A FRAMEWORK FOR LOGISTICS MANAGEMENT IN THE CLINICAL TRIALS INDUSTRY: AN EXTENSION OF THE STATE-OF-THE ART USING CONTEMPORARY SOFTWARE ARCHITECTURES AND TOOLS

by

Emad Sadoon

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
December 2018

I dedicate this thesis to

my family for their unconditional love and constant support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Logistics management in the clinical trials industry is a very challenging undertaking due to multiple stakeholders, complex processes, diverse software applications, intensive white-collar jobs, and onerous quality standards. Current business practices are inefficient and difficult to automate technologies. This thesis reviews the theories and concepts of clinical trials logistics management. Then, the inefficiencies in current logistics management in the clinical trials industry are addressed by building a conceptual framework based on contemporary software tools and architectures such as web portals, software agents, business process management system, project cards, and resource cards, all interacting with specialized software applications such as accounting, inventory, and label design software. The framework supports data analysis at multiple levels of decision making. Furthermore, a project planning tool for facilitating and optimizing the operational planning in this industry is designed and presented. Different managerial scenarios with shared, dedicated, and work group resources are investigated using this tool.

LIST OF ABBREVIATIONS USED

BPMS	Business Process Management Systems
CRO	Contract Research Organization
CT	Control Team
eSIL	Electronic Sample Inventory Log
FDA	Food and Drug Administration
Lab	Laboratory
LP	Linear Programming
OA	Operation Associate
PM	Project Manager
QC	Quality Control
RCMPSP	Resource Constrained Multi-Project Scheduling Problem
SIL	Sample Inventory Log

ACKNOWLEDGEMENTS

I would like to express my gratitude to Dr. Uday Venkatadri for encouraging me and showing me the way throughout this research journey. Without his invaluable comments, continuous support, and leadership, this thesis could not have been completed.

I would like to extend my appreciation to Dr. Alireza Ghasemi for co-supervising and guiding me through my thesis by showing me a clear vision. His feedback and support greatly enriched this endeavour.

I would like to acknowledge Clinical Logistics Inc. for giving me the opportunity to do this research at their company and sharing their valuable knowledge and expertise.

Special thanks go to my dear friend and colleague Alan Wong for his unconditional support and for sharing his knowledge of clinical trials.

Thanks to Christian Noel Samuel for helping me with the technical and programming aspects of the thesis.

Thank you to Dr. Majid Taghavi and Dr. Claver Diallo for serving on my defense committee and providing insightful critical comments, which made this a better thesis, and to Dr. Floris Goerlandt for moderating my defense session.

Chapter 1: Introduction

Logistics management in the clinical trials industry is a very challenging undertaking. The reasons for this are varied: there are multiple stakeholders, the processes are complex, many different software applications are involved, the tasks are labour intensive, and the quality standards are stringent. Current business practices are inefficient due to the lack of structure in the flow of information for coordination and collaboration, the presence of disparate software applications, and difficulties in implementing automation technologies. On the other hand, the clinical trials industry is vital for society since its output results in new treatment opportunities for an increasingly larger number of afflictions. At the same time, from an economic point of view, it is a high-value industry employing a highly skilled workforce. The motivation for this thesis is to address the inefficiencies in current logistics management practices in the industry by reviewing the state-of-the-art and developing a framework for logistics management in the industry that uses contemporary tools and technologies.

A clinical trial is usually for a new drug that is developed by a pharmaceutical company or other sponsors such as a university research centre. Due to regulatory requirements, testing for the drug needs to be at arms-length and is done by an independent laboratory. Patients are enrolled at sites (for example, hospitals) participating in the drug trial. A Contract Research Organization (CRO) is usually involved in managing the logistics of the clinical trial. This thesis takes the perspective of a CRO. The CRO works with the pharmaceutical company on developing a testing plan that involves multiple sites with multiple patients. The testing plan includes administration of the drug to the patients and collecting samples of analytes such as blood, urine, tumors, etc. The CRO in consultation with the sponsor develops manuals for the testing plan, labels, and kits. A kit includes items such as collection tubes, gel packs, and air waybills for subsequent shipment of samples to laboratories. The CRO may also provide additional logistical services such as tracking shipments from the collection sites to the testing laboratories.

In such a setting, logistics management involves complex coordination of tasks. The CRO is in touch with the sponsor both during the initial design phase of the trial as well as during implementation. Sometimes, amendments are made to the collection plans such as the addition of new sites, new patients, new assessment types, and the frequency of testing or dosage. The CRO is also in constant touch with the sites as well as the laboratories. Based on the treatment plan, there are preset collection time points which have to be managed. For example, the supplies (unexpired labels and kits) required for each enrolled patient at a site should be available at the right time (the collection tubes may have expiry dates). The CRO has to ensure that the samples are sent to the laboratories within a window of integrity and track such shipments, resolving discrepancies in a timely manner.

The clinical trials industry has stringent quality requirements. Collection and shipment have to happen exactly as specified in the collection plan. Patient and analyte labels must be error free. Kits are expensive and have a limited shelf life. Therefore, the right number of kits need to be sent to each site at the right time. Shipments in this industry are global, requiring greater attention to shipment lead times. The shortage of kits may result in samples not being collected and analyzed, while overproduction or expiration of kits increases costs. In addition, the collected samples are valuable, expensive to collect, and non-renewable. They must be maintained carefully to be analyzed by the assay labs. Therefore, a large cross-functional team needs to work together to have the right supplies at the sites and to maintain the stability of samples from sites to the assay labs.

According to Taboada and Singh Sharad (2015), redrawing samples from a patient will cause dissatisfaction and may lead to a withdrawal of the patient from the study. Replacing a patient is also challenging due to narrower patient population sizes that recent studies are focusing on and can cost up to USD 25,000.

The literature on logistics management in clinical trials is limited; in fact, most of the literature is at a very conceptual level or focuses on distinct aspects of clinical trials such as patient recruitment. This paper contributes in this regard.

The obstacles to efficient practice are first examined. In fact, the main problems can be summarized as: unstructured flow of information for coordination and collaboration, inefficiency in operations due to highly labor-intensive tasks, the need to use disparate software applications for the domain areas (document management, communication, accounting, planning, tracking, etc.) and the challenges in systems automation.

We then propose an integrative software framework for logistics management in this context. The main components of this system are a business process management system (BPMS) that integrates other business applications, a collaborative tool using project and resource cards, an intelligent software agent architecture for automation, an analytics tool for data analysis and performance measurement, a web portal for better collaboration between human and system, and a security system that controls the privacy and security of the data across the entire framework. This software framework integrates all of the management components that automate many tasks, provide a faster response to operational issues, and lastly give all parties more traceability and visibility. In addition to the main framework, a resource and project planning tool facilitating and optimizing the operational planning in this industry is designed. The resource and project planning tool works with the main framework components that are BPMS, resource and project cards to produce a detailed schedule of tasks and identify resources needed to process the tasks. The resource and project planning tool contributes to the Resource Constrained Multi-Project Scheduling Problem (RCMPSP) by considering multiple resource scenarios, multiple due dates for processes, multiple resource groups, and resource familiarity factors that are applicable to the logistics management in clinical trials industry.

Two main linear programming (LP) models are created in this tool that are task planning LP and resource planning LP. Within each model, three resource scenarios with shared, dedicated, and work group resources are investigated and compared. The objective function in both models is to minimize the total tardiness of processes and projects. Task planning LP schedules tasks with a fixed number of resources, while resource planning LP finds the optimal number of resources to finish all the tasks without any tardiness.

The rest of the thesis is organized as follows: Chapter 2 is a literature review of the theories and methods used in logistics management in this and other industries. Chapter 3 details the complex operations involved in clinical trials. Chapter 4 analyzes the challenges in current practice. Chapter 5 details our conceptual software framework solution. Chapter 6 proposes several resource and project planning tools for clinical trial logistics management. Finally, Chapter 7 summarizes this work and lists areas for future research.

Chapter 2: Literature Review

The clinical trials industry is organized around four entities: the sponsor, the CRO, the site, and the lab. Since the goal of the paper is to look at challenges in logistics management and address these through an effective framework, we classified the literature into the following categories: 1) clinical trials industry requirements and challenges, 2) service industry and servitization concepts, 3) project management, 4) contemporary software tools and architectures, and 5) visualization.

The justification for this approach is as follows: the first category covers some of the papers directly relating to the industry. Service industry concepts such as servitization were deemed important because logistics management in clinical trials is a service. Project management tools and techniques are also relevant because each clinical trial protocol, which specifies how a clinical study should be conducted, could be considered a project. During interactions with an industry partner, we realized that integration and automation using contemporary technologies is very complex. Some of the tools pertinent to this domain are business process management systems (BPMS), semantic webs, intelligent agent technologies, and electronic data capture systems. Finally, visibility of the progress of a clinical trial (protocol) is very important since four actors are involved. Logistical problems need to be quickly resolved, and issues quickly addressed to mitigate future risks. Therefore, visualization and communication techniques and tools are also relevant.

The literature review is therefore organized around these themes (Figure 1).

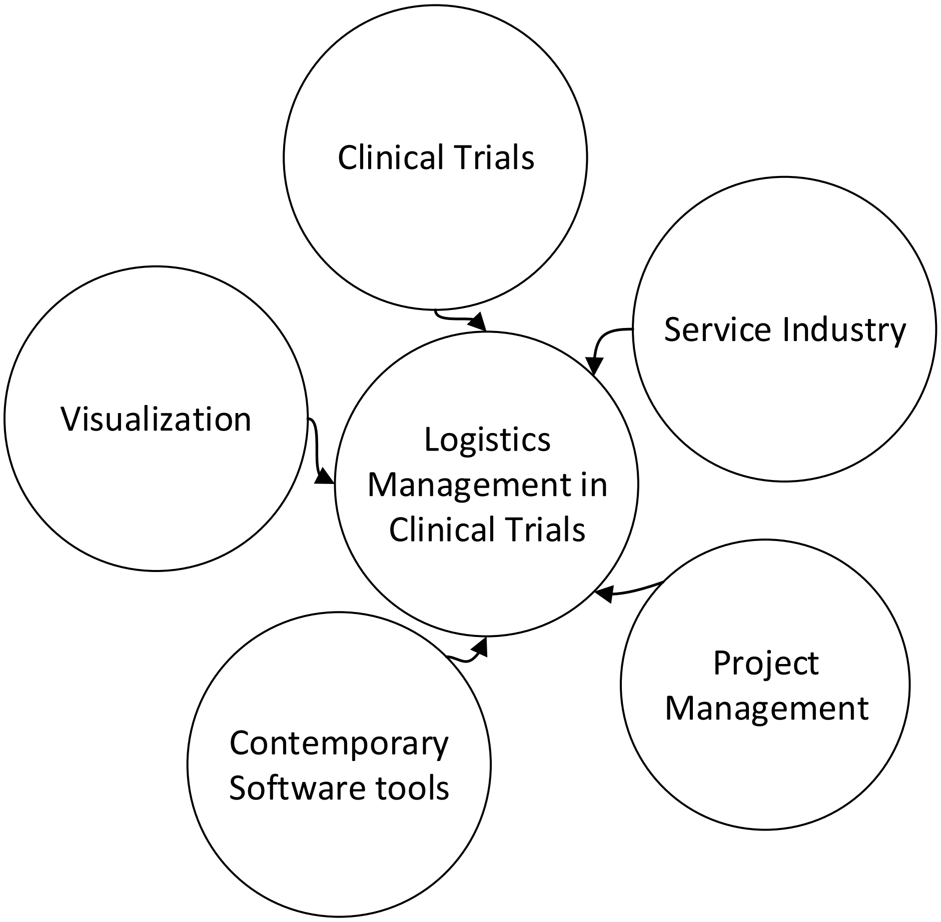

Clinical Trials
Service Industry
Visualization
Logistics Management in Clinical Trials
Project Management
Contemporary Software tools

Figure 1: The Synthesis of Concepts in the Literature for Clinical Trials

2.1 Clinical Trial Industry Background

Clinical trials are a set of experiments on individuals to test the safety and effectiveness of new drug treatments. Clinical trials are monitored by the Food and Drug Administration (FDA) authority (in the US) and if the results are safe and effective, the drug will be approved to come to the market. In Canada, Health Canada is the equivalent of the FDA; similarly, other countries have their own drug approval agencies.

U.S. FDA approves hundreds of new drugs every year. Most drugs are improved version of the existing drugs either with different dosage or generic formulation that will compete with the existing drugs in the market. The rest of these medications have unique and innovative formulations that have high impacts on clinical care. The U.S. FDA Center for Drug Evaluation and Research (2016) shows the list of novel treatments that were approved in the year 2015. According to this report, FDA approved 45 novel drugs in 2015, 16 of these drugs were First-in-

Class, meaning they used an innovative method to treat disease. 21 out of 45 drugs were for rare or orphan diseases that affect 200,000 people or less in the United States. On average 28 novel drugs per year were approved from 2006 to 2014.

According to PhRMA (2012), the number of developmental drug compounds increased from 2040 compounds in 2001 to 3240 in 2011. PhRMA (2016) reports that this number reached 7000 in 2016. They also report that the average time to develop a drug is 10 to 15 years and less than 12 % of drugs get approval after the clinical trial stage. The average cost of drug developments including the failure cost was $ 2.6 billion from 2000 to 2010.

Allison (2012) indicates that the median of the total number of procedures in each clinical trial was 105.9 in 2000-2003 and increased to 166.6 in 2008-2011. The median total work burden for each protocol was 55 in 2000-2003 and increased to 180 in 2004-2007. 54% of the sites and 78% of the subjects involved in clinical trials were located outside of US in the year 2008 (the United States. Department of Health and Human Services. Office of Inspector General, 2010).

Clinical trials are sponsored by pharmaceutical companies, governments, and universities. The individuals who choose to participate in the trials are either patients or healthy volunteers depending on the type of treatment and study. The participants are informed about the study, treatment, risks, and fill out consent forms before being enrolled in the trial.

A new trial is identified by a protocol, a document that comprises the plan and requirements to execute the study. According to the U.S. Food and Drug Administration (FDA) (2018), clinical trials (Table 1). For a drug to enter the market, the first main 3 phases are always necessary. The 4th phase, which is optional, is for post-market safety monitoring.

Table 1: Clinical Trials Phases (Retrieved from US FDA Website)

	Number of Participants	**Duration of Study**	**Purpose**
Phase 1	20 to 100	Several months	Safety and dosage
Phase 2	Several hundreds	Several months to 2 years	Efficacy and side effects
Phase 3	300 to 3000	1 to 4 years	Efficacy and monitoring of adverse reactions
Phase 4	Several thousands	Not available	Safety and efficacy (post-market safety monitoring)

2.2 Challenges and Requirements in Managing Clinical Trials

According to a study by DiMasi et al. (2003) on 68 new drugs developed by 10 pharmaceutical companies, the total capitalized cost is about 802 million dollars to develop a new drug.

Despite the importance of clinical trial management, knowledge about this subject is very limited (Farrell, Kenyon, & Shakur, 2010). The authors state that most of the failures in clinical trials are due to management problems rather than issues with study design. Therefore, project management approaches should be used to manage trial resources such as time, budget, and participants efficiently. The scope of a clinical trial is the most important component of the trial, which includes a number of engaged participants and the quality of the collected data. In commercial trials when the time to market is important, the time has a higher priority than budget (National Institute for Health Research [NHS], 2014). Communication and collaboration amongst team members, streamlined and flexible processes with computerized data capturing systems are key elements in successful trial management (Farrell et al., 2010).

Bryde and Joby (2007) state that the trend of outsourcing the management of clinical trials to clinical research organizations is increasing and more efficiency is needed in clinical trials management. According to Parrett, Sinclair, and Joby (2003), based on a survey of 28 professionals involved in outsourcing, the most common outsourced services are clinical monitoring (96% of respondents), data management (92% of respondents), and project management (81% of respondents) (as cited in Bryde and Joby, 2007, p. 364-365). This study further states that although outsourcing of clinical trial activities has been increasing, the level of satisfaction is low. The main reason for low satisfaction was concluded to be fixed priced based project management, principal-agent relationship.

According to Van Arnum (2011), as the number of treatments, procedures and foreign sites increases, the complexity of the clinical trial material (CTM) supply chain rises. The challenge is how to deliver CTM on-time to sites as the number of trials is increasing and the supply chain network is expanded globally. The main challenges of the global supply chain are different customs rules in each country, cost of transportation (especially for cold-chain CTM), and longer delivery times. One solution to these challenges is to have real-time and accurate information on deliveries and inventory at each site.

Zhang, Pawar, Shah, and Mehta (2013) evaluated outsourcing companies in the pharmaceutical industry based on the theory of dynamic capability. Dynamic capability is the ability to use resources to respond to market changes in a timely manner (Eisenhardt & Martin, 2000). The finding shows that pharmaceutical companies tend to outsource to companies that have higher dynamic capability rather than higher capacity (Zhang, Pawar, Shah, and Mehta, 2013).

2.3 Service Industry and Servitization

The service industry is an industry that provides non-tangible value, as opposed to tangible products. Sankaran and Agarwal (2012) and Lusch, Vargo, and Tanniru (2009) state that services should be created in collaboration with the client, as opposed to the traditional goods development industry such as mining, agriculture, manufacturing and construction. There are some aspects of service in these other industries such as being on time, managing resources using subcontractors, and providing warranties, but the main examples of service industries are banking, communications, wholesale and retail trade, professional services such as engineering and computer software development, medicine, etc. (Encyclopedia Britannica, 2016).

2.3.1 Service Dominant Logic

S-D (service dominant) logic implies co-creating value with clients by involving them throughout the entire supply chain. Service can be co-created by a collaboration of stakeholders in a value network (Lusch, Vargo, & Tanniru, 2009). Vargo and Lusch 2004 compare the (S-D) logic with goods dominant (G-D) logic. G-D logic is the traditional way of supply goods from manufactures through channels to customers without involving customers in the production processes. However, S-D logic alters marketing strategies from a goods-centered view to a service-centered view by encouraging partnership and relationship between customers and the providers of the value. In other words, this implies using competencies of one entity for the benefit of another entity.

Collaboration in S-D logic can be achieved by sharing information and knowledge through business processes using information technology and standardizing components. This usually, in turn, implies a modular architecture (Baldwin & Clark, 1997). However, a significant barrier to collaboration is the lack of trust amongst partners in a supply chain (Bratton, Mentzer, Foggin, Quinn, & Golicic, 2000).

2.3.2 Servitization

Many manufacturers have moved from only goods production to supplying supplemental services beside their products. This phenomenon is called servitization in much of the literature. Examples of servitization model can be seen in companies such as Xerox that provide document management through their photocopy machines, automotive companies that provide warranty and maintenance services to their customers, VLT (video lottery terminals) producers that install and maintain the VLT machines in casinos, etc. Rolls Royce provides fixed cost per hour maintenance services on its aircraft engines to its airline clients.

Frameworks for the transition to servitization have been discussed in the literature. Examples are: Opresnik and Taisch (2015) with the use of big data, Weeks (2013) for a specific framework in healthcare, and Weeks & Benade (2015) for a generic servitization framework for manufacturers. Jia, Tian, Yang, Sun, and Malik (2016) studied the effects of "Service extension, service skill improvement and product design modification" on business performance. They conclude that service extension and service skill improvements benefit business performance significantly compared to product design modifications.

Though not mentioned in the literature, the concept of servitization for clinical trials would apply to label production, kit assembly, and shipping box assembly for samples requiring refrigeration. Although these items could be thought of as products, they have to be delivered to sites or labs in a timely manner and need customization. These aspects make the industry service oriented.

2.4 Project Management

2.4.1 Project Management in Service Industry

Project management in the service industries including software development should have flexible goals and methods. Examples of new project management techniques such as agile and dynamic project management are used to design open source software such as the Linux operating system (Sankaran and Agarwal, 2012) and modular architectures (Lusch et al., 2009).

2.4.2 Project Management in Clinical Trials

Some of the challenges of managing clinical trials have been mentioned in section 2.2. This section presents some articles on how project management techniques can help to manage clinical trials.

Farrell et al. (2010) and Goodarzynejad and Babamahmoodi (2015) provide key guidelines for project management in clinical trials such as having a project manager, a project plan, communication and collaboration, and computerized data capture systems. However, neither paper proposed a technical framework for project management in clinical trials.

According to Bryde and Joby (2007), one effective approach for clinical trials is to use deliverables-based project management in which deliverables are managed instead of activities. The PRINCE2 method is an example of a deliverable-based project management methodology, which requires the creation of certain management deliverables such as plans, risk logs, and lessons learnt. Although deliverables (the outputs of the project) are more important for the pharmaceutical companies than the activities (the input of the project or resources), the budget is done based on activities currently. Contractors are paid for the activities that they perform or the time and resources that they use, not the deliverables.

PAREXEL (2015) identifies the requirements for an integrated clinical trial supply chain solution that consists of planning, storage and distribution, site and patient management, return and destruction, regulatory compliance, manufacturing, and packaging. However, a detailed technical framework is not developed. Also, the sample management component is not mentioned in their framework.

2.4.3 Similarities and Differences of Project Management in Different Industries

Cooke-Davies and Arzymanow (2003) compared project management methodologies in six different industries using qualitative criteria that are "project culture, business culture, understanding of PM, matrix strength, project authority, PM pervasiveness, project information, competent people, and flexible teams". The results show that industries that incorporated project management model earlier such as petrochemical and defense industries have higher scores for each criterion than the younger industries such as pharmaceutical R&D and financial services. The study concludes that even though the criteria used in this study are not statically proven, they can be used as qualitative characteristics of an advanced project management model.

Zwikael and Globerson (2007)'s surveys showed that project managers in the service sector excel in cost and procurement planning. However, they perform poorly in quality planning and management, which has a high impact on project success. Techniques such as cause and effect diagram, control charts, flowcharts, quality audits are recommended. Qualification of the PMs is also an indicator of the level of success. Developing PM procedures and training is suggested.

Visser and van der Merwe and Gerber (2016) studied the similarities and differences between project management in software development and R&D projects. The finding showed that there is a significant overlap in processes and principles, though success factors are significantly different. The most common processes are "Evaluation" and "Problem Definition". The most common success factors are "Client acceptance" and "Providing benefit". Some of the distinct success factors for the research project are "following plan, interpretation, minimal startup problems, meet technical goals, and negotiation". The distinct success factors for software development projects are "shared vision, risk management, project management, and adaptability".

A good project management tool should include five aspects: scientific management, humanistic management, risk management, quality management, change management, and project cost management (Tomas U. Ganiron Jr, 2015). Different methods for each aspect have been presented in this paper.

2.4.4 Planning and Scheduling Techniques in Project Management

Faghihi, Nejat, Reinschmidt, and Kang (2015) surveyed existing algorithms and techniques for solving project scheduling problem in the construction industry. Al Nasseri, Widen and Aulin (2016) prepared a taxonomy for planning and scheduling for construction project management and conclude that:

- The GC (Gantt chart) is a suitable tool for visualization of activity schedules, but no consideration is given to human resources and the trade-off between time and cost. GC has been combined with a network to handle uncertainties and control the connections between activities.

- The combination of CPM deterministic network and PERT probabilistic method in large construction projects is considered appropriate, especially when Monte Carlo simulation is used.

- Linear scheduling techniques such as LOB (line of balance) are more appropriate for projects with many repetitive tasks. However, alterations in the linear scheduling network might result in a non-linear path.

- Traditional methods are unsuitable for managing resource-constrained schedules, because of their limitations in switching among activities in terms of resource or buffer contentions and leveling.
- Traditional methods are insufficient for managing and allocating all resources at the detailed level of scheduling. This can lead to many idle buffers.
- Newer methods (advanced probabilistic algorithms such as CCPM (critical chain project management) and lean concept such as LPS (last planner system) have been integrated into the construction planning tools to supplement the previous methods.

Hartmann and Briskorn (2010) summarized and categorized existing resource-constrained project scheduling (RCPS) concepts. The popular models found to be multiple modes, generalized time lags, and net present value-based objectives. Specific examples from different industries have been discussed in this article, such as Kolisch & Meyer (2006) that applies RCPS in early stages of drug discovery in the pharmaceutical research industry.

Herroelen and Leus (2005) surveyed solution models for project scheduling under uncertainty. Their survey includes "reactive scheduling, stochastic project scheduling, fuzzy project scheduling, robust (proactive) scheduling and sensitivity analysis" (p.289). The potentials of each model have been described.

2.5 Contemporary Software Tools

2.5.1 Semantic Web (Unified System of Record)

"The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". (Hawke, Herman, Archer, & Prud'hommeaux, 2013) Semantic web was founded by the World Wide Web Consortium (W3C) and is the extension of the World Wide Web (REF). The World Wide Web is the web of documents, while the semantic web is the web of data. Semantic web integrates all the data from different sources using RDF (Resource Descriptive Framework) that can be easily read by machines. Semantic web can be seen as a larger and more robust tool than relational databases (SEMANTIC WEB, 2015). Columbus (2014) identifies five key elements to improve supply chain operations in aerospace, of which one is a unified system of record that can be achieved by semantic web technology. It is believed that the Semantic web can be a useful technology to integrate data from detached applications (such as spreadsheets and emails) in clinical trial logistics management. In what follows, the articles are synthesized in terms of their application and domain rather than detailed methodology, because more or less all of the articles use the semantic web methodology but with slightly different software and approaches.

Shankar, Martins, O'Connor, Parrish, and Das (2006) designed a high-level semantic web system for patient and specimen tracking in clinical trials. (Besana, Cuggia, Zekri, Bourde, & Burgun, 2010) use semantic web to recruit patients for the purpose of clinical trials. Dridi, Tissaoui and Sassi (2015) designed a semantic software to unify the health record information of patients to improve the health care system.

Mohammadi and Khalili (2008) propose a semantic web service-oriented model for project management. Furthermore, Niknam and Karshenas (2017) use a more detailed approach for integrating and sharing information in construction projects from heterogeneous information formats such as text, relational and XML.

Except Shankar et al. (2006), no other paper specifically applied semantic web to the logistics management of clinical trials. Semantic web can be applied as a robust query system for the entire duration of a clinical trial such as study setup, study management, sample management, and tracking.

2.5.2 Agent Technology in Project Management

Bădică, Budimac, Burkhard, and Ivanovic (2011) defines a software agent as an entity that acts autonomously on behalf of others. They mention that examples of software agents can be seen in marketing, assistance systems, search engines, chatter bots, and components of a larger software. The authors state that for the larger software, it is useful to distribute the tasks among the software agents and create a framework for them to cooperate, which in this case the system is called "multi-agent system". The agents work best in an open system such as the internet that has the characteristics of "continuous availability, extensibility, modularity, arms-length relationships, concurrency, asynchronous work, decentralized control, and inconsistent information" (Hewitt, 1985). Wooldridge and Jennings (1995) describe the main characteristics of software agents as autonomy, social ability, reactivity, and proactiveness. Bădică et al. (2011), further presents a survey of the programming languages and platforms for creating software agents. Some potential platform for building the software agents are ZEUS, JADE, JAFMAS, RETSINA, and Agent Building Shell.

Agent technology can be applied in any domain especially project management and logistics management. Software agents can be used as assistants to managers to manage projects or tasks more effectively and efficiently.

Hammouch, Medromi, and Sayouti (2015) conclude that project management guidelines such as PMBOK fail most of the time due to poor communication, lack of monitoring and non-performance measurement. They use agent technology to improve implementation of PMBOK and reduce failures.

Kiralp and Venkatadri (2010) designed a multi-agent platform for order promising and production planning in a multi-enterprise supply chain network. This model brings an integrated, proactive, and efficient approach to supply chain management.

Lim and Winsper (2012) design a framework that includes the Radio Frequency Identification (RFID) technology into a multi-agent system to maximize the efficiency of logistics management.

Kirn, Herrler, Heine, and Krempels (2003) designed the framework for a software agent system that integrates disparate business processes in a hospital setting. They showed an application of this software in a clinical trial. The focus of this framework is more in the hospital processes of a clinical trial rather than logistics management of supplies and collected specimens.

The overlap of software agents, supply chain and logistics management, hospital processes for clinical trials, and project management remains a gap.

2.5.3 Electronic Data Capture Systems

Benjamins et al. (2016) design an electronic upload and reporting system for image transfer from clinical sites to Imaging Core Laboratories (ICLs). In this system, images are uploaded into a secure web portal, are automatically quality checked under certain criteria, and if the uploaded image meets the verification criteria, then the image file will be transferred to the lab.

Other than Benjamins et al. (2016)'s work, the focus of other articles is more on the improvement of eCRF(electronic case report form) that is used to collect all of the information about a patient, the visits, medication dose, adverse events, and etc.

The literature is lacking in coverage of Sample Inventory Logs (SILs) or Sample Requisition Forms (SRFs) that are used to record the samples collected at the sites. Currently, SILs are filled out by hand on forms and are sent by emails to the CROs or labs. Hand-filled forms are prone to errors, and if forms are not corrected at the site, it takes time to catch and resolve the errors later. A similar approach to Benjamins et al. (2016) can be applied in this area.

2.5.4 Business Process Management Systems

Business Process Management Systems (BPMS) are information systems that orchestrate and control business processes (Pourmirza, Peters, Dijkman, & Grefen, 2017). Other sources such as Antonia Azzini et al. (2015) mention BPMS as a system with many flexible processes that consume real-time data from a different source. The effect of business process management systems in organizations is to streamline administrative tasks and interoperate with other applications.

Antonia Azzini et al. (2015) use semantic web approach to integrate heterogonous information to support BPMS. Baumgrac et al. (2016) designed an architecture to integrate environmental information such as unforeseen events captured by the Internet of things (IoT) with BPMS. A case study performed by Do Carmo and Albuquerque (2014) in a justice organization in Brazil showed that BPMS significantly improved project management.

Pourmirza et al. (2017) identified the common problems in most business process management systems as being in the areas of collaboration, interoperability, security, and dynamicity. Reijers, Vanderfeesten, and Van der Aalst (2016) demonstrated a longitudinal study on the effects of workflow management systems on the performance of organizations. According to this study, only half the companies successfully implement the workflow management systems in at least one of their processes. They found that the workflow management system supported only 28% of the overall processes. They further identified the failure and success factors for implementation of workflow management systems, which are development team should be separate from IT team to avoid disruption, processes must not be redesigned at the same time as implementing this technology, upper management should support the use of this technology.

Havur, Cabanillas, Mendling, and Polleres (2016) perform a comparison of papers on the resource allocation concept in three domains of operating room scheduling, project scheduling, business processes. They conclude that current BPMS disregard collaborative work and also assume constant availability of resources per period (not linking with the calendar). They propose a scheduling approach that considers resource dependencies and conflicts.

