of the changes. If the impact is very high, some less important tasks can be temporary halted and those workers are allocated to the affected tasks to ensure the timely completion. On the other hand, for low impact task, the project manager can simply request the potential replacement workers to take part on the affected task in suitable time after completion of present tasks at hand . Such decision is often needed to revise the initial task-resource allocation schedule. The schedule is revised based on the following strategies.

**Types of modification:** In this research, three types of modification is considered to revise the task-resource allocation schedule. They are summarized in Table-3.1 and

Table 3.1: Schedule modification types

| **Category** | **Initial** | **After Modification** | **Representation** |
|---|---|---|---|
| Type-1 | $sh_{ij} = t_a$ | $sh_{ij} = \phi$ | $(T_a)$ |
| Type-2 | $sh_{ij} = t_a$ | $sh_{ij} = t_b$ | $T_b(T_a)$ |
| Type-3 | $sh_{ij} = \phi$ | $sh_{ij} = t_b$ | $[T_b]$ |

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |
| P2 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |

(a) Baseline schedule

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |
| P2 | T2 | **(T2)** | T2 | T2 | T3 | T3 | | |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |

(b) Modification type-1

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | **T2(T4)** | T4 | T4 | T4 | |
| P2 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |

(c) Modification type-2

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |
| P2 | T2 | T2 | T2 | T2 | T3 | T3 | **[T3]** | |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P4 | T1 | T1 | T1 | T4 | T4 | T4 | T4 | |

(d) Modification type-3

Figure 3.6: Schedules modification types.

illustrated in Fig. 3.6.

***Type-1:*** The first category of modification refers to a task concerning a worker on one day is canceled. This category of modification is required when a worker is absent or some task of the worker should be deferred due to the delay of its precedence task. A task within first parenthesis (e.g., $(T_a)$) in the task-resource allocation schedule indicates this type of modification (Fig. 3.6(b)).

***Type-2:*** The second category of modification points to change the initial assigned task of a worker with new task. The modified entries of this type is shown as two different tasks (e.g., $T_b(T_a)$) in the task-resource allocation schedule, the first task indicates the newly assigned task and the second task within the first parenthesis means the initial assigned task (Fig. 3.6(c)). This type of modification is used to address the change impact by reallocating the tasks based on task priority.

***Type-3:*** The last category of modification is related with one day of the extra work in free day after a worker completed initial assigned task. The modified entry of this category is a single task with in a third bracket (e.g., $[T_a]$) as shown in Fig. 3.6(d). This type of modification adds extra cost for the project.

**Modification strategy:** The modification is performed in the task resource allocation schedule based on the following two strategies:

- Non-preemptive strategy
- Preemptive strategy

***Non-preemptive strategy:*** Non-preemptive modification is referred to flexible revision of the schedule that minimizes the disturbance of the remaining workers. In specific, the modification strategy allows the remaining workers to work on the affected workloads only after the current tasks at hand are completed. We can also term this strategy as non-interrupted modification as the worker will work on affected task (e.g., $t_b$) after completion of current task (e.g., $t_a$) at hand and represent mathematically as in Eq. 3.5:

$$\text{Non-preemptive modification:}\quad sh_{i(j-1)} \neq t_a \,\&\, sh_{ij} = t_a \rightarrow sh_{ij} = t_b \tag{3.5}$$

For example, if workers $p_1$ and $p_4$ in the project team is found as the replacement of the absent workers $p_2$ to work on the affected task $t_2$ as shown in the task-resource allocation schedule of Fig. 3.7. In order to minimize the disturbance of present task, non-preemption workload distribution strategy permits the workers $p_1$ and $p_4$ to work on task $t_2$ on day $d_4$ after completion of their present task $t_1$ at hand.

***Preemptive strategy:*** In contrast to non-preemptive strategy, preemptive strategy

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | **T2(T4)** | T4 | T4 | T4 | |
| P2 | T2 | **(T2)** | **(T2)** | T2 | T3 | T3 | | |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P4 | T1 | T1 | T1 | **T2(T4)** | T4 | T4 | T4 | |

Figure 3.7: Non-preemptive modification strategy

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 |
|---|---|---|---|---|---|---|---|---|
| P1 | T1 | **T2(T1)** | T1 | T4 | T4 | T4 | T4 | |
| P2 | T2 | **(T2)** | **(T2)** | T2 | T3 | T3 | | |
| P3 | T2 | T2 | T2 | T2 | T3 | T3 | | |
| P4 | T1 | **T2(T1)** | T1 | T4 | T4 | T4 | T4 | |

Figure 3.8: Preemptive modification strategy

allows more prompt action to revise the schedule to tackle high change impact. In specific, the preemptive strategy allows to revise schedule by interrupting the current tasks of the potential replacement workers in order to work on affected workloads. This modification is also referred as interrupted modification as the worker is required to stop the current task (e.g., $t_a$) at hand to work on another task (e.g., $t_b$) and mathematically defined by the Eq. 3.6. Comparatively, preemption policy is more flexible than non-preemptive policy to revise the schedule for a change scenario.

$$\text{Preemptive modification:} \quad sh_{i(j-1)} = sh_{ij} = t_a \rightarrow sh_{ij} = t_b \tag{3.6}$$

For instance, for the same change scenario of the previous example, the preemptive strategy allows the replacement workers $p_1$ and $p_4$ to suspend their present tasks $t_1$ to work on the outstanding task $t_2$ immediately on day $d_2$ as shown in Fig. 3.8.

**Change Options:** Based on the above all modification types, modification strategies and the number of workers involve in reactive action, three change options are defined as follows:

- ***Option-A:*** Considers all modification types, non-preemptive modification strategy, single or multiple replacement workers to revise the task-resource allocation

Table 3.2: Schedule change options

| Options | Modification strategy | No. of replacement workers |
|---|---|---|
| Option-A | Non-preemptive | Single or multiple |
| Option-B | Preemptive | Single |
| Option-C | Preemptive | Multiple |

schedule.

- ***Option-B:*** Considers all modification types, preemptive modification strategy, and only single replacement workers to revise the task-resource allocation schedule.

- ***Option-C:*** Considers all modification types, preemptive modification strategy, multiple replacement workers to revise the task-resource allocation schedule.

The above definitions are summarized in Table-3.2. Since Option-A allows non-preemptive actions, it has the least flexibility for schedule revision. This option seems to be more appropriate when the change impact is assessed as low. On the other hand, option-A can lead to the least disturbance to the remaining workers but may commence delay for more sensitive tasks. In contrast, option-C allows preemptive actions on multiple workers, it represents the most flexible option to revise the schedule. The weakness of this option is that it may cause some unnecessary disturbance to the remaining workers. consequently, this option is more suitable for high impact change scenario. Option-B lies between Option-A and Option-C in view of revision flexibility and disturbance to remaining workers as well as handling delay. Table-3.3 summarizes the attributes of the three change options considering revision flexibility, disturbance to remaining workers and delay.

Table 3.3: Properties of change option

| Options | Flexibility | Disruption | Delay |
|---|---|---|---|
| Option-A | Less | Low | High |
| Option-B | Moderate | Medium | Moderate |
| Option-C | High | More | Less |

### 3.3.4 Evaluation of Revised Schedules

The project managers can revise the task-worker allocation schedule with any one of the change options defined in the subsection-3.3.3 for a given change scenario. Since, the change options are defined or designed by considering different level of delays and perturbations in the existing schedule, each change option would generate a new schedule. Then, the question is which change options is the best to revise the schedule for a change scenario. In this section, we are going to discuss the evaluation of revised schedules to justify the selection of change options.

In our work, the quality of the revised schedule is evaluated based on two criteria namely, project delay and re-organization effort. Project delay is defined as the number of extended days beyond planned completion dates of the project. The key objective of the project manager is to deliver a quality product ensuring the timely completion of the project. Any project delay can imply additional cost and customer dissatisfaction. The project delay is initially triggered by the change scenario that directly causes the delay of some affected tasks. These delayed tasks can have a downstream effect that delays other tasks, leading to the delay of the entire project. For a resource constrained project scheduling problem where the number of resources is fixed and the project managers have little opportunity to hire new workers to replace the absent worker, sometimes project delay is unavoidable due to the unavailability of regular workers. Then, the subject is to examine which change options will lead to the minimum project delay in the revised schedules. In this context, project delay is defined as the number of extended days from the initial deadline of the project.

Besides the project delay, the project managers are also intended to minimize the deviation between the updated and the initial schedule. For a large and complex project, the original schedule is a result of deliberate efforts from many experts, including engineers, finance and administrative coordinators. In many cases, certain preparations have been made once the baseline schedule is established (ordering raw materials, acquiring necessary tools or equipment, fixing delivery dates, etc.). Thus, Modifications of the baseline schedule can incur different levels of re-organization. Such re-organizations may include notification of workers for changing their original tasks and intensive communications to re-structure the entire work flow of the project. Moreover, the capability and productivity of workers to the newly assign tasks is sometimes subject to proper training and adjustment with new working environment. In the above context, the project managers have a strong motive to minimize the re-organization effort to address a change scenario.

In our research, the re-organization effort is determined by the number of modified entries in the revised schedule. We consider three types of modification to revise the task-resource allocation schedule as discussed in subsection-3.3.3 and summarized in Table-3.1. The first category of modification refers to a task concerning a worker on one day is canceled due to the absent or for the delay of its precedence task. The second category of modification refers to change the previously assigned task of a worker with a affected task. The third category of modification is related with one day of the extra work after a worker completed initial assigned task. The number of modified entries demonstrates the level of the perturbation to the existing schedule as well as the solution robustness. Therefore, we use the number of total modified entries in the revised schedule to estimate the level of re-organization effort.

## 3.4 Conclusion

In this chapter, we have presented the model for task-resource allocation schedule along with possible relationships. Moreover, we also described the parameters for analyzing the change impact and strategies to revise the schedule. This general information focuses a basis that will be used in subsequent chapters. The next chapter presents the development of procedures for evaluating the change impact and selecting the appropriate change option to revise the schedule.

# Chapter 4

# Procedure for Schedule Change Management System

## 4.1 Introduction

This chapter presents the procedure of our decision based reactive scheduling system that responds to a change scenario for the absence of workers during project execution. The schedule is revised by any of the three change options as defined on the principal of preemptive or non-preemptive workload reassignment strategies. To select the appropriate change option, we classify the change scenario as *low*, *medium*, and *high* based on the change impact factors which are calculated on the basis of task sensitivity, workload sensitivity, workforce sensitivity and workload distribution policy (i.e., preemptive and non-preemptive).

To implement the procedure of our reactive scheduling system, the information from the Gantt chart and the task-resource allocation schedule ($SH$) is used. Specifically, the task-resource allocation schedule helps to find comprehensive information about absent workers, affected tasks, and to modify the schedule. The parameters (i.e., task sensitivity, workload sensitivity, and workforce sensitivity) discussed in Chapter 3 are used to calculate the change impact factor for a change scenario. The Gantt chart helps to measure the sensitivity of the tasks. The task-skill ($TS$) and resource-skill ($RS$) relationship matrices are used to find the potential replacement workers. The workload sensitivity can be easily extracted from the task-resource allocation schedule ($SH$) by finding the affected workload of each task. After identifying the change scenario (i.e., low, medium, and high), the task-resource allocation schedule is revised using appropriate change options. The task precedence matrix ($TP$) is used to respect the precedence constraints among the tasks during modification of the schedule. The Gantt chart is also updated for all sorts of changes in the task-resource allocation schedule.

In this chapter, we first present the proposed framework for our reactive scheduling system. We then explain the details of each components in the subsequence sections with necessary examples.

## 4.2 Framework for Decision Based Rescheduling

This section intends to briefly describe the procedure of the decision based rescheduling approach applying to handle change scenario in the project schedule. The framework for the proposed system is illustrated in the Fig. 4.1.

The procedure is triggered when a worker reports his absence for a specific period of time during the execution of the project. The procedure starts with the initial baseline schedule and the absent workers information. The information regarding the absent workers such as length of the absence, affected tasks and corresponding workloads are recognized by checking the task-resource allocation schedule to identify the change scenario properly. For the identified change scenario, the change impact factor is calculated on the basis of task sensitivity, workload sensitivity, workforce sensitivity and workload distribution policy (i.e., preemptive and non-preemptive). Then the multi-stage decision based change option selector module selects any three change options based on the impact of the change scenario.

The Option-A which is defined based on the non-preemptive workload distribution policy is picked for the low impact change scenario. On the other hand, the preemption based Option-B and Option-C are used to manipulate the medium and high impact change scenario respectively. Finally, the initial baseline schedule is modified with the selected change option to obtain the revised schedule.

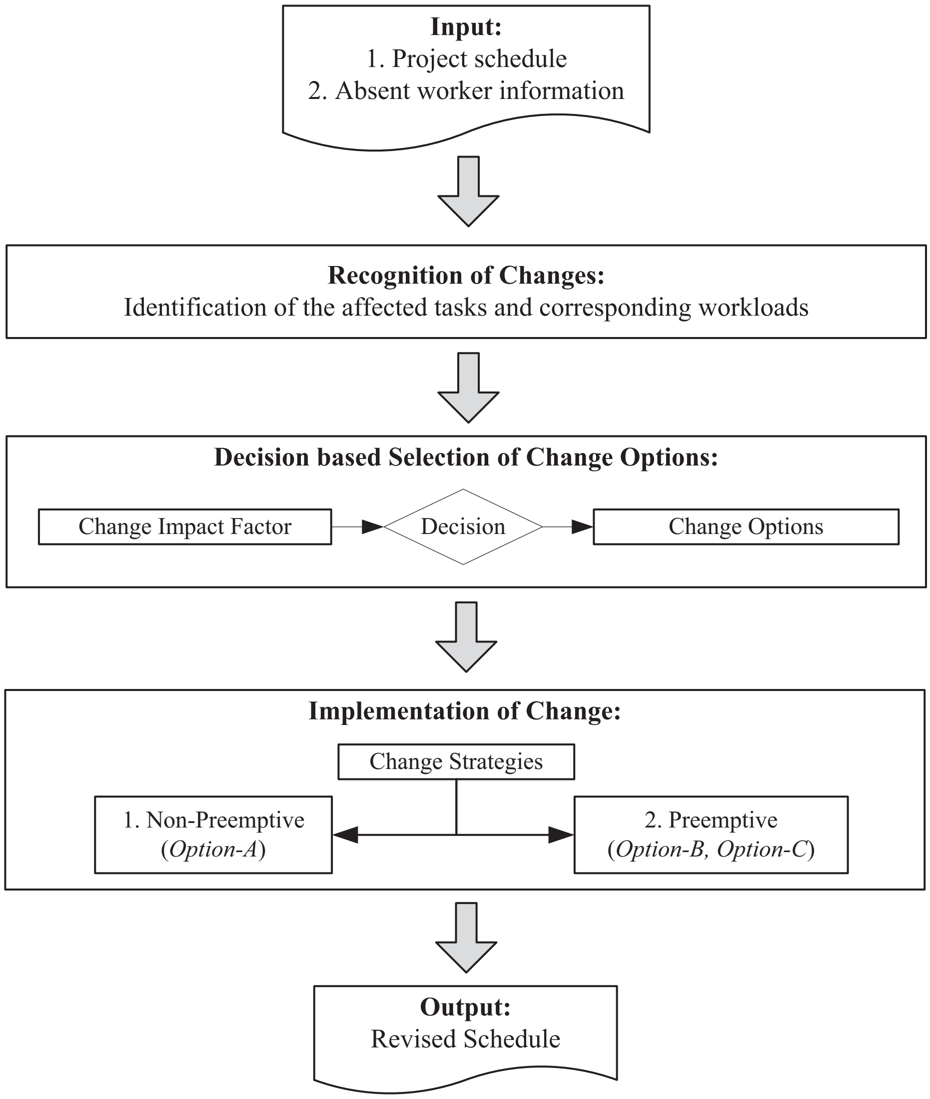



Figure 4.1: The framework for decision based rescheduling

To clarify the overall procedure, we bring out a small example situation where the procedure could be applied. The example is a portion of a large project schedule considering six tasks $t_1$, $t_2$, $t_3$, $t_4$, $t_5$ and $t_6$ with estimated effort 8, 14, 6, 6, 16, and 6 MD respectively. There are six workers $p_1$, $p_2$, $p_3$, $p_4$, $p_5$, and $p_6$ having different skills to carry out these tasks. Task $t_5$ can not start before the completion of both tasks $t_3$ and $t_4$. Task $t_6$ can only start after the completion of tasks $t_1$, $t_2$, and $t_5$. That means, these activities are subject to precedence constraints and the matrix in Eq. (4.1) presents the precedence relationship of among the tasks.

$$TP = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array} \tag{4.1}$$

$$tp_{ij} = \begin{cases} 1 & \text{the task } t_j \text{ is the successor of task } t_i, \\ 0 & \text{the task } t_j \text{ is not the successor of task } t_i. \end{cases}$$

The activity requires specific set of skills to be processed, that cannot be performed by all workers. If $S = \{s_1, s_2, , s_3\}$ be the set of required skills associated with different tasks, the skills requirement of task is represented by the Eq. (4.2).

$$TS = \begin{array}{c} \\ t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \tag{4.2}$$

$$ts_{ij} = \begin{cases} 1 & \text{task } t_i \text{ requires the skill } s_j, \\ 0 & \text{otherwise.} \end{cases}$$

The worker is not capable to do all the activities because of the specific skill requirements of some tasks. The matrix in Eq. (4.3) shows the worker with their acquired skills.

$$RS = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{array} \begin{array}{c} \begin{array}{ccc} s_1 & s_2 & s_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \end{array} \tag{4.3}$$

$$rs_{ij} = \begin{cases} 1 & \text{worker } p_i \text{ masters the skill } s_j, \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 4.2(a) shows the initial task-resource allocation schedule for this example and Fig. 4.2(b) illustrates the corresponding Gantt chart with necessary information.

## 4.3 Identification of Change Scenario

The identification of change scenario means to recognize the outstanding tasks and the corresponding amount of affected workload, when a worker reports for his absence for a specific period of time. For example, if worker $p_4$ reports to be absent from day $d_2$ to $d_4$ and worker $p_5$ will remain absent from $d_4$ to $d_6$ as shown in the task-resource allocation schedule in Fig. 4.3, then the affected tasks are $t_3$ and $t_5$. The change scenario in terms of affected tasks and corresponding workloads due to the absence of these two workers can be represented as: $T_a = \{(t_3, 4MD), (t_5, 2MD)\}$.

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P2 | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | T3 | T3 | T3 | T2 | T2 | T2 | | | | | |
| P5 | T4 | T4 | T3 | T3 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

(a) Task-reschedule allocation schedule

| Task | Effort | Duration | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 8MD | 5D | 1MD | 1MD | 2MD | 2MD | 2MD | F | F | F | F | | | |
| T2 | 14MD | 7D | 1MD | 1MD | 2MD | 2MD | 2MD | 3MD | 3MD | F | F | | | |
| T3 | 6MD | 4D | 1MD | 1MD | 2MD | 2MD | | | | | | | | |
| T4 | 6MD | 2D | 3MD | 3MD | F | F | | | | | | | | |
| T5 | 16MD | 5D | | | | | 2MD | 3MD | 3MD | 4MD | 4MD | | | |
| T6 | 6MD | 2D | | | | | | | | | | 3MD | 3MD | |

MD= Man-day, D=Day, F=Float    MD Critical task    MD Non-critical task

(b) Gantt chart

Figure 4.2: Baseline schedule for the example

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P2 | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | **(T3)** | **(T3)** | **(T3)** | T2 | T2 | T2 | | | | | |
| P5 | T4 | T4 | T3 | **(T3)** | **(T5)** | **(T5)** | T5 | T5 | T5 | T6 | T6 | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.3: Identification of change scenario.