From the literature, we understand that BPMS is a good tool to streamline execution and monitoring of business activities. However, compatibility and integration with other applications is a requirement for operational effectiveness. From the review of the literature, it appears that BPMS are execution oriented. In clinical trials logistics management, while execution is important, resource planning and scheduling still remain a gap.

2.6 Visualization and Collaboration Tools

According to Al Nasseri et al. (2016), the GC (Gantt chart) is a suitable tool for visualization of activity schedules. Jaber (2016) studied the effect of 3D visualization of project tasks and resources compared to Gantt chart and spreadsheet. This study showed a 40% increase in accuracy and a 39% decrease in monitoring time. Otjacques, Noirhomme, and Feltz (2006) created a framework to visualize the interactions and cooperation among human and non-human resources in construction projects. Spencer (2014) proposed a visual tracking system for compound management delivery in laboratories, but it can be applied to any workflow system to reduce variability.

As a project gets bigger, more disparate teams need to collaborate remotely using online tools. Chasanidou, Elvesæter, & Berre (2016) performed an exploratory study on a group of users that used Upwave, an online collaboration tool, for their project management and interviewed them for their feedback. The motivation to use this tool was found to engage with other team members remotely, access all information about the project in one place, learn a new tool, and intrinsic interest in sharing information with teams. It was found that making checklists, templates, and other artifacts are best practices that can be supported by the collaboration tools. Also, visualization, reward system such as points, badges and leaderboards are found very effective in project management collaboration tools.

Dávideková and Hvorecký (2017) build a knowledge conversion model based on Information and communication technology (ICT) for virtual teams. He mentions that collaboration in a virtual team requires motivation, supportive leadership, and suitable technology. He states that there is no single ICT tool that can serve all teams and activities and ICT tools need to be customized. This framework is a cycle with four stages that are socialization of tacit information, externalization of tacit information into explicit data, the combination of explicit data, and internalization of explicit data into a new piece of tacit information or knowledge. Tacit knowledge is our personal experience and conception that is hard to digitalize. However, explicit knowledge is any knowledge that we can codify and digitalize. Examples of socialization are conversations about experiences or brainstorming and are usually done by live communication via skype or video conferences. Examples of externalizations are creating instructions, drawings, graphs, schemes, etc. Example of the combination is to use Big Data techniques to analyze and aggregate a large amount of data. An example of internalization is interpreting or visualizing the data and learning by doing.

Nowadays, cloud computing has helped teams collaborate from anywhere. Abedi (2016) designed an integrated collaborative tool for a precast supply chain. This article mentions that the precast project requires complex coordination of tasks to be delivered successfully. An architecture was developed that is based on Context-Aware Cloud Computing Building Information Modelling (CAABIM). BIM is a centralized database for gathering information from all stakeholders during the design and build of a construction project for better collaboration (Seitz, 2016). Context is defined as any information about the situation of an entity, which can be a place, person, or an object. Context-awareness system provides users with up to date information about the context or situation of the project.

In summary, the techniques reviewed especially Spencer (2014) and Chasanidou et al. (2016) can be applied in clinical trials logistics management. The Gantt chart or network chart is useful for delivering study setup deliverables, but do not work for sample management where the CRO has no control over time and is only responsible for tracking samples. Dávideková (2017) can be a future solution that can be built after a framework is created for clinical trial logistics management.

Chapter 3: Project Management in Clinical Trials

A new clinical trial is identified by a protocol, a document that consists of the plan and the requirements to execute the research procedure. A protocol is a unique endeavor and involves many components and resources to be executed; therefore, it is considered as a unique project. Project management in a clinical trial involves complex coordination activities (i.e. logistics management) to ensure the protocol is executed correctly and within a time-frame and budget.

A clinical trial project is divided into four steps: contract award, study setup, study management, and closing (Figure 2).

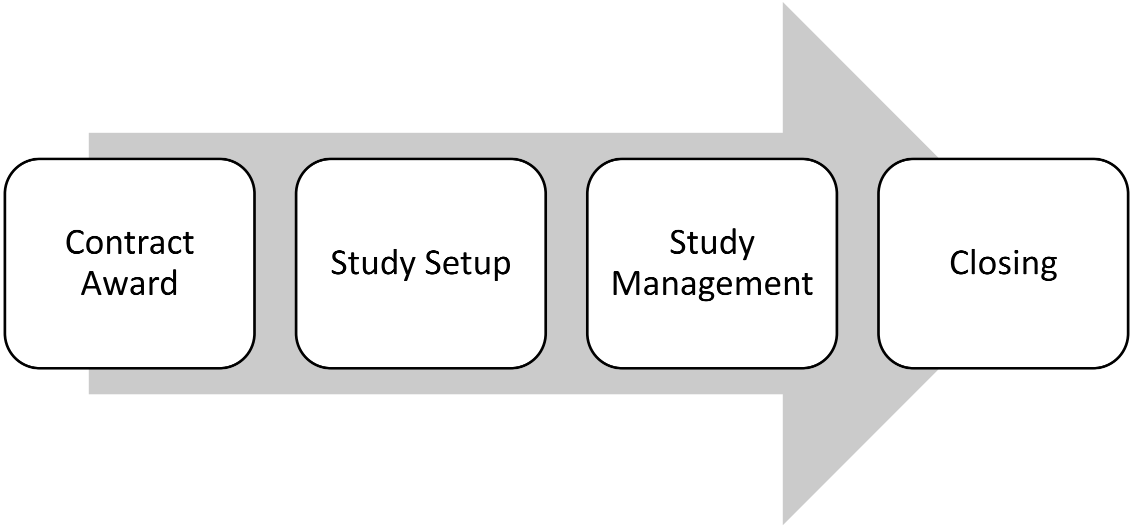


Figure 2: Components of a Clinical Trial

3.1 Contract Award

Sponsors of clinical trials outsource the management of these trials to CROs (Contract Research Organizations). The CROs submit proposals to the sponsors and the sponsors award the contract to one of the contractors. Upon approval, the CRO is responsible for providing the assigned services in the contract. The main services that are usually offered by the CROs can be categorized as study setup, study management, and closing, which will be discussed in the following.

3.2 Study Setup

In the study setup phase, the CRO designs the required supplies and coordinates the activities with the site and lab staff before starting the clinical trial. The CRO reviews the protocol and meets with the sponsor to design the required supplies (assessment kits, labels, laboratory manual, and site initiation visit presentation). Sample management services such as sample collection, storage, and shipping coordination are discussed and finalized.

A budget is created with estimates of the products and services needed to support the logistics management of the clinical trial. This includes:

- The number of patients.
- Duration of the study.
- The number of kits or other supplies.
- The frequency of shipments for each assessment.
- The number of shipments to be tracked.
- Contact names and addresses of the sites and labs.
- The number of SILs (Sample Inventory Logs) that need to be checked and verified.
- Variable and on-going fees.

Once the budget is finalized with the sponsor, the products and services are designed and customized for that specific trial. Labels are the main products used in the logistics management of clinical trials. Labels are designed in a specific label printing software. Lab manuals are designed to help the site staff handle the collection, storage, and shipment of sample shipments. Kits include labels and collection tubes, and shipping boxes to facilitate the process of transferring samples from the site to the assay lab.

3.3 Study Management

In the study management phase, the logistics provider continues to support the clinical trial by offering the agreed services. Usually, study management is a sequence of actions that starts when a new participant is entered into a clinical trial as shown in the diagram below. Participants or patients are the drivers of other management components of clinical trials. As long as participants remain enrolled in the study, the logistics management components to support the study are active. The main components of study management are: budget management, patient recruitment and management, inventory management, production and assembly, data management, compliance management. Each component will be explained in the following.

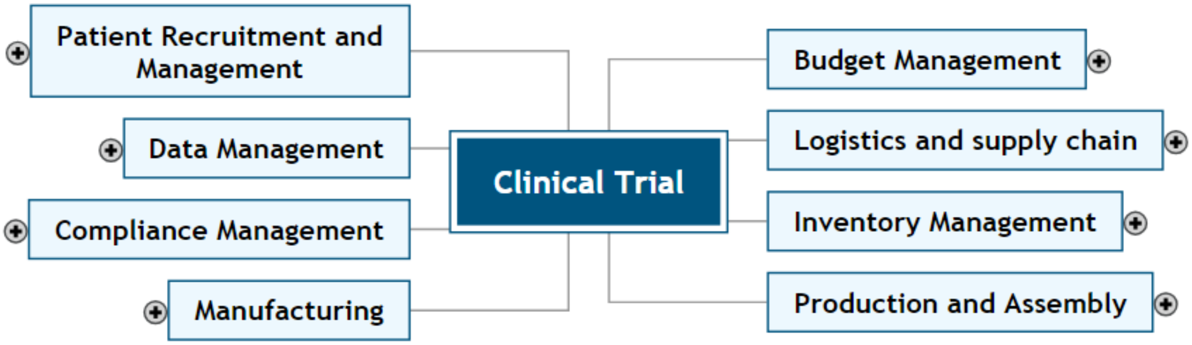


Figure 3: Components of the Study Management Phase

Figure 3 shows the components of study management, which includes:

- Budget management: Includes managing the allocated money to each activity in the protocol. Consider the effects of change orders and amendments costs on the budget.
- Participant recruitment and management: Includes recruiting the eligible participants to be entered into the clinical trial and manage their health and safety.
- Logistics and supply chain: Includes coordination of shipping supplies and treatments from the manufacturer to the sites and collected samples from the sites to the assay laboratories.
- Inventory management: Includes maintaining the appropriate level of supplies and treatments at each site and monitoring the expiration of inventories with shelf life.
- Production and Assembly: Includes designing and producing the desired supplies (sample collection kits) and treatments for clinical trials.
- Data management: Includes tracking the logistical data such as data about participants, supplies, samples, sample shipments. There are clinical data such patients' conditions after and before administration of treatments and laboratory results that are intended to the sponsors and researchers.
- Compliance management: Includes considering and maintaining regulatory compliance practices while conducting clinical research. Examples of regulators are FDA (US food and drug administration), Institutional Review Board (IRB), and Ethical Review Board (ERB) or Research Ethics Board (REB) regulations. Any major changes to the protocols such changes to the drug compounds or expanding the sample sizes must be approved by the regulatory bodies.

3.4 Closing

At this stage, the CRO ensures that all the requirements of the clinical trial have been properly addressed as planned in the contract.

Chapter 4: Challenges and Requirements of the Clinical Trials Logistics Management

In addition to the challenges and requirements of the clinical trials logistics management that have been mentioned in the Literature Review Chapter, this chapter also identifies more challenges that comes from observing the operations of an industry partner.

4.1 Protocol Amendments

Protocol amendments are very common in clinical trials, but few studies have recognized the issue. Some examples of protocol amendments are as follows:

- Changes to the study design and objectives of studies.
- Changes to the scope or sample size (number of participants).
- Changes to the duration or extension of studies.
- Changes to the budget.
- Changes to the types and frequency of assessments.
- Changes to the supplies, kits, and labeling.
- Changes to the treatments or drug compounds.
- Expanding the investigational sites.

According to a study by Getz (2011), about 60% of protocols had one or more amendments, and each amendment had an average of 6.9 changes. The most changes were identified to be changes to eligibility criteria of the participants, number and types of safety assessments, and revisions to general information such as protocol title and information about staff. About 40% of all amendments happened before dosing the first participant. The main causes of amendments were the availability of new safety information, requests from regulatory agencies, changes in the strategy, study design flaws, problems with recruiting patients. An amendment to a protocol costs $453,932 on average. The major costs are related to the increase in investigational sites and contract change orders with CROs and other third-party providers. According to this study simpler protocol design and informed decision making will reduce the amendments.

According to Chow (2011), every trial has nearly 3 to 5 protocol amendments after the trial commencement. Protocol amendments, especially change in eligibility criteria, change the target participant’s population to statistically test the differential effects of treatment on the old and new population within a desired confidence interval or statistical power.

Change to the protocols is inevitable and costly. Therefore, clinical trials require flexible processes and teams to manage the changes efficiently and effectively.

4.2 Demand for Participants

It is possible to estimate the required number of participants at every phase or stage of clinical trials to find out statistically meaningful results. For example, Chow (2011) presents some sample calculation modelling for clinical trials. However, it is difficult to find and recruit the desired level of participants especially if the clinical trial requires people with specific criteria. There are ways discussed in the literature to improve the recruitment process such as using integrated information systems (Besana et al., 2010) and marketing strategies (Francis et al., 2007). However, the level of uncertainty is still high for some studies. Thus, resource planning becomes difficult when the number of participants varies.

4.3 Data Management

There are two general categories of data in this industry, the clinical data that are intended for medical staff and researchers and the logistical data that are intended for project managers to ensure clinical trials are running as planned. This research focuses on logistical data, which most CRO’s are responsible for.

Figure 4 shows the flow of logistical data in a clinical trial. The main data elements are:

1. Enrollment Log: includes the list of enrolled patients in a site that are ready to be tested.
2. Supplies: are kits that are produced by the CRO for the patients at sites. The kits usually include supplies for collecting samples from the patients.
3. Schedule Shipments: The CRO schedules shipments to ship collected samples from the site to the central laboratory for analysis. Each type of samples has different schedules depending on the shelf life of samples.
4. Sample Shipments: The site ships sample shipments to the central laboratory for analysis.
5. Sample Inventory Log (SIL): is the manifest of samples being shipped in a shipment. SIL is used to check what samples have been shipped from the sites.

6. Notifications: are in the form of emails that are sent to inform the laboratory of incoming samples shipments. The laboratory needs to plan for resources to store or analyze the samples.
7. Electronic SIL (eSIL): is the electronic format of the SIL. The site fills out the paper SIL and the CRO transcribe the manual data into the electronic data. In the meantime, CRO checks the SIL to ensure data have been collected correctly and will send a copy to the laboratory as well.

In the study execution phase, patients are administered with a treatment and samples are collected for analysis. The main issue in this area is that most of SILs are hand-written. The reasons for hand-writing on papers are staff accountability and traceability of the actions involved in data collection and the testing plan. Any data noted on the SILs will be signed and dated according to the SOPs (Standard Operating Procedures) in clinical trials. The site staff records the information about patients, samples, and time of collection on the SILs. A copy of the SILs are shipped with the sample shipment to the labs, and another copy is usually faxed or emailed to the CRO. The CRO checks the SIL papers manually to ensure the sample collection is conducted correctly and identify and resolve discrepancies. When the number of sample increases, SIL packets becomes voluminous. Manually checking and filling out the paperwork is very time-consuming.

Most sponsors of clinical trials have their own database to store both the clinical and logistical data. CROs usually responsible to transcribe the hand-written data into electronic data and upload the electronic data into the sponsor's database.

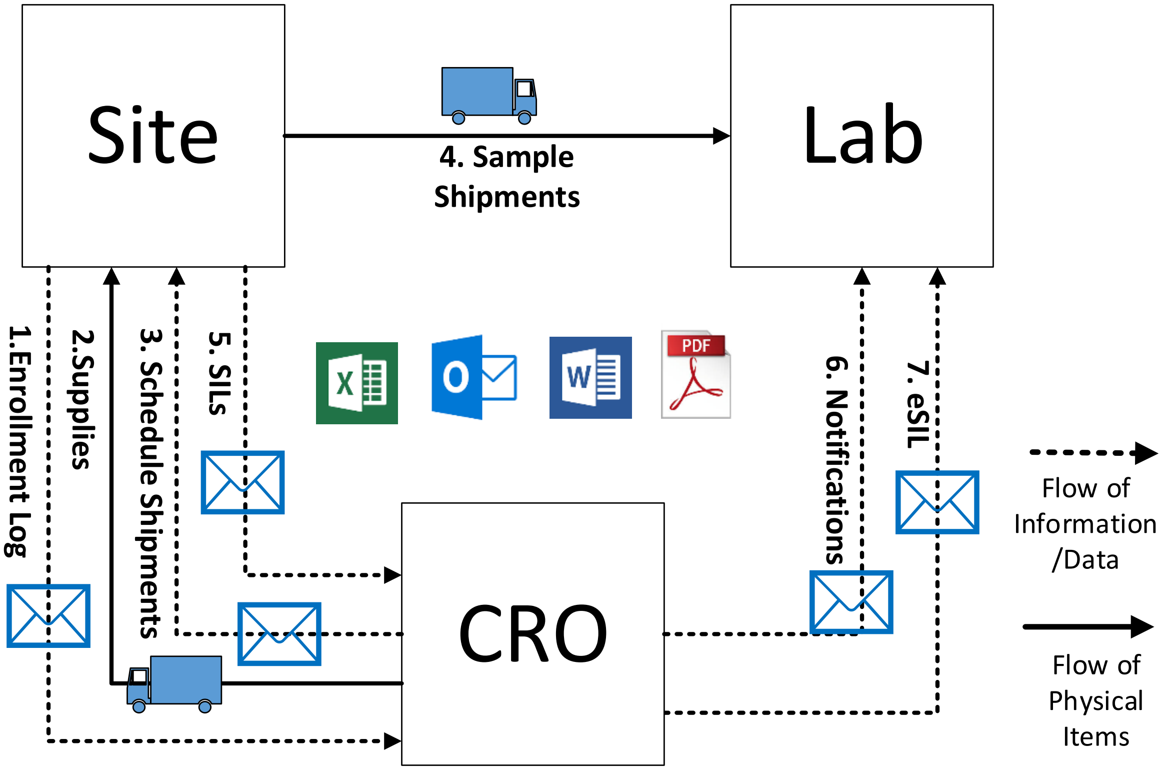


Figure 4: Existing Practice for the Study and Sample Management

4.4 Diverse Software Applications and Data Sources

Lack of integration among computer applications or sources of data can cause duplicate entries of data and consequently introduce inefficiency and errors in the system. For example, in the study setup phase where products such as labels and manuals are designed, the CRO staff copy and paste the data from the source documents such as protocol, budget or email into the label or manual design software. In the copy and paste process from the source documents, there is a high potential for human error. In other areas such as kit production, this issue is very tangible. For example, patient enrollment logs, inventory systems, production and assembly systems that are not connected create extra effort and time to transcribe data from one platform to another. Another example is sample management where multiple sources of data need to be monitored to ensure the sample management process is being performed correctly.

4.5 Challenges in Sample Management

CROs use the patient enrollment logs to schedule the sample shipments. Again, this process requires many data entries into different trackers, continual updating the trackers, problem-solving and critical thinking. And of course, as it is done manually by humans, it is very time consuming and there is a potential for human errors.

CROs input the SILs information into their tracking systems (usually Excel spreadsheets) to track the samples shipped. They must identify that correct samples have been collected, correct labels have been used, and correct information has been recorded. After data cleaning, the electronic version of SILs is created for the assay lab.

Sample tracking becomes challenging when the number of studies, sites, patients, sample types and time points increase. An estimate of the number of data points for sample tracking can be calculated by the following formula:

$$\sum_{n} P_n \sum_{m} T_m \, A_m$$

Where: n = Site number, m = Sample type, P_n = Number of patients at site n, T_m = Number of time points for sample type m, and A_m = Number of aliquots for sample type m, i.e., back up samples.

In the meantime, the CRO notifies the assay labs of the upcoming shipments. Then, it tracks the shipments to ensure the shipments reach the labs. In the end, CROs check the samples shipped against samples received at the laboratory to identify and resolve any discrepancies. All the notifications and confirmations of receipt are sent via emails. There are several issues with shipment tracking that are:

- Some labs do not confirm the receipt of samples via email.
- Confirmations are overlooked in the CRO’s email system.
- The site may use incorrect shipping boxes that leads to shipments being shipped to the wrong place.

The process of tracking is done manually by looking at the courier website and notifications are sent manually via email.

4.6 Risks in Clinical Trials

Some of the risks in clinical trials are missing collected blood samples, losing sample integrity due to storage at an incorrect temperature, losing a sample collection schedule due to not having supplies at the site, shipping the samples to the wrong laboratory, missing a patient due to the complex design of procedures, not being able to attract the targeted participants in the desired duration. Risks and errors are inevitable. Therefore, proactive actions are needed in a timely manner to reduce the impacts of errors as much as possible. For example, if a sample is being shipped from a site to the wrong laboratory, it is important to react quickly and contact the courier company to re-route the package. Another example is that a site may recruit a patient and realize that do not have enough kits, in this case, the supplier of the kits must react fast to provide the site with kits. The goal of the CROs and management team is to reduce the reactions by being more proactive instead. Proactive planning and having back-up plan to mitigate the risks are essential for clinical trials. Learning from mistakes and communications are key factors in proactive planning.

4.7 Need for Cross-Organizational Collaboration

Assuming that trust and relationship already exist among the vendors and sponsor of a clinical trial, the next step is the cooperation among them to achieve the objective of the clinical trials. Therefore, an optimum way of collaboration and coordination is essential to meet the objective within defined and acceptable constraints, which, in other words, implies optimum project management.

The sponsors of the pharmaceutical research studies usually outsource the management and conduct of studies to CROs. There are different types of CROs that each are specialized in certain services or products. The main CRO might outsource some of the works to other vendors or CROs. Figure 5 shows an example of the vendors and teams that are involved in a clinical trial.

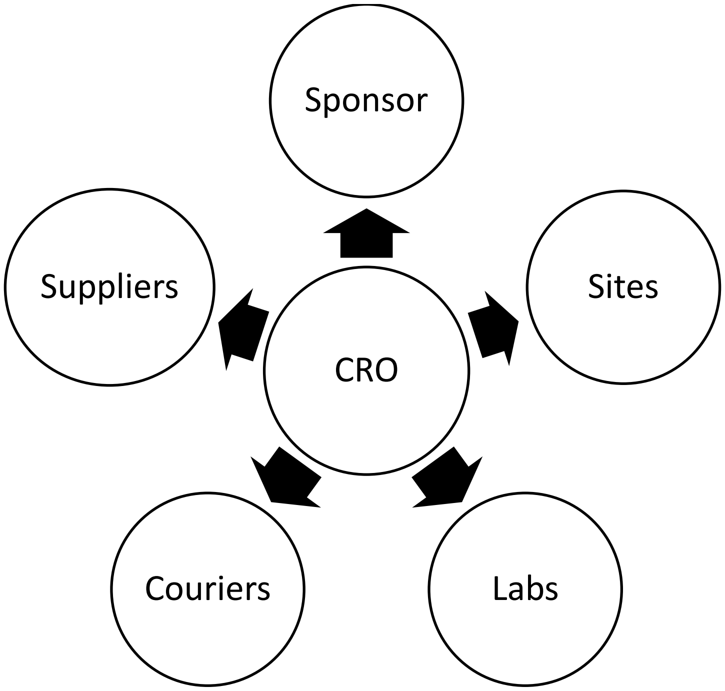


Figure 5: Main Stakeholders in a Clinical Trial

Each player or vendor in the clinical trials has its own management components such as project management, supply chain, inventory management, and data management. Each vendor might also be involved in more than one project (clinical trial). Some of the entities such as laboratories, CROs, couriers, and suppliers are independent of a clinical trial and can handle multiple clinical trials for different sponsors. Each sponsor can support one or more clinical trials. The clinical sites are usually exclusive to a clinical trial.

4.8 Collaboration Issues via Email

Almost all the collaboration among the CROs, assay labs, sponsor and sites are conducted via email. As the number of organizations and team members increases, the number of emails increases as well. Documents are shared via emails. More information might accompany the emails including the draft documents. Emails sometimes include redundant information that makes the process even less efficient. Processing all workflows via email is difficult and error-prone.

Chapter 5: Development of a Streamlined and Integrative Conceptual Framework

A conceptual framework is now proposed to improve project management in clinical trials. The high-level objectives that this framework is trying to address have been retrieved from Columbus (2014) that includes electronic traceability, manufacturing visibility, compliance with quality management standards, integration of workflows for greater visibility, and a unified system of record.

5.1 High-Level Architecture of the Framework

Figure 6 shows the high-level architecture of the framework. The lower left box shows the design phase of a clinical trial and the upper right box shows the implementation phase of a clinical trial.

In the design phase, the CRO works with the sponsor and sometimes the analytical laboratories to design the testing plan and prepare laboratory manuals for the site to follow for each type of samples that need to be collected for each assessment (Pharmacodynamics (PK), pharmacodynamics (PD), etc.). In the meantime, products such as labels, shipping boxes, and kits need to be custom designed for every clinical trial. In addition to products, certain services need to be discussed and custom designed if needed. Example of services is notifications to the sites and labs, schedule sample shipment, monitoring collected and uncollected samples and sample shipment tracking, kit inventory management, etc.

The implementation phase starts when a new site is opened to recruit patients. Then the required product is shipped to the sites. As soon as patients enter the sites, the services start to be activated. In this framework, a central database system tracks data in real time. Each stakeholder (site, lab, CRO, courier company) have a designated web portal to view relevant information and to input necessary data or communicate with other team members. Examples of this data are patient enrollment log, shipment schedules, sample inventory log, notifications, etc.

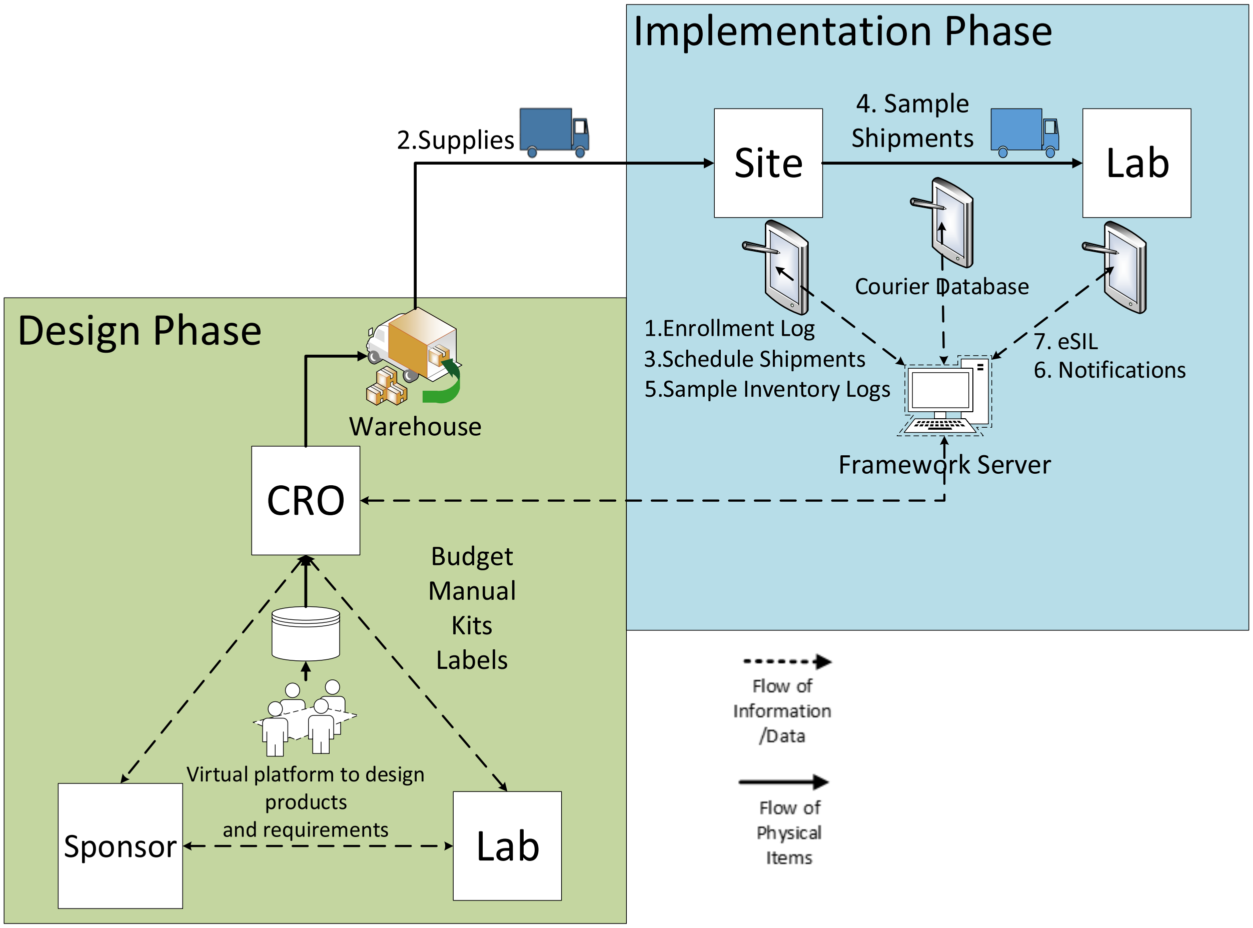


Figure 6: External Interactions of the Framework with Other Organizations

Figure 7 shows the detailed architecture (core) of the framework. This framework from top to bottom consists of:

- Web portal and its interactions with human users.
- Auto-monitoring tools implemented using intelligent software agents.
- Clinical trials BPMS to execute and orchestrate the tasks amongst resources and stakeholders.
- Project cards layer to integrate various project activities in a decentralized fashion.
- External applications (for specialized tasks).
- Document repository.
- Data repository.

- Analytics module to capture key performance indicators through the layers.

Each of these components will be explained in the following sections.