## 4.4 Method for Assessment of Change Impact

As discussed in Chapter 3, the change impact factors is calculated on the basis of task sensitivity, workload sensitivity, workforce sensitivity and workload distribution policy (i.e., preemptive and non-preemptive). Based on the context, the overall impact for a change scenario is assessed by calculating the maximum outstanding workloads that is possible to reassign among the potential replacement workers within the *recovery window*. The recovery window indicates the maximum period of time to recover the affected workload without imposing delay in the project. More precisely, if it is possible to distribute the outstanding workloads among the potential replacement workers within the recovery window, it is assume that no delay will impose on the project duration for this change scenario. The *lower limit* of the recovery window is the first affected day of the outstanding task due to absence of the worker and the *upper limit* is the Latest Finish (LF) time of that task. In general, the recovery window for non-critical task is larger than critical task for same affected date as it includes the slack time of the task. Accordingly, if the first affected day for an outstanding task $t_i$ is $d_k$ and the Latest Finish (LF) time is $d_{k+w}$; where $k \in \{1, 2, ..., t\}$ and $\{w \in R : w > 0\}$, then the recovery window $R_w$ is defined as:

$$\text{Recovery window, } R_w = d_{k+w} - d_k \tag{4.4}$$

For instance, if worker $p_4$ while working on the critical task $t_3$ is reported to be absent from day $d_2$ to $d_4$, then the lower limit and upper limit of the recovery window for $t_3$ will be $d_2$ and $d_4$ respectively as shown in Fig. 4.4(a). On the other hand, the the lower limit and upper limit of the recovery window for the non-critical task $t_2$ will be $d_5$ and $d_9$ respectively while worker $p_3$ is reported to be absent from day $d_5$ to $d_7$ as illustrated in Fig. 4.4(b).

| Task | Effort | Duration | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 8MD | 5D | 1MD | 1MD | 2MD | 2MD | 2MD | F | F | F | F | | | |
| T2 | 14MD | 7D | 1MD | 1MD | 2MD | 2MD | 2MD | 3MD | 3MD | F | F | | | |
| T3 | 6MD | 4D | 1MD | 1MD | 2MD | 2MD | | | | | | | | |
| T4 | 6MD | 2D | 3MD | 3MD | F | F | | | | | | | | |
| T5 | 16MD | 5D | | | | | 2MD | 3MD | 3MD | 4MD | 4MD | | | |
| T6 | 6MD | 2D | | | | | | | | | | 3MD | 3MD | |

MD= Man-day, D=Day, F=Float   MD Critical task   MD Non-critical task

*Recovery Window*

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P2 | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | **(T3)** | **(T3)** | **(T3)** | T2 | T2 | T2 | | | | | |
| P5 | T4 | T4 | T3 | T3 | T5 | T5 | T5 | T5 | T5 | | | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

(a) Critical task ($t_3$)

| Task | Effort | Duration | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T1 | 8MD | 5D | 1MD | 1MD | 2MD | 2MD | 2MD | F | F | F | F | | | |
| T2 | 14MD | 7D | 1MD | 1MD | 2MD | 2MD | 2MD | 3MD | 3MD | F | F | | | |
| T3 | 6MD | 4D | 1MD | 1MD | 2MD | 2MD | | | | | | | | |
| T4 | 6MD | 2D | 3MD | 3MD | F | F | | | | | | | | |
| T5 | 16MD | 5D | | | | | 2MD | 3MD | 3MD | 4MD | 4MD | | | |
| T6 | 6MD | 2D | | | | | | | | | | 3MD | 3MD | |

MD= Man-day, D=Day, F=Float   MD Critical task   MD Non-critical task

*Recovery Window*

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P2 | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | **(T2)** | **(T2)** | **(T2)** | | | | | |
| P4 | T3 | T3 | T3 | T3 | T2 | T2 | T2 | | | | | |
| P5 | T4 | T4 | T3 | T3 | T5 | T5 | T5 | T5 | T5 | | | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

(b) Non-critical task ($t_2$)

Figure 4.4: Recovery window.

Table 4.1: Classification of change impact

| **Redistribution Policy** | **Impact Factor** | **Change Impact** |
|---|---|---|
| **Non-preemptive** | $I \leq 1$ | Low |
| | $I > 1$ | Medium or High |
| **Preemptive** | $I \leq 1$ | Medium |
| | $I > 1$ | High |

In the above context, we summarize that the impact of any change scenario is highly correlated to the number of potential replace workers, and how much outstanding workload can be assigned to them without interrupting their present work or by interrupting the present work within the recovery window, $R_w$. Thus, the change impact factor ($I$) is defined as the ratio of the affected workloads $w_i^a$ and maximum distributable workloads $w_i^d$ of task $t_i$ among the potential replacement workers within the recovery window $R_w$ as:

$$\text{Change Impact factor, } I = \left| \frac{w_i^a}{w_i^d} \right|_{R_w} \tag{4.5}$$

$$\text{where,} \quad i \in \{1, 2, ..., n\}$$

In the Eq. (4.5), the affected workloads $w_i^a$ is directly linked with the workload sensitivity. On the other hand, the maximum distributable workloads $w_i^d$ is related to workforce sensitivity and workload distribution policies (i.e., preemptive or non-preemptive). Lastly, the recovery window $R_w$ is only connected with task sensitivity. Based on the preemptive and non-preemptive workload distribution policy and the corresponding impact factor value, we classify the change impact as *high*, *medium* and *low* that summarizes in the Table-4.1. We assume that if it is possible to distribute the affected workload among the potential replacement workers without interrupting their present work within the recovery window, the scenario has low impact, otherwise it may have medium or high impact.

## 4.5 Procedure for the Selection of Change Option

As we already mentioned, the main emphasis of this research work is to develop a decision based rescheduling system in response to workers absence during project execution. The major goal of the rescheduling system is to minimize the delay and perturbation in the revised schedule. The delay is highly dependent on the flexibility of the change options during schedule revision. A change option with high flexibility may impose shorter delay but cause higher perturbation in schedule and vice-versa. Based on this principle, in Chapter 3 we defined three change options named as: Option-A, Option-B and Option-C and summarize their characteristics in terms of schedule revision flexibility, perturbation or disturbance of the remaining workers and delay. The schedule possibly can be revised by any one of the three change options. To select the appropriate change option, this research classifies the change scenario as low, medium, and high based on the change impact factors. The change option-A is more appropriate for low impact change scenario and option-B is suitable to handle change scenario having medium impact value. For high impact change scenario, we should use use option-C. We describe two procedure for Option-C. The general procedure of Option-C is used to revise the schedule in response to high impact change scenario by distributing the affected workload within the recovery window $R_w$ without imposing delay. On the other hand, the extended procedure of option-C is designed to distribute the affected workload within the recovery window $R_w$ as well as after the recovery window $R_w$ by delaying the start time of the downstream tasks.

In the above context, we propose a multi-stage decision based change option selection method as illustrated in Fig. 4.5. The method starts with the change scenario for a single task and then selects the change options based on the value of change impact factor in three decision points. The step-by-step working procedure of this method is described below:

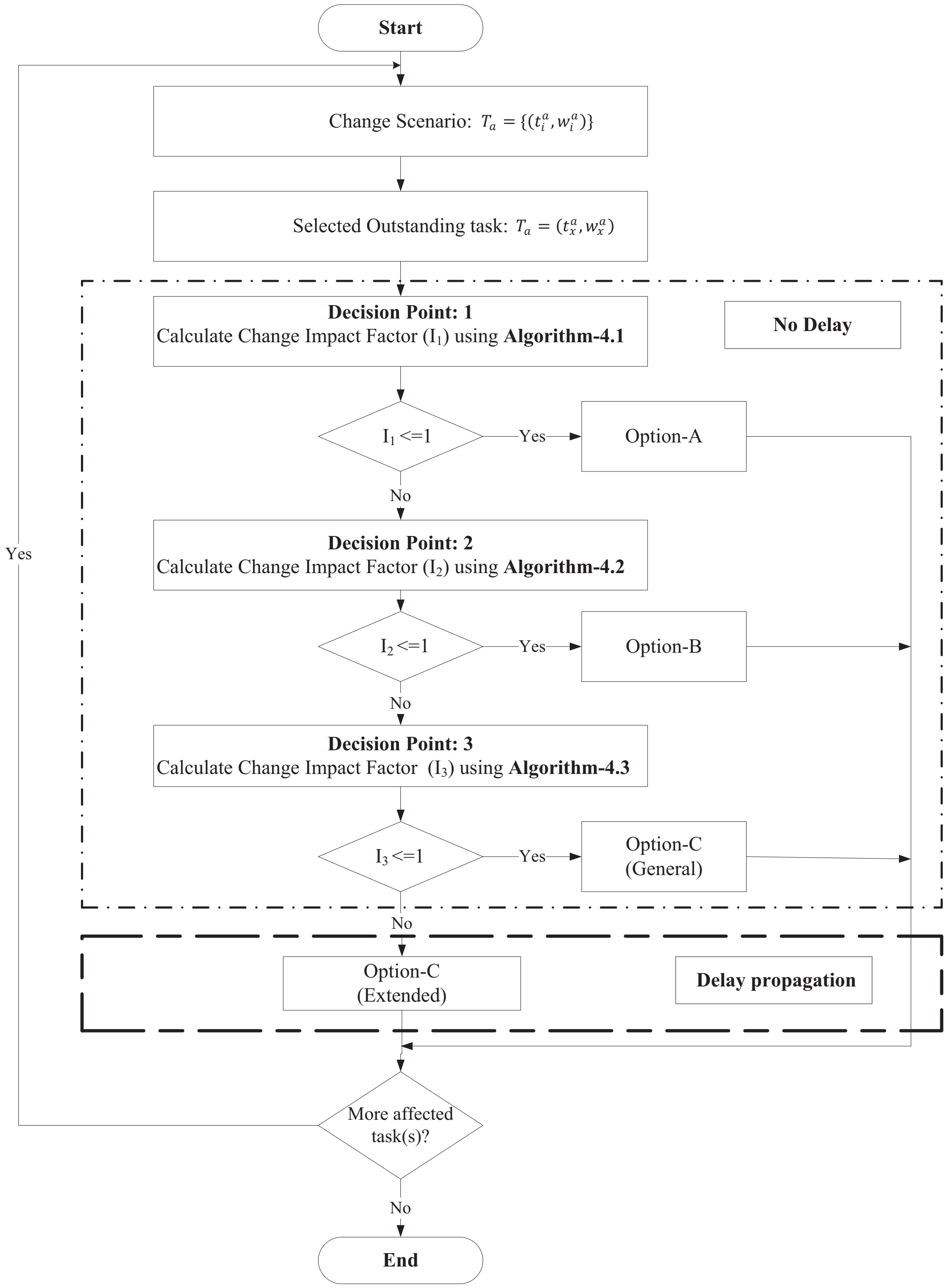



Figure 4.5: The multi-stage decision based change option selection method.

**Selection of Change Option-A:** The decision based change option selection method at first decision point or level calculates the change impact factor ($I_1$). The procedure for calculating the change impact factor ($I_1$) using non-preemptive distribution policy within the recovery window $R_w$ is described in Algorithm-4.1. This decision is made on the basis of the change impact factor ($I_1$). If the value of the change impact factor ($I_1$) is less than 1, the impact level is considered as low and Option-A is selected to revise the schedule. Otherwise, it will move to the next decision point.

---

**Algorithm 4.1** Procedure for the calculation of change impact factor ($I_1$)

***Step 1:*** Find the recovery window $R_w$ for the selected outstanding task $t_x^a$ (i.e., change scenario, $T_a = \{(t_x^a, w_x^a)\}$) by identifying its first affected day $d_k$ and its Latest Finish (LF) time $d_{k+w}$.

$$R_w = d_{k+w} - d_k$$

***Step 2:*** Find the set of potential replacement workers $P_r$ those are working in any one of the following two modes in the task-resource allocation schedule ($SH$).

**Mode-1:** $sh_{jk} = t_x \ \ \& \ \ sh_{j(k+1)} = t_y$.

**Mode-2:** $sh_{jk} = t_x \ \ \& \ \ sh_{j(k+1)} = \phi$.

where, $x, y \in \{1, 2, ..., n\}; j \in \{1, 2, ..., m\}$and, $k \in \{1, 2, ..., t\}$

and, $t_y$ = non-critical task and, $\phi$ = off period of the worker

***Step 3:*** Find the maximum outstanding workloads that can be distributed among the potential replacement workers within the recovery window $R_w$.

$$\text{Maximum redistributable workload,} \quad w_x^d = \sum_{p=1}^{|P_r|} \sum_{d=d_k}^{d_{k+w}} w_{pd}$$

where, $w_{pd} =$ Non-critical workload of worker $p$ and $p \in P_r$

***Step 4:*** Calculate the change impact factor as the ratio of the affected workloads $w_x^a$ and maximum distributable workloads $w_x^d$ of task $t_x^a$ within the recovery window $R_w$ as:

$$\text{Change impact factor, } I_1 = \left| \frac{w_x^a}{w_x^d} \right|_{R_w}$$

***Step 5:*** Finished

---

| | **D1** | **D2** | **D3** | **D4** | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | T4 | **T1** | **T1** | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P2 | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | T3 | T3 | T3 | T2 | T2 | T2 | | | | | |
| P5 | **(T4)** | **(T4)** | T3 | T3 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P6** | T4 | T4 | **T2** | **T2** | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.6: Change scenario for the absence of worker $p_5$.

For instance, suppose worker $p_5$ is absent for days $d_1$ and $d_2$ and the change scenario due to his absence is $(t_4, 2MD)$ as shown in the task-resource allocation schedule in Fig. 4.6. The change impact factor $(I_1)$ for this scenario is calculated using Algorithm-4.1 as follows:

**Step 1:** The first affected date and Latest Finish (LF) time of the outstanding task $t_4$ is $d_1$ and $d_4$ respectively. So, the range of the recovery window $R_w$ is $d_1$ to $d_4$.

**Step 2:** Only the worker $p_1$ and $p_6$ can work on the affected task $t_4$ after the completion of the already started tasks at hand. The potential replacement workers with in the recovery window $R_w$ are: $P_r = \{p_1, p_6\}$.

**Step 3:** Within the recovery window $R_w$ both $p_1$ and $p_6$ can share 2 workloads as shown in bold face in Fig. 4.6. The total maximum distributable workload, $w_4^d$ =2+2 =4.

**Step 4:** The change impact factor, $I_1 = 2/4 = 0.5$; which is less than 1.

Since the change impact factor $I_1$ in this decision point is less than 1, Option-A will be selected to revise the schedule and decision point 2 and 3 are ignored.

**Selection of Change Option-B:** When the change impact factor $(I_1)$ in decision point-1 is greater than 1, the decision based change option selection method moves to next decision point to check whether Option-B is suitable or not to revise the schedule without delaying the task duration. This decision is made on the basis of the change impact factor $(I_2)$. Algorithm-4.2 gives the details procedure to calculate the change impact factor $(I_2)$ within the recovery window $R_w$ by interrupting the present work of a *single* replacement worker. If the value of the change impact factor $(I_2)$ is less than

**Algorithm 4.2** Procedure for the calculation change impact factor ($I_2$)

***Step 1:*** Find the recovery window $R_w$ for the selected outstanding task $t_x^a$ (i.e., change scenario $T_a = \{(t_x^a, w_x^a)\}$) by identifying its first affected day $d_k$ and its Latest Finish (LF) time $d_{k+w}$.

$$R_w = d_{k+w} - d_k$$

***Step 2:*** Find the set of potential replacement workers $P_r$ for task $t_x^a$ within the recovery window $R_w$ those satisfy the following task-resource skill requirement constraints through the mapping between task-skill ($TS$) and resource-skill ($RS$) matrices.

$$ts_{xq} \geq rs_{jq}$$
$$\text{where, } x \in \{1, 2, ..., n\};\ j \in \{1, 2, ..., m\} \text{ and, } q \in \{1, 2, ..., p\}$$

***Step 3:*** Find the maximum outstanding workloads that can be distributed to *one* of the potential replacement workers within the recovery window $R_w$.

$$\text{Maximum redistributable workload, } w_x^d = \max_{p=1,2,...|P_r|} \left\{ \sum_{d=d_k}^{d_{k+w}} w_{pd} \right\}$$

$$\text{where, } w_{pd} = \text{ Non-critical workload excluding } t_x^a \text{ of worker } p \text{ and } p \in P_r$$

***Step 4:*** Calculate the change impact factor as the ratio of the affected workloads $w_x^a$ and maximum distributable workloads $w_x^d$ of task $t_x^a$ within the recovery window $R_w$ as:

$$\text{Change impact factor, } I_2 = \left| \frac{w_x^a}{w_x^d} \right|_{R_w}$$

***Step 5:*** Finished

1, then Option-B is selected to revise the schedule. Otherwise, it will move to the next decision point to check the validity of the other options.

For instance, suppose worker $p_4$ is reported to be absent from day $d_2$ to $d_4$ and the change scenario due to his absence is $(t_3, 3MD)$ as shown in the task-resource allocation schedule Fig. 4.7. The change impact factor ($I_2$) for this change scenario is calculated using procedure of Algorithm-4.2 as follows:

**Step 1:** The range of the recovery window $R_w$ for the outstanding task is $d_2$ to $d_4$ as the first affected date and Latest Finish (LF) time of task $t_3$ is $d_2$ and $d_4$ respectively.

**Step 2:** The set of potential replacement workers having skills to work on task $t_3$ is found through the mapping of task-skill ($TS$) and resource-skill ($RS$) matrices as $P_r = \{p_1, p_2, p_3, p_4, p_5\}$.

| | D1 | **D2** | **D3** | **D4** | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | **T4** | **T1** | **T1** | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P2** | T1 | **T1** | **T1** | **T1** | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| **P3** | T2 | **T2** | **T2** | **T2** | T2 | T2 | T2 | | | | | |
| **P4** | T3 | **(T3)** | **(T3)** | **(T3)** | T2 | T2 | T2 | | | | | |
| **P5** | T4 | **T4** | T3 | T3 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.7: Change scenario for the absence of worker $p_4$.

**Step 3:** Within the recovery window $R_w$, the potential replacement workers can share workloads as follows (Fig. 4.7 shows the shareable workload is in bold face ):

$p_1$ can share max 3 workloads

$p_2$ can share max 3 workloads

$p_3$ can share max 3 workloads

$p_4$ can share max 0 workload

$p_5$ can share max 1 workload

Maximum redistributable workload, $w_3^d$=max{3, 3, 3, 0, 1} = 3.

**Step 4:** The change impact factor, $I_2 = 3/3 = 1$; which is equal to 1.

Since the change impact factor $I_2$ in this decision point is equal than 1, Option-B is appropriate to revise the schedule and decision point-3 can easily be ignored.

**Selection of Change Option-C:** When the change impact factor ($I_2$) at decision point-2 is greater than 1, the decision based change option selection method moves to decision point-3 to check whether the Option-C can be applied without imposing delay or delay is unavoidable due to this change scenario.

The decision is made on the basis of the change impact factor ($I_3$). Algorithm-4.3 describes the details procedure to calculate the change impact factor ($I_3$) within the recovery window $R_w$ by interrupting the present work of multiple replacement workers. If the calculated value of the change impact factor ($I_3$) is larger than 1, we consider that the scenario has high impact with unavoidable delay.

**Algorithm 4.3** Procedure for calculation change impact factor ($I_3$)

***Step 1:*** Find the recovery window $R_w$ for the selected outstanding task $t_x^a$ (i.e., scenario $T_a = \{(t_x^a, w_x^a)\}$) by identifying its first affected day $d_k$ and its Latest Finish (LF) time $d_{k+w}$.

$$R_w = d_{k+w} - d_k$$

***Step 2:*** Find the set of potential replacement workers $P_r$ for task $t_x^a$ within the recovery window $R_w$ those satisfy the following task-resource skill requirement constraints through mapping between task-skill ($TS$) and resource-skill ($RS$) matrices.

$$ts_{xq} \geq ts_{jq}$$

$$\text{where, } x \in \{1, 2, ..., n\};\ j \in \{1, 2, ..., m\} \text{ and, } q \in \{1, 2, ..., p\}$$

***Step 3:*** Find the maximum outstanding workloads that can be distributed among the potential replacement workers within the recovery window $R_w$.

$$\text{Maximum redistributable workload,} \quad w_x^d = \sum_{p=1}^{|P_r|} \sum_{d=d_k}^{d_{k+w}} w_{pd}$$

$$\text{where, } w_{pd} = \text{ Non-critical workload excluding } t_x^a \text{ of worker } p \text{ and } p \in P_r$$

***Step 4:*** Calculate the change impact factor as the ratio of the affected workloads $w_x^a$ and maximum distributable workloads $w_x^d$ of task $t_x^a$ within the recovery window $R_w$ as:

$$\text{Change impact factor, } I_3 = \left| \frac{w_x^a}{w_x^d} \right|_{R_w}$$

***Step 5:*** Finished

**Case-1 (without delay):** For instance, suppose worker $p_4$ is reported to be absent from day $d_2$ to $d_4$ and the change scenario due to his absence is $(t_3, 3MD)$. Besides, worker $p_4$ is also informed to be absent for days $d_3$ and $d_4$ and the change scenario due to his absence is $(t_3, 2MD)$. Hence, the overall change scenario $(t_3, 5MD)$ for the absence of two workers $p_4$ and $p_5$ is shown in the task-resource allocation schedule in Fig. 4.8. The change impact factor ($I_3$) for this change scenario is calculated using the procedure of Algorithm-4.3 as follows:

**Step 1:** The first affected date and Latest Finish (LF) time of the outstanding task $t_3$ is $d_2$ and $d_4$ respectively. The range of the recovery window $R_w$ is $d_2$ to $d_4$.