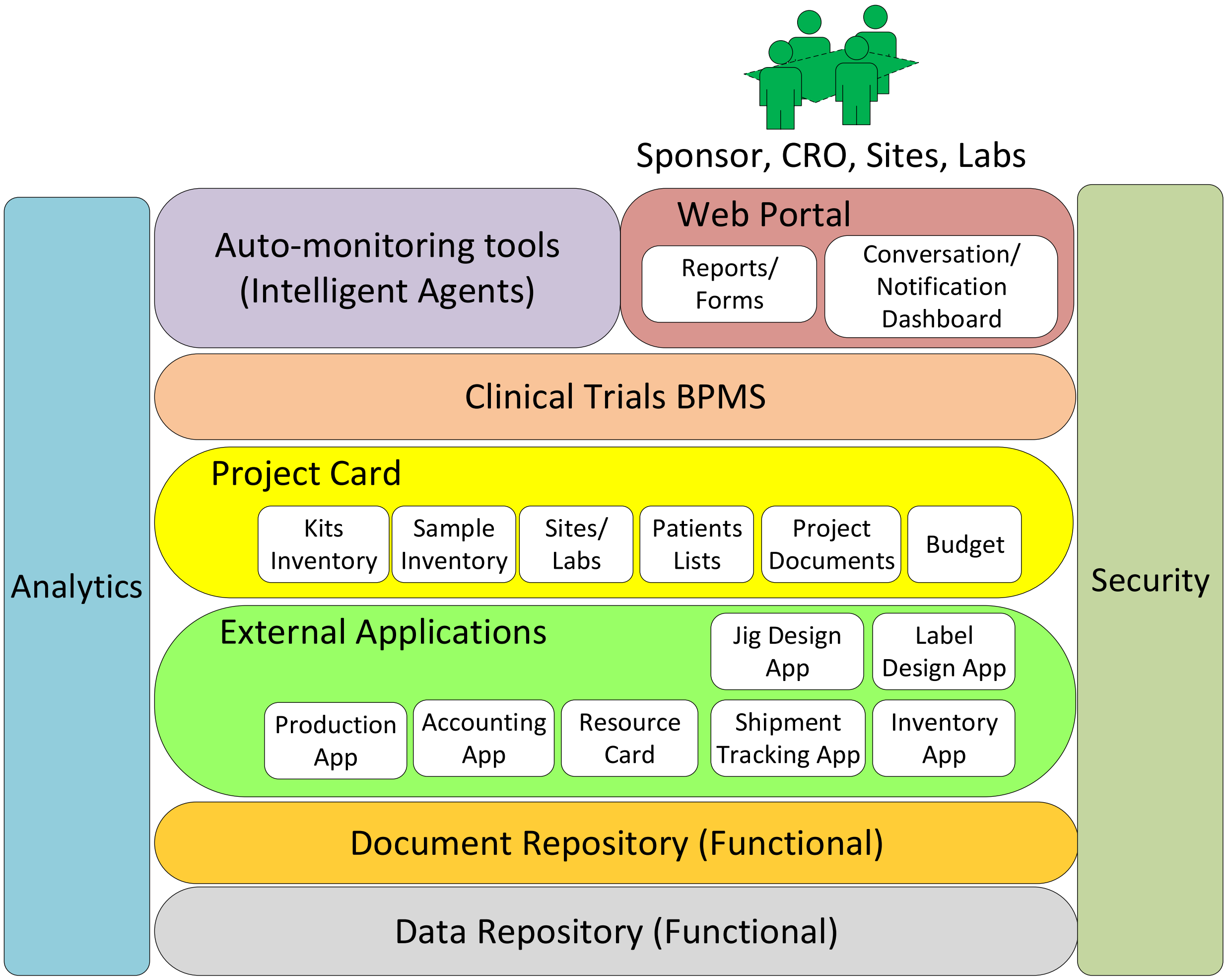


Figure 7: High-Level Architecture of the Framework

5.2 Web Portal

A web portal is incorporated into this framework to improve the communication and collaboration among the stakeholder in the clinical trials logistics management. Since clinical trials have become more global, the web portal should be available on the cloud to increase accessibility. The web portal will have forms to input the data, reports showing the data in real time to the teams, file system to store the documents and share them with other teams if needed, and dashboard for notifications and conversations with other team members.

5.2.1 Forms

In clinical trial logistics management, each process contains forms that need to be completed by the stakeholders involved that are usually CRO, Sponsor, Site, Courier, and Lab. These forms can be available online to be filled out electronically and distributed to each team member via the internet. The advantages of using online forms are that the records are saved electronically, retrieved at anytime and anywhere via the internet, and delivered faster from one team to another. In addition, the process will be more transparent and lastly less paper is used. CRO shall initiate different processes for each component of a clinical trial project using online forms. The sponsor shall sign and approve documents such as budget, change orders, and other documents using online forms. The site staff shall use the online forms to send order requests for kits and supplies from the suppliers or CRO. They shall fill out sample inventory lists using online forms and confirm the date that they are going to send the sample shipments to the laboratory. The courier shall confirm the time and date for pick-up and delivery of kits and sample shipments using online forms. The lab shall confirm the receipt of sample shipments delivery and upload the analysis reports online.

5.2.2 Reports

Since clinical trials involve many processes and organizations, detailed and customized reports shall be available for each team to improve planning and coordination of tasks. The CRO will have access to reports such as patient enrollment logs, sample inventory logs, supplies inventory for each site, sample shipments, and budgeted costs versus actual expenses. The sponsor shall have access to reports such as patient inventory, sample inventory, and budgeted costs versus actual expenses. The site shall see reports such as the inventory of kits and samples, a list of active and non-active patients and their progress through, the progress of kits orders, and status of sample shipments. The courier shall have access to pick up schedules and delivery deadlines for samples and kits. The lab shall see the collected samples inventory and the upcoming sample shipments to it to plan resources for storage and analysis of samples.

5.2.3 File System

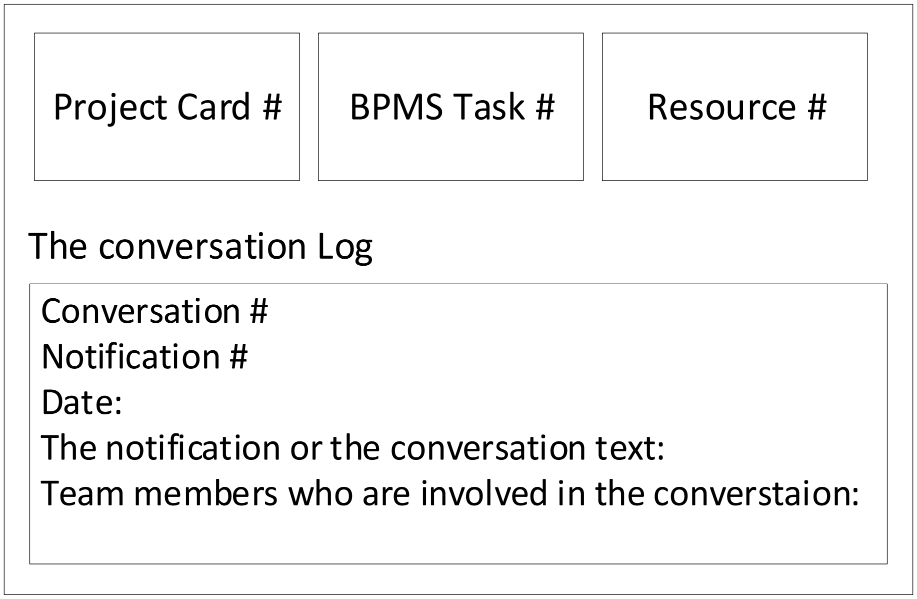
The CRO and sponsor shall access the documents associated with each project such as protocol, contract, budget, and amendments. The site and lab shall access to the most up to date laboratory manual and other necessary instructions. The courier shall access to the manifest of the shipments and customs documents needed to ship samples from one country to the labs in another country. All these documents are saved in the project card that will be explained in section 5.5.

5.2.4 Conversation and Notification Dashboard

The CRO shall communicate with other team members across the supply chain. The site shall use the dashboard to communicate with the CRO regarding sample discrepancies and shipping schedules and methods. The dashboard will send reminders to the site when to collect and ship samples to the labs. The courier shall notify pick-up, delivery, or delays to other teams via dashboard platform. Teams might communicate with the courier to mitigate or resolve delays using dashboards. The lab will receive delivery notifications of upcoming samples shipments and notify receipt of sample shipments via the dashboard. The discrepancies will be communicated and resolved via the dashboard. The conversation and notification dashboard will be illustrated more in detail in the following.

The objective of this system is to have a more structured collaborative platform compared to email. Nowadays, companies collaborate using email systems where they receive thousands of unstructured emails that they have to go through and sort. Sometimes, people miss important emails because of the large number of emails in their inbox. The emails usually include information about a document, pictures of the documents or an actual copy of those documents, which in these cases users should open that document in another platform to check or cross-reference data.

This system links the collaboration tool such as email system to the project's file system, where all of the data and documents for a project are located. Each conversation or notification is referenced to a Project Card number, where all the project files and data are located. The people in that conversation can retrieve the file/form/data that the conversation is referencing to from this platform by clicking on the project card number. This system can also refer to a task in a process where the work is being done by clicking on the BPMS Tasks number. Additionally, it can refer to the employee or the teams that are assigned on that the tasks by clicking on the Resource number. Figure 8 shows the structure of this platform.



Project Card #

BPMS Task #

Resource #

The conversation Log

Conversation #
Notification #
Date:
The notification or the conversation text:
Team members who are involved in the converstaion:

Figure 8: Structure of the Conversation and Notification Platform

There are two types of conversations:

1) System generated notifications according to the predefined rules.
2) Manual enquiry or conversation from team members.

Additional features can be added to this platform. Examples of these features are listed below.

5.2.4.1 Auto Generation of Subject Lines

Any notification or enquiry will be indexed automatically according to the project/process/ specific document that is about. This reduces the burden of defining subject lines and increases the consistencies across the platform.

5.2.4.2 Logging the Conversations and Searching Through Them

The notifications/conversations will be logged and can be searched at any time. Searching will be faster since every conversation has already been indexed.

5.2.4.3 Auto-Generation of Contact List

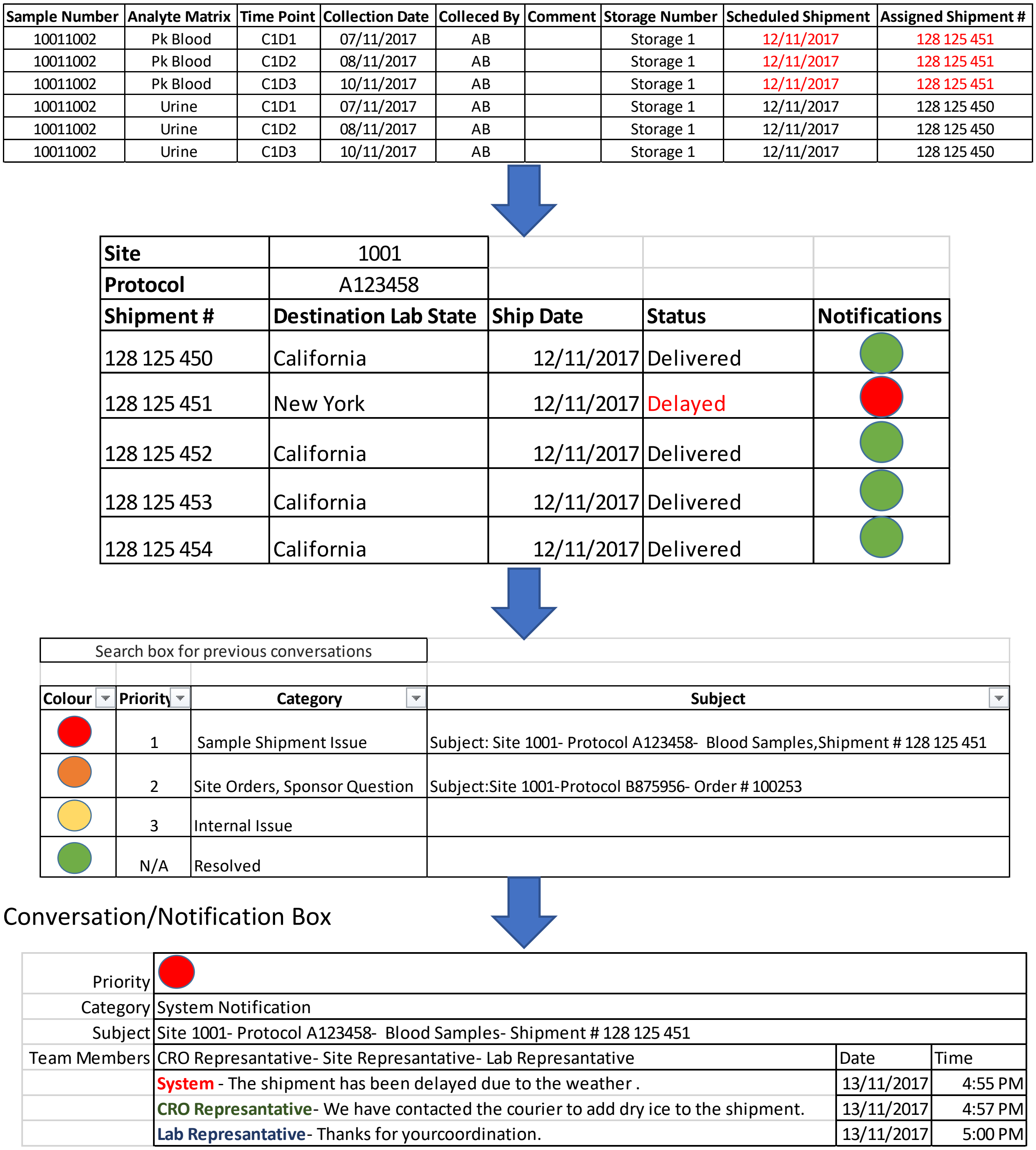
Enquiries and notifications will be sent to the responsible people that have already been identified for that process/document. For manual enquiry, the users can also choose/add/edit the contact list if needed depending on the level of autonomy of the users.

5.2.4.4 Prioritization Dashboard

According to the preprogrammed prioritization rule, notification and enquiries can be prioritized in the respective user's dashboard. In this case, users can see the prioritized enquiries/notifications on their dashboard. This will reduce the burden of reading every email and notification and manual prioritization. The platform will still show the unanswered/unread enquiries until they are read or responded. Different rules can be programmed here for how to approach different enquiries. An example of this feature is shown in Figure 9.

5.2.4.5 Accessing the Files/Tables/Processes from the Platform

Users can retrieve the files and see the workflow processes that the enquiry is about from this platform, by clicking on the subject line of the enquiry. The files can be retrieved by linking to the file system, and the processes can be seen by linking into the BPMS.



Sample Number	Analyte Matrix	Time Point	Collection Date	Colleced By	Comment	Storage Number	Scheduled Shipment	Assigned Shipment #
10011002	Pk Blood	C1D1	07/11/2017	AB		Storage 1	12/11/2017	128 125 451
10011002	Pk Blood	C1D2	08/11/2017	AB		Storage 1	12/11/2017	128 125 451
10011002	Pk Blood	C1D3	10/11/2017	AB		Storage 1	12/11/2017	128 125 451
10011002	Urine	C1D1	07/11/2017	AB		Storage 1	12/11/2017	128 125 450
10011002	Urine	C1D2	08/11/2017	AB		Storage 1	12/11/2017	128 125 450
10011002	Urine	C1D3	10/11/2017	AB		Storage 1	12/11/2017	128 125 450

Site	1001			
Protocol	A123458			
Shipment #	**Destination Lab State**	**Ship Date**	**Status**	**Notifications**
128 125 450	California	12/11/2017	Delivered	🟢
128 125 451	New York	12/11/2017	Delayed	🔴
128 125 452	California	12/11/2017	Delivered	🟢
128 125 453	California	12/11/2017	Delivered	🟢
128 125 454	California	12/11/2017	Delivered	🟢

Search box for previous conversations

Colour	Priority	Category	Subject
🔴	1	Sample Shipment Issue	Subject: Site 1001- Protocol A123458- Blood Samples,Shipment # 128 125 451
🟠	2	Site Orders, Sponsor Question	Subject:Site 1001-Protocol B875956- Order # 100253
🟡	3	Internal Issue	
🟢	N/A	Resolved	

Conversation/Notification Box

Priority	🔴		
Category	System Notification		
Subject	Site 1001- Protocol A123458- Blood Samples- Shipment # 128 125 451		
Team Members	CRO Represantative- Site Represantative- Lab Represantative	Date	Time
	System - The shipment has been delayed due to the weather .	13/11/2017	4:55 PM
	CRO Represantative- We have contacted the courier to add dry ice to the shipment.	13/11/2017	4:57 PM
	Lab Represantative- Thanks for yourcoordination.	13/11/2017	5:00 PM

Figure 9: An Example for the Auto-Notification in the Proposed Conversation and Notification Tool

5.3 Intelligent Agents

Intelligent agents have been introduced in sub-section 2.4.2 (in the Literature Review Chapter). A few examples have been discussed and potential agent software platforms were introduced as well.

Figure 10 shows the proposed architecture of the multi intelligent agent systems for clinical trial logistics management. The objective of designing intelligent software agents for clinical trial logistics management is to increase automation in this industry. The main automation is in data collection, data monitoring, and decision making. This architecture consists of the following entities:

- Intelligent agents for each stakeholder to act on behalf of the company. Each intelligent agent will be explained in tables 2-7.
- A database for each stakeholder to store the data locally.
- A central database to store the data centrally to be used by all stakeholders. Only operational data should be visible to each stakeholder. Data privacy should be agreed amongst stakeholders.
- Data will flow between the local and central databases (the dotted lines represent data in figure 10).
- Supplies that will be shipped from the CRO or other suppliers to the sites.
- Biological specimens that will be shipped from sites to the laboratories via temperature-controlled supply chain.

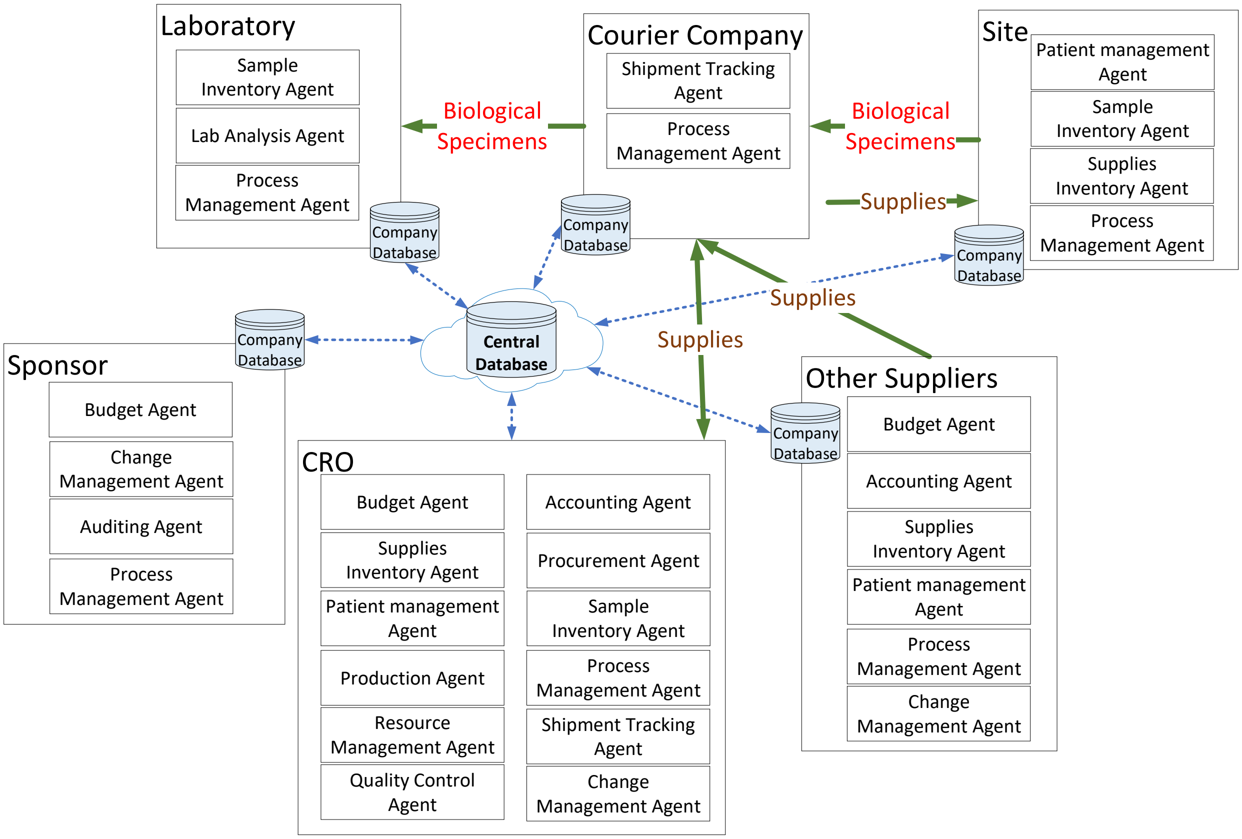


Figure 10: Proposed Multi Intelligent Agents for the Clinical Trial Logistics Management

Table 2: Description of the CRO Agents

CRO Agent	**Description**
Budget Agent	Ensures costs are and will be under the budget.
Accounting Agent	Keeps track of expenses and costs for each project.
Supplies Inventory Agent	Ensures that enough supplies with available shelf lives are available at each site before the patient's visits. This agent sends signals to the central warehouse when stock level hits a certain predefined amount.
Procurement Agent	Responsible for contacting other suppliers for procurement processes. This includes the request for quote, sending purchase orders, follow-ups, confirming lead times, etc.
Patient Management Agent	Updates the patient enrollment list to plan the upcoming supplies production.

CRO Agent	**Description**
Sample Inventory Agent	Ensures data regarding samples has been correctly collected. This also includes flagging discrepancies.
Production Agent	Coordinates all the requirements for production. This includes procurement of the raw materials, assembly schedule, and resources.
Process Management Agent	Assigns and implements the required processes for a clinical trial. There is a fixed amount of processes that each CRO is responsible for and this agent ensures processes are up to date, are capable of working with other organizations processes. This agent can proactively identify which processes will be fully utilized or underutilized.
Resource Management Agent	Plans the required resources for every task involved in a clinical trial according to the CRO’s objectives or criteria.
Shipment Tracking Agent	Coordinates and track sample shipments according to a predefined schedule and other requirements such as sample's shelf life. Tracks all the supply shipments from CRO to the sites and sample shipments from the sites to the labs.
Quality Control Agent	Checks the quality of the data associated with the supplies and services to be within the quality standards. Also, proactively finds out what current or future actions will deviate from quality standards.
Change Management Agent	Applies the required changes to the data and processes of a clinical trial. Proactively predicts the consequences of the changes and coordinates them with other system components or agents.

Table 3: Description of the Site Agents

Site Agent	**Description**
Patient Management Agent	Responsible for managing the data regarding patients. Examples of the data are patient enrollment logs and progress of each patient (Case Report Form or CRF).
Sample Inventory Agent	Ensures data regarding samples has been correctly collected. This also includes flagging discrepancy.
Supplies Inventory Agent	Ensures that enough supplies with available shelf lives are available at each site before the patient's visits. This agent sends signals to the central warehouse when stock level hits a certain predefined amount.
Process Management Agent	Assigns and implements the required processes for a clinical trial. There is a fixed amount of processes that each CRO is responsible for and this agent ensures processes are up to date, are capable of working with other organizations processes. This agent can proactively identify which processes will be fully utilized or underutilized.

Table 4: Description of the Courier Company Agents

Courier Company Agent	**Description**
Shipment Tracking Agent	Tracks shipments and coordinates the pickup and drop off of supplies and sample shipments. Handles delays for temperature dependents biological specimen shipments.
Process Management Agent	Customizes the specific processes for clinical trials shipments according to the requirements. Example of a process is a cold chain logistics for maintaining the integrity of samples from collection point to the analysis point.

Table 5: Description of the Laboratory Agents

Laboratory Agent	**Description**
Sample Inventory Agent	Keeps track of samples received and processed.
Lab Analysis Agent	Ensures the analysis for each sample is performed and results are communicated with the sponsor agent.
Process Management Agent	Customizes the specific processes for a clinical trial according to the requirements.

Table 6: Description of the Sponsor Agents

Sponsor Agent	**Description**
Budget Agent	Ensures costs are and will be under the budget.
Change Management Agent	Applies the required changes to the data and processes of a clinical trial. Proactively predicts the consequences of the changes and coordinates them with other system components and agents.
Auditing Agent	Audits the specific data points in the documents to ensure procedures meet the quality standards.
Process Management Agent	Customizes specific process for clinical trials processes according to the requirements.

Table 7: Description of the Supplies Agents

Suppliers Agent	**Description**
Budget Agent	Ensures costs are and will be under the budget.
Accounting Agent	Keeps track of expenses and costs for each project.

Suppliers Agent	**Description**
Supplies Inventory Agent	Ensures that enough supplies with available shelf lives are available at each site before the patient's visits. This agent sends signals to the central warehouse when stock level hits a certain predefined amount.
Patient Management Agent	Updates the patient enrollment list to plan the upcoming supplies production.
Process Management Agent	Assigns and implements the required processes for a clinical trial. There is a fixed amount of processes that each supplier is responsible for and this agent ensures processes are up to date, are capable to work with other organizations processes. This agent can proactively identify which processes will be fully utilized or underutilized.
Change Management Agent	Applies the required changes to the data and processes of a clinical trial. Proactively predicts the consequences of the changes and coordinates them with other system components or agents.

5.4 Clinical Trials BPMS Requirements

The aim of implementing BPMS in clinical trial logistics management is to streamline processes and automate tasks to some extent. The workflow for the BPMS is done on the computer or any other electronic device. BPMS connects the software applications by storing the data in central databases. Figure 11 shows the proposed high-level workflow diagram for clinical trials logistics management. The workflows are divided into three areas that are Study Setup, Study Management, and Sample Management. The detailed requirement of BPMS for each area will be explained in the following.

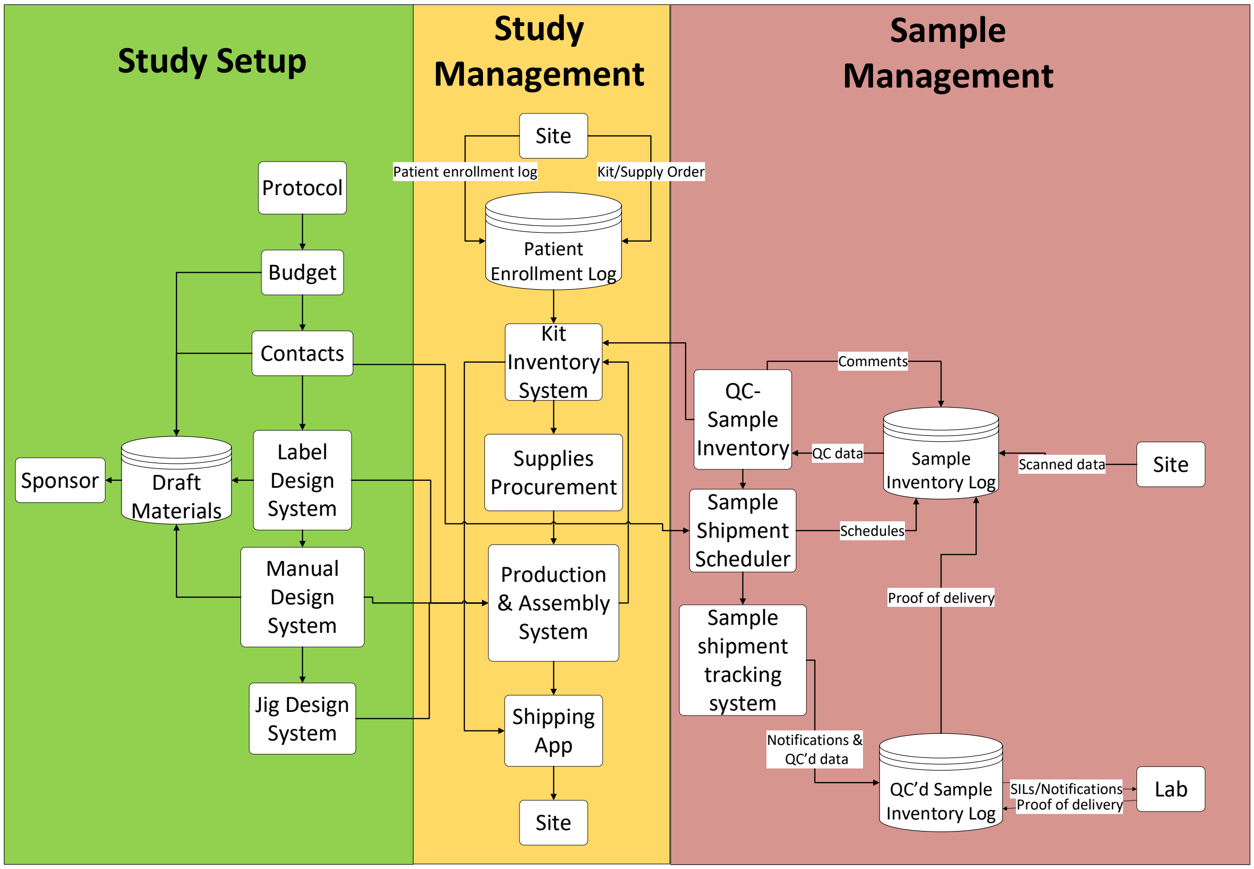


Figure 11: High-Level BPMS Requirements of the Framework

5.4.1 Study Setup

Study setup involves setting up the budget, designing laboratory manual, kits, and services specific to a protocol. The main requirements for BPMS in this phase are:

1) A platform for documenting the protocol and budget requirements.
2) A database for contacts, addresses, emails, phone number, etc. This information will be used for sample management as well.
3) A platform to design label, kit, and manual designs and share the drafts with sponsors for comments and approvals. To make the design easier, the templates or specialized applications should be integrated with the platform. Templates should be populated automatically by the available data in the budget. Figure 12 is a schematic of how data should flow from the budget into a template bank that can be used to generate labels, jigs, manuals, and bill of materials.

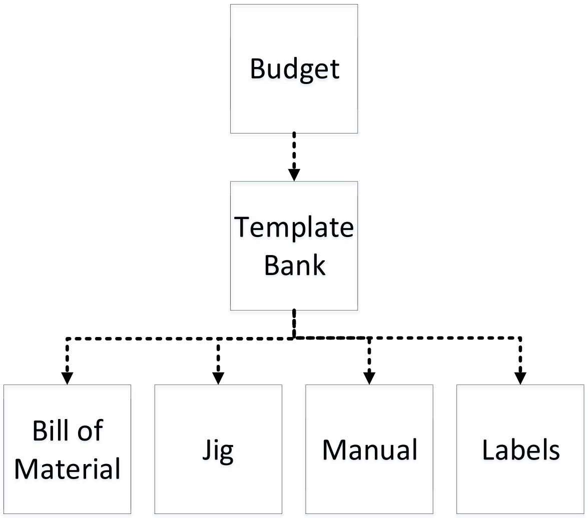


Figure 12: Flow of Data from Budget into the Templates to Streamline the Production of Kits, Labels, and Manuals

4) The platform should be able to retrieve designs of the previous projects to be used in similar future projects.
5) The proposed shipment schedules.
6) Auto-generation of bill of materials and auto-update of the inventory system for procurement.
7) A platform for site initiation visits (training materials, presentations, teleconferences).
8) Change orders documentations.
9) Change management functionality that will update the affected data and items.
10) A platform to communicate all the external communications to sponsors, sites, labs, couriers.

5.4.2 Study Management

Study management involves the implementation of logistics management in the clinical trials. In other words, producing the kits and manuals and providing the services that have been designed in the study setup. The main requirements for BPMS in this phase are:

1) Patient's enrollment database should be shared between the sites and CRO.
2) A kit inventory data system should be built for each protocol and shared between site and CROs.
3) The CRO should be notified by the system to produce and ship the required kits. Sponsors approval of the orders might be required.

4) If the kits need to be modified, the sponsor should identify in the system for which time point data should be changed and automatic notification should be sent to both the lab and site of the changes.
5) There should be a collaborative platform for changing the design of the kits between CRO and sponsor. This is because of frequent protocol amendments. Conversation systems can be incorporated in the platform.
6) Outbound shipments from CRO to sites can be tracked by the system; shipment notification should be sent to the sites automatically.
7) The approved site orders should be sent to the production department to plan inventory and production. The orders should automatically communicate with the inventory system to ensure enough kits and other material are available.
8) Sites can track their requests on the platform as well.

5.4.3 Sample Management Process

Sample management process involves managing the collected samples from patients at the sites. The samples should be stored at a certain temperature and transported to the labs to be analyzed. The main requirements for BPMS in this phase are:

1) SILs should be filled out electronically at sites. Data should be checked on the form for integrity. All the SRF fields should be on an electronic form. Back up plans should be available if the electronic forms do not work. A similar approach to Benjamins et al. (2016) can be applied here. Barcodes on sample labels can be scanned to pre-populate some field on the eSILs. Collection date and time can be either entered by the scanner, if labels are scanned at the time of collection or it can be manually entered on eSIL later. Other necessary boxes such as comment cell should be designed on the eSILs.
2) Data integrity should be checked at the time of data entry by the system. Some checks are that correct eSILs are used, correct date and time are entered, and all the necessary fields are populated by the staff at the site.
3) Conversation field should be available as a hyperlink on the form to contact the CRO or the sponsor for questions or vice versa if something goes wrong or any help is required. Once that hyperlink is clicked, a conversation box with appropriate pre-populated "subject line and responsible contacts" will pop up. The requester can write down his or her

question and press send. A notification will be shown on the respective contact's dashboard of the question. All the conversations will be logged and searchable at any time.

4) The samples data should be transferred to the CRO's central database for monitoring and other coordination.
5) After CRO monitors the data, they can schedule shipments of the collected samples if the samples are not supposed to ship on the day of collection. Shipment schedule can also be pre-programmed on the platform that automatically notifies the site staff when they should ship as soon as the required number of samples have been collected. eSILs can be virtually presented on the platform that shows what samples need to be shipped along with the instructions and documents.
6) This system can be linked to the courier database to track the shipments.
7) Automatic shipment notification along with the list of samples in the shipment and tracking number should be sent to the assay laboratory. Again, a conversation box can be available for the site to ask questions about a specific shipment.
8) Labs need to confirm receipt of the condition and number of samples on the platform. Necessary follow-ups will be done by the system if no confirmation has been received by the labs.
9) The platform should be able to handle sample discrepancies if the number of samples shipped does not match the number sample received by the lab. The required workflow can be programmed within the system to send the appropriate message to the responsible parties.

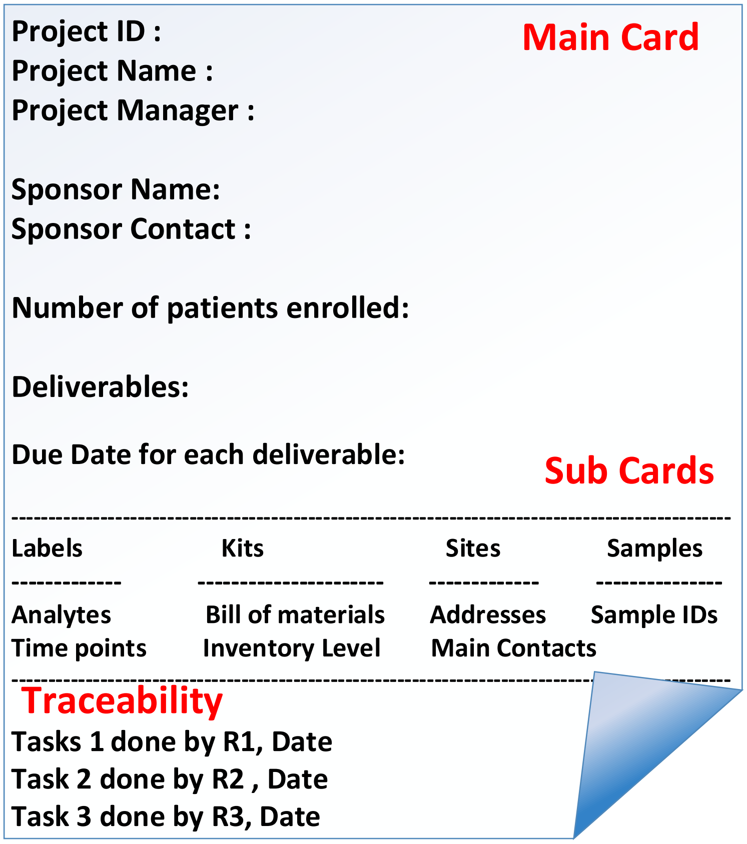
5.5 Project Card

A project card is incorporated into the system and acts as an information system for a project. Since this platform has a central data repository, queries can be performed to retrieve information about the project. Some examples of the information that can be obtained for projects are as follows:

- The progress in the entire project or a specific area. For example, a Gantt chart can be used to show the main tasks of the project.

- The resources that have been involved in any tasks. This can refer to the resource card where more information is available regarding resources. The resource card will be explained in section 5.6.
- Any signatures and dates that associate with any data or information.
- The distribution of expenses in the project.
- The documents that have been collected during the project.
- The history of change orders and amendments.
- The list of products such as kits, labels, and laboratory manual that have been used in the project.
- The lists of contacts, sites, and other vendors that have been involved in the project.
- The list of patients and samples collected for each patient.
- The log of conversations and notifications in the project.

Figure 13 shows an example of a project card that can be in either physical or electronic format. However, the electronic format will be more organized and efficient to record and retrieve data. In this example, there is a main card that has general information about a clinical trial project. There are also sub cards that can store data related to the products and services of the main project. In addition, the project card can have a traceability option, where the record of tasks will be archived.



Project ID :
Project Name :
Project Manager :

Main Card

Sponsor Name:
Sponsor Contact :

Number of patients enrolled:

Deliverables:

Due Date for each deliverable:

Sub Cards

Labels	Kits	Sites	Samples
Analytes Time points	Bill of materials Inventory Level	Addresses Main Contacts	Sample IDs

Traceability

Tasks 1 done by R1, Date
Task 2 done by R2 , Date
Task 3 done by R3, Date

Figure 13: An Example of a Project Card

Figure 14 shows how project cards will be moved through the tasks in the BPMS. Each project may go through the same or different tasks and resources depending on the requirements and availability of the resources. For example, in figure 14, project cards 1, 2, and 3 goes through tasks 1, 2, and 3. Task 1 requires resource 1 and 2, task 2 requires resource 3 and 4, and task 3 requires resource 1, 3, and 5.

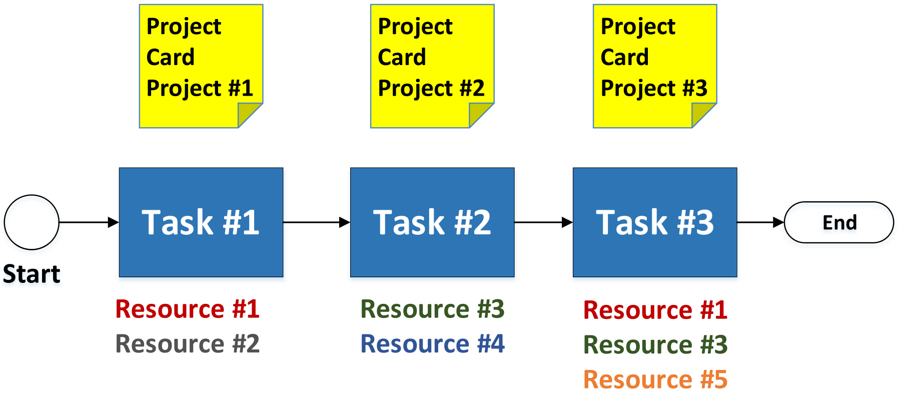


Figure 14: Movement of Project Cards through Tasks of a Process

5.6 External Applications

External applications can be any specialized applications that need to be connected to the BPMS to streamline the workflow. Examples of external applications are ERP system, label design system, inventory software, and accounting software. To distinguish between Project Card mentioned in the previous section and external applications is that external applications are here to serve the project (clinical trial), which has its own application (project card). Any data or document that is produced for a specific project by an external application will be stored in the project card. Conversely, any data that external applications require for a project will come from the project card. This research paper proposes an external application called Resource Card that will work with the Project Card and BPMS. Resource Card will have all the information about all the company's employees.

5.6.1 Resource Card

A resource card can be incorporated into this platform for multiple purposes such as a curriculum vitae (CV), skills and personal development, and a bridge between resource management and project management. Like other applications listed as external, a resource card has its own requirements. The resource card should consist of the following components:

- A log system that keeps track of activities that a team member has been involved in.
- The acknowledgments and contributions of the team member in the projects.
- The availability of the team member for another upcoming project.
- The interests, knowledge, skills, preferences of a team member that can be used to match to suitable activities and projects. Or the potential candidates can be automatically identified for certain tasks or projects.
- The areas that a team member requires training according to the history of skills and knowledge and the required skills of the new project.

Figure 15 shows an example of a resource card. Like project card, the resource card can be in physical or electronic format. However, the electronic format is preferred.

Resource ID:
Resource Name:

Available hours to work per day:
Time to take time off:
Availability each day:

Experience: Worked on this type of projects
Competencies: Computer skills

Qualified tasks: 1, 2,3, etc.

Assigned Projects: 1001, 1002, 1006, etc.

Figure 15: An Example of a Resource Card

Figure 16 shows the relationship between project cards, tasks in BPMS, and resource cards. A resource can perform one or more tasks for one or more projects. So, there can be a many-to-many relationship between resource and project cards if a shared pool of resources is used for projects. For example, in figure 16, project A has tasks 1, 2, and 5 and project B has tasks 1and 2. There are two resources (resource # 1001 and 1002) that each is qualified for a task. Tasks 1 and 5 of project A is processed by resource 1001 and task 2 of the same project is processed by resource 1002. Task 1 and 2 of project B is processed by resource 1001 and 1002 respectively.

Resource card can be useful for planning and managing human resources in the long-term (monthly or yearly planning horizon) and short-term (daily or weekly planning horizon). This problem is known as RCMPSP (Resource Constrained Multi-Project Scheduling Problem) in the operations research that can be solved using Linear Programming (LP). To solve this problem, the manager's objectives are incorporated into the LP model. Examples of objectives are minimizing total time it takes to finish all projects (makespan), assigning similar projects to one team, or having a shared pool of resources versus dedicated teams for projects. More examples of management objectives for resource and project planning will be discussed in Chapter 6.

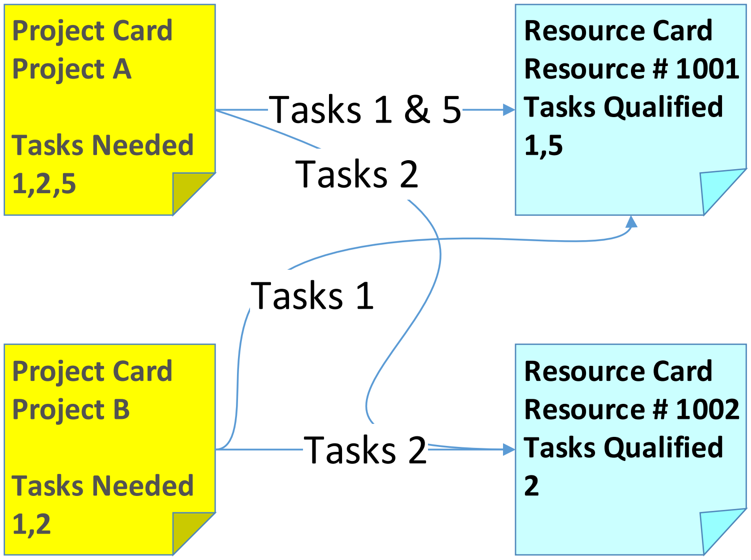


Figure 16: Relationship between Project Cards, Tasks, and Resource Cards

5.7 Document Repository

The document repository is a place to store any company's non-project related documents. The project-related documents will be stored in the project card. The business-related or functional documents such as accounting documents and documents regarding the operations will be stored in the document repository.

5.8 Data Repository

The data repository is a place where all the company's non- project related data will be stored. The project related data will be stored in the project card. The business-related or functional data such as accounting and operational data will be stored in the data repository.

5.9 Analytics

Analytics can be used to analyze the collected data from the framework and find meaningful data patterns or answer to questions that can improve the company's performance.

Examples of the questions or data that can be investigated using analytics in the clinical trial logistics management are as follows:

- Which sites are making errors that require training? Which suppliers and partner companies are making errors and reasons?
- Which human resources are making errors and require training?

- What is the source of common errors in QC? Source of reworks?
- Which areas are consuming more costs/time? Identify resource bottlenecks in the company.
- Identify the common trends for supplies in all clinical trials. SKU wise, size wise, design wise, quantity wise.
- Rank the protocols, sponsors, or type of services in terms of profit.
- Identify the historical demand for supplies and services and forecast future demand.

5.10 Security

The role of the security is to control access to all components of the main framework. For example, operations managers and employee need access to the web portal, but they do not need access to the structure of the data repository or BPMS. IT department needs access to data repository or BPMS to be able to modify the existing structure or add new components. The IT manager will be responsible to manage the security and control access of different users or stakeholders to different components of the framework. In addition, the clinical trials industry requires stringent data security and privacy standard.

Chapter 6: Resource and Project Planning in the Clinical Trial Logistics Management

In Chapter 5, we have designed a framework for clinical trials logistics management that integrates data sources, automates workflows to some extent, generate performance reports, and eventually makes the tasks easier for the human resources. In this framework, Project Card holds data about a clinical trial project, Resource Card holds information about resources available in the organization to perform the tasks, and BPMS distributes the projects' tasks to the resources. As the number of projects and tasks increases, there will be a need for a resource and project planning tool to plan tasks before being executed by the BPMS. Without proper planning, resources might not be able to finish their tasks on time and the whole project will be delayed.

In clinical trials, the logistics provider or the CRO has many ongoing projects to manage with its limited resources. In Operation Research, this problem is known as RCMPSP (Resource Constrained Multi-Project Scheduling Problem). To solve this problem, usually, a linear programming is used with respects to the company or management's objectives and preferences. In this chapter, some examples of these preferences will be discussed.

Figure 17 shows how the resource and project planning tool will be incorporated into the main framework that was discussed in Chapter 5. The proposed resource and project planning tool will be fed by the Project Card and the Resource Card. The resource and project planning tool will also communicate with the BPMS to find out the tasks that need to be done for each project and lastly will schedule resources and tasks that need to be executed in the BPMS. In the following, the resources and tasks involved in the clinical trials logistics management will be described followed by our conceptual solution to plan resources and tasks.

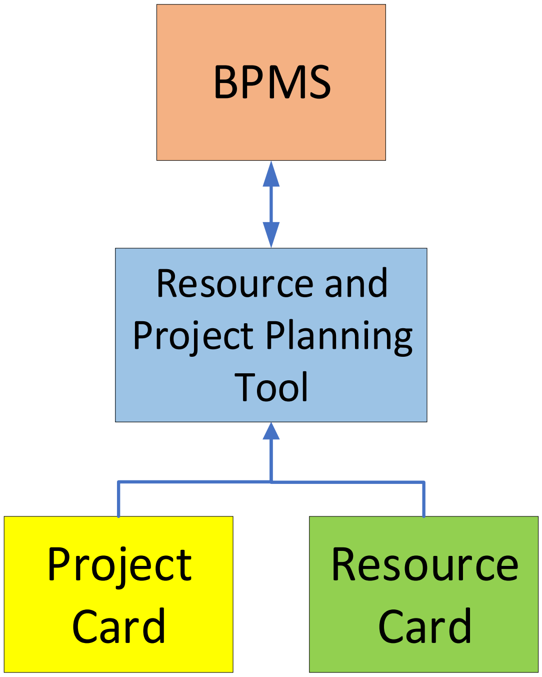


Figure 17: Incorporating a Resource and Project Planning Tool into the Proposed Framework

6.1 Description of the Project Tasks and Resources

The resources that are needed for a clinical trial logistics project is broken down to Contract Team (CT), Project Manager (PM), Operation Associate (OA), and Quality Control (QC). PM attends meetings with external study team members, find out the requirements and coordinate deliverables and changes with internal team members (OAs). OA performs simpler and more structured tasks, while project manager and contract team perform more complex tasks with less or sometimes no structure. QC checks the quality of the work through the entire processes of a project. Table 8 shows the resources that are involved in each phase of a clinical trial logistics management project.

Table 8: The Resources for Each Phase of a Clinical Trial Project

Project Phase	**Human Resources**
Contract and budget phase	CT and QC
Study setup (labels, lab manuals, kits)	PM, OA, and QC
Study management (kits management, samples management, change orders)	PM, OA, QC, and CT
Closing	PM

6.1.1 Contract and Budget Phase

In this phase, the CT meets with the sponsor to identify requirements and statement of work (SOW) for the protocol. This phase typically takes 1-2 weeks to complete.

6.1.2 Study Setup Phase

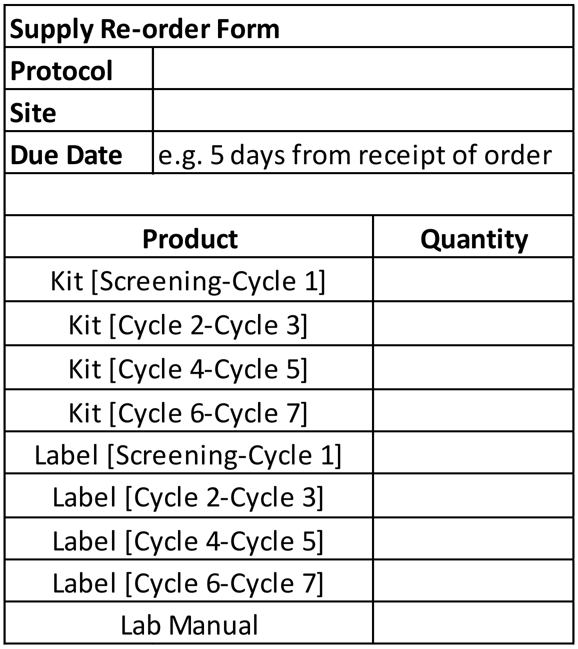
In the study setup, all the required materials such as labels, collection manual, and kits are designed and approved by the sponsor. Study setup takes a month typically and starts right after the contract phase. Study setup triggers most processes such as label design and production, manuals design and print, site initiation visit presentation, Jig design, and kit design and production.

6.1.3 Study Management Phase

In this phase, the resources must work on demand. In other words, the resources should be available when a new patient is recruited into a site or when each patient progresses from initial cycles to the others. Examples of the tasks in this phase are replenishing sample collection supplies, handling protocol amendments, sample shipment tracking, and sample management.

6.1.3.1 Replenishing Sample Collection Supplies

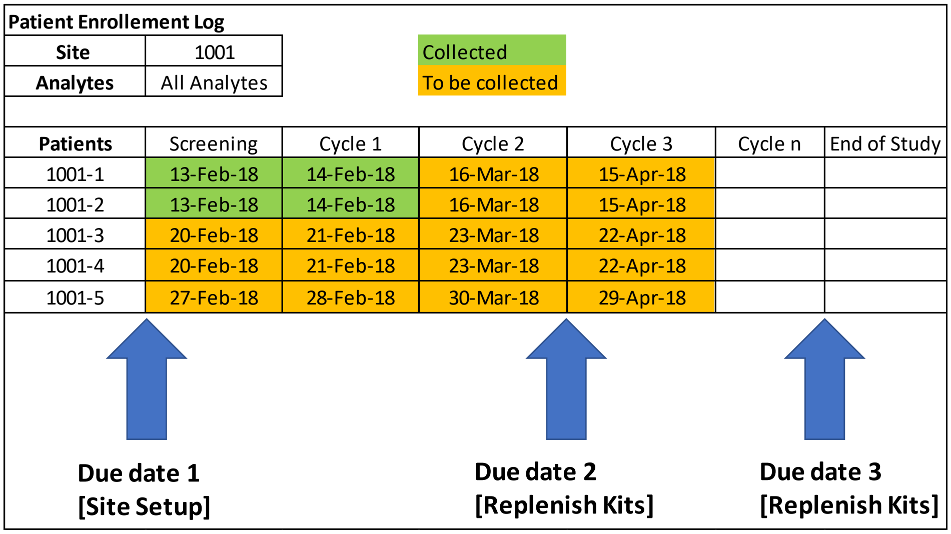
Replenishing sample collection supplies will be based on the patients enrolled in each site. For the duration of the study, each site continues to accept new patients to reach or maintain their desired number of patients. New patients require kits for initial time points, and therefore, the site will ask CROs for kits. In the meantime, existing patients may choose to continue through the study or withdraw from the study. Each period requires its own supplies or collection kits which needs to be prepared and sent by the CROs. Sites will send re-order forms to ask the CRO to replenish the supplies or kits required for new patients or existing patients progressing through collection time points. Figure 18 shows an example of the re-order form. Some sponsors might ask for bulk supplies instead of kitted supplies, for example, bulk labels, or collection tubes. The re-order forms are used to control or monitor the number of kits used by each site. CROs tend to monitor the kit expenses against the budget and communicate that with the sponsors. The re-order forms are also used to order the right number of kits with longer expiry dates. The due date to replenish re-order forms varies from 2 to 5 business days.



Supply Re-order Form	
Protocol	
Site	
Due Date	e.g. 5 days from receipt of order
Product	**Quantity**
Kit [Screening-Cycle 1]	
Kit [Cycle 2-Cycle 3]	
Kit [Cycle 4-Cycle 5]	
Kit [Cycle 6-Cycle 7]	
Label [Screening-Cycle 1]	
Label [Cycle 2-Cycle 3]	
Label [Cycle 4-Cycle 5]	
Label [Cycle 6-Cycle 7]	
Lab Manual	

Figure 18: An Example of a Supply Re-order Form

The patient enrollment log helps to order the right number of items listed on a re-order form. Figure 19 shows an example of patient enrollment log. The arrival rate of re-order forms depends on patient recruitment rate, and number and duration of cycles in a study. The higher recruitment rate leads to ordering screening kits more frequently. The longer the study duration, the more frequent orders will be, because of the expiry date of some kits.



Patient Enrollement Log

Site	1001	🟩 Collected
Analytes	All Analytes	🟧 To be collected

Patients	Screening	Cycle 1	Cycle 2	Cycle 3	Cycle n	End of Study
1001-1	🟩 13-Feb-18	🟩 14-Feb-18	🟧 16-Mar-18	🟧 15-Apr-18		
1001-2	🟩 13-Feb-18	🟩 14-Feb-18	🟧 16-Mar-18	🟧 15-Apr-18		
1001-3	🟧 20-Feb-18	🟧 21-Feb-18	🟧 23-Mar-18	🟧 22-Apr-18		
1001-4	🟧 20-Feb-18	🟧 21-Feb-18	🟧 23-Mar-18	🟧 22-Apr-18		
1001-5	🟧 27-Feb-18	🟧 28-Feb-18	🟧 30-Mar-18	🟧 29-Apr-18		

Due date 1
[Site Setup]
Due date 2
[Replenish Kits]
Due date 3
[Replenish Kits]

Figure 19: An Example of a Patient Enrollment Log and Multiple Due Dates to Replenish Kits

6.1.3.2 Amendments

Amendments or change orders are very common in a clinical trial. The sponsor might decide to change the collection manual, labels, and kits anytime during the study. These changes should be reflected in the budget and any other subsequent supplies and materials (kits, labels, manual) that will be affected by the change. New patients enrolled in the study after any amendments, must be treated according to the new amendment and receive amended collection materials.

6.1.3.3 Sample Shipment Scheduling and Tracking

Sample shipments should be scheduled to ship the collected samples from the site to the laboratory. The frequency of the shipments varies from the day of collection to monthly and quarterly. The frequency of the shipment depends on the window of integrity of the sample, and the capacity of the site and lab. The content of each shipment can be determined from the patient's enrollment tracker and sample collection tracker. Special shipping kits suitable for maintaining the desired temperature is used to transfer samples from the site to the laboratory. Shipments should be tracked, and labs should be notified of upcoming shipments to prepare the analysis. Any delays should be resolved and documented.

6.1.3.4 Sample Management

Sample management means keeping track of the collected samples and ensure they have been shipped from the site and received by the laboratory. Information about collected and shipped samples are documented in a sample inventory log (SIL) for the sponsor and laboratories. Any discrepancy should be resolved and documented.

6.1.3.5 Closing

At this stage, the project manager ensures all the requirements for the clinical trial have been addressed. The discrepancies should be resolved, outstanding fees should be paid, and all documents should be archived for future references.

6.2 Task Flow

The outputs of clinical trials logistics management are products and services such as labels, kits, inventory management, and sample management. During each clinical trial, the CRO or the logistics provider is responsible to provide the products and services as they are demanded by the sites, labs, and the sponsor.

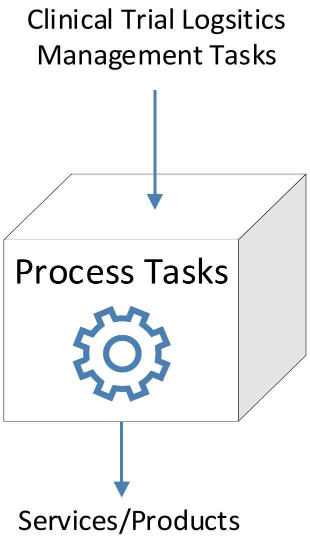


Figure 20: CRO Processes Tasks to Produce Services and Products for Clinical Trials

The workflow consists of three levels that are project phases, processes for each project phase, and tasks for each process (as shown in Figure 21). The processes and tasks in this industry are often undergone revisions.

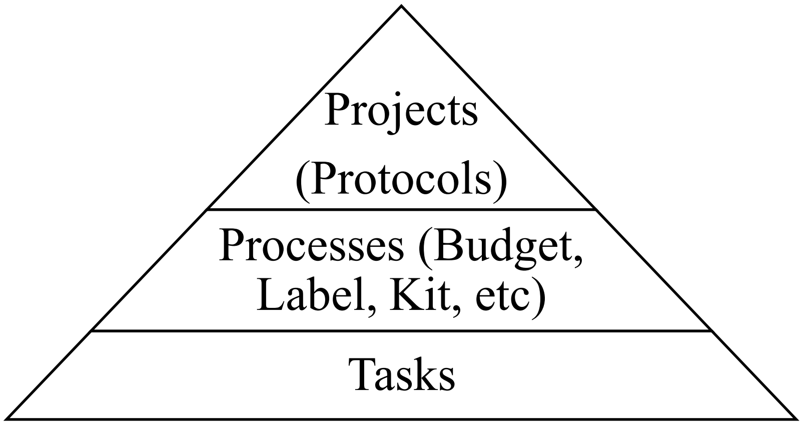


Figure 21: The Workflow Hierarchy for Clinical Trial Logistics Management

Figure 22 shows the main project phases of a clinical trial logistics management, and some examples of processes and the tasks within them such as processes for budget creation, label creation, and sample management. In the next section, we propose a conceptual solution on how to plan the tasks and resources in this setting.

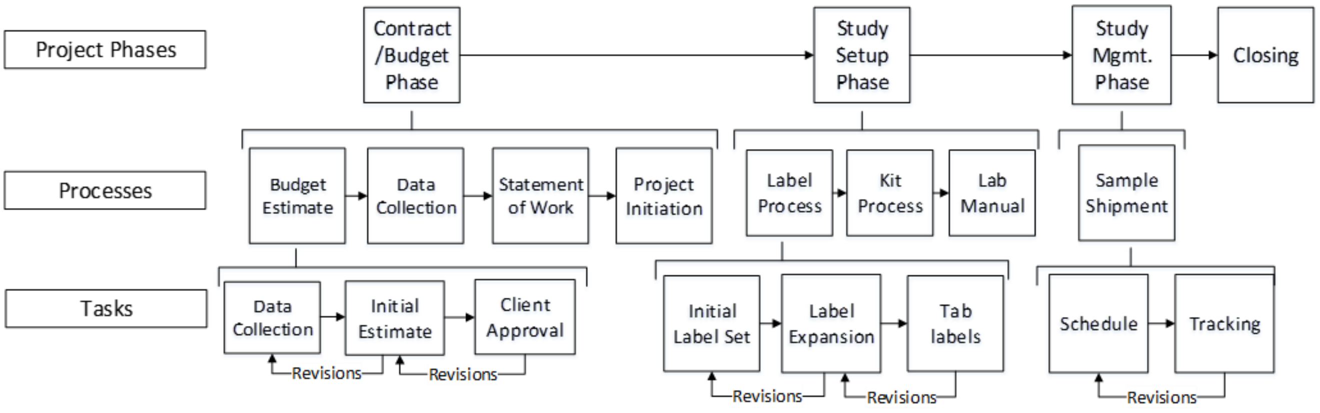


Figure 22: An Example of a Workflow for Clinical Trial Logistics Management

Table 9 shows an example of how due dates will be assigned to each process within each project phase. The due date for each process is the required finish time of the last task of a process. For example, for Budget phase, if we want to initiate the Budget Estimate process, we need to assign a due date for Client Approval task, which is the last task of the Budget Estimate process (as shown in Figure 22). Many of the processes can be re-initiated for a project because of many reasons such as revisions, amendments, setting up a new site, and recruiting new patients. To reinitiate, a new due date should be assigned for that process of any phase.

Table 9: Due Dates for each Phase of a Clinical Trial Logistics Management Project

Project Phase	**Processes**	**Due Dates**
Budget (Amendments or change orders)	Budget Estimate Process	Required finish time of the last task in the budget process (usually a few due dates due to amendments).
Study Setup	Label Process	Required finish time of the last task in the Project Setup Process (usually a few due dates due to amendments).
Study Management	Kit Production Process	Required finish time of the last task in Kit Production process.
	Sample Shipment Process	Required finish time of the last task in Sample Management Process.
Closing	Closing Process	Required finish time of the last task in Closing Process.

6.3 Conceptual Solution for Planning the Tasks and Resources

This solution consists of two parts. The first part is a planning and scheduling tool that plans the tasks and resources. The second part is a task execution tool, BPMS, that executes the planned tasks.

6.3.1 Planning and Scheduling Tool

The planning and scheduling tool will have input data that needs to be entered by the management and output data that the system will generate. The input and output data are explained in the following.