**Step 2:** The set of potential replacement workers having skills to work on task $t_3$ is $P_r = \{p_1, p_2, p_3, p_4, p_5\}$.

**Step 3:** Within the recovery window $R_w$, the potential replacement worker can share workloads as follows (Fig. 4.8 shows shareable workload in bold face):

$p_1$ can share max 3 workloads

$p_2$ can share max 3 workloads

$p_3$ can share max 3 workloads

$p_4$ can share max 0 workloads

$p_5$ can share max 1 workloads

The total maximum redistributable workload, $w_3^d$ =(3+3+3+0+1)= 10.

**Step 4:** The change impact factor, $I_3 = 5/10 = 0.5$; which is less than 1.

Since, the change impact factor $I_3$ is less than 1, the Option-C (general) can be used to revise the schedule.

| | D1 | **D2** | **D3** | **D4** | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | **T4** | **T1** | **T1** | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P2** | T1 | **T1** | **T1** | **T1** | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| **P3** | T2 | **T2** | **T2** | **T2** | T2 | T2 | T2 | | | | | |
| **P4** | T3 | **(T3)** | **(T3)** | **(T3)** | T2 | T2 | T2 | | | | | |
| **P5** | T4 | **T4** | **(T3)** | **(T3)** | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.8: Change scenario for the absence of workers $p_4$ and $p_5$.

**Case-2 (Delay propagation):** Here, we consider another example where worker $p_4$ will remain absent from day $d_6$ to $d_9$ while working on task $t_5$. The change scenario due to his absence is $(t_5, 4MD)$ as shown in the task-resource allocation schedule in Fig. 4.9. The change impact factor ($I_3$) for this change scenario is calculated using the procedure of Algorithm-4.3 as follows:

**Step 1:** The first affected date and Latest Finish (LF) time of the outstanding task $t_5$ is $d_6$ and $d_9$ respectively. The range of the recovery window $R_w$ is $d_6$ to $d_9$.

**Step 2:** The set of potential replacement workers having skills to work on task $t_5$ is $P_r = \{p_1, p_2, p_5, p_6\}$.

**Step 3:** Within the recovery window $R_w$, the potential replacement worker can share workloads as follows (Fig. 4.9 shows shareable workload in bold face):

- $p_1$ can not share any workload as he is assigned the same affected task $t_5$ during this recovery window.

- $p_2$ can share maximum 2 workloads on $d_6$ and $d_7$

- $p_4$ can not share any workload as he will remain absent during this period.

- $p_5$ also can not share any workload as he is assigned the same affected task $t_5$ during this period.

The total maximum redistributable workload, $w_5^d$ =(0+2+0+0)= 2.

**Step 4:** The change impact factor, $I_3 = 4/2 = 2$; which is greater than 1.

Since, the change impact factor $I_3$ is greater than 1, the Option-C (extended) should be used to revise the schedule.

| | D1 | D2 | D3 | D4 | D5 | **D6** | **D7** | **D8** | **D9** | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P2** | T1 | T1 | T1 | T1 | T1 | **T2** | **T2** | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | T3 | T3 | T3 | T2 | T2 | T2 | | | | | |
| **P5** | T4 | T4 | T3 | T3 | T5 | **(T5)** | **(T5)** | **(T5)** | **(T5)** | T6 | T6 | |
| **P6** | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.9: Change scenario for the absence of worker $p_5$.

## 4.6 Implementation of Change Option

After selecting the appropriate change option using the selection method as described in section 4.5, the baseline schedule is modified with the selected change option to obtain the revised schedule. The three change options (i.e., Option-A, Option-B, and Option-C) are developed on the basis of preemptive and non-preemptive workload distribution strategy as already mentioned in Chapter 3. In this section we are going to describe four procedures based on the definition of Option-A, Option-B, and Option-C to revise the task-resource allocation schedule in a systematic way. Algorithm-4.4 describes the step-by-step procedures for Option-A and Algorithm-4.5 describes the overall method for

Option-B to revise the task-resource allocation schedule. We define the Option-C by two procedures based on the workload distribution policy within the recovery window (i.e., $R_w$) or after the recovery window. The procedure described in Algorithm-4.6 is based on the definition of Option-C to distribute the workload among the multiple potential replacement workers within the recovery window without imposing delay. In contrast, the method defined in Algorithm-4.7 is also expanded from the definition of Option-C to distribute the workload among the multiple potential replacement workers both within the recovery window and after the recovery window considering the delay propagation. The working principles of the four procedures are described below with necessary illustrations.

**Implementation of Change Option-A:** The step-by-step procedure based on the principle (i.e., non-preemptive workload distribution policy) as described in Chapter 3 is described in Algorithm-4.4. To clarify the working principle of this procedure, we consider the change scenario $(t_4, 2MD)$ due to the absence of worker $p_5$ as shown in the task-resource allocation schedule Fig. 4.6. We can revise the schedule using the procedure of Option-A as described in Algorithm-4.4 as follows:

***Step 1:*** The first affected date and Latest Finish (LF) time of the outstanding task $t_4$ is $d_1$ and $d_4$ respectively. So the range of the recovery window $R_w$ is $d_1$ to $d_4$.

***Step 2:*** within the recovery window $R_w$ that is for days $d_1$ to $d_4$.

- Days $d_1$ and $d_2$: No one is working either mode-1 or mode-2 in the task-resource allocation schedule $(SH)$.

- Day $d_3$: The potential replacement workers are $p_1$ and $p_6$ and they are working in mode-1 in the task-resource allocation schedule $(SH)$. We can assign 2MD workloads of $t_4$ to them. Hence, the remaining affected workload becomes, $w_x^a$=2-2=0. The revised schedule is shown in Fig. 4.10

  Newly affected task, $T_{new}^a$={$(t_1, 1MD)$,$(t_2, 1MD)$}

  Since, $w_x^a$=0, distribution of the affected workload for task $t_4$ is completed and stop the procedure.

**Algorithm 4.4** Procedure for the change Option-A

***Step 1:*** Find the recovery window $R_w$ for the selected outstanding task $t_x^a$ (i.e., scenario $T_a = \{(t_x^a, w_x^a)\}$) by identifying its first affected day $d_k$ and its Latest Finish (LF) time $d_{k+w}$.

$$R_w = d_{k+w} - d_k$$

***Step 2:*** For day $d = d_k$ to $d_{k+w}$ perform the following operations on each day.

a) Find the set of potential replacement workers $P_r$ those are working in any one of the following two modes in the task-resource allocation schedule ($SH$).

**Mode-1:** $sh_{jk} = t_x \;\; \& \;\; sh_{j(k+1)} = t_y.$

**Mode-2:** $sh_{jk} = t_x \;\; \& \;\; sh_{j(k+1)} = \phi.$

where, $x, y \in \{1, 2, ..., n\}; j \in \{1, 2, ..., m\}$and, $k \in \{1, 2, ..., t\}$

and, $t_y$ = non-critical task and, $\phi$ = off period of the worker

b) Assign the affected workload $w_x^a$ of $t_x^a$ to the replacement workers $P_r$ and deduct the reassigned workload $w_x^d$ from $w_x^a$ accordingly and keep the record of newly affected tasks for future processing.

$$w_x^a \leftarrow (w_x^a - w_x^d)$$
$$T_{new}^a = \{(t_i^a, w_i^a)\}$$

c) if $w_x^a = 0$ then stop; otherwise continue.

***Step 3:*** Finished

| | **D1** | **D2** | **D3** | **D4** | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | T4 | **T4(T1)** | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P2 | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | T3 | T3 | T3 | T2 | T2 | T2 | | | | | |
| P5 | **(T4)** | **(T4)** | T3 | T3 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P6** | T4 | T4 | **T4(T2)** | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.10: Implementation of change Option-A

**Implementation of Change Option-B:** Algorithm-4.5 describes the overall procedure based on the principle of preemptive workload distribution policy as described in Chapter 3 to revise the task-resource schedule using Option-B with one replacement worker. For instance, we consider the change scenario $(t_3, 3MD)$ due to the absent of worker $p_4$ as shown in the task-resource allocation schedule in Fig. 4.7. We can revise the schedule using the procedure of Option-B as described in Algorithm-4.5 for this change scenario as follows:

***Step 1:*** The range of the recovery window $R_w$ is $d_2$ to $d_4$ as the first affected date and Latest Finish (LF) time of the outstanding task $t_3$ is $d_2$ and $d_4$ respectively.

***Step 2:*** At first, we search for the potential replacement workers through the mapping of task-skill ($TS$) and resource-skill ($RS$) matrix as defined in Eq. (4.2) and Eq. (4.3). Hence, the set of potential replacement workers having skills to work on task $t_3$ is $P_r = \{p_1, p_2, p_3, p_4, p_5\}$.

***Step 3:*** Within the recovery window, the potential replacement worker can share workloads as follows (Fig. 4.7 ):

$p_1$ can share max 3 workloads

$p_2$ can share max 3 workloads

$p_3$ can share max 3 workloads

$p_4$ can share max 0 workloads

$p_5$ can share max 1 workloads

Finally, we sort the list of the potential replacement workers based on the amount of workload they can share in descending order as follows:

$P_r$=$\{(p_1, 3MD), (p_2, 3MD), (p_3, 3MD), (p_5, 1MD), (p_4, 0MD)\}$

a) Although workers $p_1$, $p_2$, and $p_3$ can share the same amount of additional workload (i.e, 3MD), we assign lower priority to worker $p_1$ as he is working on two tasks (e.g., $t_4$ and $t_1$) compare to workers $p_2$ and $p_3$ who are working only single task within the recovery window.

b) Both workers $p_2$ and $p_3$ can share same amount of additional workload (i.e., 3MD), we select the worker $p_3$ as the most suitable replacement worker because the task (e.g., $t_2$) assigned to him has higher earliest finish (EF) time compare to task $t_1$ of $p_2$.

**Algorithm 4.5** Procedure for the change Option-B

***Step 1:*** Find the recovery window $R_w$ for the selected outstanding task $t_x^a$ (i.e., scenario $T_a = \{(t_x^a, w_x^a)\}$) by identifying its first affected day $d_k$ and its Latest Finish (LF) time $d_{k+w}$.

$$R_w = d_{k+w} - d_k$$

***Step 2:*** Find the set of potential replacement workers $P_r$ for task $t_x^a$ within the recovery window $R_w$ those satisfy the following task-resource skill requirement constraints through mapping between task-skill ($TS$) and resource-skill ($RS$) matrices.

$$ts_{xq} \geq rs_{jq}$$
$$\text{where, } x \in \{1, 2, ..., n\};\ j \in \{1, 2, ..., m\} \text{ and, } q \in \{1, 2, ..., p\}$$

***Step 3:*** Find the maximum additional workload $w_x^d$ that can be assigned to each potential replacement worker in $P_r$ within the recovery window $R_w$ and *sort* them in descending order based on the amount of workload.

$$P_r = SORT\{(p_r, w_x^d)\}$$

where, $w_d$ = Non-critical workload *excluding* the same affected workload of task $t_x^a$

a) In case of a tie, that is when two or more replacement workers have the same workload value, priority is given to those working least number of tasks within the recovery window $R_w$.

b) For further tie, priority is given to those working on task that has the largest Early Finish (EF) time.

***Step 4:*** Assign the affected workload $w_x^a$ of $t_x^a$ to the sorted *first* worker in $P_r$ and subtract the reassigned workload $w_x^d$ from $w_x^a$ accordingly and keep the record of newly affected tasks for future processing.

$$w_x^a \leftarrow (w_x^a - w_x^d)$$
$$T_{new}^a = \{(t_i^a, w_i^a)\}$$

***Step 5:*** Finished

***Step 4:*** Finally, the affected workload $(t_3, 3MD)$ is assigned to worker $p_3$ as shown in Fig. 4.11 and record the newly affected tasks. Newly affected workload, $T_{new}^a$=$\{(t_2, 3MD)\}$

***Step 5:*** Finished.

| | D1 | **D2** | **D3** | **D4** | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P2** | T1 | T1 | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| **P3** | T2 | **T3(T2)** | **T3(T2)** | **T3(T2)** | T2 | T2 | T2 | | | | | |
| **P4** | T3 | **(T3)** | **(T3)** | **(T3)** | T2 | T2 | T2 | | | | | |
| **P5** | T4 | T4 | T3 | T3 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.11: Implementation of change Option-B

**Implementation of Change Option-C (General):** The procedure defined in Algorithm-4.6 is extended from the principle of preemptive workload distribution policy as described in Chapter 3 to distribute the affected workload among the multiple potential replacement workers within the recovery window $R_w$. For the change scenario $(t_3, 5MD)$ due to the absent of workers $p_4$ and $p_5$ as shown in the task-resource allocation schedule (Fig. 4.8), we can revise the schedule using the procedure of Algorithm-4.6 without imposing delay as follows:

***Step 1:*** The range of the recovery window $R_w$ is $d_2$ to $d_4$ as the first affected date and latest finish (LF) time of the outstanding task $t_3$ is $d_2$ and $d_4$ respectively.

***Step 2:*** within the recovery window $R_w$ that is for days $d_2$ to $d_4$.

- **Day** $d_2$**:** a) we search for the potential replacement workers through the mapping of task-skill $(TS)$ and resource-skill $(RS)$ matrix as defined in Eq. (4.2) and Eq. (4.3). The set of potential replacement workers having skills to work on task $t_3$ is $P_r = \{p_1, p_2, p_3, p_4, p_5\}$.

  b) Since, worker $p_4$ is not available, we exclude him from $P_r$ and the updated list is: $P_r = \{p_1, p_2, p_3, p_5\}$.

  c) Now, we assign 4MD workloads of $t_3$ to the workers $p_1$, $p_2$, $p_3$, $p_5$. Hence, the remaining affected workload becomes, $w_x^a$=5-4=1.

  Newly affected task, $T_{new}^a$={$(t_1, 1MD)$,$(t_2, 1MD)$,$(t_4, 2MD)$}

  d) Since, $w_x^a \neq 0$, we distribute the remaining workload on the next day.

- **Day** $d_3$**:** a) The set of potential replacement workers having skills to work on task

**Algorithm 4.6** Procedure for the change Option-C (General)

***Step 1:*** Find the recovery window $R_w$ for the selected outstanding task $t_x^a$ (i.e., scenario $T_a = \{(t_x^a, w_x^a)\}$) by identifying its first affected day $d_k$ and its Latest Finish (LF) time $d_{k+w}$.

$$R_w = d_{k+w} - d_k$$

***Step 2:*** For day $d = d_k$ to $d_{k+w}$ perform the following operations on each day.

a) Find the set of potential replacement workers $P_r$ for task $t_x^a$ within the recovery window $R_w$ those satisfy the following task-resource skill requirement constraints through mapping between task-skill ($TS$) and resource-skill ($RS$) matrices:

$$ts_{xq} \geq rs_{jq}$$
$$\text{where, } x \in \{1, 2, ..., n\};\ j \in \{1, 2, ..., m\} \text{ and, } q \in \{1, 2, ..., p\}$$

b) Exclude those workers from $P_r$ whose are not available, or working on critical tasks or the outstanding task $t_x^a$ .

c) Assign the affected workload $w_x^a$ of $t_x^a$ to the replacement workers $P_r$ and deduct the reassigned workload $w_x^d$ from $w_x^a$ accordingly and keep the record of newly affected tasks for future processing.

$$w_x^a \leftarrow (w_x^a - w_x^d)$$
$$T_{new}^a = \{(t_i^a, w_i^a)\}$$

d) if $w_x^a = 0$ then stop; otherwise continue.

***Step 3:*** Finished

$t_3$ is $P_r = \{p_1, p_2, p_3, p_4, p_5\}$.

b) Since, workers $p_4$ and $p_5$ are not available, we exclude them from $P_r$ and the final list is: $P_r = \{p_1, p_2, p_3\}$.

c) Now, we assign 1MD workloads of $t_3$ to the workers $p_1$. The remaining affected workload becomes, $w_x^a$=1-1=0.

Newly affected task, $T_{new}^a$={$(t_1, 2MD)$,$(t_2, 1MD)$,$(t_4, 2MD)$}

d) As $w_x^a$=0, the distribution of the affected workload for task $t_3$ is completed and stop the procedure. The revised schedule is shown in Fig. 4.12.

***Step 3:*** Finished.

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | **T3(T4)** | **T3(T1)** | T1 | T1 | T5 | T5 | T5 | T5 | T6 | T6 | |
| **P2** | T1 | **T3(T1)** | T1 | T1 | T1 | T2 | T2 | T5 | T5 | T6 | T6 | |
| **P3** | T2 | **T3(T2)** | T2 | T2 | T2 | T2 | T2 | | | | | |
| **P4** | T3 | **(T3)** | **(T3)** | **(T3)** | T2 | T2 | T2 | | | | | |
| **P5** | T4 | **T3(T4)** | **(T3)** | **(T3)** | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |
| P6 | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | T6 | T6 | |

Figure 4.12: Implementation of change Option-C (General)

**Implementation of Change Option-C (Extended):** The procedure described in Algorithm-4.7 is also expanded from the definition of Option-C which ables to distribute the affected workload among the multiple potential replacement workers within the recovery window (i.e., $R_w$) as well as after the recovery window by delaying the start time of the successor tasks as a consequence of delay propagation.

**Algorithm 4.7** Procedure for Change Option-C (Extended)

***Step 1:*** First distribute the affected workload of the $w_x^a$ of the task $t_x^a$ within the recovery window $R_w$ using the procedure of Algorithm-4.6.
***Step 2:*** Distribute the remaining affected workload of the $w_x^a$ for this task $t_x^a$ after the recovery window $R_w$ using the following steps:

a) Find the potential replacement workers after the recovery window $R_w$ and first assign the remaining affected workload $w_x^a$ to the workers who are working on the immediate successor of the outstanding task $t_x^a$ and then the remaining workers *until* $w_x^a = 0$.

b) If the workers who are working on the successor task is not eligible to work on the affected task $t_x^a$, then successor task needed to be canceled up to the date the affected task is completed.

c) Finally the *finished* time of $t_x^a$ as well as the *start* time of the successor of $t_x^a$ is updated.

***Step 3:*** Finished

For the change scenario $(t_5, 4MD)$ due to the absent of workers $p_5$ for days $d_6$ to $d_9$ as shown in the task-resource allocation schedule in Fig. 4.9, we can revise the schedule using the procedure of Option-C (Extended) defined in Algorithm-4.7 considering the delay propagation as follows:

***Step 1:*** First, the affected workload of the task $t_5$ is distributed within the recovery window $R_w$ using Algorithm-4.6 as follows:

- The upper and lower limit of the recovery window $R_w$ is $d_6$ to $d_9$ based on the first affected date and Latest Finish (LF) time of the outstanding task $t_5$.
- The set of potential replacement workers having skills to work on task $t_5$ is $P_r = \{p_1, p_2, p_5, p_6\}$. But, only worker $p_2$ is eligible to work on the affected task as others are working on the same affected task $t_5$.
- The affected workload $(t_5, 2MD)$ is assigned to worker $p_2$ on days $d_6$ and $d_7$ as shown in Fig. 4.13 and the remaining workload $(t_5, 2MD)$ is required to distribute after the recovery window that is after day $d_9$.
- Newly affected workload, $T^a_{new}$=$\{(t_2, 2MD)\}$

***Step 2:*** Now, we have to distribute the remaining affected workload $w^a_5 = 2MD$ of task $t_5$ after the recovery window $R_w$ that is after day $d_9$ as follows:

a) At first, we search for the potential replacement workers through the mapping of task-skill ($TS$) and resource-skill ($RS$) matrix as defined in Eq. (4.2) and Eq. (4.3). The set of potential replacement workers having skills to work on task $t_5$ is $P_r = \{p_1, p_2, p_5, p_6\}$. Since all potential replacement workers are working on the successor of the task $t_5$ (i.e., task $t_6$), we assign the remaining $(t_5, 2MD)$ workload to workers $p_1$ and $p_2$ on day $d_{10}$ as shown Fig. 4.13.

b) We need to cancel the initially assigned task $t_6$ of the workers $p_5$ and $p_6$ for the day $d_{10}$ to maintain the precedence relationship between task $t_5$ and $t_6$ as as illustrated in Fig. 4.13.

c) Finally the *finished* time of $t_5$ is updated as $d_{10}$ and the *start* time of the task $t_6$ is as $d_{11}$.

d) The overall newly affected workload will be:, $T^a_{new}$=$\{((t_2, 2MD)$, $(t_6, 4MD)\}$

***Step 3:*** Finished.

| | D1 | D2 | D3 | D4 | D5 | **D6** | **D7** | **D8** | **D9** | D10 | D11 | D12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | T4 | T4 | T1 | T1 | T1 | T5 | T5 | T5 | T5 | **T5(T6)** | T6 | |
| **P2** | T1 | T1 | T1 | T1 | T1 | **T5(T2)** | **T5(T2)** | T5 | T5 | **T5(T6)** | T6 | |
| P3 | T2 | T2 | T2 | T2 | T2 | T2 | T2 | | | | | |
| P4 | T3 | T3 | T3 | T3 | T2 | T2 | T2 | | | | | |
| **P5** | T4 | T4 | T3 | T3 | T5 | **(T5)** | **(T5)** | **(T5)** | **(T5)** | **(T6)** | T6 | |
| **P6** | T4 | T4 | T2 | T2 | T5 | T5 | T5 | T5 | T5 | **(T6)** | T6 | |

Figure 4.13: Implementation of change Option-C (Extended) .