6.3.1.1 Input Data

1. The main processes of each phase of a project should be identified in a Gantt chart inside the project card. A predefined list of processes can be used with blank due dates that need to be filled out later by the project manager of that clinical trial project. Note that the processes are listed in the Gantt chart while the tasks of that process are listed in the BPMS.
2. The due date for each process should be identified by the Project Manager of that project. Then, the planning tool will schedule the tasks of each process to be finished by its due date.
3. The duration for each task that is in the BPMS can be estimated by a time study of the current practice or expert's opinions. The tasks dependencies are also known and are reflected in the business processes that are located on the BPMS, for example, "confirm lab addresses" should be done before "order preprinted airwaybill".

6.3.1.2 Output Data

4. The planning tool shall assign human resources such as the Project Managers, Quality Controls, and Operations Associates to the project according to the resource card. The resource card will have the list of resources with their availabilities, qualifications, and type of tasks and projects that the resources prefer to work with.
5. The planning tool shall also identify the start and finish date for each task automatically according to the due date, the duration of each task, and task dependencies (processors and successors), and resource availability.

6.3.2 Task Execution Steps in BPMS

6. The start and finish dates that were generated from the planning tool shall be communicated with the BPMS. The breakdown for each process (or sub-tasks) is listed in the BPMS to be executed. Currently, the due date for each task is defined as a fixed unit, for example, 1 hour, 1 day, 1 week, etc. For example, if a task is initiated today at noon and must be finished in 1 day (according to the predefined fixed duration), then the task must be done by tomorrow at noon; otherwise, the task is considered late. However, we want to have a variable start and due dates in the BPMS. The start and finish date for each task that was generated from the planning tool should be communicated with the BPMS to assign the start and finish date of each task in the BPMS. Note that the system will assign the finish dates for each task except the last task in the process, which will have the same due date as the main process.
7. Currently, in BPMS, there are departments or groups that each is responsible for a set of tasks. For example, the contract group is responsible for contract and budget tasks, or QC group is responsible for quality control tasks. The tasks will be entered into the group's pool of tasks and someone in the group should take that tasks. The BPMS can assign multiple people to a task if needed. The preference is that the task stays with the person who started it until end of the process. Currently, the BPMS does not assign the tasks automatically to the people in a group, and it is not capable of knowing the workload of each individual in the group. The scheduling system will consider the workloads and other objectives to distribute the tasks to individuals in a department or group.

6.3.3 Planning and Scheduling Objectives

The resource and project planning tool shall assign tasks to resources according to required criteria to achieve the objectives. Below are some examples of management choices and scenarios that can be used to plan tasks and resources for a CRO or any other logistics provider in the clinical trials.

6.3.3.1 Management Choices

The management choices and objectives can be varied from one organization to the other and from time to time depending on the priorities of the company. The planning tool should be flexible enough that allows the management to combine different choices or objectives at any time and for a given planning horizon. Although the choices are limitless, important choices that can benefit the operation of clinical trials logistics management are listed below. These choices have been categorized as company-wide objectives and task assignment choices.

Company-wide objectives:

- Finish all of the processes of all projects by their due dates. For example, the budget shall be done by its due date, supplied shall be delivered by its due date, schedules and notification shall be sent in a timely manner.
- Optimize the company's capacity. Increase resource utilization.
- Increase accuracy and efficiency throughout the entire projects.
- Determine the appropriate level of resources for a given planning horizon to meet the company's objectives.

Task Assignment Choices:

- Assign tasks of any project (protocol) to people in a group according to their qualifications. For example, in the QC group, some people can only QC labels, while some are qualified to QC budgets as well as labels.
- Include familiarity factors for resources.
- Assign similar projects to one project manager for more efficiency.
- Assign urgent or high priority tasks to more experienced people to process them faster.
- Assign tasks according to the resource's workload.
- Transfer of resources from one resource group to the other for cross training and when work demand moves from one department to the other.
- Project's tasks stay with the same person or group who started the first task of that job within that process to increase the familiarity factor.
- Assign simpler tasks to new employees.

6.4 The Linear Programing Model

To design a planning tool that will address the management choices mentioned in sub-section 6.3.3.1, two main planning models are suggested to be built for clinical trials logistics management. The first model is a task scheduling LP and the second model is a resource planning LP. The task scheduling LP will be used when the operations manager has a fixed number of resources and wants to know how to schedule tasks accordingly. However, the resource planning tool will help the manager not only schedule the tasks, but to find out the required number of resources needed for every resource group in order to finish all tasks without any tardiness. Each of these models will be explained in the following sub-sections.

6.4.1 Model 1– Task Scheduling LP

Schedule sub-tasks by building a scheduling LP (Basic Scenario). Three scenarios will be tested within this model to see the effect of each scenario on the objective function. The scenarios are: fixed shared resources among all projects, fixed dedicated resources for each project, and fixed dedicated resources in project work groups.

6.4.1.1 Scenario I – Fixed Shared Resources among All Projects

In a shared or non-dedicated resource model, resources are shared amongst projects (one resource should work on multiple projects). This model is feasible and economical when each project does not demand dedicated resources and resources have time to process tasks for multiple parallel projects. In most projects, there are slacks between tasks and processes that the resources dedicated to them become idle during those situations. Therefore, it is economical to share the resources amongst projects. In this model shared resources will be fixed to see its effects on the objective function. The fixed resources represent the resources on hand in a company.

6.4.1.2 Scenario II – Fixed Dedicated Resources for Each Project

In the dedicated model, the resource will focus on one or a set of similar projects and cannot be shared with all other projects. This makes him or her work more efficiently and accurately and as a result, will benefit clinical trials logistics more. Clinical trials are performed under stringent quality standards. Therefore, quality is as important as efficiency. However, the drawback of this model is that the dedicated resources are not always utilized, because of the slacks involved in one or more projects that they are assigned to. Another drawback is that in the dedicated model resources are not flexible enough (require longer learning curve) to be moved from task to the other. In this scenario, the number of dedicated resources for each project shall be determined by the management.

6.4.1.3 Scenario III – Fixed Dedicated Resources in Project Work Groups

In this model, resources will work within work groups. Each work group will have its own dedicated resources. Similar projects or projects from the same sponsor will be assigned to a work group. This model takes the characteristics of both shared and dedicated resource models since it dedicates a group of resources to several projects and resources can be shared amongst those projects if needed. In this scenario, the number of resources in each work group will be fixed by the management.

6.4.2 Model 2– Resource Planning LP

This model determines the optimal number of resources to meet the objective function within the planning horizon.

6.4.2.1 Scenario IV – Variable Shared Resources

Same as the scenario I, but with a variable number of resources.

6.4.2.2 Scenario V – Variable Dedicated Resources

Same as the scenario II, but with a variable number of resources.

6.4.2.3 Scenario VI – Variable Dedicated Work Groups

Same as the scenario III, but with a variable number of resources.

6.4.3 Background Research on RCMPSP

Some of the research articles in the area of project planning and scheduling have already been mentioned in the Literature Review Chapter. For example, Hartmann and Briskorn (2010) surveyed plenty of articles regarding RCPSP, Resource Constrained Project Scheduling Problem. Herroelen and Leus (2005) surveyed solution models for project scheduling under uncertainty. Faghihi et al. (2015) surveyed existing algorithms and techniques for solving project scheduling problem in the construction industry. Al Nasseri et al. (2016) prepared a taxonomy for planning and scheduling for construction project management. They review theories and concepts and their key features, usability and suitability, and benefits and limitations.

In addition to the above articles, we found Talbot (1982) and Besikci, Bilge, and Ulusoy (2015) that are very applicable to project management in clinical trials. Talbot (1982) formulated a mixed integer linear programming model to the RCPSP (Resource Constrained Project Scheduling Problem) using the multi-mode approach, which means that a task can be done in different modes and in each mode, number of resources and duration for each task is different. Talbot's model has been a reference to many other articles. Besikci et al. (2015) extended the Talbot (1982)'s formulation to solve RCMPSP (Resource Constrained Multi-Project Scheduling Problem), which facilitates the scheduling of multi projects. In addition, they dedicate resources to each project while Talbot (1982) shares the resources. We chose Talbot (1982) and Besikci et al. (2015)'s models as our reference models in this thesis. Talbot (1982) provides a scheduling model with shared renewable and non-renewable resources, which we wanted to use as one of our scenarios in clinical trials task and resource planning. Talbot also provides a clear example that we could use to validate our model against. Besikci et al. (2015) provide a multi-project scheduling with dedicated resources to each project; this model gave us a starting point to build a multi-project scheduling model and a dedicated resource configuration.

6.4.4 Our Approach to RCMPSP

Our model uses both Talbot (1982) and Besikci et al. (2015) with extensions to use processes in addition to projects, having process precedence relationship in addition to task precedence relationship, using only one resource type for each mode, having deadlines for processes in addition to projects, and running several scenarios with fixed and shared number of resources that was mentioned in section 6.4. Our approach to solving the RCMPSP for the clinical trial logistics management is shown in table 10. In this table, there are two main linear programming models that are used for task planning and resource planning.

Three scenarios (non-dedicated resources, dedicated resources, resources in work groups) are incorporated in each model to be used by the managers in the clinical trials industry. In task planning model, we fix the number of resources in each scenario. However, for resource planning model, we let the number of resources be variables and let the LP model determine the optimal level of resources. BR_g is a variable that represents the total number of shared resources from each resource group to complete all projects. BR_{vg} is a variable that represents the total number of dedicated resources to each project (v) to complete all projects. BR_g is used in the scenario I, III, IV, and VI, because resources are shared amongst the projects in one or more work groups. BR_{vg} is used in scenario II and IV, because resources are dedicated to each project.

In scenarios III and IV (resources in work groups), a multiple number of work groups can be used, while no work groups (or in other words, all resource groups are part of one work group) is used in non-dedicated and dedicated resource configurations models. In all resource configuration, each mode represents one or more resource group (g). In other words, a mode represents a unique combination of resources and the rate at which that task will be processed in that mode. In all scenarios, one to many projects and processes can be used.

Table 10: The Proposed RCMPSP Models for Clinical Trials Logistics Management

Purpose of Model	Planning Clinical Trial Projects		
Number of Work Group	None		w
Number of Projects	v		
Number of Processes	p, where p > v		
Source Model	Talbot (1982) & Besikci et. al (2015)		
Resource Configuration	One or more resource groups in each mode		
	Non-dedicated Resources	Dedicated Resources	Resources in Work Groups
Task Planning Model	Fixed (BR_g) - Scenario I	Fixed (BR_{vg}) - Scenario II	Fixed (BR_g) - *Scenario III*
Resource Planning Model	Variable (BR_g) - Scenario IV	Variable (BR_{vg}) - Scenario V	Variable (BR_g) - *Scenario VI*

The formulation for task planning model (scenarios I and II) and task planning model (scenario III) will be shown in sub-sections 6.4.5 and 6.4.6 respectively. The formulation for resource planning model (scenarios IV and V) and resource planning model (scenario VI) will be shown in sub-sections 6.4.7 and 6.4.8 respectively.

6.4.5 Task Planning Model (Scenarios I and II)

Sets and indices

V set of projects, $v \in V$

P_v set of processes of project $v, p \in P_v$

J_{vp} set of tasks of process p of project $v, j \in J_{vp}$

p_{vN} last process (indexed with N) of project v

j_{vpL} last task (indexed with L) of process p of project v

j_{vpF} first task (indexed with F) of process p of project v

S_v set of all process precedence relationships of project v

Q_{vp} set of all task precedence relationship of process p of project v

M_{vpj} set of modes for task j of process p of project v

G set of resource groups, $g \in G$

K_g set of renewal resources for group g, $k \in K_g$

T set of time periods of project, $t \in T$

Parameters

EE_{vpj} earliest finish time of task j of process p of project v

LL_{vpj} latest finish time of task j of process p of project v

E_{vp} earliest finish time of process p of project v

L_{vp} latest finish time of process p of project v

d_{vpjm} duration of tasks j of process p of project v, operating on mode m

r_{vpjgkm} usage of tasks j of process p of project v,
operating on mode m from renewable resource k of group g

dd_v assigned due date for project v

ddd_{vp} due date for the process p of project v

c_v relative weight of project v

c_{vp} relative weight of process p of project v

b_{gk} unit cost of renewable resource k belonging to group g

BR_g number of renewable resources in group g per period t

Or

BR_{vg} number of renewable resources in group g per period t dedicated to project v

Decision variables

$$x_{vpjmt} = \begin{cases} 1, \text{if task } j \text{ of process p of project } v \text{ operating on mode m is finished at period } t \\ 0, otherwise \end{cases}$$

$TC_v =$ Weighted tardiness cost of project v

$TC_{vp} =$ Weighted tardiness cost of process p of project v

Mathematical model

$$\text{Min z} = \sum_{v\,\in\,V} TC_v + \sum_{v\,\in\,V}\ \sum_{p\,\in\,P_v} TC_{vp} \qquad (1.1)$$

Subject to

$$\sum_{m\,\in\,M_{vpj}}\ \sum_{t=EE_{vpj}}^{LL_{vpj}} x_{vpjmt} = 1 \qquad \forall v \in V, \forall p \in P_v, \text{and } \forall j \in J_{vp} \qquad (1.2)$$

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpb}}^{LL_{vpb}} \left(t - d_{vpjm}\right) x_{vpbmt} \geq \sum_{m \in M_{vpj}} \sum_{t=EE_{vpa}}^{LL_{vpa}} t\, x_{vpamt} \;\; \forall\, (a,b) \in Q_{vp} \text{ and } \forall v \in V \tag{1.3}$$

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj_{vbF}}}^{LL_{vpj_{vbF}}} (t - d_{vpj_{vbF}m}) x_{vbj_{vbF}mt} \geq \sum_{m \in M_{vpj}} \sum_{t=E_{va}}^{L_{va}} t\, x_{vaj_{vaL}mt} \;\; \forall\, (a,b) \in S_v \text{ and } \forall v \in V \tag{1.4}$$

$$\sum_{v \in V} \sum_{p \in P_v} \sum_{j \in J_{vp}} \sum_{q=\max\{t, EE_{vpj}\}}^{\min\{t+d_{vpjkg}-1, LL_{vpj}\}} r_{vpjgkm} x_{vpjmq} \leq BR_g \quad \forall\, g \in G, and\ \forall\, t \in T \tag{1.5}$$

OR

$$\sum_{p \in P_v} \sum_{j \in J_{vp}} \sum_{q=\max\{t, EE_{vpj}\}}^{\min\{t+d_{vpjkg}-1, LL_{vpj}\}} r_{vpjgkm} x_{vpjmq} \leq BR_{vg} \quad \forall\, g \in G, \forall\, t \in T, \text{and } \forall \in V \tag{1.6}$$

$$TC_v \geq C_v \left(\sum_{m \in M_{vpj}} \sum_{t=E_{vp_{vN}}}^{L_{vp_{vN}}} t\, x_{vp_{vN}j_{vpL}mt} - dd_v \right) \quad \forall\, v \in V \tag{1.7}$$

$$TC_{vp} \geq C_{vp} \left(\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj_{vpL}}}^{LL_{vpj_{vpL}}} t\, x_{vpj_{vpL}mt} - ddd_{vp} \right) \quad \forall v \in V, \text{and } \forall p \in P_v \tag{1.8}$$

$$x_{vpjmt} \in \{0,1\} \quad \forall v \in V, \forall p \in P_v, \forall\, j \in J_{vp}, \forall\, m \in M, \text{and } \forall\, t \in T_v \tag{1.9}$$

$$TC_v, TC_{vp} \in Z^{+} \quad \forall p \in P_v, \text{and } \forall v \in V \tag{1.10}$$

The objective function (1.1) minimizes the total weighted tardiness over all projects and processes. Constraint set (1.2) ensures that all tasks are scheduled only once for all projects and processes. Constraint set (1.3) enforces the precedence relationships amongst tasks within each process and project. Constraint set (1.4) enforces the precedence relationship amongst processes of each project. Constraint set (1.5) determines the maximum number of renewable shared resources per period across all projects during the planning horizon. Constraint set (1.6) determines the maximum number of renewable dedicated resources per period to each project during the planning horizon. Constraint 5 is used in the scenario I and constraint 6 is used in scenario II. Constraint sets (1.7) and (1.8) calculate the weighted tardiness for projects and processes respectively. Constraint sets (1.9) and (1.10) specifies the feasible range for decision variables.

6.4.6 Task Planning (Scenario III)

Sets and indices

V set of projects, $v \in V$

W set of project work groups, $w \in W$

P_v set of processes of project $v, p \in P_v$

J_{vp} set of tasks of process p of project $v, j \in J_{vp}$

p_{vN} last process (indexed with N) of project v

j_{vpL} last task (indexed with L) of process p of project v

j_{vpF} first task (indexed with F) of process p of project v

S_v set of all process precedence relationships of project v

Q_{vp} set of all task precedence relationship of process p of project v

M_{vpj} set of modes for task j of process p of project v

M_w set of modes in project work group w

G set of resource groups, $g \in G$

K_g set of renewal resources for group g, $k \in K_g$

T set of time periods of project, $t \in T$

Assumption: k belongs to one g only. In other words, one person belongs only to one group and cannot be in more than one groups.

Parameters

EE_{vpj} earliest finish time of task j of process p of project v

LL_{vpj} latest finish time of task j of process p of project v

E_{vp} earliest finish time of process p of project v

L_{vp} latest finish time of process p of project v

d_{vpjm} duration of tasks j of process p of project v, operating on mode m

r_{vpjgkm} usage of tasks j of process p of project v,
operating on mode m from renewable resource k of group g

dd_v assigned due date for project v

ddd_{vp} due date for the process p of project v

c_v relative weight of project v

c_{vp} relative weight of process p of project v

b_{gk} unit cost of renewable resource k belonging to group g

BR_{gk} number of renewable resources k in group g per period t

Decision variables

$$x_{vpjmt} = \begin{cases} 1, \text{if task } j \text{ of process p of project } v \text{ operating on mode m is finished at period } t \\ 0, \text{otherwise} \end{cases}$$

$TC_v =$ Weighted tardiness cost of project v

$TC_{vp} =$ Weighted tardiness cost of process p of project v

$$y_{vw} = \begin{cases} 1, \text{if project } v \text{ is in the work group } w \\ 0, \text{otherwise} \end{cases}$$

Mathematical model

$$\text{Min z} = \sum_{v \in V} TC_v + \sum_{v \in V} \sum_{p \in P_v} TC_{vp} \tag{2.1}$$

Subject to

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj}}^{LL_{vpj}} x_{vpjmt} = 1 \quad \forall v \in V, \forall p \in P_v, \text{and } \forall j \in J_{vp} \tag{2.2}$$

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpb}}^{LL_{vpb}} (t - d_{vpjm})\, x_{vpbmt} \geq \sum_{m \in M_{vpj}} \sum_{t=EE_{vpa}}^{LL_{vpa}} t\, x_{vpamt} \;\forall\, (a,b) \in Q_{vp} \text{ and } \forall v \in V \tag{2.3}$$

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj_{vbF}}}^{LL_{vpj_{vbF}}} (t - d_{vpj_{vbF}m}) x_{vbj_{vbF}mt} \geq \sum_{m \in M_{vpj}} \sum_{t=E_{va}}^{L_{va}} t\, x_{vaj_{vaL}mt} \;\forall\, (a,b) \in S_v \text{ and } \forall v \in V \tag{2.4}$$

$$\sum_{v\in V}\sum_{p\in P_v}\sum_{j\in J_{vp}}\sum_{q=\max\{t,EE_{vpj}\}}^{\min\{t+d_{vpjkg}-1,LL_{vpj}\}} r_{vpjgkm}x_{vpjmq} \leq BR_{gk} \quad \forall\, k \in K_g, \forall\, g \in G, and\ \forall\, t \in T \tag{2.5}$$

$$TC_v \geq C_v\left(\sum_{m}\sum_{t=E_{vp_{vN}}}^{L_{vp_{vN}}} t\, x_{vp_{vN}j_{vpL}mt} - dd_v\right) \quad \forall\, v \in V \tag{2.6}$$

$$TC_{vp} \geq C_{vp}\left(\sum_{m}\sum_{t=EE_{vpj_{vpL}}}^{LL_{vpj_{vpL}}} t\, x_{vpj_{vpL}mt} - ddd_{vp}\right) \quad \forall v \in V, \text{and } \forall p \in P_v \tag{2.7}$$

$$\sum_{w\in \mathrm{W}} y_{v,w} = 1 \quad \forall v \in V \tag{2.8}$$

$$x_{vpjmt} \leq y_{v,w}\ \forall v \in V, \forall p \in P_v, \forall j \in J_{vp}, \forall m \in M_w, \forall w \in W, \text{and } \forall\, t \in T \tag{2.9}$$

$$x_{vpjmt} \in \{0,1\} \quad \forall v \in V, \forall p \in P_v, \forall\, j \in J_{vp}, \forall\, m \in M_w, \text{and } \forall\, t \in T \tag{2.10}$$

$$TC_v, TC_{vp} \in Z^+ \quad \forall p \in P_v, \text{and } \forall v \in V \tag{2.11}$$

$$y_{vw} \in \{0,1\} \quad \forall v \in V, \text{and } \forall w \in W \tag{2.12}$$

The objective function (2.1) minimizes the total weighted tardiness over all projects and processes. Constraint set (2.2) ensures that all tasks are scheduled only once for all projects and processes. Constraint set (2.3) enforces the precedence relationships amongst tasks within each process and project. Constraint set (2.4) enforces the precedence relationship amongst processes of each project. Constraint set (2.5) determines the maximum number of renewable resources per period during the planning horizon. Constraint sets (2.6) and (2.7) calculate the weighted tardiness for projects and processes respectively. Constraint set (2.8) ensures each project is only in one work group. Constraint set (2.9) ensures that each task of a project is assigned a mode that belongs to the work group of that project. Constraint sets (2.10), (2.11), and (2.12) specifies the feasible range for decision variables.

6.4.7 Resource Planning Model (Scenario IV and V)

Sets and indices

V set of projects, $v \in V$

P_v set of processes of project $v, p \in P_v$

J_{vp} set of tasks of process p of project $v, j \in J_{vp}$

p_{vN} last process (indexed with N) of project v

j_{vpL} last task (indexed with L) of process p of project v

j_{vpF} first task (indexed with F) of process p of project v

S_v set of all process precedence relationships of project v

Q_{vp} set of all task precedence relationship of process p of project v

M_{vpj} set of modes for task j of process p of project v

G set of resource groups, $g \in G$

K_g set of renewal resources for group g, $k \in K_g$

T set of time periods of project, $t \in T$

Parameters

EE_{vpj} earliest finish time of task j of process p of project v

LL_{vpj} latest finish time of task j of process p of project v

E_{vp} earliest finish time of process p of project v

L_{vp} latest finish time of process p of project v

d_{vpjm} duration of tasks j of process p of project v, operating on mode m

r_{vpjgkm} renewable resource k of group g usage of task j of process p of project v, operating on mode m

dd_v assigned due date for project v

ddd_{vp} due date for the process p of project v

c_v relative weight of project v

c_{vp} relative weight of process p of project v

b_{gk} unit cost of renewable resource k belonging to group g

Decision variables

$$x_{vpjmt} = \begin{cases} 1, \text{if activity } j \text{ of process p of project } v \text{ operating on mode m is finished at period } t \\ 0, otherwise \end{cases}$$

$TC_v =$ Weighted tardiness cost of project v

TC_{vp} = Weighted tardiness cost of process p of project v

BR_g = Number of renewable resources in group g per period t

Or

BR_{vg} = Number of renewable resources in group g per period t dedicated to project v

Mathematical model

$$\text{Min z} = \sum_{v \in V} TC_v + \sum_{v \in V} \sum_{p \in P_v} TC_{vp} \tag{3.1}$$

Subject to

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj}}^{LL_{vpj}} x_{vpjmt} = 1 \quad \forall v \in V, \forall p \in P_v, \forall j \in J_{vp} \tag{3.2}$$

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpb}}^{LL_{vpb}} \left(t - d_{vpjm}\right) x_{vpbmt} \geq \sum_{m \in M_{vpj}} \sum_{t=EE_{vpa}}^{LL_{vpa}} t\, x_{vpamt} \;\; \forall\, (a,b) \in Q_{vp}, \forall v \in V, and\; \forall p \in P_v \tag{3.3}$$

$$\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj_{vbF}}}^{LL_{vpj_{vbF}}} (t - d_{vpj_{vbF}m}) x_{vbj_{vbF}mt} \geq \sum_{m \in M_{vpj}} \sum_{t=E_{va}}^{L_{va}} t\, x_{vaj_{vaL}mt} \;\; \forall\, (a,b) \in S_v \text{ and } \forall v \in V \tag{3.4}$$

$$\sum_{v \in V} \sum_{p \in P_v} \sum_{j \in J_{vp}} \sum_{q=\max\{t, EE_{vpj}\}}^{\min\{t + d_{vpjkg} - 1, LL_{vpj}\}} r_{vpjgkm} x_{vpjmq} \leq BR_g \;\; \forall\, g \in G, and\; \forall\, t \in T \tag{3.5}$$

OR

$$\sum_{p \in P_v} \sum_{j \in J_{vp}} \sum_{q=\max\{t, EE_{vpj}\}}^{\min\{t + d_{vpjkg} - 1, LL_{vpj}\}} r_{vpjgkm} x_{vpjmq} \leq BR_{vg} \;\; \forall\, g \in G, \forall\, t \in T, \text{and } \forall \in V \tag{3.6}$$

$$TC_v \geq C_v \left(\sum_{m \in M_{vpj}} \sum_{t=E_{vp_{vN}}}^{L_{vp_{vN}}} t\, x_{vp_{vN} j_{vpL} mt} - dd_v \right) \quad \forall\, v \in V \tag{3.7}$$

$$TC_{vp} \geq C_{vp} \left(\sum_{m \in M_{vpj}} \sum_{t=EE_{vpj_{vpL}}}^{LL_{vpj_{vpL}}} t\, x_{vpj_{vpL}mt} - ddd_{vp} \right) \quad \forall v \in V, \text{and } \forall p \in P_v \tag{3.8}$$

$x_{vpjmt} \in \{0,1\} \quad \forall v \in V, \forall p \in P_v, \forall\, j \in J_{vp}, \forall\, m \in M_{vpj}, \text{and } \forall\, t \in T_v$ (3.9)

$TC_v, TC_{vp}, BR_g, BR_{vg} \in Z^+ \quad \forall p \in P_v, \text{and } \forall v \in V$ (3.10)

The objective function (3.1) minimizes the total weighted tardiness over all projects and processes. Constraint set (3.2) ensures that all tasks are scheduled only once for all projects and processes. Constraint set (3.3) enforces the precedence relationships amongst tasks within each process and project. Constraint set (3.4) enforces the precedence relationship amongst processes of each project. Constraint set (3.5) determines the maximum number of renewable shared resources per period across all projects during the planning horizon. Constraint set (3.6) determines the maximum number of renewable dedicated resources per period to each project during the planning horizon. Constraint sets (3.7) and (3.8) calculate the weighted tardiness for projects and processes respectively. Constraint sets (3.9) and (3.10) specify the feasible range for decision variables.

6.4.8 Resource Planning Model (Scenario VI)

Sets and indices

V set of projects, $v \in V$

W set of project work groups, $w \in W$

P_v set of processes of project $v, p \in P_v$

J_{vp} set of tasks of process p of project $v, j \in J_{vp}$

p_{vN} last process (indexed with N) of project v

j_{vpL} last task (indexed with L) of process p of project v

j_{vpF} first task (indexed with F) of process p of project v

S_v set of all process precedence relationships of project v

Q_{vp} set of all task precedence relationship of process p of project v

M_{vpj} set of modes for task j of process p of project v

M_w set of modes in project work group w

G set of resource groups, $g \in G$

K_g set of renewal resources for group g, $k \in K_g$

T set of time periods of project, $t \in T$

Assumption: k belongs to one g only. In other words, one person belongs only to one group and cannot be in more than one groups.

Parameters

EE_{vpj} earliest finish time of task j of process p of project v

LL_{vpj} latest finish time of task j of process p of project v

E_{vp} earliest finish time of process p of project v

L_{vp} latest finish time of process p of project v

d_{vpjm} duration of tasks j of process p of project v, operating on mode m

r_{vpjgkm} renewable resource k of group g usage of task j of process p of project v, operating on mode m

dd_v assigned due date for project v

ddd_{vp} due date for the process p of project v

c_v relative weight of project v

c_{vp} relative weight of process p of project v

b_{gk} unit cost of renewable resource k belonging to group g

Decision variables

$$x_{vpjmt} = \begin{cases} 1, \text{if activity } j \text{ of process p of project } v \text{ operating on mode m is finished at period } t \\ 0, \text{otherwise} \end{cases}$$

$TC_v =$ Weighted tardiness cost of project v

$TC_{vp} =$ Weighted tardiness cost of process p of project v

$$y_{vw} = \begin{cases} 1, \text{if project } v \text{ is in the work group } w \\ 0, \text{otherwise} \end{cases}$$

$BR_{gk} =$ Number of renewable resource k of group g per period t

Mathematical model

$$\text{Min z} = \sum_{v \in V} TC_v + \sum_{v \in V} \sum_{p \in P_v} TC_{vp} \tag{4.1}$$

Subject to

$$\sum_{m} \sum_{t=EE_{vpj}}^{LL_{vpj}} x_{vpjmt} = 1 \qquad \forall v \in V, \forall p \in P_v, \forall j \in J_{vp} \tag{4.2}$$

$$\sum_{m}\sum_{t=EE_{vpb}}^{LL_{vpb}}\left(t-d_{vpjm}\right)x_{vpbmt} \;\geq\; \sum_{m}\sum_{t=EE_{vpa}}^{LL_{vpa}} t\,x_{vpamt}\;\forall\,(a,b)\in\; Q_{vp}\text{ and }\forall v\in V \tag{4.3}$$

$$\sum_{m}\sum_{t=EE_{vpj_{vbF}}}^{LL_{vpj_{vbF}}}(t-d_{vpj_{vbF}m})x_{vbj_{vbF}mt} \;\geq \sum_{m}\sum_{t=E_{va}}^{L_{va}} t\,x_{vaj_{vaL}mt}\;\forall\,(a,b)\in\; S_v\text{ and }\forall\in V \tag{4.4}$$

$$\sum_{p\,\in\,P_v}\sum_{j\,\in\,J_{vp}}\sum_{q=\max\{t,EE_{vpj}\}}^{\min\{t+d_{vpjkg}-1,LL_{vpj}\}} r_{vpjgkm}x_{vpjmq}\leq\; BR_{vgk}\;\;\forall\,k\in K_g,\forall\,g\in G, and\,\forall\,t\in T,\;\;\forall v \in V \tag{4.5}$$

$$TC_v\;\geq\;C_v\left(\sum_{m}\sum_{t=E_{vp_{vN}}}^{L_{vp_{vN}}} t\,x_{vp_{vN}j_{vpL}mt}-\;dd_v\right)\quad\forall\,v\in V \tag{4.6}$$

$$TC_{vp}\;\geq\;C_{vp}\left(\sum_{m}\sum_{t=EE_{vpj_{vpL}}}^{LL_{vpj_{vpL}}} t\,x_{vpj_{vpL}mt}-\;ddd_{vp}\right)\quad\forall v\in V,\forall p\in P_v \tag{4.7}$$

$$\sum_{w\,\in\,\mathrm{W}} y_{v,w}=1\quad\forall v\in V \tag{4.8}$$

$$x_{vpjmt}\leq\; y_{v,w}\;\forall v\in V,\forall p\in P_v,\forall j\in J_{vp},\forall m\in M_w,\forall w\in W,\forall\,t\in T \tag{4.9}$$

$$x_{vpjmt}\;\in\{0,1\}\qquad\forall v\in V,\forall p\in P_v,\forall\,j\in J_{vp},\forall\,m\in M_w,\forall\,t\in T \tag{4.10}$$

$$TC_v,TC_{vp},BR_{gk}\in Z^{+}\quad\forall p\in P_v,\text{and }\forall v\in V \tag{4.11}$$

$$y_{vw}\in\{0,1\}\quad\forall v\in V,\text{and }\;\forall w\in W \tag{4.12}$$

The objective function (4.1) minimizes the total weighted tardiness over all projects and processes. Constraint set (4.2) ensures that all tasks are scheduled only once for all projects and processes. Constraint set (4.3) enforces the precedence relationships amongst tasks within each process and project. Constraint set (4.4) enforces the precedence relationship amongst processes of each project. Constraint set (4.5) determines the maximum number of renewable resources per period during the planning horizon. Constraint sets (4.6) and (4.7) calculate the weighted tardiness for projects and processes respectively. Constraint set (4.8) ensures each project is only in one work group. Constraint set (4.9) ensures that each task of a project is assigned a mode that belongs to the work group of that project. Constraint sets (4.10), (4.11), and (4.12) specify the feasible range for decision variables.