## 4.7 Conclusion

This chapter has presented the working principle of our proposed decision based reactive scheduling system. The detailed procedures for calculating the change impact and the implementation of change options to revise the schedule has also been presented with examples. Chapter 5 presents the application of our reactive scheduling procedure using a case study.

# Chapter 5

# Application

## 5.1 Introduction

In this chapter, we examine the performance of our decision-based reactive scheduling approach to address changes due to the absence of workers during project execution. We present a case study related to software development projects to evaluate the performance. For a change scenario, the proposed reactive scheduling approach would revise the baseline schedule by selecting the appropriate change options (i.e., option-A, option-B and option-C) to minimize the overall project delay while limiting the number of modifications. The selection is done by calculating change impact factor at three different points in the decision module. Finally, the selected change option is used to revise the schedule. The quality of the revised schedule is evaluated in terms of project delay and re-organization efforts. The simple case study related to software development project is introduced in section 5.2. In section 5.3, we describe the test cases that will be used to demonstrate our proposed approach. Finally, in section 5.4, we evaluate the performance of our approach for the cases under study.

## 5.2 Case Study Implementation

The case study (Wysocki, 2006) is about the development of software application for a pizza company to support automated in-store operation and home delivery services. Pizza Delivered Quickly (PDQ) is a local chain (40 stores) of eat-in and home delivery pizza stores decided to promote program that guarantees 30-45 minute delivery service from order entry to home delivery by upgrading their existing IT infrastructure. The basic functionality of the automated system will be to receive orders, prepare, and deliver the pizzas. The factory location nearest the customer's location will receive the order from a central ordering facility, process, and deliver the order within 30 or 45 minutes of order entry depending on whether the customer orders their pizza ready for the oven or already baked by using their own logistic system. The software development team defined the scope of the full system by identifying five sub-systems as follows:

- **Sub-system-1 (Order Entry):** The order entry subsystem will support the store and factory operations. The telephone orders coming from the customers will be received and inputted here and then routed to the appropriate store or factory electronically for further processing. This subsystem deals with the information related to customer, order, delivery, price and payment.

- **Sub-system-2 (Order Fulfillment):** The subsystem decides where to prepare the orders (i.e., a store, factory or pizza van) based on the current workloads and then transmits the order to the right place accordingly.

- **Sub-system-3 (Order Routing):** This software application will be a routing subsystem for the delivery trucks. This application will probably involve having GPS systems installed in all the delivery trucks.

- **Sub-system-4 (Logistics Management):** This sub-system was just a database that keeps the record of all current operational data and would have to be constantly updated. The subsystem decides how to deliver the order by computing real time route based on the delivery instructions and current workloads.

- **Sub-system-5 (Inventory Management):** This sub-system monitors real-time inventory levels at all locations and automatically issues replenishment orders to the trucks to replenish location inventories and automatically re-orders inventory from the vendor.

The project manager decided to use the well known *waterfall* (Sommerville, 2011) model as the system development life cycle. The manager and his team start with the project charter and scope statement and develop the Work Breakdown Structure (WBS) for requirement, analysis, design, implementation, testing and documentation phases. Then the project team develops a detailed list of tasks and their attributes such as efforts in man-day, duration and dependencies as shown in Table-5.1. For demonstration purpose, the effort and duration has been reduced from the actual calculation, but the relationships remained unchanged. The task precedence diagram is shown in Fig. 5.1. The team classifies the tasks based on the skill requirements in different phases. The skills required by the tasks include the following types:

- Analysis: it represents the skills of requirement analysis technique, such Object Oriented Analysis (OOA), Structured Analysis (RA) and communication skills.

- Design: it represents the skills of design technique, such as Object Oriented Design (OOD), database design, GUI design.

- Programming: it represents the skills of coding language, such as C/C++, Java, C# etc.

- Database: it represents the skills of data analysis, data mining, database design, SQL, database security.

- Quality: it represents the skills for unit testing, integration testing and system testing.

- Technical writing: it represents the skills for writing the operation manual or user manual.

Table 5.1: Attributes of the tasks for the case study

| Sl. | Tasks | Task Description | Effort (MD) | Duration (day) | Predecessor(s) |
|---|---|---|---|---|---|
| 1 | T1 | Requirements for the Order Entry Sub-system | 5 | 5 | - |
| 2 | T2 | Requirements for the Order Routing Sub-system | 8 | 4 | - |
| 3 | T3 | Requirements for the Order Fulfillment Sub-system | 7 | 5 | - |
| 4 | T4 | Design the Customer Profile Sub-system | 13 | 7 | T1 |
| 5 | T5 | Design the Order Taking Sub-system | 13 | 7 | T1 |
| 6 | T6 | Design Order Routing Sub-system | 12 | 5 | T2 |
| 7 | T7 | Design Order Fulfillment Sub-system | 8 | 4 | T3 |
| 8 | T8 | Design Integration T4 & T5 | 12 | 2 | T4,T5 |
| 9 | T9 | Design Integration | 15 | 3 | T6,T7,T8 |
| 10 | T10 | Coding of the Order Entry Sub-system | 16 | 8 | T9 |
| 11 | T11 | Coding of the Order Routing Sub-system | 39 | 13 | T9 |
| 12 | T12 | Coding of the Order Fulfillment Sub-system | 22 | 10 | T9 |
| 13 | T13 | Database creation | 22 | 8 | T9 |
| 14 | T14 | Security and transaction module | 10 | 5 | T9 |
| 15 | T15 | Middle-ware Development | 46 | 10 | T13,T14 |
| 16 | T16 | Integration Testing | 16 | 4 | T10,T11,T12,T15 |
| 17 | T17 | System Testing | 20 | 5 | T16 |
| 18 | T18 | Operational Manual | 15 | 5 | T16 |
| 19 | T19 | User Manual | 15 | 6 | T16 |
| 20 | T20 | Acceptance Testing | 9 | 3 | T17 |
| | | **Total effort** | **323** | | |

The set of required skills associated with different tasks are listed in Table-5.2. A project team with 20 technical workers (analyst, designer, coder, tester, technical writer) is formed to accomplish the tasks on time by properly distributing the workload among the team members. The Table-5.3 shows the list of workers with their expertize in different skills.

After preparing task and resource list, the project team first develop the task worker allocation schedule and then the corresponding Gantt chart. The Gant chart captures the information of start and completion times as well as the slack time for each task. To keep the project within space limit, the Gantt chart is excluded in this chapter but it is assumed that Gantt chart is always updated for all types of modification in task resource allocation schedule. The tasks $t_1$, $t_4$, $t_8$, $t_9$, $t_{13}$, $t_{15}$, $t_{16}$, $t_{17}$ and $t_{20}$ form the critical path,

Table 5.2: Required skills for the tasks

| Task | Programming | Algorithm | Database | Analyst | Design | Quality Control | Tech. writing |
|---|---|---|---|---|---|---|---|
| T1 | | | | 1 | | | |
| T2 | | | | 1 | | | |
| T3 | | | | 1 | | | |
| T4 | | 1 | 1 | | 1 | | |
| T5 | | 1 | 1 | | 1 | | |
| T6 | | 1 | 1 | | 1 | | |
| T7 | | 1 | 1 | | 1 | | |
| T8 | | 1 | | | 1 | | |
| T9 | | 1 | 1 | | 1 | | |
| T10 | 1 | | | | | | |
| T11 | 1 | | | | | | |
| T12 | 1 | | | | | | |
| T13 | | | 1 | | | | |
| T14 | 1 | | 1 | | | | |
| T15 | 1 | | 1 | | | | |
| T16 | | | | | | 1 | |
| T17 | | | | 1 | | 1 | |
| T18 | | | | | | | 1 |
| T19 | | | | | | | 1 |
| T10 | | | | 1 | | 1 | |

Table 5.3: Workers skills capability

| Workers | Programming | Algorithm | Database | Analyst | Design | Quality Control | Tech. writing |
|---|---|---|---|---|---|---|---|
| P1 | | | | 1 | | 1 | |
| P2 | 1 | 1 | | | | | |
| P3 | 1 | 1 | 1 | | 1 | | |
| P4 | 1 | | | | 1 | | |
| P5 | | 1 | 1 | 1 | | 1 | |
| P6 | 1 | | | | | | |
| P7 | | 1 | 1 | | 1 | | |
| P8 | 1 | | | | | | |
| P9 | 1 | 1 | 1 | | 1 | | |
| P10 | 1 | | 1 | | | | |
| P11 | 1 | 1 | 1 | | 1 | | |
| P12 | | | | 1 | | 1 | |
| P13 | 1 | | 1 | | | 1 | |
| P14 | | | | 1 | | | |
| P15 | 1 | 1 | 1 | | 1 | 1 | |
| P16 | | | | | | | 1 |
| P17 | | | | | | | 1 |
| P18 | | | | | | | 1 |
| P19 | | | | | | | 1 |
| P20 | | | | | | | 1 |

which leads to 47 days of the project duration. To indicate, the tasks on the critical path are highlighted with bold face in the task precedence graph in Fig. 5.1. Fig. 5.2

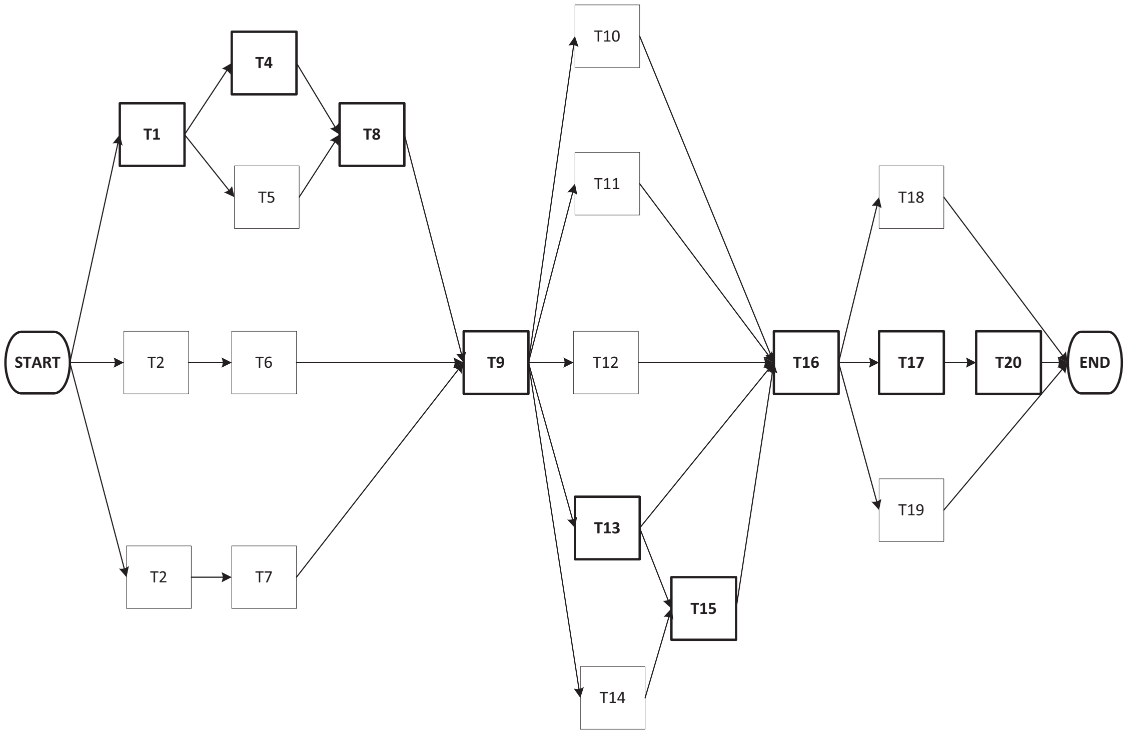



Figure 5.1: Task precedence diagram.

shows the task worker allocation schedule of the Pizza Delivered Quickly (PDQ) software development project. On this schedule, the top row lists the days of the project, and the left column displays the workers involved in the project. Then, each schedule entry indicates the responsible task of the worker on the specific day. For instance, Fig. 5.2 shows that worker $p_1$ is responsible for task $t_1$ starting from Day $d_1$ to $d_5$.

| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 | D16 | D17 | D18 | D19 | D20 | D21 | D22 | D23 | D24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | T1 | T1 | T1 | T1 | T1 | | | | | | | | | | | | | | | | | | | |
| P2 | | | | | | | | | | | | | | | | | | T10 | T10 | T10 | T10 | T10 | T10 | T10 |
| P3 | | | | | T6 | T6 | T6 | T6 | T6 | T4 | T4 | T4 | T8 | T8 | T9 | T9 | T9 | T10 | T10 | T10 | T10 | T10 | T10 | T10 |
| P4 | | | | | | | | | | | | | | | | | | T11 | T11 | T11 | T11 | T11 | T11 | T11 |
| P5 | T2 | T2 | T2 | T2 | T3 | | | | | | | | | | | | | | | | | | | |
| P6 | | | | | T6 | T6 | T6 | T6 | T6 | T4 | T4 | T4 | T8 | T8 | T9 | T9 | T9 | T11 | T11 | T11 | T11 | T11 | T11 | T11 |
| P7 | | | | | T6 | T4 | T4 | T4 | T4 | T4 | T4 | T4 | T8 | T8 | | | | | | | | | | |
| P8 | | | | | | | | | | | | | | | | | | T12 | T12 | T12 | T12 | T12 | T12 | T12 |
| P9 | | | | | T6 | T5 | T5 | T5 | T5 | T5 | T5 | T4 | T8 | T8 | T9 | T9 | T9 | T12 | T12 | T12 | T12 | T12 | T12 | T12 |
| P10 | | | | | | | | | | | | | | | | | | T13 | T13 | T13 | T13 | T13 | T13 | T13 |
| P11 | | | | | | T7 | T7 | T7 | T7 | T5 | T5 | T4 | T8 | T8 | T9 | T9 | T9 | T13 | T13 | T13 | T13 | T13 | T13 | T13 |
| P12 | T2 | T2 | T2 | T2 | T3 | | | | | | | | | | | | | | | | | | | |
| P13 | | | | | | | | | | | | | | | | | | T14 | T14 | T14 | T14 | T14 | T13 | T13 |
| P14 | T3 | T3 | T3 | T3 | T3 | | | | | | | | | | | | | | | | | | | |
| P15 | | | | | | T7 | T7 | T7 | T7 | T5 | T5 | T4 | T8 | T8 | T9 | T9 | T9 | T14 | T14 | T14 | T14 | T14 | T13 | T13 |
| P16 | | | | | | | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | | | | | | | |

(a) First part (days $d_1$ to $d_{24}$)

| | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 | D36 | D37 | D38 | D39 | D40 | D41 | D42 | D43 | D44 | D45 | D46 | D47 | D48 | D49 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | T17 | T17 | T17 | T17 | T17 | T20 | T20 | T20 | | |
| P2 | T10 | T11 | T11 | T11 | T11 | T11 | | | | | | | | | | | | | | | | | | | |
| P3 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | | | | | | | | | | | | | | |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | | | | | | | | | | | | | | | | | | | |
| P5 | | | | | | | | | | | | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | T20 | T20 | T20 | | |
| P6 | T11 | T11 | T11 | T11 | T11 | T11 | | | | | | | | | | | | | | | | | | | |
| P7 | | | | | | | | | | | | | | | | | | | | | | | | | |
| P8 | T12 | T12 | T12 | T11 | T11 | T11 | | | | | | | | | | | | | | | | | | | |
| P9 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | | | | | | | | | | | | | | |
| P10 | T13 | T11 | T11 | T11 | T11 | | | | | | | | | | | | | | | | | | | | |
| P11 | T13 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | | | | | | | | | | | | | | |
| P12 | | | | | | | | | | | | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | T20 | T20 | T20 | | |
| P13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | | | | | |
| P14 | | | | | | | | | | | | | | | | | | | | | | | | | |
| P15 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | | | | | |
| P16 | | | | | | | | | | | | | | | | T18 | T18 | T18 | T18 | T18 | T19 | | | | |
| P17 | | | | | | | | | | | | | | | | T18 | T18 | T18 | T18 | T18 | T19 | | | | |
| P18 | | | | | | | | | | | | | | | | T18 | T18 | T18 | T18 | T18 | T19 | | | | |
| P19 | | | | | | | | | | | | | | | | T19 | T19 | T19 | T19 | T19 | T19 | | | | |
| P20 | | | | | | | | | | | | | | | | T19 | T19 | T19 | T19 | T19 | T19 | | | | |

(b) Second part (days $d_{25}$ to $d_{49}$)

Figure 5.2: Task worker allocation schedule.

## 5.3 Classification of the Cases

The purpose of this section is to categorize the possible cases concerning the worker absent on the basis of change impact factor value in the three decision points and the change options selection method. Finally the cases are used to demonstrate the functionality of the procedure and validate the decision based selection method.

As we discussed in section-4.5 in Chapter 4, the schedule can be revised by any one of the change options (i.e., Option-A, Option-B, and Option-C (General, extended)). To select the appropriate change options, the decision module uses the calculated values of the change impact factors (i.e., $I_1$, $I_2$ and $I_3$) in three hierarchy decision points. If the value of the change impact factor $I_1$ at decision point-1 is less than 1, the impact is considered as low and option-A is selected to revise the schedule. On the other hand, the impact is assumed to be medium when the change impact factor $I_2$ is less than 1 at decision point-2. In this case, option-B is used to revise the schedule. If change impact factor $I_2$ is greater than 1, the option-C is the best selection to revise the schedule for this high impact change scenario. In the above context, we set down four cases denoted to case-1, case-2, case-3 and case-4 based on different change option selection situations and summarize in Table-5.4.

Table 5.4: Classification of the cases based on change options selection.

| Sl. | Impact Factor (I) | Change Impact | Selected Options | Case Types |
|---|---|---|---|---|
| 1. | $I_1 \leq 1$ | Low | Option-A | Case-1 |
| 2. | $I_1 > 1$ and $I_2 \leq 1$ | Medium | Option-B | Case-2 |
| 3. | $I_2 > 1$ and $I_3 \leq 1$ | High | Option-C (General) | Case-3 |
| 4. | $I_3 > 1$ | High (Inevitable delay ) | Option-C (Extended) | Case-4 |

## 5.4 Demonstration and Validation

In this section, we demonstrate the procedures of decision-based reactive scheduling approach discussed in Chapter 4 for the four cases described in previous section 5.3. After demonstration, we revise the schedule using the other two alternative options for the same change scenario to benchmark the decision making method in selecting the appropriate change option. More specific, the purpose of evaluating the quality of the revised schedule is to examine whether the decision based change option selection module helps to select a proper change option that incurs least delay with minimum number of reorganization efforts. Project delay is the most important criteria that needs to be considered during schedule modification as delay can lead to customer dissatisfaction as well as add extra cost. Meanwhile, the project manager also wish to keep the perturbation in a minimum level in terms of modified entries in the revised schedule. The proposed reactive approach revise the schedule taking the delay in first consideration and then the reorganization efforts. By evaluating the delay and number of modifications in the revised schedule, we can examine the performance of the decision module in selection the appropriate change options.

### 5.4.1 Demonstration of Case-1

In the first case, worker $p_8$ working on non-critical task $t_{12}$ is absent for day $d_{18}$ to day $d_{25}$ and the change scenario due to his absence is $(t_{12}, 8MD)$ as shown in the task-resource allocation schedule in Fig. 5.3. The step-by-step procedure for the selection of change option and implementation of the selected change option on the task worker schedule are described below:

**Part-A (Selection of the change option):** The basic steps for the decision based change option selection method described in Fig. 4.5 in Chapter 4 is used to find the appropriate change option to revise the schedule as follows:

***Decision point-1:*** At the decision point-1, the change impact factor $(I_1)$ is calculated using Algorithm-4.1 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{35}$ as first affected date and latest finish (LF) time of the outstanding task $t_{12}$ is $d_{18}$ and $d_{35}$ respectively.

Step 2: With in the recovery window only worker $p_8$ can work on the affected task $t_{12}$ starting from day $d_{28}$ after completing his current tasks at hand. Worker $p_9$ can not work on the affected task $t_{12}$ as he was assigned to critical task $t_{15}$ next. So, the potential replacement workers with in the recovery window $R_w$ is: $P_r = \{p_8\}$.
Step 3: Within the recovery window $R_w$, $p_8$ can share only 8 workloads. The total maximum distributable workload, $w^d_{12}$ =8.
Step 4: The change impact factor, $I_1 = 8/8 = 1$; which is equal to 1.
Since the change impact factor $I_1$ in this decision point is equal to 1, Option-A is selected to revise the schedule and decision points 2 and 3 are ignored.

**Part-B (Implementation of the change Option-A):** Now we will revise the schedule using the procedure of Option-A as described in Algorithm-4.4 as follows:
***Iteration-1:*** We will distribute the affected 8MD workloads of the outstanding task $t_{12}$ using option-A.
Step 1: The first affected date and latest finish (LF) time of the outstanding task $t_{12}$ is $d_{18}$ and $d_{35}$ respectively. The range of the recovery window $R_w$ is $d_{18}$ to $d_{35}$.
Step 2: Within the recovery window $R_w$ that is for days $d_{18}$ to $d_{35}$, we distribute the workload per day as follows:

- Days $d_{18}$ to $d_{27}$: No one is working either mode-1 or mode-2 in the task-resource allocation schedule ($SH$).

- Days $d_{28}$ to $d_{35}$: The only potential replacement worker is $p_8$ as he is working in non-critical task after completion of his present task in the task-resource allocation schedule ($SH$). We assign 3MD workloads of $t_{12}$ by canceling his previously assigned task $t_{11}$ for days $d_{28}$ to $d_{30}$ . After that he is requested to work on the remaining 5MD affected load as an extra work from days $d_{31}$ to $d_{35}$. Hence, the remaining affected workload becomes, $w^a_{12}$=8-8=0. The revised schedule is shown in Fig. 5.3.
  Newly affected workload, $T^a_{new}$=$\{(t_{11}$,3MD)$\}$
  Since $w^a_{12}$=0, distribution of the affected workload for task $t_{12}$ is completed.

Step 3: Finished.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | | | |
| P2 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| P3 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| P5 | | | | | | | | | | | | | | | | | | |
| P6 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| P7 | | | | | | | | | | | | | | | | | | |
| P8 | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | T12 | T12 | **T12(T11)** | **T12(T11)** | **T12(T11)** | **[T12]** | **[T12]** | **[T12]** | **[T12]** | **[T12]** |
| P9 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P10 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T11 | T11 | T11 | T11 | T11 | | | | | |
| P11 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | | | | | |
| P13 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | | | | | |
| P15 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | |

Figure 5.3: Revised schedule using option-A for case-1

***Iteration-2:*** In the second stage, we found that Option-A is suitable for the change scenario $T^{a}_{new}$=$\{(t_{11}$,3MD)$\}$ by following the same procedure as described above. Finally we distribute the 3MD affected workload of $t_{11}$ using option-A on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$ and $p_6$ as an extra load. Since there are no more affected workload, the schedule shown in Fig. 5.3 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3) as described in Table-3.1 in Chapter 3. Table-5.5 illustrates the over all performance of change option-A for the revised schedule in Fig. 5.3 in terms of delay and reorganization efforts.

Table 5.5: Performance of Option-A for case-1

| **Option-A** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 0 | 8 | 3 | 8 | 19 |

### 5.4.2 Validation of Case-1

To compare the results of applying different change options to revise the schedule, we demonstrate two other schedule revision processes by applying the remaining two change options (i.e. option-B and Option-C) as follows:

**Implementation of the change Option-B:** Now we will revise the schedule by using the procedure of Option-B as described in Algorithm-4.5 as follows:

***Iteration-1:*** We will distribute the affected 8MD workloads of the outstanding task $t_{12}$ using option-B.

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{35}$ as the first affected date and latest finish (LF) time of the outstanding task $t_{12}$ is $d_{18}$ and $d_{35}$ respectively.

Step 2: At first, we search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. Hence, the set of potential replacement workers having skills to work on task $t_{12}$ is $P_r = \{p_2, p_3, p_4, p_6, p_8, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

Step 3: Since we need only one replacement workers for option-B, workers $p_2$, $p_4$ and $p_6$ are considered first as they can share maximum 18 MD workloads among the 10 potential replacement workers within the recovery window. Although workers $p_2$, $p_4$, and $p_6$ can share the same amount workload, we should assign lower priority to worker $p_2$ as he is working on two tasks (i.e., $t_{10}$ and $t_{11}$) as compare to workers $p_4$ and $p_6$ those are working only single task within the recovery window. The list of potential candidates who are selected to work on the affected task is sorted descending order as: $P_r$=$\{(p_6, 8MD), (p_4, 8MD), (p_2, 8MD)\}$

Step 4: Finally, the affected workload $(t_{12}, 8MD)$ is assigned to worker $p_6$ as shown in Fig.

5.4 and record the newly affected tasks. Newly affected workload, $T^a_{new}$={$(t_{11}, 8MD)$}

Step 5: Finished.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | | | |
| P2 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T11 | T11 | T11 | T11 | T11 | **[T11]** | **[T11]** | | | |
| P3 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | **[T11]** | | | |
| P5 | | | | | | | | | | | | | | | | | | |
| P6 | **T12(T11)** | **T12(T11)** | **T12(T11)** | **T12(T11)** | **T12(T11)** | **T12(T11)** | **T12(T11)** | **T12(T11)** | T11 | T11 | T11 | T11 | T11 | **[T11]** | **[T11]** | | | |
| P7 | | | | | | | | | | | | | | | | | | |
| P8 | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | T12 | T12 | T11 | T11 | T11 | **[T11]** | | | | |
| P9 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P10 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| P11 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | | | | | |
| P13 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | | | | | |
| P15 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | |

Figure 5.4: Revised schedule using option-B for case-1

***Iteration-2:*** We will distribute the newly affected task $T^a_{new}$={$(t_{11}, 8MD)$} by finding the suitable change options through decision module. In this stage, we found that Option-A is suitable for this change scenario $T^a_{new}$={($t_{11}$, 8MD)} by following the decision module as described in the beginning. Finally, we distribute the 5MD affected workload of $t_{11}$ on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$, $p_6$, $p_8$, and $p_{10}$ and the remaining 3MD workload are assigned to workers $p_2$, $p_4$, and $p_6$ on day $d_{32}$ as an extra load. Since there are no more affected workload, the schedule shown in Fig. 5.4 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.6 illustrates the over all performance of change option-B for the revised schedule in Fig. 5.4 in terms of delay and the number of modified entries.

Table 5.6: Performance of Option-B for case-1

| **Option-B** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 0 | 8 | 8 | 8 | 24 |

**Implementation of the change Option-C:** Now we will revise the schedule by using the procedure of Option-C as described in Algorithm-4.6 as follows:

***Iteration-1:*** We will distribute the affected 8MD workloads of the outstanding task $t_{12}$ using option-C.

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{35}$ as the first affected date and latest finish (LF) time of the outstanding task $t_{13}$ is $d_{18}$ and $d_{35}$ respectively.

Step 2: within the recovery window $R_w$ that is for days $d_{18}$ to $d_{35}$, we will distribute the workload per day as follows:

- Day $d_{18}$: At first, we search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. Hence, the set of capable workers having skills to work on task $t_{12}$ is $P_r = \{p_2, p_3, p_4, p_6, p_8, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$. Finally, we find the potential replace working on this day by excluding those are working on outstanding task $t_{12}$ or any critical task. Since, workers $p_9$ is working on task $t_{12}$ and workers $p_{10}$ and $p_{11}$ are working on critical task $t_{13}$, we exclude them from $P_r$ to obtain the final potential replacement workers list as: $P_r = \{p_2, p_3, p_4, p_6, p_{13}, p_{15}\}$. Now, we assign the 6MD workload of task $t_{12}$ to them by canceling their present task. The list of newly affected tasks due to this reassignment is: $T^a_{new}$={($t_{10}$,2MD), ($t_{11}$,2MD), ($t_{14}$,2MD)}.

- Day $d_{19}$: The potential replacement workers are same as day $d_{18}$. We assign remaining 2MD workloads of $t_{12}$ to workers $p_2$ and $p_3$ by putting off their previously assigned task $t_{10}$ and the newly affected task is $t_{10}$ with 2MD workload. Since $w^a_{12}$=0, the distribution of the affected workload for task $t_{12}$ is completed. The updated newly affected task list is: $T^a_{new}$={($t_{10}$,4MD), ($t_{11}$,2MD), ($t_{14}$,2MD)}.

Step 3: Finished.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | | | |
| P2 | **T12(T10)** | **T12(T10)** | T10 | T10 | T10 | T10 | T10 | T10 | **T10(T11)** | **T10(T11)** | **T10(T11)** | **T10(T11)** | T11 | **[T11]** | **[T11]** | | | |
| P3 | **T12(T10)** | **T12(T10)** | T10 | T10 | T10 | T10 | T10 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P4 | **T12(T11)** | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | **[T11]** | | | |
| P5 | | | | | | | | | | | | | | | | | | |
| P6 | **T12(T11)** | **T14(T11)** | **T14(T11)** | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | **[T11]** | | | |
| P7 | | | | | | | | | | | | | | | | | | |
| P8 | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | **(T12)** | T12 | T12 | T11 | T11 | T11 | **[T11]** | | | | |
| P9 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P10 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| P11 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | | | | | |
| P13 | **T12(T14)** | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | | | | | |
| P15 | **T12(T14)** | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | |

Figure 5.5: Revised schedule using option-C for case-1

***Iteration-2:*** We will distribute the newly affected task $T^a_{new}$={($t_{10}$,4MD), ($t_{11}$,2MD), ($t_{14}$,2MD)} by finding the suitable change options through decision module. Since there are multiple affected tasks, we need to process one by one as decision module is designed for the change scenario for a single task. We will sort the newly affected tasks list in ascending order based on the early finish (EF) time of the tasks as: $T^a_{new}$={($t_{14}$,2MD), ($t_{10}$,4MD), ($t_{11}$,2MD)}. In this stage, we will distribute the affected workload of task $t_{14}$ only. We found that Option-B is suitable for the change scenario ($t_{14}$, 2MD) by following the decision module. Finally we assign the 2MD affected workload of $t_{14}$ to workers $p_6$ on day $d_{19}$ and $d_{20}$ and update the affected task list by recording the newly affected 2MD workload of task $t_{11}$ as: $T^a_{new}$={($t_{10}$,4MD), ($t_{11}$,4MD)}.

***Iteration-3:*** In this stage, we assign the total 4MD affected workload of $t_{10}$ to worker $p_2$ from day $d_{26}$ to $d_{29}$ using option-A by canceling the previously assigned task $t_{11}$ and update the newly affected task list accordingly. The updated newly affected task list is: $T^a_{new}$={($t_{11}$,8MD)}.

Table 5.7: Performance of Option-C for case-1

| **Option-C** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 0 | 8 | 14 | 8 | 30 |

***Iteration-4:*** We distribute the 5MD affected workload of $t_{11}$ on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$, $p_6$, $p_8$, and $p_{10}$ and the remaining 3MD workload are assigned to workers $p_2$, $p_4$, and $p_6$ on day $d_{32}$ as an extra load. Since there are no more affected workload, the schedule shown in Fig. 5.5 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the updated schedule of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.7 illustrates the over all performance of change option-C for the revised schedule Fig. 5.5 in terms of delay and number of modified entries.

**Analysis of performance:** Based on the above result, we will examine the quality of the revised schedule by analyzing the delay and reorganization efforts for option-A, option-B and option-C for the case-1. Table-5.8 summarizes the results of the revised schedule for the three change options. Each option has same response in terms of delay. By comparing the number of total modification, we came to the conclusion that Option-A is the best choice among three schedule revision options in this case. This result matches the one acquired by applying the decision based reactive scheduling approach to deal with this change scenario. Thus, this approach is considered as efficient to deal with this change scenario in this case. Moreover, we generally consider that change option-A is a better change option to address such kind of low impact change scenario.

Table 5.8: Performance of the three change options for case-1

| Option | Delay | No. of Modifications | | | |
|---|---|---|---|---|---|
| | | Type-1 | Type-2 | Type-3 | Total |
| A | 0 | 8 | 3 | 8 | 19 |
| B | 0 | 8 | 8 | 8 | 24 |
| C | 0 | 8 | 14 | 8 | 30 |

### 5.4.3 Demonstration of Case-2

In the second case, worker $p_{10}$ working on critical task $t_{13}$ is absent for day $d_{18}$ to day $d_{23}$ and the change scenario due to his absence is $(t_{13}, 6MD)$ as shown in the task-resource allocation schedule Fig. 5.6. The step-by-step procedure for the selection of change option and implementation of the selected change option on the task worker schedule are described below:

**Part-A (Selection of the change option):** The basic steps for the decision based system described in Fig. 4.5 in Chapter 4 is used to find the appropriate change option to revise the schedule as follows:

***Decision point-1:*** At the decision point-1, the change impact factor $(I_1)$ is calculated using Algorithm-4.1 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{25}$ as first affected date and latest finish (LF) time of the outstanding task $t_{13}$ is $d_{18}$ and $d_{25}$ respectively.

Step 2: With in the recovery window, only workers $p_{13}$ and $p_{15}$ will finish their present task $t_{14}$ and going to start task $t_{13}$ on day $d_{23}$. Since they are working on the same affected task $t_{13}$, it is meaningless to assign the same affected task. We can say that the number of potential replacement workers in this case is almost zero.

Step 3: Within the recovery window $R_w$, the total maximum distributable workload, $w_{13}^d$ =0.

Step 4: The change impact factor, $I_1 = 6/0 = \infty$; which is larger than 1.

Since, the change impact factor $I_1$ in this decision point is greater than 1, Option-A is not suitable to revise the schedule. We should move to the decision point 2 and 3 for the other change options.

***Decision point-2:*** At decision point-2, the change impact factor ($I_2$) is calculated using Algorithm-4.3 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{25}$ as first affected date and latest finish (LF) time of the outstanding task $t_{13}$ is $d_{18}$ and $d_{25}$ respectively.

Step 2: Now, we search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of potential replacement workers having skills to work on task $t_{13}$ is, $P_r = \{p_3, p_8, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

Step 3: Within the recovery window $R_w$, the potential replacement workers can share workloads as follows:

- $p_3$, $p_8$ and $p_9$ can share maximum 8 workloads;
- $p_{10}$ and $p_{11}$ can share 0 workloads as one is absent and other one is working on the same affected task $t_{13}$ on this duration;
- $p_{13}$ and $p_{15}$ can share maximum 5 workloads;

Maximum redistributable workload, $w_{13}^d$=max$\{8, 8, 8, 0, 0, 5, 5\} = 8$.

Step 4: The change impact factor, $I_2 = 6/8 = 0.75$; which is less than 1.

Since, the change impact factor $I_2$ in this decision point is less than 1, Option-B is appropriate to revise the schedule and decision point-3 can easily be ignored.

**Part-B (Implementation of the change Option-B):** Now, we will revise the schedule using the procedure of Option-B as described in Algorithm-4.5 by using the following steps:

***Iteration-1:*** We will distribute the affected 6MD workloads of the outstanding task $t_{13}$ using option-B.

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{25}$ as the first affected date and latest finish (LF) time of the outstanding task $t_{13}$ is $d_{18}$ and $d_{25}$ respectively.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | | | |
| P2 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P3 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P5 | | | | | | | | | | | | | | | | | | |
| P6 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P7 | | | | | | | | | | | | | | | | | | |
| P8 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | **T12(T11)** | **T12(T11)** | **T12(T11)** | **[T12]** | **[T12]** | **[T12]** | | |
| P9 | **T13(T12)** | **T13(T12)** | **T13(T12)** | **T13(T12)** | **T13(T12)** | **T13(T12)** | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P10 | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | T13 | T13 | T11 | T11 | T11 | T11 | T11 | | | | | |
| P11 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | | | | | |
| P13 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | | | | | |
| P15 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | |

Figure 5.6: Revised schedule using option-B for case-2

Step 2: Now, we search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of potential replacement workers having skills to work on task $t_{12}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

Step 3: Within the recovery window $R_w$, the potential replacement worker can share workloads as follows:

- $p_3$ and $p_9$ can share maximum 8 workloads;
- $p_{10}$ and $p_{11}$ can share 0 workloads as one is absent and other one is working on the same affected task $t_{13}$ on this duration;
- $p_{13}$ and $p_{15}$ can share maximum 5 workloads;

Step 4: As we need only one replacement worker for option-B, workers $p_3$ and $p_9$ is considered first as they can share maximum 8MD workloads among the 6 potential replacement workers within the recovery window. Although they can share the same amount of maximum additional workload, we should select worker $p_9$ as the best suitable potential replacement worker as the task (i.e., $t_{12}$) he is working now has largest early finish (EF) time compare to the tasks $t_{10}$ of the worker $p_3$. So, the final list of potential candidates

Table 5.9: Performance of Option-B for case-2

| **Option-B** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 0 | 6 | 9 | 6 | 21 |

after sorting in descending order is: $P_r$={$(p_9, 8MD), (p_3, 8MD)$}

Step 5: The affected workload $(t_{13}, 6MD)$ is assigned to worker $p_9$ as shown in Fig. 5.6 and record the newly affected tasks as: $T^a_{new}$={$(t_{12}, 6MD)$}

Step 6: Finished.

***Iteration-2:*** In this stage, we will distribute the newly affected task $T^a_{new}$={$(t_{12}, 6MD)$} by finding the suitable change options through decision module. We found that Option-A is suitable for this change scenario $T^a_{new}$={($t_{12}$, 6MD)}. So, we distribute the 3MD affected workload of $t_{12}$ to worker $p_8$ from days $d_{28}$ to $d_{30}$ using option-A by canceling the previously assigned task $t_{11}$ and the remaining 3MD affected workload as an extra load from days $d_{31}$ to $d_{33}$. The updated newly affected task list is: $T^a_{new}$={($t_{11}$,3MD)}.

***Iteration-3:*** We distribute the 3MD affected workload of $t_{11}$ on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$, and $p_6$ as an extra load using option-A. Since, there are no more affected workload, the schedule shown in Fig. 5.6 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to updated schedule of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.9 illustrates the over all performance of change option-B for the revised schedule in Fig. 5.6 in terms of delay and number of modified entries.

### 5.4.4 Validation of Case-2

To compare the results of applying different change options to revise the schedule, we demonstrate two other schedule revision processes by applying the remaining two change options (i.e. option-A and Option-C) as follows:

**Implementation of the change option-A:** We will revise the schedule using the procedure of Option-A as described in Algorithm-4.4 by using the following steps:

***Iteration-1:*** We will distribute the affected 6MD workloads of the outstanding task $t_{13}$ using option-A.

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{25}$.

Step 2: within the recovery window $R_w$:

- Days $d_{18}$ to $d_{25}$: Only workers $p_{13}$ and $p_{15}$ will finish their present task $t_{14}$ and going to start new task $t_{13}$ on day $d_{23}$. As they will work on the same affected task $t_{13}$, it is meaningless to assign them the same affected task.

- Day $d_{26}$: We have to distribute the affected workload $w^a_{13} = 6MD$ of task $t_{13}$ after the recovery window $R_w$. At first, we search for the capable replacement workers through the mapping of task-skill and resource-skill matrix as defined in Table-5.2 and Table-5.3. The set of potential replacement workers is found by keeping those workers who previously worked on the affected task $t_{13}$ or now working on its successor tasks. So, the list of potential replacement worker is $P_r = \{p_3, p_{10}, p_{11}, p_{13}, p_{15}\}$. We assign the affected workload $(t_{13}, 5MD)$ to workers $p_3$, $p_{10}$, $p_{11}$, $p_{13}$, and $p_{15}$ as shown in Fig. 5.7 and record the newly affected task as: $T^a_{new}$={($t_{11}$,1MD),($t_{12}$,1MD),($t_{15}$,3MD)}.