6.4.9 Verification of Scenarios I, II, IV, and V

To verify and validate our task and resource planning models, we used the project network example from Elmaghraby (1977) and data from Talbot (1982) with renewable resources only. The data for this example is shown in Appendix D. Talbot's example has 1 project with 7 tasks and the objective function is to minimize project completion time given a fixed number of resources. Talbot uses multiple resources in each mode. Since no work groups and processes have been used in this example, we set the number of work groups and processes to 1 in our models. Task planning model (Scenarios I and II) and resource planning model (Scenarios IV and V) have been tested using Talbot's experiment. Table 11 shows the details of this verification. Scenarios III and VI have not been validated with Talbot's model, since there is no work group feature in Talbot's model to be able to compare the results with.

Table 11: Verification Models Using Talbot (1982)'s Experiment

Purpose of Model	Verification	
Number of Work Group	1	
Number of Projects	1	
Number of Processes	1	
Source Model	Talbot (1982)	
Resource Configuration	Multiple resources in each mode	
	Non-dedicated Resources	Dedicated Resources
Task Planning Model	Fixed (BR_k) - *Scenario I*	Fixed (BR_{vk}) - *Scenario II*
Resource Planning Model	Variable (BR_k) - *Scenario IV*	Variable (BR_{vk}) - *Scenario V*

The results of our validation test are shown in table 12. The verification results match Talbot (1982)'s results (task schedules and objective function).

For task planning (scenario I- non-dedicated resources), the number of resources BR for each resource type (k1, k2, k3, and k4) was fixed and it resulted in project completion time of 9 days. For task planning (scenario II- dedicated resources), the number of allocated resources for each project BR_{vk} was fixed, for each resource type k and it resulted in the project completion time of 9 days.

For resource planning model (scenario IV- non-dedicated resources), the number of resources BR for each resource type k was set as a variable in the model. The model outputted completion time of 9 days with 2, 0, 5, and 7 as the number of resources for resources k1, k2, k3, k4 respectively. This problem example has more than one feasible solution, because we have previously fixed different number of resources (k) in scenario I and II and we got the completion time of 9 days. One reason for having multiple solutions for the number of resources is that we do not consider the costs of resources in this example, which will result in solutions that are economical as well.

Table 12: The Verification Results by Talbot's Example

Purpose of Model	Verification	
Number of Work Group (w)	1	
Number of Projects (v)	1	
Number of Processes (p)	1	
Source Model	Talbot (1982) & Besikci et. al (2015)	
Data	Talbot (1982) Example	
Number of Modes (m)	2	
Number of Resource Types (k)	4	
Resource Configuration	Multiple resources in each mode	
	Non-dedicated Resources	Dedicated Resources
Task Planning Model	**Scenario I**	**Scenario II**
	Input: BR[k=1]=1; BR[k=2]=2; BR[k=3]=6; BR[k=4]=8 and Tasks	Input: BR[v=1,k=1]=1; BR[v=1,k=2]=2; BR[v=1,k=3]=6; BR[v=1,k=4]=8 and Tasks
	Output: Project completion time = 9	Output: Project completion time = 9
Resource Planning Model	**Scenario IV**	**Scenario V**
	Input: Tasks	Input: Tasks
	Output: BR[k1]=2; BR[k=2]=0; BR[k=3]=5; BR[k=4]=7 Project completion time = 9	Output: BR[v=1,k=1]=2; BR[v=1,k=2]=0; BR[v=1,k=3]=5; BR[v=1,k=4]=7, Project completion time = 9

6.5 A Case Study for Resource and Project Planning

This thesis chose to study a real case in a CRO to create a resource and project planning tool. There are many tasks and processes in clinical trials logistics management. But for simplicity, the data for one of the processes (Label Production for Clinical Trials Sample Collection) was retrieved and studied. Labels have information such as patient ID, sample ID, date and time of sample collection, type of assessment, and customized barcodes. The labels are affixed on the sample collection tubes and have important information about the samples. Incorrect information on the labels might cause samples to be mishandled. Therefore, design and production of labels must meet stringent quality standards.

The workflow for label production process is shown in figure 23. For a new protocol, the label process starts with Create Label Data (task 1). However, for the existing protocol, the process does not always start from the beginning, because the label has already been designed and it only needs to be replicated if more labels are needed. For replicating exiting design, the process only recurs from Replicate Label Data (tasks 6). Often, information on the labels needs to be changed because of protocol amendments and the label process should start from the beginning (task 1) to create new label designs or revision of the existing ones. The label process can be broken down into 3 sub-processes: Label Design and Production (task 1 to task 10), Label Design Modification (task 1 to task 5), and Label Reproduction (task 6 to task 10).

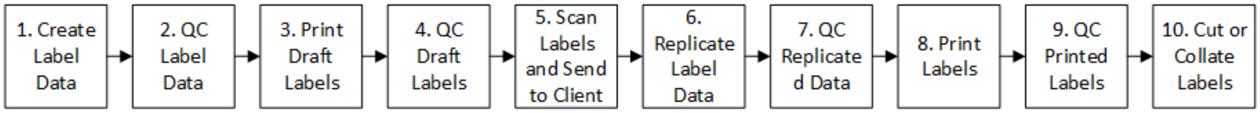


Figure 23: The Workflow for Label Production in Clinical Trials

6.5.1 Case Study Data

The full set of data consists of 70 projects that have been active in a 3-month period. A sample rows of data regarding label process is shown in table 13. The data includes 10 clinical trial projects (protocols) and within each project, there are multiple processes. Each process represents either Label Design and Production (process 1) or Label Design Modification (process 2) or label reproduction (process 3). The same kind of a process might be repeated within the planning horizon. Each process has an arrival date that represents when the request is made, and a due date that represents when that process is needed to be done by. Having arrival date and due date for each process makes this example different than the common RCMPSP examples that have multiple projects with one due date at the end. The objective function in this example is to minimize the total tardiness of all processes.

In this example, within each project, processes are assumed to be in sequence. Each process is given a time window (early finish date and late finish date) to be finished. The earliest finish time is calculated using CPM (Critical Path) forward path. The latest finish time is calculated using the backward path starting from the end of the planning horizon. The tasks within each process is also sequential in this example and each task also has an early finish date and late finish date.

Table 13: Case Study Sample Rows

Project #	Process #	Duration	Arrival Date	Due Date	Early Finish	Late Finish
1	1.1	11	6	19	16	23
1	3.1	5	18	24	22	28
1	3.2	5	23	29	27	33
2	1.1	11	6	19	16	28
2	3.1	5	18	24	22	33
3	3.1	5	14	20	18	18
3	3.2	5	19	25	23	23
3	3.3	5	24	30	28	28
3	3.4	5	29	35	33	33
4	3.1	5	15	21	19	33
5	3.1	5	14	20	18	33
6	1.1	11	9	23	19	33
7	3.1	5	16	22	20	33
8	3.1	5	15	21	19	33
9	3.1	5	16	22	20	23
10	3.1	5	17	23	21	28
10	3.2	5	22	28	26	33

For simplicity, processing times are assumed constant regardless of the size of the job and is shown in table 14. The resources needed to perform each job has also been identified in table 14.

Table 14: Processing Times for Each Task and Resources

	Processing Time (Days)									
Role	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10
PM	2				1					1
QC		1		1			1		1	
OA			1			1		1		1

6.5.2 Results

6.5.2.1 Inter-Scenarios Comparisons

The results of our experiments that were explained in sub-section 6.4.4 are shown in this section in table 15. The LP formulation for each scenario has been solved using GLPK and Gurobi optimization solvers. Each scenario is compared with its adjacent scenario in table 15. To be able to compare scenarios, the number of projects has been fixed to 10 (17 processes total) for all scenarios and number of resources has been fixed to 6 for scenarios I, II, and III. The weight of tardiness is set to 1 for all the scenarios.

The source model is from Talbot (1982) and Besikci et al. (2015) with minor modifications and extensions that have been explained in section 6.4. The data is from a clinical trials case study that was explained in section 7.2.

The number of modes used in scenarios I and IV is 1, while the number of modes used in scenarios II, III, V, and VI is 2. The mode in all scenarios represents one or more resource group (g) with a defined number of resources needed and the time it takes to process the task. In other words, a mode represents a unique combination of resources and the rate at which that task will be processed in that mode.

In scenarios I and IV, only one mode (resource group) is allocated to every task, because it is assumed that all resources within each resource group work at the same speed. Therefore, the processing times that are in table 14 were used for mode 1.

In scenario II, two modes were used that each consists of three resource types. For example, m1 consists of PM1, QC1, OA2, and m2 consists of PM2, QC2, and OA2. We have set the BR (number of resources) equal to 1 to make each resource type as a unique resource from that mode. For example, PM1 represent an individual project manager in scenario II. Scenario V is the same as scenario II, except that BR is a variable. In this scenario BR can represent one or many people depending on the work demand. For example, the results might end up having 2 resources for PM1, which implies that 1 project manager is not enough for that role. The number of modes depends on the number of people in each department. For example, if we have three project managers, then we need three modes.

However, in scenario III and VI, 2 modes are allocated to each task, because we have 2 work groups that consist of different resources. Mode 1 and 2, belong to work group 1and 2 respectively. The processing times for each work group is the same as every other scenario that comes from table 14. Each resource group is assigned a subscript that shows the work group and mode that it belongs to. For example, PM_1 means project managers group from work group 1 or mode 1. Each inter-scenario comparison will be explained in the following sub-sections.

Table 15: Inter-Scenario Comparisons

Purpose of Model	Planning Clinical Trial Projects (Baseline)		
Number of Work Groups (w)	1		2
Number of Projects (v)	10		
Number of Processes (p)	17		
Source Model	Talbot (1982) & Besikci et al. (2015)		
Data	Clinical Trials Example		
Number of Modes (m)	1	2	2
Resource Configuration	One or more resource types in each mode		
	Non-dedicated Resources	Dedicated Resources	Resources in Work Groups
Task Planning Model	**Scenario I**	**Scenario II**	**Scenario III**

	Input:
	10 Projects
	6 Resources
	BR[PM] = 2 , BR[QC] = 2 , BR[OA] = 2
		Input:
		10 Projects
		6 Resources
		BR[PM1] = 1, BR[PM2]=1
		BR[QC1]=1, BR[QC2]=1
		BR[OA1]=1, BR[OA2]=1
		Projects of PM1={1,2,3,4,5}
		Project of PM2={6,7,8,9,10}
		Projects of QC1={1,2,3,4,5}
		Project of QC2={6,7,8,9,10}
		Projects of OA1={1,2,3,4,5}
		Project of OA2={6,7,8,9,10}
			Input:
			10 Projects
			6 Resources
			w1={PM1, QC1, OA1}, w2={PM2, QC2, OA2}
			BR[PM1] = 1 , BR[QC1] = 1 , BR[OA1] = 1
			BR[PM2] = 1 , BR[QC2] = 1 , BR[OA2] = 1

	Output: Total Tardiness = 66 days	Output: Total Tardiness = 96 days	Output: Projects of work group 1={1,5,6,8,10} Projects of work group 2 ={2,3,4,7,9} Total Tardiness = 68 days
Resource Planning Model	**Scenario IV**	**Scenario V**	**Scenario VI**

	Input:
	10 Projects
		Input:
		10 Projects
		6 Type of Resources
		Projects of PM1={1,2,3,4,5}
		Project of PM2={6,7,8,9,10}
		Projects of QC1={1,2,3,4,5}
		Project of QC2={6,7,8,9,10}
		Projects of OA1={1,2,3,4,5}
		Project of OA2={6,7,8,9,10}
			Input:
			10 Projects
			2 Work Groups

	Output:
	BR[PM] = 2, BR[QC] = 3, BR[OA] = 4
	Total Tardiness = 0
		Output:
		BR[PM1] = 3 , BR[PM2] = 2
		BR[QC1] = 2 , BR[QC2] = 3
		BR[OA1] = 3, BR[OA2] = 2
		Total Tardiness = 0
			Output:
			BR[PM1] = 2 , BR[QC1] = 4 , BR[OA1] = 3
			BR[PM2] = 1 , BR[QC2] = 2 , BR[OA2] = 3
			Projects of work group 1={2,3,5,6,9,10}
			Projects of work group 2 ={1,4,7,8}
			Total Tardiness = 0

6.5.2.1.1 I vs IV

In scenario I, 6 shared resources have been assigned (2 PM, 2 QC, 2 OA), the system output task schedule with 66 days tardiness. In scenario IV, 10 projects were given to the LP to minimize the total tardiness of the projects by finding the optimal number of resources for each department. The LP output total of 9 resources (2 PM, 3 QC, 4 OA).

6.5.2.1.2 I vs II

In Scenario I, 6 shared resources were used, however, in scenario II, 6 dedicated resources were used. For example, in scenario II, resource type PM1 is assigned to projects 1,2,3,4, and 5, while PM2 is assigned to projects 6,7,8,9, and10. The output of scenario II shows that the total tardiness increased to 96 days compared to scenario I that has tardiness of 66 days. This is because in scenario I, resources are shared across all projects, so each project has access to 2 resource types. However, in scenario II, each project has access to only 1 resource type because of the dedication policy.

6.5.2.1.3 II vs V

In scenarios II and V, resource types are dedicated to projects. The only difference is that, scenario V has a variable number of resources for each resource type, while scenario II has a fixed number of resources for each resource type. In scenario II, 6 resources have been used, the output came out to be 96 days of tardiness. Same tasks were given to scenario V to minimize the tardiness, the output showed an increase of resources to a total of 15 resources (PM1=3, QC1=2, OA1=3, PM2=2, QC2=3, OA2=2).

6.5.2.1.4 II vs III

In scenario III, 2 work groups have been created that each includes 3 resource types, each resource type has given only one resource (only one person in each resource's type, a total of 6 resources). In scenario II, there are 6 resources that have been dedicated to one to many projects. In scenario III, resources are being assigned to a work group by the LP. The results show less tardiness for scenario III (68 days) than scenario II (96 days). One reason for the better performance of scenario III is that the assignment of projects to the work groups is done optimally by the LP rather than manual assignments of projects by management.

6.5.2.1.5 III vs VI

In scenario III, number of resources for each work group has been fixed (3 resources for work group 1 and 3 resources for work group 2), while in scenario VI, number of resources are variables. Scenario III, output tardiness of 68 days with 6 resources, while scenario IV output tardiness of 0 with a total of 15 resources.

The second difference is in the assignment of projects to each work group in each scenario. In scenario III, 5 projects have been assigned to each work group, while in scenario VI, work group 1 received 6 projects and work group 2 received 4 projects.

6.5.2.1.6 IV vs V

Scenario IV determines the optimal number of resources for a shared resource structure, while scenario V determines the optimal number of resources for a dedicated resource structure. The results show that scenario IV performs better (9 resources) than scenario V (15 resources) with respect to the total number of resources. The reason for that is that resources are being shared in scenario IV and fewer overall resources are needed as opposed to the dedicated resource model.

6.5.2.1.7 V vs VI

Scenario V and VI both showed 0 tardiness and 15 total number of resources. However, the breakdown of the number of resources for each resource type is different in each scenario. For example, in scenario V, PM1 has 3 and PM2 has 2 resources, while in scenario VI, PM1 has 2 and PM2 has 1 resource.

6.5.2.2 Intra-Scenario Comparisons

In this subsection, each scenario will be given two datasets, baseline and second example, and results will be compared within that scenario. The baseline is coming from table 15 that was explained in subsection 6.5.3.2. The second example, one of the parameters will be changed to see the behavior of the scenario, for example, the number of resources or projects will be changed.

6.5.2.2.1 Scenario I (6 Resources vs. 9 Resources)

The number of resources has been increased from 6 in the baseline to 9 in the second example scenario. 9 resources have been evenly distributed to each resource type (3 PMs, 3 QCs, 3 OAs). The output tardiness decreased to 8 days from 66 days. This validates the behavior of this scenario, because as the number of resources increases, tardiness must decrease if all other parameters such as the number of projects stay constant. The details of this comparison are in table 16.

Table 16: Intra-Scenario I Comparisons

Scenario I (Baseline)	Scenario I (Second Example)
Input: 10 Projects 6 Resources BR[PM] = 2 , BR[QC] = 2 , BR[OA] = 2	Input: 10 Projects 9 Resources BR[PM] = 3 , BR[QC] = 3 , BR[OA] = 3
Output: Total Tardiness = 66 days	Output: Total Tardiness = 8 days

6.5.2.2.2 Scenario II (6 Resources vs 30 Resources)

The number of resources has been increased from 6 in the baseline to 30 in the second example scenario. Each project has been given only 1 dedicated resource of each type (for example for project 1, $BR_{1, PM} = BR_{1, QC} = BR_{1, OA}= 1$) that cannot be shared with other projects. The tardiness has decreased from 96 days in the baseline to 0 in the second example. The results validate the behavior of this scenario, because in our dataset, processes are in sequence in each project. Therefore, each project requires only 1 dedicated resource of each resource type at a time to finish its processes by their due dates. The details of this comparison are in table 17.

Table 17: Intra-Scenario II Comparisons

Scenario II (Baseline)	Scenario II (Second Example)

Input:
10 Projects
6 Resources
BR[PM1] = 1, BR[PM2]=1
BR[QC1]=1, BR[QC2]=1
BR[OA1]=1, BR[OA2]=1
Projects of PM1={1,2,3,4,5}
Project of PM2={6,7,8,9,10}
Projects of QC1={1,2,3,4,5}
Project of QC2={6,7,8,9,10}
Projects of OA1={1,2,3,4,5}
Project of OA2={6,7,8,9,10}
	Input:
	10 Projects
	30 Resources
	BR[v,g] = 1 for every g and v

Output: Total Tardiness = 96 days	Output: Total Tardiness = 0

6.5.2.2.3 Scenario III (6 Resources vs. 18 Resources)

The number of resources has been increased from 6 in the baseline to 18 in the second example scenario. 18 resources have been distributed evenly to each work group in the second example. The results show that the second example has a lower tardiness (10 days) than the baseline (68 days). Projects have been distributed to all three work groups in the second example. This implies that as the number of resources increases, the tardiness decreases. The details of this comparison are in table 18.

Table 18: Intra-Scenario III Comparisons

Scenario III (Baseline)	Scenario III (Second Example)

Input:
10 Projects
6 Resources
w1={PM1, QC1, OA1}, w2={PM2, QC2, OA2}
BR[PM1] = 1 , BR[QC1] = 1 , BR[OA1] = 1
BR[PM2] = 1 , BR[QC2] = 1 , BR[OA2] = 1
	Input:
	10 Projects
	18 Resources
	w1={PM1, QC1, OA1},
	w2={PM2, QC2, OA2},
	w3= {PM3, QC3, OA3}
	BR[PM1] = 2 , BR[QC1] = 2 , BR[OA1] = 2
	BR[PM2] = 2 , BR[QC2] = 2 , BR[OA2] = 2
	BR[PM3] = 2 , BR[QC3] = 2 , BR[OA3] = 2

Output: Total Tardiness = 68 days Projects of work group 1={1,5,6,8,10} Projects of work group 2 ={2,3,4,7,9}	Output: Total Tardiness = 10 days Projects of work group 1={1,7,8} Projects of work group 2 ={4,5,6,10} Projects of work group 3 = {2,3,9}

6.5.2.2.4 Scenario IV (10 Projects vs. 20 Projects)

The number of projects has been changed from 10 in the baseline to 20 in the second example. The output shows the number of resources has been increased from 9 to 20 resources. This implies that as the number of projects increases, the number of resources increases. The details of this comparison are in table 19.

Table 19: Intra-Scenario IV Comparisons

Scenario IV (Baseline)	Scenario IV (Second Example)
Input: 10 Projects	Input: 20 Projects
Output: BR[PM] = 2, BR[QC] = 3, BR[OA] = 4 Total Tardiness = 0	Output: BR[PM] = 3, BR[QC] = 7, BR[OA] = 10 Total Tardiness = 0

6.5.2.2.5 Scenario V (6 Type of Resources vs. 3 Type of Resources)

This scenario is very similar to scenario II with the only difference that the number of resources is variable in this scenario. In the baseline model, one resource is assigned to more than one project, while in the second example each resource is assigned to one and only one project. The number of resource types has been decreased from 6 in the baseline to 3 in the second example, because in the second example each project will be assigned only 1 dedicated resource of each type and there is no need to differentiate resources. The output of the second example shows an assignment of 1 resource to every resource type and project (total of 30 resources), while the baseline has a total of 15 resources. This implies that more resource dedication to projects requires more resources. The details of this comparison are in table 20.

Table 20: Intra-Scenario V Comparisons

Scenario V (Baseline)	Scenario V (Second Example)

Input:
10 Projects
6 Type of Resources
Projects of PM1={1,2,3,4,5}
Project of PM2={6,7,8,9,10}
Projects of QC1={1,2,3,4,5}
Project of QC2={6,7,8,9,10}
Projects of OA1={1,2,3,4,5}
Project of OA2={6,7,8,9,10}
	Input:
	10 Projects
	3 Type of Resources (PM,QC,OA)

Output:
BR[PM1] = 3 , BR[PM2] = 2
BR[QC1] = 2 , BR[QC2] = 3
BR[OA1] = 3, BR[OA2] = 2
Total Tardiness = 0
	Output:
	BR[v,g] = 1 for every g and v
	Total Tardiness = 0

6.5.2.2.6 Scenario VI (2 Work Groups vs. 3 Work Groups)

The number of work groups has been increased from 2 in the baseline to 3 in the second example scenario. The output of the second example shows 17 resources while the baseline has 15 resources. This implies that more work groups require more resources to finish all the projects with 0 tardiness. The details of this comparison are in table 21.

Table 21: Intra-Scenario VI Comparisons

Scenario VI (Baseline)	Scenario VI (Second Example)
Input: 10 Projects 2 Work Groups	Input: 10 Projects 3 Work Groups
Output: BR[PM1] = 2 , BR[QC1] = 4 , BR[OA1] = 3 BR[PM2] = 1 , BR[QC2] = 2 , BR[OA2] = 3 Projects of work group 1={2,3,5,6,9,10} Projects of work group 2 ={1,4,7,8} Total Tardiness = 0	Output: BR[PM1] = 1 , BR[QC1] = 2 , BR[OA1] = 1 BR[PM2] = 2 , BR[QC2] = 3 , BR[OA2] = 3 BR[PM3] = 1 , BR[QC3] = 2 , BR[OA3] = 2 Projects of work group 1={1,7} Projects of work group 2 ={2,3,6,8,10} Projects of work group 3 = {4,5,9} Total Tardiness = 0

6.5.2.3 CPU Times for Resource and Task Planning Models

The full set of data for the label process consists of 70 projects. Table 22 shows the solution time for the case with 70 projects for scenarios IV, V, and VI. Since scenarios I, II, and III take the number of resources as input, a similar comparison for these scenarios is not practical. In fact, for the 70-projects problem using only 6 resources, as in Table 15, the solver could not even find a feasible solution in 30 minutes. This is because the resource level needed to finish 70 projects is much higher.

The LP model was solved using Gurobi optimizer version 8.1.0. The computer that was used has the following specifications:

- Windows 10 Home.
- A 64-bit operating system, x64-based processor.
- Intel Core i7-6700 (3.4 GHz) processor.
- Installed RAM 16.0 GB.

Table 22: CPU Times for Scenarios IV, V, and VI for 70 Projects

Scenario	Number of Projects	CPU Time (Seconds)	MIP Gap
IV	70	0.50	0.0001
V	70	0.70	0.0001
VI	70	5.85	0.0001

6.5.2.4 Costs Analysis for Scenario IV

Scenarios IV, V, and VI were chosen to perform an analysis on the effects of resource cost and tardiness costs on the number of resources and total tardiness of projects. These scenarios minimize the tardiness costs by changing the number of resources and tardiness of each project or process. Scenarios I, II, and III were not considered for this analysis, because the number of resources is fixed in these scenarios.

For this analysis, resource costs were also added to the objective function of each scenario to minimize the total resource cost as well as the total tardiness of projects. The new objective function is as follows:

$$\text{Min z} = \sum_{v \in V} TC_v \times c_v + \sum_{v \in V} \sum_{p \in P_v} TC_{vp} \times c_{v,p} + \sum_{g \in G} \sum_{k \in K_g} BR_{g,k} \times b_{g,k}$$

Where: c_v = unit cost of tardiness for projects, $c_{v,p}$ = unit cost of tardiness for processes, and $b_{g,k}$ = unit cost of each resource k that belongs to group g.

In practice, costs of tardiness in logistics management of clinical trials can be found by identifying the consequences of delays in the logistics processes. For example, delays in shipping of the collection kits to the site will cause to miss a sample collection, or delays in coordinating of a sample shipment with a site will delay the sample shipment to the lab, and in a worst-case scenario, unsatisfactory services will cause a patient to withdraw from a clinical trial.

In this analysis, $\sum_{v \in V} TC_v \times c_v$ is not used, because our clinical trials example data do not include due dates on projects. The $c_{v,p}$ value was fixed to 1 as the tardiness cost for processes. The $b_{g,k}$ value is set at 1.5 for PM, 1 for QC, and 1 for OA. $b_{g,k}$ was multiplied by {1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100} during each LP run. The output tardiness and the number of resources were recorded for each run and are displayed in figures 24, 25, and 26 for scenarios IV, V, and VI respectively.

The x-axis in figures 24, 25, and 26 represents the ratio of resource cost to tardiness cost, which starts from 1 and is incremented by 10. The left y-axis represents tardiness and the right y-axis represents the number of resources.

For example, for scenario IV that is shown in figure 24, at ratios between 0 and less than or equal to 1, the number of resources stays constant at 9 and the total tardiness stays at 0. At the ratio of 10, the number of resources decreases to 6 and tardiness increase to 26. The number of resources and tardiness stays unchanged at ratios of 20, 30, and 40. At the ratio of 50, the number of resources decreases to 5 and tardiness increases to 72. The number of resources and tardiness stays unchanged at ratios of 60, 70, and 80. At the ratio of 90, the number of resources decreases to 4 and tardiness increases to 153. At the ratio of 100, the number of resources decreases to its lowest level that is 3 and tardiness increases to 253.

Also, in figure 25 and 26, as the ratio of resource cost to tardiness cost increases, the number of resources decreases and tardiness increases. This is because the resources become so costly that the LP tend to delay the tasks instead of using more resources. Inversely, as the ratio of resource cost to tardiness cost decreases, the number of resources increases and tardiness decreases. In this case, the LP uses more human resources than time resources (tardiness).

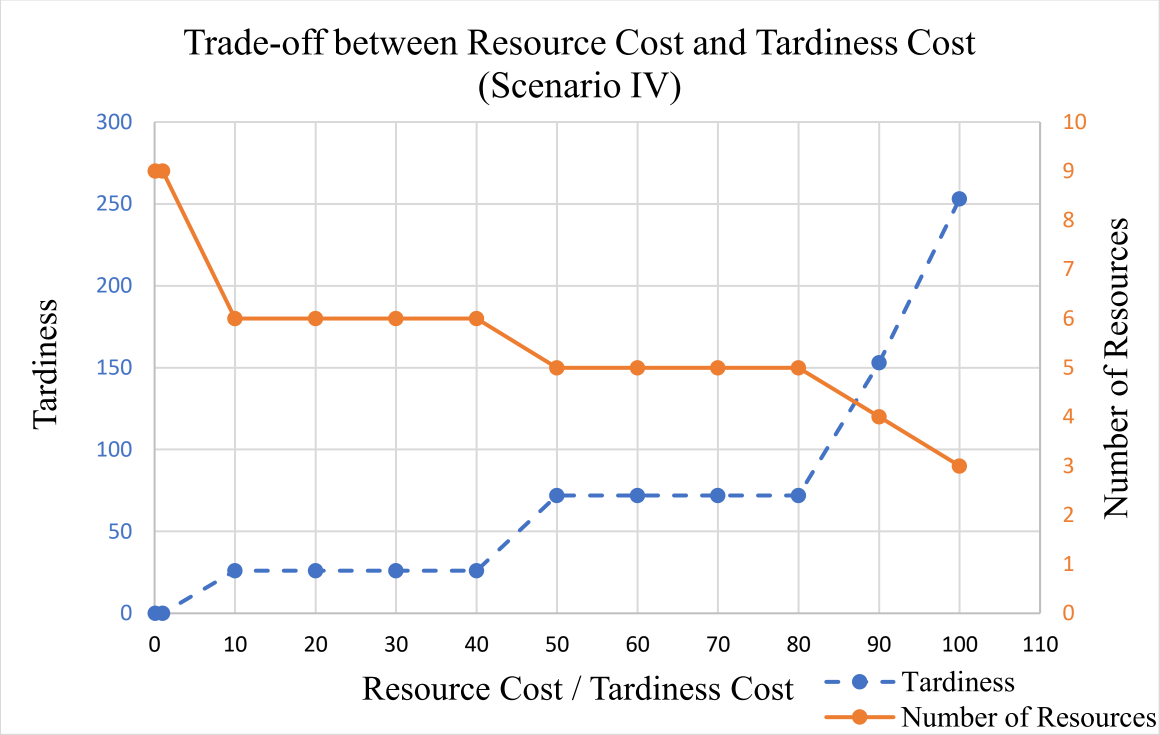


Figure 24: Trade-off between Resource and Tardiness Costs for Scenario IV

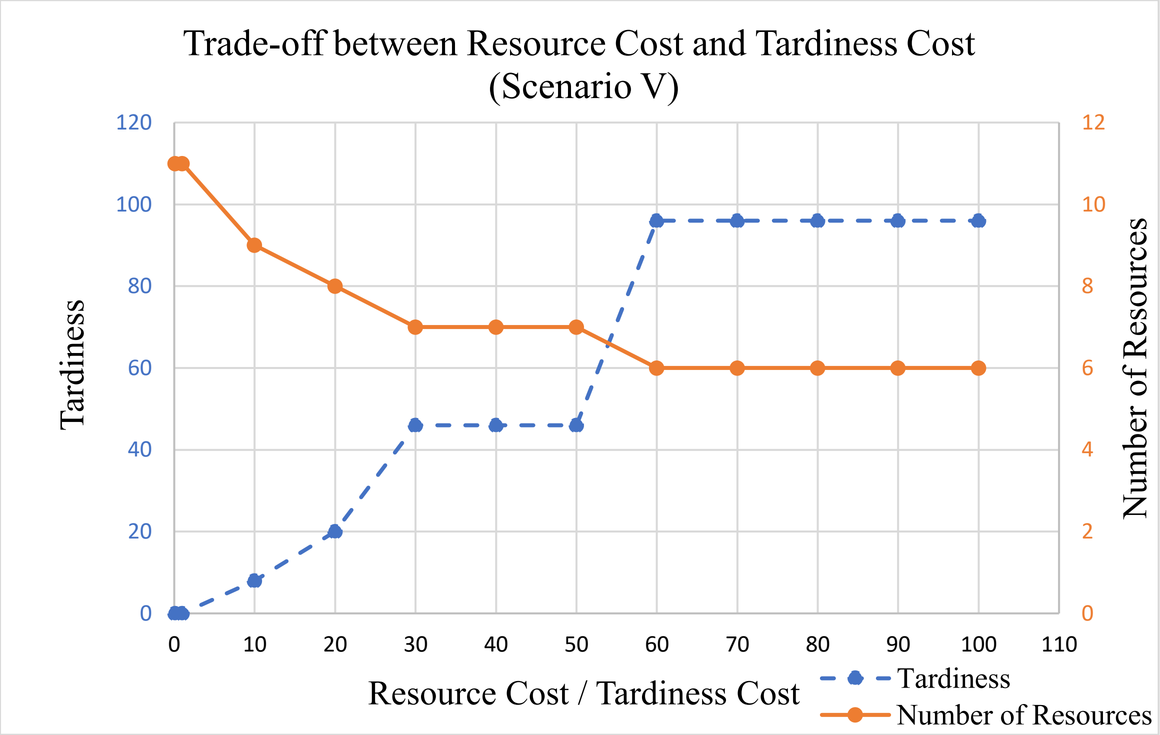


Figure 25: Trade-off between Resource and Tardiness Costs for Scenario V

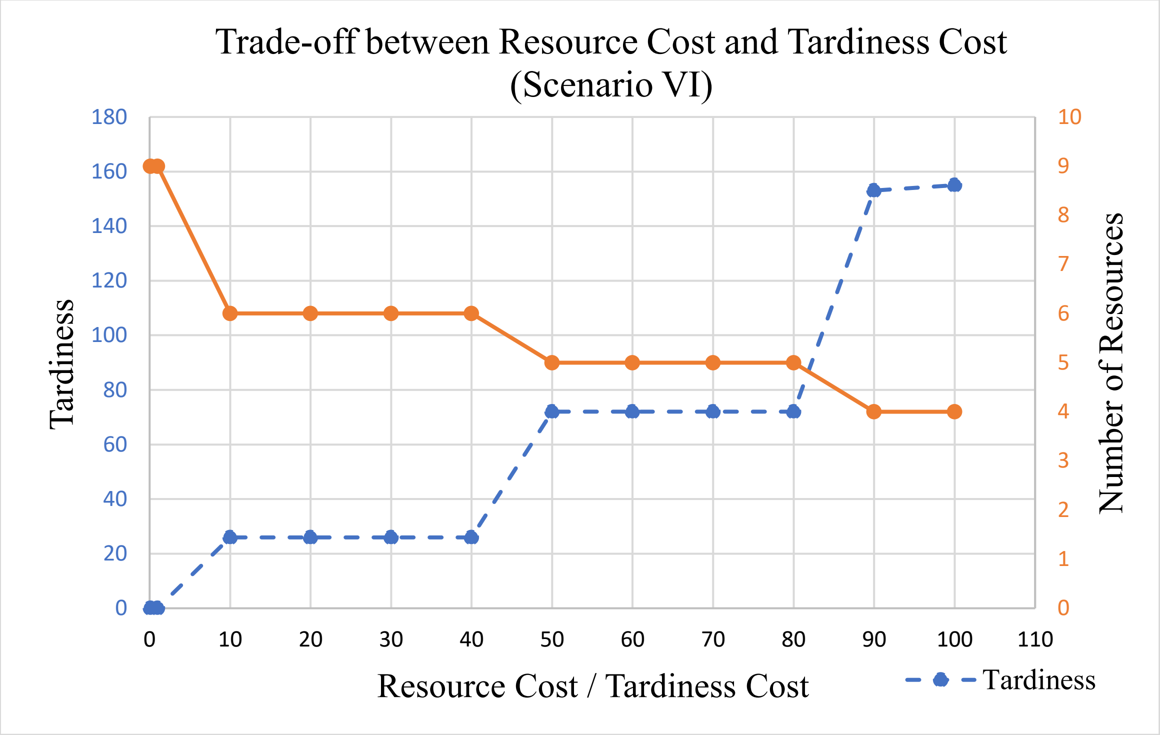


Figure 26: Trade-off between Resource and Tardiness Costs for Scenario VI

Chapter 7: Conclusion

Clinical trials are the last stage of drug discovery, where new medical treatments are tested on patients for their safety and effectiveness. Clinical trials are important for finding better treatments and constantly improving our health worldwide. In addition, from an economic point of view, clinical trials employ many highly skilled workers such as medical researchers, medical doctors, nurses, etc.

Clinical trials require complex coordination of tasks and involve many resources that need to be planned and managed carefully. Currently, logistics management of clinical trials is seen to be inefficient. Data are hand-written on forms; computer programs and databases are disparate. This makes it difficult to streamline the flow of data from one computer program to the other or from one organization to the other. Disparate data sources also increase the chance of errors when data are manually moved from one data source to other. Resources cannot be planned properly when all the data are not available in real time. In clinical trials, many organizations need to be collaborating to solve day to day problems, and the proper tool is not used for that yet. Above all the issues, the clinical trials industry has stringent quality requirements to meet the FDA standards.

The literature includes conceptual solutions on managing clinical trials or articles focused on areas other than logistics management of the clinical trials. This thesis focused on the issues in the logistics management of the clinical trials and proposed a framework using contemporary software tools and technology.

The main components of this framework from top to bottom are web portal, intelligent agents, BPMS, project card, external applications, document repository, data repository, and analytics. web portal facilitates the use of BPMS for the users and customizes forms, reports, and dashboards for each stakeholder. Intelligent agents work in parallel with the web portal or the human users to automate processes and decrease the burden of work from human users to some extent. BPMS holds all of the business processes for clinical trial logistics management and orchestrates the tasks among respective human resources. Project card maintains all of the data regarding a clinical trial and communicates that with the BPMS and other applications if needed. External applications are the software applications that are available for a particular purpose such as label design software and courier's applications that need to communicate with the project card and BPMS. Document repository store the entire business-related documents for the stakeholders involves in the clinical trial industry, these documents are non-project related documents and are used for accounting purposes. The project related documents will be stored in the project card. The data repository is similar to the document repository, but it maintains the business data instead of documents. There is an analytics component, which enables the stakeholders to analyze the data to identify the areas of improvements or measure the performance. Lastly, the security component is responsible for privacy and security of the entire system, since the clinical trials industry requires a high degree of network security.

This management framework can be used for service industries such as clinical trial logistics management where their main works are done in an office setting and with software applications and have many stakeholders and outsourced companies across the globe where communication and concise collaboration is a must. Clinical trial logistics management was studied as a case study in this research to focus on a real problem than an abstract one. This framework brings an integrative computer platform to exchange data among computers faster and remove many redundant works and possible errors. Most of the tasks can be automated using intelligent software agents. It also facilitates easy and efficient collaboration across organizations. Data can be analyzed using the analytics component to find meaningful data and eventually improve the workflows to better manage the projects.

In addition to the main framework, this thesis improves resource and project planning in clinical trial logistics management by incorporating a linear programming planning tool into the main framework. This tool works with the project card and resource card to obtain its input data and outputs tasks and resource plan into the BPMS for execution. This planning tool also contributes to RCMPSP by incorporating several different resource scenarios such as a shared pool of resources, dedicated resources for each project, or several work groups with dedicated resources. Additional features of this tool that are tailored to the logistics management of clinical trials are multiple due dates for processes, resource familiarity factors with tasks and projects, and an objective function to minimize the total tardiness of processes and projects.

In this tool, there are two main LP models, a task planning LP and a resource planning LP. Task planning LP schedules tasks and resources based on the fixed number of resources, while the resource planning LP finds the optimal number of resources to finish all tasks without any tardiness.

Three different task planning LP scenarios have been created that are task planning with shared resources, dedicated resources, and work groups. Among these three scenarios, task planning with shared resources had the least total tardiness. In addition, work group resource scenario performed significantly better than the dedicated resource scenario under tardiness criterion.

Also, three resource planning scenarios have been created that are resource planning with shared resources, dedicated resources, and work groups. Resource planning with shared resources scenario output the least number of resources amongst all. Resources planning scenarios with dedicated and work group structure output the same total number of resources. Although the total number of resources across all departments is the same in the dedicated and work group scenarios, the number of resources allocated to each department (PM, QC, OA) is different in each scenario. For example, a total of 5 project managers are assigned in the dedicated scenario, while a total of 3 project managers are assigned in the work group scenario.

In addition to the inter-scenario comparisons, intra-scenario comparisons were performed for each scenario to observe and test its behavior. Each scenario was given a different dataset and the results were compared with the baseline scenario. The findings validated the behavior of the resource and project planning model.

The CPU times for resource and project planning scenarios that consisted of 70 projects have shown a very efficient solving performance. In addition, the behavior of each scenario was analyzed by incorporating the resource and tardiness costs into the LP model. The results show, as the ratio of the resource cost to the tardiness costs increases, the total number of resources increases and the total tardiness increases. This resource and project planning tool will help managers to plan clinical trial logistics projects more accurately and efficiently.

The main framework and the resource and project planning tool can be further improved by adding more components and features to it. Some of these components are mentioned in the flowing sub-section.

7.1 Future Research

Existing components of this framework can be improved, modified, and new components can be added to it to better fit an organization in the clinical trial industry, the whole clinical trial industry, or any similar industry of interest. Some of the future research directions on this framework are as follows:

Web Portal

The conceptual requirements of the web portal were briefly mentioned in this framework, and it was mostly designed for the clinical trial industry. More requirements and features can be added to the conversation and notification tool, reports, and forms to better suit the organization and industry of interest.

Project Card

This framework proposed some conceptual requirements for the project card. Requirements can be modified to better serve the organization and industry of interest. In addition, the software application for it can be developed.

Semantic Web

Also, semantic web can be used as the architecture for the data repository component of this platform to facilitate more automation (machine-readable data) and data analysis.

Servitization

Servitization concept can be used in clinical trials logistics management to design the services and products that communicate with each other. Servitization can be used in the design of kits and labels to automate the tracking of sample shipments. Examples of this would be using sensors on the labels of kits and track them using the Internet of Things.

Multi-Agent System

Only the conceptual requirements of the intelligent multi-agent system were proposed in this framework. This component can be built in a software agent platform.

Resource and Project Planning Tool

This framework is a foundation for developing many more computer applications on top of it. One application that can very beneficial to clinical trials logistics provider is a resource and project planning tool that was designed in this thesis. Further development to this tool can be done in future by incorporating more management objectives and scenarios. Some examples are as follows:

- Incorporating autonomy factors into the model to give human resources autonomy to some extent. This can be done by scheduling all the tasks but give choices to employees to start earlier or to sequence their tasks in a way that they like as long as the deadlines are met.
- Investigating what if analysis scenarios, for example, to see the effect of reduced processing times on the company's capacity.
- Ensuring that employees are cross-trained in different areas to learn about the entire business and be able to function when more resources are needed for a specific task.
- Growth path in the long term for employees. In other words, ensuring that employees are trained towards higher responsibilities and positions such as project management in the future if they prefer. Developing skills such as (analytical, project management, problem-solving skills, people skills, active listening, communication, system knowledge). For example, employees should complete training for all operations and eventually become project managers.
- Meet other business objectives such as handling more projects, improving customer satisfaction and service quality, and increasing revenue.

Longitudinal Study

A longitudinal study can be performed to test the applicability of this framework in industries. This can be done by implementing this framework in a company in the clinical trial industry or similar industries and surveying users' opinions on the areas of improvement.

REFERENCES

Abedi, M. (2016). Integrated collaborative tools for precast supply chain management. *Scientia Iranica, 23*(2), 429-448. doi:10.24200/sci.2016.2129

Al Nasseri, H. A., Widen, K., & Aulin, R. (2016). A taxonomy of planning and scheduling methods to support their more efficient use in construction project management. *Journal of Engineering, Design and Technology, 14*(3), 580-601. doi:10.1108/JEDT-11-2013-0078

Allison, M. (2012). Reinventing clinical trials. *Nature Biotechnology, 30*(1), 41. doi:10.1038/nbt.2083

Antonia Azzini, Paolo CeravoloB, Ernesto Damiani, Nello Scarabottolo, Francesco Zavatarelli, Claudia Vicari, & Vincenzo Savarino. (2015). *Heterogeneous business process management: A metamodel-based approach* doi:10.1007/978-3-319-21009-4

Bădică, C., Budimac, Z., Burkhard, H., & Ivanovic, M. (2011). Software agents: Languages, tools, platforms. *Computer Science and Information Systems, 8*(2), 255-298. doi:10.2298/CSIS110214013B

Baldwin, C. Y., & Clark, K. B. (1997). Managing in an age of modularity. *Harvard Business Review, 75*(5), 84-93. Retrieved from http://ezproxy.library.dal.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=9709112720&site=ehost-live

Baumgraß, A., Botezatu, M., Di Ciccio, C., Dijkman, R., Grefen, P., Hewelt, M., . . . Völzer, H. (2016). Towards a methodology for the engineering of event-driven process applications. Paper presented at the *Business Process Management Workshops,* 501-514.

Benjamins, J. W., Hummel, Y. M., Busman, J. P., Riepma, F., Dorhout, B., & Van Melle, J. P. (2016). Design of an electronic upload and reporting system aimed at corelab tasks and responsibilities in multi-center clinical trials. Paper presented at the *43rd Computing in Cardiology Conference, CinC 2016, September 11, 2016 - September 14, , 43* 145-148.

Besana, P., Cuggia, M., Zekri, O., Bourde, A., & Burgun, A. (2010). Using semantic web technologies for clinical trial recruitment. Paper presented at the *International Semantic Web Conference,* 34-49.

Besikci, U., Bilge, U., & Ulusoy, G. (2015). Multi-mode resource constrained multi-project scheduling and resource portfolio problem. *European Journal of Operational Research, 240*(1), 22-31. doi:10.1016/j.ejor.2014.06.025

Bratton, T. W., Mentzer, J. T., Foggin, J. H., Quinn, F. J., & Golicic, S. L. (2000). Supply chain collaboration: The enablers, impediments, and benefits. Paper presented at the *Council of Logistics Management Fall MeetingCouncil of Logistics Management.*

Bryde, D. J., & Joby, R. (2007). Product-based planning: The importance of project and project management deliverables in the management of clinical trials. *R&D Management, 37*(4), 363-377. doi:10.1111/j.1467-9310.2007.00482.x

Chasanidou, D., Elvesaeter, B., & Berre, A. (2016). Enabling team collaboration with task management tools. Paper presented at the *12th International Symposium on Open Collaboration, OpenSym 2016, August 17, 2016 - August 19,* Google; Wikimedia Foundation. doi:10.1145/2957792.2957799 Retrieved from http://dx.doi.org.ezproxy.library.dal.ca/10.1145/2957792.2957799

Chow, S. (2011). Sample size calculations for clinical trials. *Wiley Interdisciplinary Reviews: Computational Statistics, 3*(5), 414-427. doi:10.1002/wics.155

Columbus, L. (2014). Five strategies for improving aerospace supply chain quality management and performance. *SAE Int.J.Aerosp., 7*(1), 142-145. doi:10.4271/2014-01-2231

Do Carmo, W. C., & Albuquerque, A. B. (2014). Project management suported by business process management: A case study in a brazilian justice organization. Paper presented at the *9th International Conference on the Quality of Information and Communications Technology, QUATIC 2014, September 23, 2014 - September 26,* 236-241. doi:10.1109/QUATIC.2014.39 Retrieved from http://dx.doi.org.ezproxy.library.dal.ca/10.1109/QUATIC.2014.39

Dávideková, M., & Hvorecký, J. (2017). Collaboration tools for virtual teams in terms of the SECI model. Paper presented at the *Interactive Collaborative Learning,* 97-111.

DiMasi, J. A., Hansen, R. W., & Grabowski, H. G. (2003). The price of innovation: New estimates of drug development costs. *Journal of Health Economics, 22*(2), 151-185. doi:10.1016/S0167-6296(02)00126-1

Dridi, A., Tissaoui, A., & Sassi, S. (2015). The medical project management (MPM) system. Paper presented at the *2015 Global Summit on Computer & Information Technology (GSCIT),*1-6. doi:10.1109/GSCIT.2015.7353336 Retrieved from http://ieeexplore.ieee.org/document/7353336

Elmaghraby, S. E. (1977). *Activity networks : Project planning and control by network models*. New York : Wiley.

Faghihi, V., Nejat, A., Reinschmidt, K., & Kang, J. (2015). Automation in construction scheduling: A review of the literature. *The International Journal of Advanced Manufacturing Technology, 81*(9), 1845-1856. doi:10.1007/s00170-015-7339-0

Farrell, B., Kenyon, S., & Shakur, H. (2010). Managing clinical trials. *Trials, 11*(1), 78. doi:10.1186/1745-6215-11-78

Francis, D., Roberts, I., Elbourne, D. R., Shakur, H., Knight, R. C., Garcia, J., . . . Campbell, M. K. (2007). Marketing and clinical trials: A case study. *Trials, 8*(1), 37. doi:10.1186/1745-6215-8-37

Getz, K. A. (2011). Protocol Amendments: a Costly Solution. Retrieved from: http://www.appliedclinicaltrialsonline.com/protocol-amendments-costly-solution?id=&sk=&date=&&pageID=1

Goodarzynejad, H., & Babamahmoodi, A. (2015). Project management of randomized clinical trials: A narrative review. *Iranian Red Crescent Medical Journal, 17*(8), e11602. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/26430517

Hammouch, H., Medromi, H., & Sayouti, A. (2015). Toward an intelligent system for project management based on the multi agents systems. Paper presented at the *10th International Conference on Intelligent Systems: Theories and Applications, SITA 2015, October 20, 2015 - October 21,* doi:10.1109/SITA.2015.7358412 Retrieved from http://dx.doi.org.ezproxy.library.dal.ca/10.1109/SITA.2015.7358412

Hartmann, S., & 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. doi:10.1016/j.ejor.2009.11.005

Havur, G., Cabanillas, C., Mendling, J., & Polleres, A. (2016). Resource allocation with dependencies in Business process management systems. Paper presented at the *Business Process Management Forum,* 3-19.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research, 165*(2), 289-306. doi: 10.1016/j.ejor.2004.04.002

Hewitt, C. (1985). Challenge of open systems. *Byte, 10*(4), 13.

Jia, Y., Tian, Y., Yang, Y., Sun, H., & Malik, S. A. (2016). The relationship between servitization improvements and business performance of manufacturing companies: A strategic fit perspective. *International Journal of U- and E- Service, Science and Technology, 9*(5), 117-134. doi:10.14257/ijunesst.2016.9.5.11

Kiralp, R., & Venkatadri, U. (2010). DSOPP: A platform for distributed simulation of order promising protocols in supply chain networks. *Production Planning & Control, 21*(6), 562-580. doi:10.1080/09537287.2010.489251

Kirn, S., Herrler, R., Heine, C., & Krempels, K. -. (2003). Agent.hospital - agent-based open framework for clinical applications. Paper presented at the *12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2003,* 36-41. doi:10.1109/ENABL.2003.1231379 Retrieved from http://dx.doi.org/10.1109/ENABL.2003.1231379

Kolisch, R., & Meyer, K. (2006). Selection and scheduling of pharmaceutical research projects. In J. Józefowska, & J. Weglarz (Eds.), *Perspectives in modern project scheduling* (pp. 321-344). Boston, MA: Springer US. doi:10.1007/978-0-387-33768-5_13 Retrieved from https://doi.org/10.1007/978-0-387-33768-5_13

United States. Department of Health and Human Services. Office of Inspector General. (2010). *Challenges to FDA's ability to monitor and inspect foreign clinical trials*. Washington, D.C.: Department of Health and Human Services, Office of Inspector General. Retrieved from http://purl.fdlp.gov/GPO/gpo71045

Lim, M. K., & Winsper, M. (2012). Maximising the efficiency of logistics operations with radio frequency identification technology. Paper presented at the *2012 IEEE International Conference on RFID-Technologies and Applications (RFID-TA)*, Nice, 2012, 275-280. doi:10.1109/RFID-TA.2012.6404529 Retrieved from https://ieeexplore.ieee.org/document/6404529

Lusch, R. F., Vargo, S. L., & Tanniru, M. (2009). *Service, value networks and learning*. Journal of Academy of Marketing Science: doi:10.1007/s11747-008-0131-z

Mohammadi, S., & Khalili, A. (2008). A semantic web service-oriented model for project management. Paper presented at the *2008 IEEE 8th International Conference on Computer and Information Technology Workshops*, Sydney, QLD, 2008, 667-672. doi:10.1109/CIT.2008.Workshops.6 Retrieved from http://ieeexplore.ieee.org/document/4568580

Niknam, M., & Karshenas, S. (2017). A shared ontology approach to semantic representation of BIM data. *Automation in Construction, 80*, 22-36. doi:10.1016/j.autcon.2017.03.013

Opresnik, D., & Taisch, M. (2015). The value of big data in servitization. *International Journal Production Economics,* doi:10.1016/j.ijpe.2014.12.036

Otjacques, B., Noirhomme, M., & Feltz, F. (2006). *Taxonomy of the visualization techniques of project related interactions*

PAREXEL International Corporation. (2015). Optimizing the clinical trial supply chain. Retrieved from https://www.parexel.com/files/3014/2772/8130/Optimizing_Clinical_Trial_Supply_Chain_WP_Mar-15.pdf

Pharmaceutical Research and Manufacturing of America (PhRMA). (2012). *2012 profile pharmaceutical industry*. Retrieved from http://phrma-docs.phrma.org/sites/default/files/pdf/phrma_industry_profile.pdf

Pharmaceutical Research and Manufacturing of America (PhRMA). (2016). *2016 profile biopharmaceutical research industry*. Retrieved from http://phrma-docs.phrma.org/sites/default/files/pdf/biopharmaceutical-industry-profile.pdf

Pourmirza, S., Peters, S., Dijkman, R., & Grefen, P. (2017). A systematic literature review on the architecture of business process management systems. *Information Systems, 66*, 43-58. doi:10.1016/j.is.2017.01.007

Reijers, H. H., Vanderfeesten, I. I., & Aalst, van der, WMP Wil. (2016). The effectiveness of workflow management systems: A longitudinal study. *International Journal of Information Management, 36*(1), 126-141. doi:10.1016/j.ijinfomgt.2015.08.003

Seitz, P. (2016). Understanding BIM - what does it mean and what can it be used for. Paper presented at the *19th IABSE Congress Stockholm 2016: Challenges in Design and Construction of an Innovative and Sustainable Built Environment, September 21, 2016 - September 23,* 545-551.

Shankar, R. D., Martins, S. B., O'Connor, M., Parrish, D. B., & Das, A. K. (2006). A knowledge-based system for managing complex clinical trials. Paper presented at the *19th IEEE International Symposium on Computer-Based Medical Systems, CBMS 2006, June 22, 2006 - June 23,2006,* 270-275. doi:10.1109/CBMS.2006.107 Retrieved from http://dx.doi.org/10.1109/CBMS.2006.107

Taboada, M., & Singh Sharad, S. (2015). Opportunities & Challenges in Central Lab Logistics for Clinical Trial Success. Retrieved from https://www.covance.com/content/dam/covance/assetLibrary/whitepapers/Opportunities%20and%20Challenges%20in%20Central%20Lab%20Logistics%20for%20Clinical%20Trial%20Success%20(2).pdf

Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science, 28*(10), 1197-1210. doi:10.1287/mnsc.28.10.1197

Cooke-Davies, T., & Arzymanow, A. (2003). The maturity of project management in different industries: An investigation into variations between project management models. *International Journal of Project Management; Selected Papers from the Fifth Biennial Conference of the International Research Network for Organizing by Projects.Held in Renesse, Seeland, the Netherlands, 28-31 May 2002., 21*(6), 471-478. doi://doi.org/10.1016/S0263-7863(02)00084-4

Tomas U. Ganiron Jr. (2015). Towards model integrations of tools and techniques for project management success. *International Journal of U- and E- Service, Science and Technology, 8*(7), 49-62. doi:10.14257/ijunesst.2015.8.7.05

U.S. Food and Drug Administration. (2018). The drug development process. Retrieved from https://www.fda.gov/ForPatients/Approvals/Drugs/default.htm

U.S. Food and Drug Administration, Center for Drug Evaluation and Research. (2016). *NOVEL DRUGS 2015 SUMMARY*. Retrieved from https://www.fda.gov/downloads/drugs/developmentapprovalprocess/druginnovation/ucm481709.pdf

Van Arnum, P. (2011). Managing the clinical-trial-material supply chain. *Pharmaceutical Technology, 35*(11), 34-36.

Visser, D., van der Merwe, A., & Gerber, A. (2016). A comparison of project management in system and research projects. Paper presented at the *IST-Africa Week Conference, 2016,* 1-10.

Weeks, R. (2013). The convergence of systemic threads shaping a future South African healthcare dispensation: A technology management perspective: Original research. *Acta Commercii, 13*(1), 1-8. Retrieved from http://reference.sabinet.co.za/sa_epublication_article/acom_v13_n1_a10

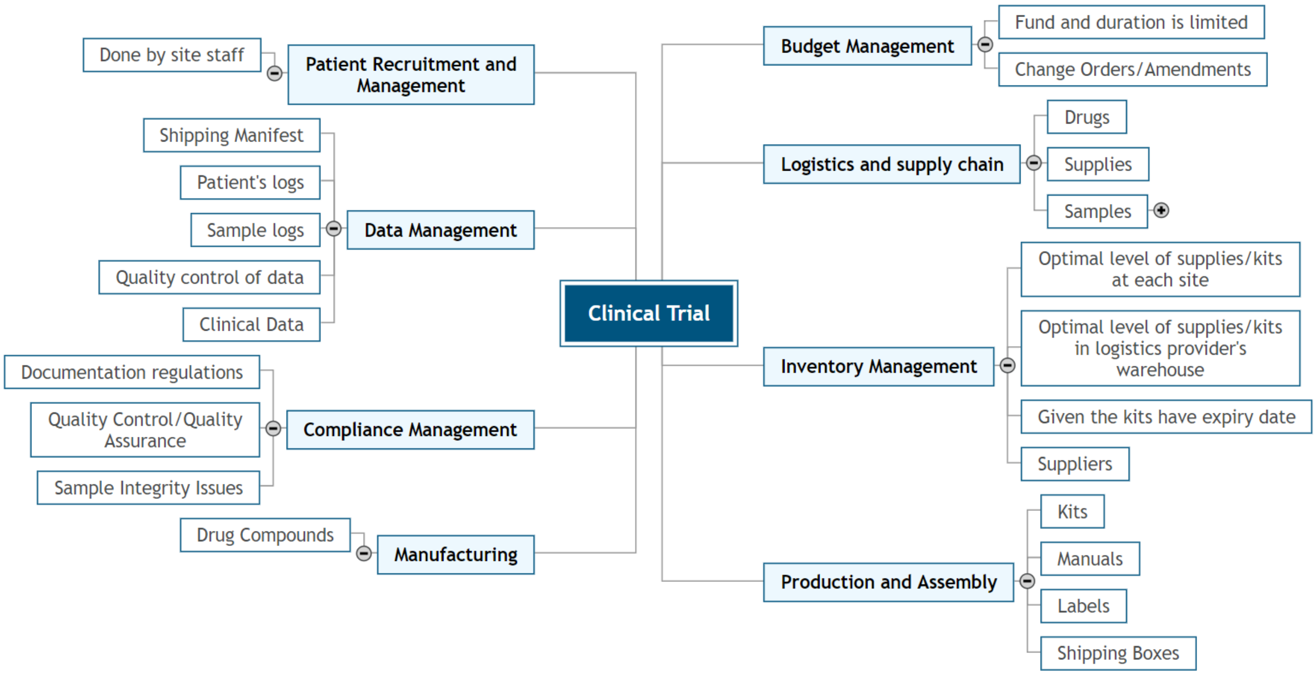
Weeks, R., & Benade, S. (2015). The development of a generic servitization systems framework. *Technology in Society, 43*, 97-104. doi:10.1016/j.techsoc.2015.09.003

Wooldridge, M., & Jennings, N. R. (1995). Agent theories, architectures, and languages: A survey. Paper presented at the *Intelligent Agents,* 1-39.