- Day $d_{27}$: We distribute the remaining 1MD workload of task $t_{13}$ to worker $p_3$. Besides, we need to defer the initially assigned task $t_{15}$ of the workers $p_{13}$ and $p_{15}$ to maintain the precedence relationship between task $t_{13}$ and $t_{15}$. Since $w^a_{13}$=0, the distribution of the affected workload for task $t_{13}$ is completed. The updated newly affected task list is: $T^a_{new}$={($t_{11}$,1MD),($t_{12}$,1MD),($t_{15}$,6MD)}.

***Iteration-2:*** We will distribute the newly affected task $T^a_{new}$ by finding the suitable change options with the helps of the decision module. Since there are multiple affected tasks, we need to process one by one because the decision module is designed for the change scenario for a single task. We sort the newly affected tasks list in ascending order based on the early finish (EF) time of the tasks as: $T^a_{new}$={($t_{12}$,1MD),($t_{11}$,1MD),($t_{15}$,6MD)}. Now, we will distribute the 1MD affected workload of $t_{12}$ using option-A to worker $p_8$ on day $d_{28}$ by interrupting his previously assigned task $t_{11}$. The updated newly affected task list: $T^a_{new}$={($t_{11}$,2MD), ($t_{15}$,6MD)}.

***Iteration-3:*** In this stage, we assign the total 2MD affected workload of $t_{11}$ to worker $p_2$ and $p_4$ on day $d_{31}$ using option-A as an extra load. The remaining affected task list is: $T^a_{new}$={($t_{15}$,6MD)}.

***Iteration-4:*** The recovery window for the affected task $t_{15}$ is day $d_{26}$ to $d_{35}$. From the task worker allocation schedule and using the decision module, it is found that option-A or option-B are not suitable for the change scenario ($t_{15}$,6MD). We have to distribute the affected workload of $t_{15}$ using option-C (Extended) on days $d_{36}$ and $d_{37}$ and defer the task $t_{16}$ by two days . The list of newly affected task is: $T^a_{new}$={($t_{16}$,8MD)}.

***Iteration-5:*** Similarly, we distribute the affected workload 8MD of $t_{16}$ using option-C (Extended) on days $d_{40}$ and $d_{41}$ and defer the tasks $t_{17}$, $t_{18}$ and $t_{19}$. The list of newly affected tasks is: $T^a_{new}$={($t_{17}$,10MD),($t_{18}$,6MD),($t_{19}$,4MD)}.

***Iteration-6:*** The affected workload of $t_{17}$ is distributed using option-C (Extended) on days $d_{45}$ and $d_{46}$ which defers the task $t_{20}$ for two days. The updated list of newly affected task, $T^a_{new}$={($t_{18}$,6MD),($t_{19}$,4MD),($t_{20}$,6MD)}.

***Iteration-7:*** The affected workload of tasks $t_{18}$ and $t_{19}$ is distributed using option-A as shown in Fig. 5.7 and the list of newly affected task is: $T^a_{new}$={($t_{20}$,6MD)}.

***Iteration-8:*** Finally, The affected workload of tasks ($t_{20}$,6MD) is distributed using option-C (Extended) on days $d_{48}$ and $d_{49}$. Since, there are no more affected workload, the schedule shown in Fig. 5.7 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | | | |
| P2 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P3 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T13(T15) | T13(T15) | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P5 | | | | | | | | | | | | | | | | | | |
| P6 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | | | | | |
| P7 | | | | | | | | | | | | | | | | | | |
| P8 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12(T11) | T11 | T11 | | | | | |
| P9 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P10 | (T13) | (T13) | (T13) | (T13) | (T13) | (T13) | T13 | T13 | T13(T11) | T11 | T11 | T11 | T11 | | | | | |
| P11 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13(T12) | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | | | | | |
| P13 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T13(T15) | (T15) | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | | | | | |
| P15 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T13(T15) | (T15) | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | |

(a) First part (days $d_{18}$ to $d_{35}$)

| | D36 | D37 | D38 | D39 | D40 | D41 | D42 | D43 | D44 | D45 | D46 | D47 | D48 | D49 | D50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | T16(T17) | T16(T17) | T17 | T17 | T17 | T17(T20) | T17(T20) | T20 | [T20] | [T20] | |
| P2 | | | | | | | | | | | | | | | |
| P3 | [T15] | | | | | | | | | | | | | | |
| P4 | | | | | | | | | | | | | | | |
| P5 | (T16) | (T16) | T16 | T16 | T16(T17) | T16(T17) | T17 | T17 | T17 | T17(T20) | T17(T20) | T20 | [T20] | [T20] | |
| P6 | | | | | | | | | | | | | | | |
| P7 | | | | | | | | | | | | | | | |
| P8 | | | | | | | | | | | | | | | |
| P9 | [T15] | | | | | | | | | | | | | | |
| P10 | | | | | | | | | | | | | | | |
| P11 | | | | | | | | | | | | | | | |
| P12 | T15(T16) | T15(T16) | T16 | T16 | T16(T17) | T16(T17) | T17 | T17 | T17 | T17(T20) | T17(T20) | T20 | [T20] | [T20] | |
| P13 | T15(T16) | (T16) | T16 | T16 | T16(T17) | (T17) | T17 | T17 | T17 | [T17] | [T17] | | | | |
| P14 | | | | | | | | | | | | | | | |
| P15 | T15(T16) | (T16) | T16 | T16 | T16(T17) | (T17) | T17 | T17 | T17 | [T17] | [T17] | | | | |
| P16 | | | | | (T18) | (T18) | T18 | T18 | T18 | T18(T19) | [T18] | [T19] | | | |
| P17 | | | | | (T18) | (T18) | T18 | T18 | T18 | T18(T19) | [T18] | [T19] | | | |
| P18 | | | | | (T18) | (T18) | T18 | T18 | T18 | T18(T19) | [T18] | [T19] | | | |
| P19 | | | | | (T19) | (T19) | T19 | T19 | T19 | T19 | [T19] | [T19] | | | |
| P20 | | | | | (T19) | (T19) | T19 | T19 | T19 | T19 | [T19] | [T19] | | | |

(b) Second part (days $d_{36}$ to $d_{50}$)

Figure 5.7: Revise schedule using option-A for case-2.

Table 5.10: Performance of Option-A for case-2

| **Option-A** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 2 | 24 | 27 | 24 | 75 |

summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.10 illustrates the over all performance of change option-A for the revised schedule in Fig. 5.7 in terms of delay and number of modified entries.

**Implementation of the change option-C:** We will revise the schedule by using the procedure of Option-C as described in Algorithm-4.6 as follows:

***Iteration-1:*** We will distribute the affected 6MD workloads of the outstanding task $t_{13}$ using option-C.

Step 1: The range of the recovery window $R_w$ is $d_{18}$ to $d_{25}$.

Step 2: Within the recovery window $R_w$ that is for days $d_{18}$ to $d_{25}$, we will distribute the workload per day as follows:

- Day $d_{18}$: We search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of potential replacement workers having skills to work on task $t_{13}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

  Finally, we find the eligible replacement workers on this day by excluding those working on task $t_{13}$ or any critical task. As workers $p_{11}$ is working on task $t_{13}$ and workers $p_{10}$ is absent, we exclude them from $P_r$ to obtain the final potential replacement workers list as: $P_r = \{p_3, p_9, p_{13}, p_{15}\}$. We assign the 4MD workload of task $t_{13}$ to them by interrupting their present tasks. The list of newly affected tasks due to this reassignment is: $T^a_{new}$=$\{$($t_{10}$,1MD), ($t_{12}$,1MD), ($t_{14}$,2MD)$\}$.

- Days $d_{19}$: The potential replacement workers are same as day $d_{18}$. We assign the remaining 2MD workloads of $t_{13}$ to workers $p_3$ and $p_9$. Since the remaining affected workload of task $t_{13}$ is $w^a_{13}$=0, the distribution of the affected workload for task $t_{13}$

is completed. The updated newly affected task list is: $T^{a}_{new}$={($t_{10}$,2MD),($t_{12}$,2MD), ($t_{14}$,2MD)}.

Step 3: Finished.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | | | | | | | | | | | | | | | | | | |
| **P2** | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | **T10(T11)** | **T10(T11)** | T11 | T11 | T11 | **[T11]** | **[T11]** | | | |
| **P3** | **T13(T10)** | **T13(T10)** | T10 | T10 | T10 | T10 | T10 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P4** | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| **P5** | | | | | | | | | | | | | | | | | | |
| **P6** | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| **P7** | | | | | | | | | | | | | | | | | | |
| **P8** | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | **T12(T11)** | **T12(T11)** | **T12(T11)** | **[T12]** | | | | |
| **P9** | **T13(T12)** | **T13(T12)** | **T14(T12)** | **T14(T12)** | T12 | T12 | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P10** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | T13 | T13 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| **P11** | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P12** | | | | | | | | | | | | | | | | | | |
| **P13** | **T13(T14)** | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P14** | | | | | | | | | | | | | | | | | | |
| **P15** | **T13(T14)** | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P16** | | | | | | | | | | | | | | | | | | |
| **P17** | | | | | | | | | | | | | | | | | | |
| **P18** | | | | | | | | | | | | | | | | | | |
| **P19** | | | | | | | | | | | | | | | | | | |
| **P20** | | | | | | | | | | | | | | | | | | |

Figure 5.8: Revised schedule using option-C for case-2

***Iteration-2:*** We will distribute the newly affected task $T^{a}_{new}$={($t_{10}$,2MD), ($t_{12}$,2MD), ($t_{14}$,2MD)} by finding the suitable change options with the help of our decision module. We sort the newly affected tasks list in ascending order based on the early finish (EF) time of the tasks as: $T^{a}_{new}$={($t_{14}$,2MD), ($t_{10}$,2MD), ($t_{12}$,2MD)}. At first, we will distribute the 2MD affected workload of task $t_{14}$ only. We found that Option-B is suitable for this change scenario ($t_{14}$, 4MD) by following the decision module procedure. We distribute the 2MD affected workload of $t_{14}$ to workers $p_9$ on days $d_{20}$ and $d_{21}$ and update the affected task list by recording the newly affected 2MD workload of task $t_{12}$ as: $T^{a}_{new}$={($t_{10}$,2MD), ($t_{12}$,4MD)}.

***Iteration-3:*** In this stage, we assign the total 2MD affected workload of $t_{10}$ to worker $p_2$ for days $d_{26}$ and $d_{27}$ using option-A by suspending their previously assigned task $t_{11}$ and update the newly affected task list accordingly. The updated newly affected task list is: $T^{a}_{new}$={($t_{12}$,4MD),($t_{11}$,2MD)}.

Table 5.11: Performance of Option-C for case-2

| **Option-C** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 0 | 6 | 13 | 6 | 25 |

***Iteration-4:*** We distribute the 3MD affected workload of $t_{12}$ to worker $p_8$ from days $d_{28}$ to $d_{30}$ using option-A by canceling the previously assigned task $t_{11}$ and the remaining 1MD affected workload as an extra load on day $d_{31}$. So, the updated newly affected task list is: $T^a_{new}$={($t_{11}$,5MD)}.

***Iteration-5:*** Finally, we assign the 4MD affected workload of $t_{11}$ on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$, $p_6$, and $p_{10}$ and the remaining 1MD workload to workers $p_2$ on day $d_{32}$ as an extra load. Since, there are no more affected workload, the schedule shown in Fig. 5.8 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.11 illustrates the over all performance of change option-C for the revised schedule in Fig. 5.8 in terms of delay and number of modified entries.

**Analysis of performance:** Based on the above result, we will examine the quality of the revised schedule by analyzing the delay and reorganization efforts for option-A, option-B and option-C for case-2. Table-5.12 summarizes the results of the revised schedule for the three change options. Option-B and Option-C do not impose any delay for this change scenario. In contrast, the changes cause the 2-day delay of the task $t_{13}$ when we applied option-A to revise the schedule and the delay is propagated to the downstream tasks and finally its delay the project duration by 2 days. Thus, option-B and option-C show better performance in terms of handling the delay than option-A. By comparing the number of total modification, we observe that option-B is better than option-C. So,

Table 5.12: Performance of the three change options for case-2

| Option | Delay | No. of Modifications | | | |
|---|---|---|---|---|---|
| | | Type-1 | Type-2 | Type-3 | Total |
| A | 2 | 24 | 27 | 24 | 75 |
| B | 0 | 6 | 9 | 6 | 21 |
| C | 0 | 6 | 13 | 6 | 25 |

we can conclude that option-B is the best choice among three schedule revision options in this case. This result matches the one acquired by applying the decision based reactive scheduling approach to deal with this change scenario. This approach is considered as efficient to deal with this change scenario in this case. Moreover, we generally consider that change option-B is a better change option to address such kind of medium impact change scenario.

### 5.4.5 Demonstration of Case-3

For the third case, worker $p_{10}$ working on critical task $t_{13}$ is absent for day $d_{19}$ to day $d_{25}$ and worker $p_{11}$ working on same critical task reports to be absent from days $d_{23}$ to $d_{25}$. The change scenario due to their absence is $(t_{13}, 10MD)$ as shown in the task-resource allocation schedule in Fig. 5.9. The step-by-step procedure for the selection of change option and implementation of the selected change option on the task worker schedule are described below:

**Part-A (Selection of the change option):** The basic steps for the decision based system described in Fig. 4.5 in Chapter 4 is used to find the appropriate change option to revise the schedule as follows:

***Decision point-1:*** At decision point-1, the change impact factor $(I_1)$ is calculated using Algorithm-4.1 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{19}$ to $d_{25}$ as first affected date and latest finish (LF) time of the outstanding task $t_{13}$ is $d_{19}$ and $d_{25}$ respectively.

Step 2: With in the recovery window, only workers $p_{13}$ and $p_{15}$ will finish their present task $t_{14}$ and going to start new task $t_{13}$ on day $d_{23}$. Since, they will start the same affected task $t_{13}$, it is meaningless to assign them the same task. We can say that the number of potential replacement workers in this case is zero.

Step 3: Within the recovery window $R_w$, the total maximum distributable workload, $w_{13}^d$ =0.

Step 4:The change impact factor, $I_1 = 10/0 = \infty$; which is larger than 1.

Since the change impact factor $I_1$ in this decision point is greater than 1, Option-A is not suitable to revise the schedule. We should move to the decision point 2 and 3 for the other change options.

***Decision point-2:*** At this decision point, the change impact factor ($I_2$) is calculated using Algorithm-4.2 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{19}$ to $d_{25}$ as first affected date and Latest Finish (LF) time of the outstanding task $t_{13}$ is $d_{19}$ and $d_{25}$ respectively.

Step 2: We search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of potential replacement workers having skills to work on task $t_{13}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

Step 3: Within the recovery window $R_w$, the potential replacement worker can share workloads as follows:

- $p_3$ and $p_9$ can share maximum 7 workloads;
- $p_{10}$ and $p_{11}$ can share 0 workloads as one is absent and other one is working on the same affected task $t_{13}$ on this duration;
- $p_{13}$ and $p_{15}$ can share maximum 4 workloads;

Maximum redistributable workload, $w_{13}^d$=max{7, 7, 0, 0, 4, 4} = 7.

Step 4: The change impact factor, $I_2 = 10/7 = 1.428$; which is greater than 1.

Since, the change impact factor $I_2$ in this decision point is greater than 1, the decision based change option selection method moves to decision point-3 to check whether the Option-C can be applied without imposing delay or delay is unavoidable due to this change scenario.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | | | | | |
| P2 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10(T11) | T10(T11) | T10(T11) | T11 | T11 | [T11] | [T11] | | | |
| P3 | T10 | T13(T10) | T13(T10) | T13(T10) | T10 | T10 | T10 | T10 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] | [T11] | | | |
| P5 | | | | | | | | | | | | | | | | | | |
| P6 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P7 | | | | | | | | | | | | | | | | | | |
| P8 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12(T11) | T12(T11) | T12(T11) | [T12] | [T12] | [T12] | [T12] | |
| P9 | T12 | T13(T12) | T13(T12) | T13(T12) | T14(T12) | T14(T12) | T14(T12) | T14(T12) | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P10 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T11 | T11 | T11 | T11 | T11 | [T11] | | | | |
| P11 | T13 | (T13) | (T13) | (T13) | (T13) | (T13) | (T13) | (T13) | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | | | | | |
| P13 | T14 | T13(T14) | T13(T14) | T14 | T14 | (T13) | (T13) | (T13) | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | | | | | |
| P15 | T14 | T13(T14) | T13(T14) | T14 | T14 | T13 | T13 | T13 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | | | | | |

Figure 5.9: Revised schedule using option-C for case-3

***Decision point-3:*** At the decision point-3, the change impact factor ($I_3$) is calculated using Algorithm-4.2 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{19}$ to $d_{25}$.

Step 2: We search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of potential replacement workers having skills to work on task $t_{13}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

Step 3: Within the recovery window $R_w$, the potential replacement worker can share workloads as follows:

- $p_3$ and $p_9$ can share maximum 7 workloads;
- $p_{10}$ and $p_{11}$ can share 0 workloads as one is absent and other one is working on the same affected task $t_{13}$ during this time;
- $p_{13}$ and $p_{15}$ can share maximum 4 workloads;

Maximum redistributable workload, $w_{13}^a$=7+7+0+0+4+4= 22.

Step 4: The change impact factor, $I_3 = 10/22 = 0.45$; which is less than 1.

Since, the change impact factor $I_3$ is less than 1, Option-C can be used to revise the schedule.

**Part-B (Implementation of the change Option-C):** We will revise the schedule by using the procedure of Option-C as described in Algorithm-4.6 as follows:

***Iteration-1:*** We will distribute the affected 10MD workloads of the outstanding task $t_{13}$ using Option-C.

Step 1: The range of the recovery window $R_w$ is $d_{19}$ to $d_{25}$.

Step 2: We search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of potential replacement workers having skills to work on task $t_{13}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

Step 3: within the recovery window $R_w$ that is for days $d_{19}$ to $d_{25}$, we will distribute the workload per day as follows:

- Day $d_{19}$: We search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. The set of replacement workers having skills to work on task $t_{13}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

  Finally, we find the potential replace working on this day by excluding those who are working on task $t_{13}$ or any critical task. Since, workers $p_{10}$ is working on task $t_{13}$ and workers $p_{11}$ is absent, we exclude them from $P_r$ to obtain the final potential replacement workers list as: $P_r = \{p_3, p_9, p_{13}, p_{15}\}$. We assign the 4MD workload of task $t_{13}$ to them by canceling their initial assigned task. The list of newly affected tasks due to this reassignment is: $T^a_{new}$={($t_{10}$,1MD), ($t_{12}$,1MD), ($t_{14}$,2MD)}.

- Day $d_{20}$: The potential replacement workers are same as day $d_{19}$. We assign 4MD workloads of $t_{13}$ to workers $p_3$, $p_9$, $p_{13}$, $p_{15}$ again and update the newly affected task list as: $T^a_{new}$={($t_{10}$,2MD), ($t_{12}$,2MD), ($t_{14}$,4MD)}.

- Day $d_{21}$: We assign remaining 2MD workloads of $t_{13}$ to workers $p_3$ and $p_9$ by canceling his previously assigned tasks. Since $w^a_{13}$=0, the distribution of the affected workload for task $t_{13}$ is completed. The updated newly affected task list is: $T^a_{new}$={($t_{10}$,3MD), ($t_{12}$,3MD), ($t_{14}$,4MD)}.

Step 4: Finished.

***Iteration-2:*** We have to distribute the newly affected tasks $T^a_{new}$={($t_{10}$,3MD), ($t_{12}$,3MD), ($t_{14}$,4MD)} by finding the suitable change options for each task through decision module. Since there are multiple affected tasks, we will sort the newly affected tasks list in ascending order based on the early finish (EF) time of the tasks as: $T^a_{new}$={($t_{14}$,4MD), ($t_{10}$,3MD), ($t_{12}$,3MD)}. In this stage, we will distribute the affected work of $t_{14}$ only. We found that Option-B is suitable for this change scenario ($t_{14}$, 4MD) by following the decision module. Finally, we assign the 4MD affected workload of $t_{14}$ to workers $p_9$ from day $d_{22}$ and $d_{25}$ and update the affected task list by including the newly affected 4MD workload of task $t_{12}$ as: $T^a_{new}$={($t_{10}$,3MD), ($t_{12}$,7MD)}.

***Iteration-3:*** In this stage, we assign the total 3MD affected workload of $t_{10}$ to worker $p_2$ from day $d_{26}$ to $d_{28}$ using option-A by canceling the previously assigned task $t_{11}$ and update the newly affected task list accordingly. The updated newly affected task list is: $T^a_{new}$={($t_{12}$,7MD),($t_{11}$,3MD)}.

***Iteration-4:*** We distribute the 3MD affected workload of $t_{12}$ to worker $p_8$ from days $d_{28}$ to $d_{30}$ using option-A by canceling the previously assigned task $t_{11}$ and the remaining 4MD affected workload as an extra load from days $d_{31}$ to $d_{34}$. The updated newly affected task list is: $T^a_{new}$={($t_{11}$,6MD)}.

***Iteration-5:*** Finally, the 4MD affected workload of $t_{11}$ is distributed on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$, $p_6$, and $p_{10}$ and the remaining 2MD workload are assigned to workers $p_2$ and $p_4$ on day $d_{32}$ as an extra load. Since, there are no more affected workload, the schedule shown in Fig. 5.9 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.13 illustrates the over all performance of change option-C for the revised schedule in Fig. 5.9 in terms of delay and number of modified entries.

Table 5.13: Performance of Option-C for case-3

| **Option-C** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 0 | 10 | 20 | 10 | 40 |

### 5.4.6 Validation of Case-3

To compare the results of applying different change options to revise the schedule, we demonstrate two other schedule revision processes by applying the remaining two change options (i.e. option-A and Option-B) as follows:

**Implementation of the change Option-A:** First, we will revise the schedule by using the procedure of Option-A as described in Algorithm-4.4 as follows:

***Iteration-1:*** We will distribute the affected 10MD workloads of the outstanding task $t_{13}$ using Option-A.

Step 1: The first affected date and latest finish (LF) time of the outstanding task $t_{13}$ is $d_{19}$ and $d_{25}$ respectively. Hence, the range of the recovery window $R_w$ is $d_{19}$ to $d_{25}$.

Step 2: Within the recovery window $R_w$, we will distribute the workload per day as follows:

- Days $d_{18}$ to $d_{25}$: Only workers $p_{13}$ and $p_{15}$ will finish their present task $t_{14}$ and switch to new task $t_{13}$ on day $d_{23}$. Since they will start the same affected task $t_{13}$, it is meaningless to assign them the same task.

- Day $d_{26}$: We have to distribute the affected workload $w_{13} = 10MD$ of task $t_{13}$ after the recovery window $R_w$. At first, we search for the capable replacement workers through the mapping of task-skill and resource-skill matrix as defined in Table-5.2 and Table-5.3. The set of potential replacement workers is found by keeping those workers who previously worked on the affected task $t_{13}$ or now working on its successor tasks. The list of potential replacement worker is $P_r = \{p_3, p_{10}, p_{11}, p_{13}, p_{15}\}$. We assign the affected workload $(t_{13}, 5MD)$ to workers $p_3$, $p_{10}$, $p_{11}$, $p_{13}$, and $p_{15}$ as shown in Fig. 5.10 and record the newly affected task as: $T^a_{new}$={($t_{11}$,1MD),($t_{12}$,1MD),($t_{15}$,3MD)}.

- Day $d_{27}$: We distribute the remaining 5MD workload of task $t_{13}$ to worker $p_3$, $p_{10}$, $p_{11}$, $p_{13}$, and $p_{15}$ again and record the newly affected tasks as: $T^a_{new}$={($t_{11}$,2MD), ($t_{12}$,2MD), ($t_{15}$,6MD)}. Since $w^a_{13}$=0, the distribution of the affected workload for task $t_{13}$ is completed.

***Iteration-2:*** We will distribute the newly affected task $T^a_{new}$ by finding the suitable change options with the helps of the decision module. We will sort the newly affected tasks list in ascending order based on the early finish (EF) time of the tasks as: $T^a_{new}$={($t_{12}$,2MD),($t_{11}$,2MD),($t_{15}$,6MD)}.

Now, we will distribute the 2MD affected workload of $t_{12}$ using option-A to worker $p_8$ on days $d_{28}$ and $d_{29}$ by interrupting his previously assigned task $t_{11}$. The newly affected task list: $T^a_{new}$={($t_{11}$,4MD), ($t_{15}$,6MD)}.

***Iteration-3:*** In this stage, we assign the total 4MD affected workload of $t_{11}$ to workers $p_2$, $p_4$, $p_8$ and $p_{10}$ on day $d_{31}$ using option-A as an extra load. The remaining affected task list is: $T^a_{new}$={($t_{15}$,6MD)}.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 | D32 | D33 | D34 | D35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | | | | | | | | | | | | | | | | | | |
| **P2** | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| **P3** | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | **T13(T15)** | **T13(T15)** | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P4** | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | **[T11]** | | | | |
| **P5** | | | | | | | | | | | | | | | | | | |
| **P6** | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | | | | | |
| **P7** | | | | | | | | | | | | | | | | | | |
| **P8** | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | **T12(T11)** | **T12(T11)** | **T11** | **[T11]** | | | | |
| **P9** | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P10** | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | **T13(T11)** | **T13(T11)** | T11 | T11 | T11 | **[T11]** | | | | |
| **P11** | T13 | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **(T13)** | **T13(T12)** | **T13(T12)** | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P12** | | | | | | | | | | | | | | | | | | |
| **P13** | T14 | T14 | T14 | T14 | T14 | **(T13)** | **(T13)** | **(T13)** | **T13(T15)** | **T13(T15)** | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P14** | | | | | | | | | | | | | | | | | | |
| **P15** | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | **T13(T15)** | **T13(T15)** | T15 | T15 | T15 | T15 | T15 | T15 | T15 | T15 |
| **P16** | | | | | | | | | | | | | | | | | | |
| **P17** | | | | | | | | | | | | | | | | | | |
| **P18** | | | | | | | | | | | | | | | | | | |
| **P19** | | | | | | | | | | | | | | | | | | |
| **P20** | | | | | | | | | | | | | | | | | | |

(a) First part (days $d_{18}$ to $d_{35}$)

| | D36 | D37 | D38 | D39 | D40 | D41 | D42 | D43 | D44 | D45 | D46 | D47 | D48 | D49 | D50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P1** | | | | | **T16(T17)** | **T16(T17)** | T17 | T17 | T17 | **T17(T20)** | **T17(T20)** | T20 | **[T20]** | **[T20]** | |
| **P2** | | | | | | | | | | | | | | | |
| **P3** | **[T15]** | | | | | | | | | | | | | | |
| **P4** | | | | | | | | | | | | | | | |
| **P5** | **(T16)** | **(T16)** | T16 | T16 | **T16(T17)** | **T16(T17)** | T17 | T17 | T17 | **T17(T20)** | **T17(T20)** | T20 | **[T20]** | **[T20]** | |
| **P6** | | | | | | | | | | | | | | | |
| **P7** | | | | | | | | | | | | | | | |
| **P8** | | | | | | | | | | | | | | | |
| **P9** | **[T15]** | | | | | | | | | | | | | | |
| **P10** | | | | | | | | | | | | | | | |
| **P11** | | | | | | | | | | | | | | | |
| **P12** | **T15(T16)** | **T15(T16)** | T16 | T16 | **T16(T17)** | **T16(T17)** | T17 | T17 | T17 | **T17(T20)** | **T17(T20)** | T20 | **[T20]** | **[T20]** | |
| **P13** | **T15(T16)** | **(T16)** | T16 | T16 | **T16(T17)** | **(T17)** | T17 | T17 | T17 | **[T17]** | **[T17]** | | | | |
| **P14** | | | | | | | | | | | | | | | |
| **P15** | **T15(T16)** | **(T16)** | T16 | T16 | **T16(T17)** | **(T17)** | T17 | T17 | T17 | **[T17]** | **[T17]** | | | | |
| **P16** | | | | | **(T18)** | **(T18)** | T18 | T18 | T18 | **T18(T19)** | **[T18]** | **[T19]** | | | |
| **P17** | | | | | **(T18)** | **(T18)** | T18 | T18 | T18 | **T18(T19)** | **[T18]** | **[T19]** | | | |
| **P18** | | | | | **(T18)** | **(T18)** | T18 | T18 | T18 | **T18(T19)** | **[T18]** | **[T19]** | | | |
| **P19** | | | | | **(T19)** | **(T19)** | T19 | T19 | T19 | T19 | **[T19]** | **[T19]** | | | |
| **P20** | | | | | **(T19)** | **(T19)** | T19 | T19 | T19 | T19 | **[T19]** | **[T19]** | | | |

(b) Second part (days $d_{36}$ to $d_{50}$)

Figure 5.10: Revise schedule using option-A for case-3.

***Iteration-4:*** The recovery window for the affected task $t_{15}$ is day $d_{26}$ to $d_{35}$. From the task worker allocation schedule and using the decision module, it is found that option-A, option-B or option-C could not revise the schedule without imposing delay within the recovery window for the change scenario ($t_{15}$,6MD). We distribute the affected workload of $t_{15}$ using extended procedure of Option-C (i.e. Algorithm-4.7)on days $d_{36}$ and $d_{37}$ and defer the task $t_{16}$ by two days . The list of newly affected task is: $T^a_{new}$={($t_{16}$,8MD)}.

***Iteration-5:*** Similarly, we distribute the affected workload of ($t_{16}$,8MD) using option-C (extended) on days $d_{40}$ and $d_{41}$ and defers the tasks $t_{17}$, $t_{18}$ and $t_{19}$. The list of newly affected tasks is: $T^a_{new}$={($t_{17}$,10MD),($t_{18}$,6MD),($t_{19}$,4MD)}.

***Iteration-6:*** The affected workload of $(t_{17}, 8MD)$ is distributed using option-C (extended) on days $d_{45}$ and $d_{46}$ which defers the task $t_{20}$ for two days. The updated list of newly affected task, $T^a_{new}$={($t_{18}$,6MD),($t_{19}$,4MD),($t_{20}$,6MD)}.

***Iteration-7:*** The affected workload of tasks ($t_{18}$,6MD) and ($t_{19}$,4MD) are distributed using option-A on days $d_{46}$ and $d_{47}$ and the list of newly affected task is: $T^a_{new}$={($t_{20}$,6MD)}.

***Iteration-8:*** Finally, The affected workload of tasks ($t_{20}$,6MD) is distributed using Option-C (extended) on days $d_{48}$ and $d_{49}$. Since, there are no more affected workload, the schedule shown in Fig. 5.10 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.14 illustrates the over all performance of change option-A for the revised schedule in Fig. 5.10 in terms of delay and number of modified entries.

Table 5.14: Performance of Option-A for case-3

| **Option-A** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 2 | 26 | 33 | 26 | 85 |

**Part-B (Implementation of the change option-B):** Lastly, we will revise the schedule using the procedure of Option-B as described in Algorithm-4.5 by using the following steps:

***Iteration-1:*** We will distribute the affected 10MD workloads of the outstanding task $t_{13}$ using option-B as follows:

Step 1: The range of the recovery window $R_w$ is $d_{19}$ to $d_{25}$.

Step 2: We search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. Hence, the set of potential replacement workers having skills to work on task $t_{13}$ is $P_r = \{p_3, p_9, p_{10}, p_{11}, p_{13}, p_{15}\}$.

**Step 3:** Within the recovery window $R_w$, the potential replacement workers can share workloads as follows:

- $p_3$ and $p_9$ can share maximum 7 workloads;

- $p_{10}$ and $p_{11}$ can share 0 workloads as one is absent and other one is working on the same affected task $t_{13}$ on this duration;

$p_{13}$ and $p_{15}$ can share maximum 4 workloads;

Step 3: Since, only one replacement workers is needed for option-B, workers $p_3$ and $p_9$ is considered first as they can share maximum 7MD workloads among the 6 potential replacement workers within the recovery window. Although they can share the same amount of maximum additional workload (i.e., 7MD), we should select worker $p_9$ as the best suitable potential replacement worker as the task $t_{12}$, he is working has largest early finish (EF) time compare to the tasks $t_{10}$ of the workers $p_3$. The list of potential candidates who are selected to work on the affected task is sorted descending order as: $P_r$=$\{(p_9, 7MD), (p_3, 7MD)\}$

Step 4: Finally, the affected workload $(t_{13}, 7MD)$ is assigned to worker $p_9$ within the recovery window as shown in Fig. 5.11. On the other hand, we need to distribute the remaining affected workload of task $t_{13}$ after the recovery window by finding the potential workers workers who previously worked on the affected task $t_{13}$ or now working on its successor tasks. Now, we assign the affected workload $(t_{13}, 3MD)$ to workers $p_3$, $p_{13}$ and $p_{15}$ on day $d_{26}$ and defer the task $t_{15}$ by one day as shown in Fig. 5.11 and record the newly affected task as: $T^a_{new}$={($t_{12}$,7MD), ($t_{15}$,3MD)}. Since $w^a_{13}$=0, the distribution of the affected workload for task $t_{13}$ is completed.

Step 5: Finished.

***Iteration-2:*** We will distribute the newly affected task $T^a_{new}$={($t_{12}$,7MD)} by finding the suitable change options through decision module. In this stage, we found that Option-A is suitable for this change scenario $T^a_{new}$={($t_{12}$,7MD)}. Finally, we distribute the 3MD affected workload of $t_{12}$ to worker $p_8$ from days $d_{28}$ to $d_{30}$ using option-A by canceling the previously assigned task $t_{11}$ and the remaining 4MD affected workload as an extra load from days $d_{31}$ to $d_{34}$. The updated newly affected task list is: $T^a_{new}$={($t_{11}$,3MD), ($t_{15}$,3MD)}.

***Iteration-3:*** We distribute the 3MD affected workload of $t_{11}$ on day $d_{31}$ among the potential replacement workers $p_2$, $p_4$, and $p_6$ as an extra load using option-A. The remaining list of tasks is:$T^a_{new}$={($t_{15}$,3MD)}.

***Iteration-4:*** The recovery window for the affected task $t_{15}$ is day $d_{26}$ to $d_{35}$. From the task worker allocation schedule and using the decision module, it is found that option-A, option-B or option-C could not revise the schedule without imposing delay within the recovery window for the change scenario ($t_{15}$,3MD). We need to distribute the affected workload of $t_{15}$ using option-C (Extended) on day $d_{36}$ and defer the task $t_{16}$ by one day. The list of newly affected task is: $T^a_{new}$={($t_{16}$,4MD)}.

***Iteration-5:*** Similarly, we distribute the affected workload of ($t_{16}$,4MD) using option-C (Extended) on day $d_{40}$ and defers the tasks $t_{17}$, $t_{18}$ and $t_{19}$. The list of newly affected tasks is: $T^a_{new}$={($t_{17}$,5MD),($t_{18}$,3MD),($t_{19}$,2MD)}.

***Iteration-6:*** The affected workload of $(t_{17}, 5MD)$ is distributed using option-C (Extended) on day $d_{45}$ which defers the task $t_{20}$ for one day. The updated list of newly affected task, $T^a_{new}$={($t_{18}$,3MD),($t_{19}$,2MD),($t_{20}$,3MD)}.

| | D18 | D19 | D20 | D21 | D22 | D23 | D24 | D25 | D26 | D27 | D28 | D29 | D30 | D31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | | | | | | |
| P2 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T11 | T11 | T11 | T11 | T11 | [T11] |
| P3 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T10 | T13(T15) | T15 | T15 | T15 | T15 | T15 |
| P4 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] |
| P5 | | | | | | | | | | | | | | |
| P6 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | T11 | [T11] |
| P7 | | | | | | | | | | | | | | |
| P8 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12 | T12(T11) | T12(T11) | T12(T11) | [T12] |
| P9 | T12 | T13(T12) | T13(T12) | T13(T12) | T13(T12) | T13(T12) | T13(T12) | T13(T12) | T12 | T12 | T15 | T15 | T15 | T15 |
| P10 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T13 | T11 | T11 | T11 | T11 | T11 | |
| P11 | T13 | (T13) | (T13) | (T13) | (T13) | (T13) | (T13) | (T13) | T12 | T12 | T15 | T15 | T15 | T15 |
| P12 | | | | | | | | | | | | | | |
| P13 | T14 | T14 | T14 | T14 | T14 | (T13) | (T13) | (T13) | T13(T15) | T15 | T15 | T15 | T15 | T15 |
| P14 | | | | | | | | | | | | | | |
| P15 | T14 | T14 | T14 | T14 | T14 | T13 | T13 | T13 | T13(T15) | T15 | T15 | T15 | T15 | T15 |
| P16 | | | | | | | | | | | | | | |
| P17 | | | | | | | | | | | | | | |
| P18 | | | | | | | | | | | | | | |
| P19 | | | | | | | | | | | | | | |
| P20 | | | | | | | | | | | | | | |

(a) First part (days $d_{18}$ to $d_{31}$)

| | D32 | D33 | D34 | D35 | D36 | D37 | D38 | D39 | D40 | D41 | D42 | D43 | D44 | D45 | D46 | D47 | D48 | D49 | D50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | | | | (T17) | T17 | T17 | T17 | T17 | T17(T20) | T20 | T20 | [T20] | | |
| P2 | | | | | | | | | | | | | | | | | | | |
| P3 | T15 | T15 | T15 | T15 | | | | | | | | | | | | | | | |
| P4 | | | | | | | | | | | | | | | | | | | |
| P5 | | | | | (T16) | T16 | T16 | T16 | T16(T17) | T17 | T17 | T17 | T17 | T17(T20) | T20 | T20 | [T20] | | |
| P6 | | | | | | | | | | | | | | | | | | | |
| P7 | | | | | | | | | | | | | | | | | | | |
| P8 | [T12] | [T12] | [T12] | | | | | | | | | | | | | | | | |
| P9 | T15 | T15 | T15 | T15 | | | | | | | | | | | | | | | |
| P10 | | | | | | | | | | | | | | | | | | | |
| P11 | T15 | T15 | T15 | T15 | [T15] | | | | | | | | | | | | | | |
| P12 | | | | | (T16) | T16 | T16 | T16 | (T17) | T17 | T17 | T17 | T17 | T17(T20) | T20 | T20 | [T20] | | |
| P13 | T15 | T15 | T15 | T15 | T15(T16) | T16 | T16 | T16 | T16(T17) | T17 | T17 | T17 | T17 | [T17] | | | | | |
| P14 | | | | | | | | | | | | | | | | | | | |
| P15 | T15 | T15 | T15 | T15 | T15(T16) | T16 | T16 | T16 | T16(T17) | T17 | T17 | T17 | T17 | [T17] | | | | | |
| P16 | | | | | | | | | (T18) | T18 | T18 | T18 | T18 | T18(T19) | [T19] | | | | |
| P17 | | | | | | | | | (T18) | T18 | T18 | T18 | T18 | T18(T19) | [T19] | | | | |
| P18 | | | | | | | | | (T18) | T18 | T18 | T18 | T18 | T18(T19) | [T19] | | | | |
| P19 | | | | | | | | | (T19) | T19 | T19 | T19 | T19 | T19 | [T19] | | | | |
| P20 | | | | | | | | | (T19) | T19 | T19 | T19 | T19 | T19 | [T19] | | | | |

(b) Second part (days $d_{32}$ to $d_{50}$)

Figure 5.11: Revise schedule using option-B for case-3.

***Iteration-7:*** The affected workload of tasks ($t_{18}$,3MD) and ($t_{19}$,2MD) is distributed using option-A and the list of newly affected task is: $T^a_{new}$={($t_{20}$,3MD)}.

***Iteration-8:*** Finally, the affected workload of tasks ($t_{20}$,3MD) is distributed using option-C (Extended)) on day $d_{48}$. Since there are no more affected workload, the schedule shown in Fig. 5.11 is the final revised schedule due to this change scenario.

To examine the quality of the revised schedules, we check the number of extended days pertaining to the final task(s) of the project in view of project delay and the number of modified entries in the worker allocation schedule. The number of modified entries is the summation of the three types of modification (i.e., type-1, type-2 and type-3). Table-5.15 illustrates the overall performance of change option-B for the revised schedule in Fig. 5.11 in terms of delay and number of modified entries.

Table 5.15: Performance of Option-B for case-3

| **Option-B** | **Delay** | **No. of Modifications** | | | |
|---|---|---|---|---|---|
| | | **Type-1** | **Type-2** | **Type-3** | **Total** |
| | 1 | 19 | 24 | 18 | 61 |

**Analysis of performance:** Now we will examine the quality of the revised schedule by analyzing the delay and reorganization efforts for option-A, option-B and option-C for the case-3. Table-5.16 summarizes the results of the revised schedule for the three change options. Both Option-A and Option-B impose delay for this case scenario in the revise schedule. In contrast, Option-C does not incur any delay. By comparing the number of total modifications, we observe that option-C has least number reorganization effort. So, we can conclude that option-C is the best choice among three schedule revision options in this case. This result matches the one acquired by applying the decision based reactive scheduling approach to deal with this change scenario. This approach is considered as efficient to deal with this change scenario in this case. Moreover, we generally consider that change option-C is a better change option to address such kind of high impact change scenario.

Table 5.16: Performance of the three change options for case-3

| Option | Delay | No. of Modifications | | | |
|---|---|---|---|---|---|
| | | Type-1 | Type-2 | Type-3 | Total |
| A | 2 | 26 | 33 | 26 | 85 |
| B | 1 | 19 | 24 | 18 | 61 |
| C | 0 | 10 | 20 | 10 | 40 |

### 5.4.7 Demonstration of Case-4

In the fourth case, worker $p_{12}$ working on critical task $t_{17}$ is absent for days $d_{40}$ to $d_{44}$. The change scenario due to his absence is $(t_{17}, 5MD)$ as shown in the task-resource allocation schedule in Fig. 5.12. The step-by-step procedure for the selection of change option and implementation of the selected change option on the task worker schedule are described below:

**Part-A (Selection of the change option):** The basic steps for the decision based system described in Fig. 4.5 in Chapter 4 is used to find the appropriate change option to revise the schedule as follows:

***Decision point-1:*** At the decision point-1, the change impact factor $(I_1)$ is calculated using Algorithm-4.1 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{40}$ to $d_{44}$.

Step 2: With in the recovery window, workers $p_1$, $p_5$, $p_{12}$, $p_{13}$ and $p_{15}$ are available and consider as potential replacement workers. But they are working on the same affected task $t_{17}$ within the recovery window. In this context, we can say that it is worthless to assign them the same affected task. We can say that the number of potential replacement workers in this case is zero.

Step 3: Within the recovery window $R_w$, the total maximum distributable workload, $w_{17}^d$ =0.

Step 4: The change impact factor, $I_1 = 5/0 = \infty$; which is larger than 1.

Since the change impact factor $I_1$ in this decision point is greater than 1, Option-A is not suitable to revise the schedule. We should move to the decision point 2 and 3 for the other change options.

***Decision point-2:*** At the decision point-2, the change impact factor ($I_2$) is calculated using Algorithm-4.2 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{40}$ to $d_{44}$.

Step 2: Now, we search for the potential replacement workers through the mapping of skill requirements for the tasks defined in Table-5.2 and skills master by the workers defined in Table-5.3. Hence, the set of potential replacement workers having skills to work on task $t_{17}$ is $P_r = \{p_1, p_5, p_{12}, p_{13}, p_{15}\}$.

Step 3: Within the recovery window $R_w$, the potential replacement worker can not share any workload as they are assigned the same task $t_{17}$. So, the maximum redistributable workload, $w^d_{17}$=max{0, 0, 0, 0, 0} = 0.

Step 4: The change impact factor, $I_2 = 5/0 = \infty$; which is larger than 1.

Since the change impact factor $I_2$ in this decision point is greater than 1, the decision based change option selection method moves to decision point-3 to check whether the Option-C can be applied without imposing delay or delay is unavoidable due to this change scenario.

***Decision point-3:*** At the decision point-3, the change impact factor ($I_3$) is calculated using Algorithm-4.3 as follows:

Step 1: The range of the recovery window $R_w$ is $d_{40}$ to $d_{44}$.

Step 2: The set of potential replacement workers having skills to work on task $t_{17}$ is $P_r = \{p_1, p_5, p_{12}, p_{13}, p_{15}\}$.

Step 3: Within the recovery window $R_w$, the potential replacement worker can not share any workload as they are assigned the same task $t_{17}$. So, the maximum redistributable workload, $w^d_{17}$={0 + 0 + 0 + 0 + 0} = 0.

Step 4: The change impact factor, $I_3 = 5/0 = \infty$; which is larger than 1.

Since the change impact factor $I_3$ is greater than 1, it indicates that the delay due to this change scenario is unavoidable. Hence, The extended procedure of change Option-C (i.e., Algorithm-4.7) should be used to handle the delay propagation in a systematic way.

| | D35 | D36 | D37 | D38 | D39 | D40 | D41 | D42 | D43 | D44 | D45 | D46 | D47 | D48 | D49 | D50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P1 | | | | | | T17 | T17 | T17 | T17 | T17 | **T17(T20)** | T20 | T20 | **[T20]** | | |
| P2 | | | | | | | | | | | | | | | | |
| P3 | T15 | | | | | | | | | | | | | | | |
| P4 | | | | | | | | | | | | | | | | |
| P5 | | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | **T17(T20)** | T20 | T20 | **[T20]** | | |
| P6 | | | | | | | | | | | | | | | | |
| P7 | | | | | | | | | | | | | | | | |
| P8 | | | | | | | | | | | | | | | | |
| P9 | T15 | | | | | | | | | | | | | | | |
| P10 | | | | | | | | | | | | | | | | |
| P11 | T15 | | | | | | | | | | | | | | | |
| P12 | | T16 | T16 | T16 | T16 | **(T17)** | **(T17)** | **(T17)** | **(T17)** | **(T17)** | **T17(T20)** | T20 | T20 | **[T20]** | | |
| P13 | T15 | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | **[T17]** | | | | | |
| P14 | | | | | | | | | | | | | | | | |
| P15 | T15 | T16 | T16 | T16 | T16 | T17 | T17 | T17 | T17 | T17 | **[T17]** | | | | | |
| P16 | | | | | | T18 | T18 | T18 | T18 | T18 | T19 | | | | | |
| P17 | | | | | | T18 | T18 | T18 | T18 | T18 | T19 | | | | | |
| P18 | | | | | | T18 | T18 | T18 | T18 | T18 | T19 | | | | | |
| P19 | | | | | | T19 | T19 | T19 | T19 | T19 | T19 | | | | | |
| P20 | | | | | | T19 | T19 | T19 | T19 | T19 | T19 | | | | | |

Figure 5.12: Revised schedule using option-C (Extended) for case-4

**Implementation of the change option-C (Extended version):** Now we will revise the schedule using the extended procedure of option-C as described in Algorithm-4.7 by using the following steps:

***Iteration-1:*** First, we will try to distribute the affected 5MD workloads of the outstanding task $t_{17}$ using general procedure of change Option-C (i.e., Algorithm-4.6).

Step 1: The first affected date and Latest Finish (LF) time of the outstanding task $t_{17}$ is $d_{40}$ and $d_{44}$ respectively. So, the range of the recovery window $R_w$ is $d_{40}$ to $d_{44}$.

Step 2: Within the recovery window $R_w$, the set of potential replacement workers having skills to work on task $t_{17}$ is $P_r = \{p_1, p_5, p_{12}, p_{13}, p_{15}\}$. But they could not share any workload during this period as they are assigned in the same task $t_{17}$. So, we are not able to distribute any workload within the recovery window.

Step 3: In this stage, we have to distribute the affected workload $w^a_{17} = 5MD$ of task $t_{17}$ after the recovery window $R_w$. At first, we search for the capable replacement workers through the mapping of task-skill and resource-skill matrix as defined in Table-5.2 and Table-5.3. The set of potential replacement workers is found by keeping those workers who previously worked on the affected task $t_{17}$ or now working on its successor tasks. The list of potential replacement worker is $P_r = \{p_1, p_5, p_{12}, p_{13}, p_{15}\}$. We assign the affected workload $(t_{17}, 3MD)$ to workers $p_1$, $p_5$ and $p_{12}$ by interrupting their task $t_{20}$ on day $d_{45}$. Besides, we distribute the affected workload $(t_{17}, 2MD)$ to workers $p_{13}$ and $p_{15}$ as an extra load on the same day as shown in Fig. 5.12 and record the newly affected task as: $T^a_{new}$=$\{(t_{20}$,3MD$)\}$.

***Iteration-2:*** Finally, The affected workload of tasks ($t_{20}$,3MD) is distributed using option-C (Extended) on day $d_{48}$ as an extra load to workers $p_1$, $p_5$ and $p_{12}$ . Since there are no more affected workload, the schedule shown in Fig. 5.12 is the final revised schedule due to this change scenario.

## 5.5 Summary

In this chapter, we have demonstrated through an illustrative example how the proposed framework can be applied to address changes due to the absence of workers during project execution. We also examined the performance of the rescheduling procedures in repairing the baseline schedule whenever a change occurs. By exploring various cases of workers absence, the decision-based reactive scheduling approach proved to be efficient and practical in selecting the correct change options and providing a revised schedule. From the above demonstration and the experimental result, it is observed that if less flexible change options (i.e., Option-A and Option-B) are used to handle high change impact, the effected task incurred delays which propagated to the subsequent tasks, causing a delay to the project as a whole. Moreover, this outcome introduced an unexpected number of re-organization efforts. On the other hand, implementing the most flexible options (i.e, Option-C) to handle less impact changes caused extra perturbation in the schedule. By comparing the outcomes of three revised schedules using two criteria (e.g., project delay and the re-organization effort), the results of most cases matched our expectation. That is, implementing the change option selected by decision based module of the reactive scheduling framework are able to minimize project delay and re-organization efforts.

# Chapter 6

# Conclusion

## 6.1 Summary

Project schedules are key documents for proper management of projects. But, the schedules are subject to be changed due to uncertain aspect of a project. In general, it is extremely difficult to deal with schedule change management due to the extensive interaction between activities, resources and project stakeholders besides time and budget constraints. In real-world environments, most challenges include measuring the impact of changes, selecting the right time to react, and determining the appropriate corrective actions to control and minimize the impact of changes. This research work should be viewed as an attempt to address such issues due to the absence of workers in resource limited project scheduling environment.

In this research, we have proposed a systematic reactive scheduling approach to deal with changes due to absence of workers. The change management related to worker unavailability is extremely difficult due to multi-skill workers and exponential number of ways of assigning activities to workers. Moreover, it becomes more challenging when a project is executing with limited proprietary resources, i.e., without sharing resources of other projects or third parties. In such cases, the unexpected absence or sudden turnover of worker(s) can significantly affect the timely delivery of the projects. Traditionally, the project managers update the baseline schedule considering such unexpected events or changes based on their experiences and skills. This practice is typically based on intuition and its reliability varies from case to case. Most of the time, the project managers will face challenges to make proper decisions for change management especially when the project has new contents and complex structures, and the required skill sets become more specific. Generally, the initiatives made by the project managers in such situation are the product of a poor evaluation of the changes and result in improper corrective actions. Finally, this may lead to project delay, cost overrun, employee malcontent, etc. Our proposed systematic reactive scheduling approach will help the project managers to deal with such changes in project managements in an effective and efficient ways. To the best of our knowledge, there are not much rescheduling tools that provide approaches to systematically update the schedule to cope with changes for the absence of workers.

In the first phase of our research, we placed great emphasis on developing a standard model for a baseline schedule. A task-resource allocation schedule was modeled by exploiting the features of a general multi-mode and multi-skill resource constraint project scheduling problems. The next component of our research centered on the development of reactive strategies and methods that could be used when some aspect of a project schedule is changed. The emphasis was on both modeling and developing methods for rescheduling problems. We proposed a systematic approach to modify the baseline schedule to address changes due to the absent of workers during project execution time. The objective of the proposed method was to revise the schedule in a way that limits the increasing of the project duration from its deadline without changing too many task-worker assignments. To address the objective of the reactive scheduling, we defined different change options to revise the schedule according to the different levels of change impact. The change impact is assessed based on the importance of the absent worker, the length of the absence, and the criticality of the affected tasks. The suggested approach deals with changes systematically by analyzing their significance and selecting the appropriate change options through a decision module to control and minimizes the impact of changes. By exploring various cases concerning the absence of the workers in software development projects, the decision-based reactive scheduling approach proved to be efficient. That is, implementing the change option selected by the decision based approach results in minimizing project delays and re-organization efforts. Our proposed systematic rescheduling approach will help project managers to deal with changes in projects effectively and efficiently. To our knowledge, our proposed approach is the first that allows project managers to cope with changes in systematic way.

## 6.2 Contributions

In this thesis, we have developed a systematic reactive scheduling approach to modify the task-resource allocation schedule to address the changes due to the absence of worker(s) during project execution time. Based on the current development in the literature, this research makes a number of contributions:

- **Standard task-resource allocation schedule:** This research has introduced a standard baseline schedule model that assists to intuitively represent the relationship among tasks, resources with respect to time. It also provides helpful environment for solving resource allocation and workload distribution problems for both multi-mode and multi-skill resource schedules. Moreover, the schedule enables the project management team to identify changes, evaluate change impact, and finally, modify the schedule. Additionally, the extended Gantt chart of this research provides additional feature to keep track of the workload distribution of various tasks on a daily basis.

- **Reactive scheduling approach for multi-mode and multi-skill baseline schedule:** The developed reactive scheduling approach is able to revise baseline schedule having multi-mode activity durations. Moreover, the approach can also handle multi-skill resources properly.

- **Different reactive solution procedures for different level of changes:** This research work has taken into account the impact of changes in reactive solution strategy. We defined different change options to revise the schedule for different levels of change impact (i.e., low, medium, and high). The reactive procedure can modify the schedule by selecting the best change options that minimizes the project delay as well as the overall re-organization effort in the revised schedule.

- **Partial reactive scheduling approach:** The approach performed partial rescheduling within a recovery window that helps to control the delay more locally in each task level.

## 6.3 Future Work

The proposed reactive scheduling framework serves as a starting point for a solution that will assist project managers in dealing with unexpected events during project execution. Many important issues in change management are yet to be considered and solved. There are several possible avenues for refining and extending the capability of our presented approach. The followings are a few possible directions for future enhancement to our work.

The first major research direction concerns the model used. The scope of the model used could be extended by capturing more essence of real world project problems and practices. More complex interdependencies among the entities of the project, such as minimum and maximum time lags and dependency between activities other than finish to start relationship, should be considered. One other direction is to include additional resources (e.g., machine and materials) and more constraints to obtain a more precise model. Furthermore, additional ways of describing the scheduling and resource allocation requirements are needed to include more real-world situations, such as over time work and multiple calenders. Another important direction is to investigate the use of our proposed approach in multi-project scheduling environment where resources are shared among several projects. At present, our proposed approach takes the unavailability of resources as the only unexpected events. Due to the various unexpected events in real project execution, the capability of the proposed method should be extended to handle other changes like network disruptions (e.g., adding or removing of tasks, altering the precedence relationship due to urgency of some tasks), unexpected progress of the tasks (e.g., task behind the planned schedule), and rework.

The second major research direction concerns the reactive solution approach. In particular, more specific change options should be identified to cope with more specific change scenarios. Another direction would be to find a reactive scheduling approach that modifies the schedule offering a fair trade-off between time, cost and quality.

The existing application of the proposed reactive scheduling approach modifies the schedule manually. A prototype (i.e., series of codes) could be developed for this reactive scheduling method so that all the procedures are integrated in a user-friendly automated environment. Moreover, it might be more effective to link such prototype to existing commercial project management software in order to analyze schedule delays and perturbations and finally performing necessary schedule modifications.

Another important direction of future research is to apply the ideas developed in the thesis to other application domain, such as medical, airlines, railway scheduling.

# Bibliography

Alcaraz, J., Maroto, C., and Ruiz, R. (2003). Solving the multi-mode resource-constrained project scheduling problem with genetic algorithms. *Journal of the Operational Research Society*, 54:614 –626.

Artigues, C., Demassey, S., and Nron, E. (2010). *Resource-Constrained Project Scheduling: Models, Algorithms, Extensions and Applications.* ISTE,UK, Wiley.

Aytuga, J., Lawleyb, M. A., McKayc, K., Mohand, S., and Uzsoye, R. (2005). Executing production schedules in the face of uncertainties: A review and some future directions. *European Journal of Operational Research*, 161(1):86 –110.

Bagheri, M., Jolai, F., and Aryanezhad, M. B. (2012). Stability analysis in project resource levelling problem with stochastic activity durations. *World Applied Sciences Journal*, 16(1):126 –134.

Bellenguez, G. and Nron, E. (2005). Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills. *Lecture Notes in Computer Science*, 3616:229 –243.

Bellenguez, O. (2008). Method to solve multi-skilled project scheduling problem. *4OR: A Quarterly Journal of Operation Research*, 6(1).

Bellenguez, O. and Neron, E. (2004). An exact method for solving the multi-skill project scheduling problems. *Operational Research Conference*, pages 97 –98.

Brucker, P., Knust, S., and Nron, E. (2012). *Complex Scheduling.* Springer Berlin Heidelberg.

Chen, J. (2010). Development of the change management method for small and medium construction projects. Master's thesis, Concordia University.

Dablaere, F., Demeulemeester, E., and Herroelen, W. (2011). Proactive policies for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 214(2):308 –316.

Deblaere, F., Demeulemeester, E., and Herroelen, W. (2011). Reactive scheduling in the multi-mode rcpsp. *Journal of computer and Operational Research*, 38:63 –74.

Drezet, L. E. and Billaut, J. (2007). A project schedule problem with labour constraints and time-dependent activities requirements. *International Journal of Production Economics*, 112:217 –225.

Hartmann, S. (2012). Project scheduling with resource capacities and requests varying with time: a case study. *Flexible Services and Manufacturing Journal*.

Hartmann, S. and Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1):1 –14.

Herroelen, W. and Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289 –306.

Katragjini, K., Vallada, E., and Ruiz, R. (2012). Flow shop rescheduling under different types of disruption. *International Journal of Production Research*, pages 1 –12.

Kuster, J., jannach, D., and Friendrich, G. (2010). Applying local rescheduling in response to schedule disruptions. *Annals of Operations Research*, 180(1):265 –282.

Lambrechts, O., Demeulemeester, E., and Herroelen, W. (2008a). Proactive and reactive strategies for resource-constrained project scheduling with uncertain resource availabilities. *Journal of Scheduling*, 11(2):121 –136.

Lambrechts, O., Herroelen, W., and Demeulemeester, E. (2008b). A tabu search procedure for developing robust predictive project schedules. *International Journal of Production Economics*, 111:493 –508.

Li, Z. and Ierapetritou, M. (2008). Process scheduling under uncertainty: Review and challenges. *Computers and Chemical Engineering*, 32:715 –727.

Liu, S. and Shih, K. (2009). Construction rescheduling based on a manufacturing rescheduling framework. *Automation in Construction*, 18(6):715 –723.

PMI, P. M. I. (2008). *A Guide to the Project Management Body of Knowledge:(PMBOK Guide).* Project Management Institute, fourth edition.

Poder, E., Beldiceanu, N., and Sanlaville, E. (2004). Computing a lower approximation of the compulsory part of a task with varying duration and varying resource consumption. *European Journal of Operational Research*, 153:239 –254.

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

Schwalbe, K. (2010). *Information Technology Project Management.* Course Technology, sixth edition.

Sommerville, I. (2011). *Software Engineering.* Project Management Institute, ninth edition.

Van de Vonder, S., Demeulemeester, E., and Herroelen, W. (2007). A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling*, 10(3):195 –207.

Van de Vonder, S., Demeulemeester, E., and Herroelen, W. (2008). Proactive heuristic procedures for robust project scheduling: An experimental analysis. *European Journal of Operational Research*, 189(3):723 –733.

Vieira, G. E., Herrmann, J. W., and Lin, E. (2003). Rescheduling manufacturing systems: A framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1):39 –62.

Vonder, S. V., Ballestn, F., Demeulemeester, E., and Herroelen, W. (2007). Heuristic procedures for reactive project scheduling. *Journal of Computers and Industrial Engineering*, 5:11 –28.

Wysocki, R. K. (2006). *Effective Software Project Management*. Wiley Desktop Editions.

Xiao, J., Osterweil, L. J., Wang, Q., and Li, M. (2010). Dynamic resource scheduling disruption-prone software development environments. *International Conference on Fundamental Approaches to Software Engineering (FASE 2010)*, (2):107 –122.

Xiong, J., Xing, L., and Chen, Y. (2012). Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *International Journal of Production Economics*.

Zhu, G., Bard, J. F., and Yu, G. (2005). Disruption management for resource-constrained project scheduling. *Journal of Operations Research Society*, 56:365 –381.

Zhu, G., Bard, J. F., and Yu, G. (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, 18(3):377 –390.