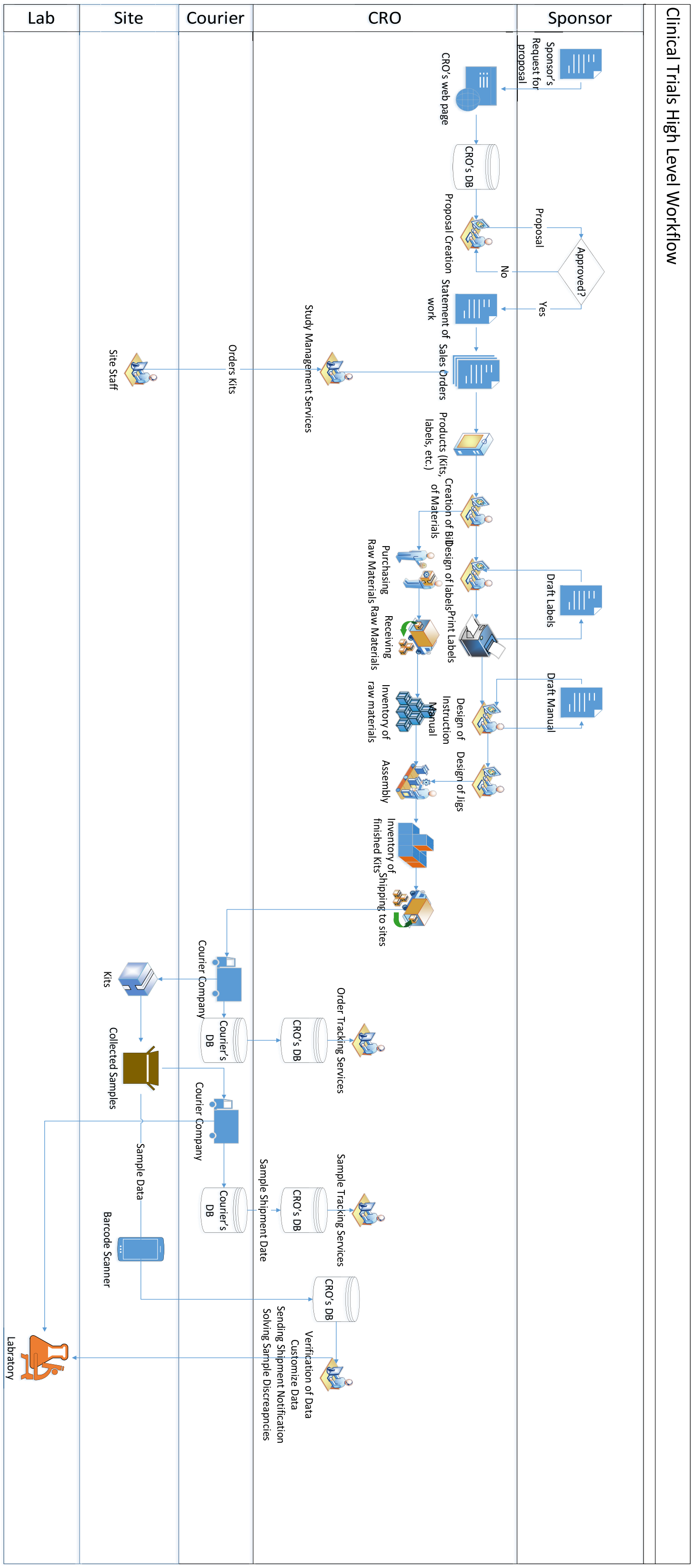
Zhang, M., S. Pawar, K., Shah, J., & Mehta, P. (2013). Evaluating outsourcing partners' capability: A case study from the pharmaceutical supply chain. *Journal of Manufacturing Technology Management, 24*(8), 1080-1101. doi:10.1108/JMTM-02-2012-0023

Zwikael, O., & Globerson, S. (2007). Quality management: A key process in the service industries. *The Service Industries Journal, 27*(8), 1007-1020. doi:10.1080/02642060701673661

Appendix A: Detailed Breakdown of the Clinical Trial Logistics Management



Appendix B: High Level Workflow of the Clinical Trial Logistics Management



Appendix C: Tasks Breakdown of the Clinical Trial Logistics Management

Main Processes	Number	Sub-Tasks	Task Frequency	Who can do it?	Dependency
Budget Initiation	1	Analysis	Once [before seting up the fist site]	CT	
	2	Scope of Work		CT	1
	3	SOW QC		QC	2
	4	SOW Revision		CT	3
	5	SOW Finalization		CT	4
Project Initiation	6	SOW Review, create spec doc and schedule Kick off meeting, confirm the details of SOW with the s		PM	5
	7	Confirm SIV (Site Initiation Visit)- a date on which the first site become active		PM & Site	6
	8	Confirm site and Lab addresses		PM & Sponsor	7
Label Design	9	SOW Review and Reconciliation		PM	8
	10	Create Draft Spreadsheet and Label Design		PM	9
	11	QC		QC	10
	12	Ist Subject File Prep and Printing		OA	11
	13	QC		QC	12
	14	Scan and Send		PM	13
	15	Revise		PM	14
Manual Design	16	SOW Review		PM	8
	17	Develop Manual		PM	16
	18	QC		QC	17
	19	Revise Manual		PM	18
	20	Prepare/PDF/Print		PM	19
	21	QC		QC	20
Kit Setup	24	SOW Review and Kit Blueprint Development		PM	8
	25	QC		QC	24
	26	Order collection supplies and shipping kits		OA	25
	27	Design Shipping kits labels		PM	26
	28	QC shipping kits		QC	27
	29	Kit Label Production		OA	28
	30	QC		QC	29
	31	Design Jig		PM	30
	32	QC		QC	31
Manual Print	33	Prepare/PDF/Print	[1] manual for each site	OA	21
	34	QC		QC	33
	35	Bind/Tab/Collate		QA	34
	36	QC		QC	35
	37	Shipping		OA	36
	38	QC		QC	37
Label Production	39	Expand	[x] kits for every [y] cycles/ or bulk order at once	OA	15
	40	QC		QC	39
	41	Print Label		OA	41
	42	QC		QC	42
	43	Collate and Bind		OA + PM	43
	44	QC		QC	44
	45	Shipping		OA	45
	46	QC		QC	46
Kit Production	47	Affix Labels to Packaging	[x] kits for every [y] cycles	OA	32
	48	Kit Assembly		OA	47
	49	QC		QC	48
	50	Store		OA	49
	51	Shipping		OA	50
	52	QC		QC	51
Shipping	53	Ship Labels , Manual, Kits	Once [before seting up the first site]	OA	52
	54	QC		QC	53
Sample Oversight Initiaition	55	SOW Review	Once	PM	52
	56	Determine sample shipping schedule for each analyte		PM	55
Sample Oversight- Sample Shipment Scheduling and Tracking	57	Request enrolment tracker from all sites	Every [x] weeks	PM	56
	58	Coordinate with site to schedule shipments 1-2 weeks before ship date [with follow ups] [for all Analytes, all sites]	1-2 weeks before samlpe shipping schedule	PM	57
	59	Confirm ship date with site [follow ups- 5 minute each shipment]	48 hours after	SSTP	58
	60	Site ships samples on ship date- print SRFs- follow up for missing paperwork [1.5 hours per day for all shipments]	On every ship date	SSTP	59
	61	Notify lab [10 min per shipment]	On every ship date	SSTP	60
	62	Track shipment- Handle delays [1 hour a day]	One day after every	SSTP	61
	63	Notify site of delivery [5 min per shipment]	One day after every	SSTP	62
Sample Oversight- Sample Management	64	Complete REC 009 with the data for all samples shipped in this shipment	On every ship date	SSTP	60
	65	Create electronic Sample Inventory Log (e-SIL)	On every ship date	SSTP	64
	66	QC e-SIL	On every ship date	PM	65
	67	Ensure number of samles shipped match the number of samples received at the lab	On every ship date	QC	66
	68	Handle discrepancies	On every ship date	SSTP	67
Closing	69	Closing Tasks	Every Site	PM	68

Appendix D: Talbot (1982)'s Example Model

To validate our model, the example problem (a six-job project) mentioned in Talbot (1982) was used. The non-renewable resource (money) was excluded from the problem, since our model does not consider non-renewable resources.

			Renewable Resource Requirements (r_{jkm})						
Job (j)	Mode (m)	Duration (d_{jm})	1	2	3	4	5	(k)	Total Cost (w_{j1m})
1	1	2	1	0	2	1	$135		$270
	2	3	0	1	2	1	65		195
2	1	1	1	0	3	2	160		160
	2	3	0	1	3	2	90		270
3	1	3	1	0	1	4	170		510
	2	4	0	1	1	4	100		400
4	1	5	1	0	1	3	155		775
	2	7	0	1	1	3	85		595
5	1	4	1	0	2	2	150		600
	2	6	0	1	2	2	80		480
6	1	1	1	0	3	4	190		190
	2	4	0	1	3	4	120		480
7	1	0	0	0	0	0	0		0
Resource Available Each Period (R_{kt})			1	2	6	8	$300		

Appendix E: GLPK Code to Validate Model (I)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede
task j
set M{v in V, p in P[v],j in J[v,p]}, default {}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]}; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 1000; /* duration of tasks j of process p of
project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k
usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 3; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR1 {w in W, g in G, k in K[g]},default 0; /* Amount of renewable resource k of group g
dedicated to project work group w*/
param BR{g in G, k in K[g]}, default 0; /* Amount of renewable resource k of group g*/
param a{v in V, p in P[v], j in J[v,p],g in G}, default 0; /* Qualification for resource k of group g on task j
of process p of project v - Binary */

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]},>=0;/*Weighted tardiness cost of process p of project v*/

minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */


s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 resource (k) from group h(p,v,j) for every project, process and
task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in E[v,p]..L[v,p],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,p]..L[v,p],m in M[v,p,jL]}(t)*x[v,s,jL,m,t];/*
predence relationship for processes*/

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V, p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[g,k]; /* capacity
constraint for each resource k of group g   */


solve;

display z, x;

data;
param projects :=1;
param resourceGroup := 1;
param timePeriods := 11;

set P[1] := a ;

set J[1,a] := t_a0 t_a1 t_a2 t_a3 t_a4 t_a5 t_a6 t_a7;

set M[1,a,t_a0] := 1;
set M[1,a,t_a1] := 1 2;
set M[1,a,t_a2] := 1 2;
set M[1,a,t_a3] := 1 2;
set M[1,a,t_a4] := 1 2;
set M[1,a,t_a5] := 1 2;
set M[1,a,t_a6] := 1 2;
set M[1,a,t_a7] := 1;

set Q[1,a,t_a1] := t_a0;
set Q[1,a,t_a2] := t_a0;
set Q[1,a,t_a3] := t_a0;
set Q[1,a,t_a5] := t_a1;
set Q[1,a,t_a4] := t_a1;
set Q[1,a,t_a6] := t_a4 t_a2;
set Q[1,a,t_a7]:= t_a5 t_a6 t_a3;

set p_N[1] := a;
set j_L[1,a] := t_a7;
set j_F[1,a] := t_a0;

set G:= PM;
set K[PM] := K1 K2 K3 K4;

param EE :=
1 a t_a0 0
1 a t_a1 2
1 a t_a2 1
1 a t_a3 3
1 a t_a4 7
1 a t_a5 6
1 a t_a6 8
1 a t_a7 9;


param LL :=
1 a t_a0 0
1 a t_a1 4
1 a t_a2 9
1 a t_a3 10
1 a t_a4 9
1 a t_a5 10
1 a t_a6 10
1 a t_a7 11;

param d :=
1 a t_a0 1 0
1 a t_a1 1 2
1 a t_a2 1 1
1 a t_a3 1 3
1 a t_a4 1 5
1 a t_a5 1 4
1 a t_a6 1 1
1 a t_a7 1 0
1 a t_a1 2 3
1 a t_a2 2 3
1 a t_a3 2 4
1 a t_a4 2 7
1 a t_a5 2 6
1 a t_a6 2 4;

param r :=
1 a t_a1 PM K1 1 1
1 a t_a1 PM K2 1 0
1 a t_a1 PM K3 1 2
```

```
1 a t_a1 PM K4 1 1
1 a t_a1 PM K1 2 0
1 a t_a1 PM K2 2 1
1 a t_a1 PM K3 2 2
1 a t_a1 PM K4 2 1
1 a t_a2 PM K1 1 1
1 a t_a2 PM K2 1 0
1 a t_a2 PM K3 1 3
1 a t_a2 PM K4 1 2
1 a t_a2 PM K1 2 0
1 a t_a2 PM K2 2 1
1 a t_a2 PM K3 2 3
1 a t_a2 PM K4 2 2
1 a t_a3 PM K1 1 1
1 a t_a3 PM K2 1 0
1 a t_a3 PM K3 1 1
1 a t_a3 PM K4 1 4
1 a t_a3 PM K1 2 0
1 a t_a3 PM K2 2 1
1 a t_a3 PM K3 2 1
1 a t_a3 PM K4 2 4
1 a t_a4 PM K1 1 1
1 a t_a4 PM K2 1 0
1 a t_a4 PM K3 1 1
1 a t_a4 PM K4 1 3
1 a t_a4 PM K1 2 0
1 a t_a4 PM K2 2 1
1 a t_a4 PM K3 2 1
1 a t_a4 PM K4 2 3
1 a t_a5 PM K1 1 1
1 a t_a5 PM K2 1 0
1 a t_a5 PM K3 1 2
1 a t_a5 PM K4 1 2
1 a t_a5 PM K1 2 0
1 a t_a5 PM K2 2 1
1 a t_a5 PM K3 2 2
1 a t_a5 PM K4 2 2
1 a t_a6 PM K1 1 1
1 a t_a6 PM K2 1 0
1 a t_a6 PM K3 1 3
1 a t_a6 PM K4 1 4
1 a t_a6 PM K1 2 0
1 a t_a6 PM K2 2 1
1 a t_a6 PM K3 2 3
1 a t_a6 PM K4 2 4
;

param BR :=
PM K1 1
PM K2 2
PM K3 6
PM K4 8;

end;
```

```
z.val = 9
x[1,a,t_a7,1,9].val = 1
x[1,a,t_a7,1,10].val = 0
x[1,a,t_a7,1,11].val = 0
x[1,a,t_a0,1,0].val = 1
x[1,a,t_a1,1,2].val = 0
x[1,a,t_a1,2,2].val = 0
x[1,a,t_a1,1,3].val = 0
x[1,a,t_a1,2,3].val = 1
x[1,a,t_a1,1,4].val = 0
x[1,a,t_a1,2,4].val = 0
x[1,a,t_a2,1,1].val = 0
x[1,a,t_a2,2,1].val = 0
x[1,a,t_a2,1,2].val = 0
x[1,a,t_a2,2,2].val = 0
x[1,a,t_a2,1,3].val = 0
x[1,a,t_a2,2,3].val = 0
x[1,a,t_a2,1,4].val = 0
x[1,a,t_a2,2,4].val = 0
x[1,a,t_a2,1,5].val = 0
x[1,a,t_a2,2,5].val = 0
x[1,a,t_a2,1,6].val = 0
x[1,a,t_a2,2,6].val = 0
x[1,a,t_a2,1,7].val = 0
x[1,a,t_a2,2,7].val = 1
x[1,a,t_a2,1,8].val = 0
x[1,a,t_a2,2,8].val = 0
x[1,a,t_a2,1,9].val = 0
x[1,a,t_a2,2,9].val = 0
x[1,a,t_a3,1,3].val = 1
x[1,a,t_a3,2,3].val = 0
x[1,a,t_a3,1,4].val = 0
x[1,a,t_a3,2,4].val = 0
x[1,a,t_a3,1,5].val = 0
x[1,a,t_a3,2,5].val = 0
x[1,a,t_a3,1,6].val = 0
x[1,a,t_a3,2,6].val = 0
x[1,a,t_a3,1,7].val = 0
x[1,a,t_a3,2,7].val = 0
x[1,a,t_a3,1,8].val = 0
x[1,a,t_a3,2,8].val = 0
x[1,a,t_a3,1,9].val = 0
x[1,a,t_a3,2,9].val = 0
x[1,a,t_a3,1,10].val = 0
x[1,a,t_a3,2,10].val = 0
```

```
x[1,a,t_a4,1,7].val = 0
x[1,a,t_a4,2,7].val = 0
x[1,a,t_a4,1,8].val = 1
x[1,a,t_a4,2,8].val = 0
x[1,a,t_a4,1,9].val = 0
x[1,a,t_a4,2,9].val = 0
x[1,a,t_a5,1,6].val = 0
x[1,a,t_a5,2,6].val = 0
x[1,a,t_a5,1,7].val = 0
x[1,a,t_a5,2,7].val = 0
x[1,a,t_a5,1,8].val = 0
x[1,a,t_a5,2,8].val = 0
x[1,a,t_a5,1,9].val = 0
x[1,a,t_a5,2,9].val = 1
x[1,a,t_a5,1,10].val = 0
x[1,a,t_a5,2,10].val = 0
x[1,a,t_a6,1,8].val = 0
x[1,a,t_a6,2,8].val = 0
x[1,a,t_a6,1,9].val = 1
x[1,a,t_a6,2,9].val = 0
x[1,a,t_a6,1,10].val = 0
x[1,a,t_a6,2,10].val = 0
Model has been successfully processed
>Exit code: 0    Time: 0.419
```

The objective function gives the same answer as the Talbot (1982)'s model. The solutions are all the same except for Job 2, which in our model finishes at time 7 instead of time 3 in Talbot's model.

Appendix F: GLPK Code to Validate Model (II)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede
task j
set M{v in V, p in P[v],j in J[v,p]}, default {}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]}; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 1000; /* duration of tasks j of process p of
project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k
usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 3; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
param BR{v in V, g in G, k in K[g]}, default 0; /* Amount of renewable resource k of group g*/

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]},>=0;/*Weighted tardiness cost of process p of project v*/
#var BR {g in G, k in K[g]},integer;
```

```
minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */


s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 resource (k) from group h(p,v,j) for every project, process and
task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in E[v,p]..L[v,p],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,p]..L[v,p],m in M[v,p,jL]}(t)*x[v,s,jL,m,t];/*
predence relationship for processes*/

s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource k of group g   */


solve;

display z,x,BR;

data;
param projects :=1;
param resourceGroup := 1;
param timePeriods := 20;

set P[1] := a ;

set J[1,a] := t_a0 t_a1 t_a2 t_a3 t_a4 t_a5 t_a6 t_a7;

set M[1,a,t_a0] := 1;
set M[1,a,t_a1] := 1 2;
set M[1,a,t_a2] := 1 2;
set M[1,a,t_a3] := 1 2;
set M[1,a,t_a4] := 1 2;
set M[1,a,t_a5] := 1 2;
set M[1,a,t_a6] := 1 2;
set M[1,a,t_a7] := 1;

set Q[1,a,t_a1] := t_a0;
set Q[1,a,t_a2] := t_a0;
set Q[1,a,t_a3] := t_a0;
set Q[1,a,t_a5] := t_a1;
set Q[1,a,t_a4] := t_a1;
set Q[1,a,t_a6] := t_a4 t_a2;
set Q[1,a,t_a7]:= t_a5 t_a6 t_a3;
set p_N[1] := a;
set j_L[1,a] := t_a7;
set j_F[1,a] := t_a0;


set G:= PM;
set K[PM] := K1 K2 K3 K4;

param EE :=
1 a t_a0 0
1 a t_a1 2
1 a t_a2 1
1 a t_a3 3
1 a t_a4 7
1 a t_a5 6
1 a t_a6 8
1 a t_a7 9
;

param LL :=
1 a t_a0 0
1 a t_a1 4
1 a t_a2 9
1 a t_a3 10
1 a t_a4 9
1 a t_a5 10
1 a t_a6 10
1 a t_a7 11;

param d :=
1 a t_a0 1 0
1 a t_a1 1 2
1 a t_a2 1 1
1 a t_a3 1 3
1 a t_a4 1 5
1 a t_a5 1 4
1 a t_a6 1 1
1 a t_a7 1 0
1 a t_a1 2 3
1 a t_a2 2 3
1 a t_a3 2 4
1 a t_a4 2 7
1 a t_a5 2 6
1 a t_a6 2 4;

param r :=
1 a t_a1 PM K1 1 1
1 a t_a1 PM K2 1 0
1 a t_a1 PM K3 1 2
1 a t_a1 PM K4 1 1
1 a t_a1 PM K1 2 0
```

```
1 a t_a1 PM K2 2 1
1 a t_a1 PM K3 2 2
1 a t_a1 PM K4 2 1
1 a t_a2 PM K1 1 1
1 a t_a2 PM K2 1 0
1 a t_a2 PM K3 1 3
1 a t_a2 PM K4 1 2
1 a t_a2 PM K1 2 0
1 a t_a2 PM K2 2 1
1 a t_a2 PM K3 2 3
1 a t_a2 PM K4 2 2
1 a t_a3 PM K1 1 1
1 a t_a3 PM K2 1 0
1 a t_a3 PM K3 1 1
1 a t_a3 PM K4 1 4
1 a t_a3 PM K1 2 0
1 a t_a3 PM K2 2 1
1 a t_a3 PM K3 2 1
1 a t_a3 PM K4 2 4
1 a t_a4 PM K1 1 1
1 a t_a4 PM K2 1 0
1 a t_a4 PM K3 1 1
1 a t_a4 PM K4 1 3
1 a t_a4 PM K1 2 0
1 a t_a4 PM K2 2 1
1 a t_a4 PM K3 2 1
1 a t_a4 PM K4 2 3
1 a t_a5 PM K1 1 1
1 a t_a5 PM K2 1 0
1 a t_a5 PM K3 1 2
1 a t_a5 PM K4 1 2
1 a t_a5 PM K1 2 0
1 a t_a5 PM K2 2 1
1 a t_a5 PM K3 2 2
1 a t_a5 PM K4 2 2
1 a t_a6 PM K1 1 1
1 a t_a6 PM K2 1 0
1 a t_a6 PM K3 1 3
1 a t_a6 PM K4 1 4
1 a t_a6 PM K1 2 0
1 a t_a6 PM K2 2 1
1 a t_a6 PM K3 2 3
1 a t_a6 PM K4 2 4
;
param BR :=
1 PM K1 1
1 PM K2 2
1 PM K3 6
1 PM K4 8;

end;
```

```
z.val = 9
x[1,a,t_a7,1,9].val = 1
x[1,a,t_a7,1,10].val = 0
x[1,a,t_a7,1,11].val = 0
x[1,a,t_a0,1,0].val = 1
x[1,a,t_a1,1,2].val = 0
x[1,a,t_a1,2,2].val = 0
x[1,a,t_a1,1,3].val = 0
x[1,a,t_a1,2,3].val = 1
x[1,a,t_a1,1,4].val = 0
x[1,a,t_a1,2,4].val = 0
x[1,a,t_a2,1,1].val = 0
x[1,a,t_a2,2,1].val = 0
x[1,a,t_a2,1,2].val = 0
x[1,a,t_a2,2,2].val = 0
x[1,a,t_a2,1,3].val = 0
x[1,a,t_a2,2,3].val = 0
x[1,a,t_a2,1,4].val = 0
x[1,a,t_a2,2,4].val = 0
x[1,a,t_a2,1,5].val = 0
x[1,a,t_a2,2,5].val = 0
x[1,a,t_a2,1,6].val = 0
x[1,a,t_a2,2,6].val = 0
x[1,a,t_a2,1,7].val = 0
x[1,a,t_a2,2,7].val = 1
x[1,a,t_a2,1,8].val = 0
x[1,a,t_a2,2,8].val = 0
x[1,a,t_a2,1,9].val = 0
x[1,a,t_a2,2,9].val = 0
x[1,a,t_a3,1,3].val = 1
x[1,a,t_a3,2,3].val = 0
x[1,a,t_a3,1,4].val = 0
x[1,a,t_a3,2,4].val = 0
x[1,a,t_a3,1,5].val = 0
x[1,a,t_a3,2,5].val = 0
x[1,a,t_a3,1,6].val = 0
x[1,a,t_a3,2,6].val = 0
x[1,a,t_a3,1,7].val = 0
x[1,a,t_a3,2,7].val = 0
x[1,a,t_a3,1,8].val = 0
x[1,a,t_a3,2,8].val = 0
x[1,a,t_a3,1,9].val = 0
x[1,a,t_a3,2,9].val = 0
x[1,a,t_a3,1,10].val = 0
x[1,a,t_a3,2,10].val = 0
x[1,a,t_a4,1,7].val = 0
x[1,a,t_a4,2,7].val = 0
x[1,a,t_a4,1,8].val = 1
x[1,a,t_a4,2,8].val = 0
x[1,a,t_a4,1,9].val = 0
```

```
x[1,a,t_a4,2,9].val = 0
x[1,a,t_a5,1,6].val = 0
x[1,a,t_a5,2,6].val = 0
x[1,a,t_a5,1,7].val = 0
x[1,a,t_a5,2,7].val = 0
x[1,a,t_a5,1,8].val = 0
x[1,a,t_a5,2,8].val = 0
x[1,a,t_a5,1,9].val = 0
x[1,a,t_a5,2,9].val = 1
x[1,a,t_a5,1,10].val = 0
x[1,a,t_a5,2,10].val = 0
x[1,a,t_a6,1,8].val = 0
x[1,a,t_a6,2,8].val = 0
x[1,a,t_a6,1,9].val = 1
x[1,a,t_a6,2,9].val = 0
x[1,a,t_a6,1,10].val = 0
x[1,a,t_a6,2,10].val = 0
BR[1,PM,K1] = 1
BR[1,PM,K2] = 2
BR[1,PM,K3] = 6
BR[1,PM,K4] = 8
Model has been successfully processed
>Exit code: 0    Time: 0.231
```

Appendix G: GLPK Code to Validate Model (IV)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede
task j
set M{v in V, p in P[v],j in J[v,p]}, default {}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]}; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 1000; /* duration of tasks j of process p of
project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k
usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 3; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR1 {w in W, g in G, k in K[g]},default 0; /* Amount of renewable resource k of group g
dedicated to project work group w*/
#param BR{g in G, k in K[g]}, default 0; /* Amount of renewable resource k of group g*/
param a{v in V, p in P[v], j in J[v,p],g in G}, default 0; /* Qualification for resource k of group g on task j
of process p of project v - Binary */

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]},>=0;/*Weighted tardiness cost of process p of project v*/
var BR {g in G, k in K[g]},integer;

minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */


s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 resource (k) from group h(p,v,j) for every project, process and
task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in E[v,p]..L[v,p],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,p]..L[v,p],m in M[v,p,jL]}(t)*x[v,s,jL,m,t];/*
predence relationship for processes*/

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V, p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[g,k]; /* capacity
constraint for each resource k of group g   */


solve;

display z, x,BR;

data;
param projects :=1;
param resourceGroup := 1;
param timePeriods := 20;

set P[1] := a ;

set J[1,a] := t_a0 t_a1 t_a2 t_a3 t_a4 t_a5 t_a6 t_a7;

set M[1,a,t_a0] := 1;
set M[1,a,t_a1] := 1 2;
set M[1,a,t_a2] := 1 2;
set M[1,a,t_a3] := 1 2;
set M[1,a,t_a4] := 1 2;
set M[1,a,t_a5] := 1 2;
set M[1,a,t_a6] := 1 2;
set M[1,a,t_a7] := 1;

set Q[1,a,t_a1] := t_a0;
set Q[1,a,t_a2] := t_a0;
set Q[1,a,t_a3] := t_a0;
set Q[1,a,t_a5] := t_a1;
set Q[1,a,t_a4] := t_a1;
set Q[1,a,t_a6] := t_a4 t_a2;
set Q[1,a,t_a7]:= t_a5 t_a6 t_a3;

set p_N[1] := a;
set j_L[1,a] := t_a7;
set j_F[1,a] := t_a0;

set G:= PM;
set K[PM] := K1 K2 K3 K4;

param EE :=
1 a t_a0 0
1 a t_a1 2
1 a t_a2 1
1 a t_a3 3
1 a t_a4 7
1 a t_a5 6
1 a t_a6 8
1 a t_a7 9;


param LL :=
1 a t_a0 0
1 a t_a1 4
1 a t_a2 9
1 a t_a3 10
1 a t_a4 9
1 a t_a5 10
1 a t_a6 10
1 a t_a7 11;

param d :=
1 a t_a0 1 0
1 a t_a1 1 2
1 a t_a2 1 1
1 a t_a3 1 3
1 a t_a4 1 5
1 a t_a5 1 4
1 a t_a6 1 1
1 a t_a7 1 0
1 a t_a1 2 3
1 a t_a2 2 3
1 a t_a3 2 4
1 a t_a4 2 7
1 a t_a5 2 6
1 a t_a6 2 4;

param r :=
1 a t_a1 PM K1 1 1
1 a t_a1 PM K2 1 0
```

```
1 a t_a1 PM K3 1 2
1 a t_a1 PM K4 1 1
1 a t_a1 PM K1 2 0
1 a t_a1 PM K2 2 1
1 a t_a1 PM K3 2 2
1 a t_a1 PM K4 2 1
1 a t_a2 PM K1 1 1
1 a t_a2 PM K2 1 0
1 a t_a2 PM K3 1 3
1 a t_a2 PM K4 1 2
1 a t_a2 PM K1 2 0
1 a t_a2 PM K2 2 1
1 a t_a2 PM K3 2 3
1 a t_a2 PM K4 2 2
1 a t_a3 PM K1 1 1
1 a t_a3 PM K2 1 0
1 a t_a3 PM K3 1 1
1 a t_a3 PM K4 1 4
1 a t_a3 PM K1 2 0
1 a t_a3 PM K2 2 1
1 a t_a3 PM K3 2 1
1 a t_a3 PM K4 2 4
1 a t_a4 PM K1 1 1
1 a t_a4 PM K2 1 0
1 a t_a4 PM K3 1 1
1 a t_a4 PM K4 1 3
1 a t_a4 PM K1 2 0
1 a t_a4 PM K2 2 1
1 a t_a4 PM K3 2 1
1 a t_a4 PM K4 2 3
1 a t_a5 PM K1 1 1
1 a t_a5 PM K2 1 0
1 a t_a5 PM K3 1 2
1 a t_a5 PM K4 1 2
1 a t_a5 PM K1 2 0
1 a t_a5 PM K2 2 1
1 a t_a5 PM K3 2 2
1 a t_a5 PM K4 2 2
1 a t_a6 PM K1 1 1
1 a t_a6 PM K2 1 0
1 a t_a6 PM K3 1 3
1 a t_a6 PM K4 1 4
1 a t_a6 PM K1 2 0
1 a t_a6 PM K2 2 1
1 a t_a6 PM K3 2 3
1 a t_a6 PM K4 2 4
;

/*param BR :=
PM K1 1
PM K2 2
PM K3 6
PM K4 8; */

end;


z.val = 9
x[1,a,t_a7,1,9].val = 1
x[1,a,t_a7,1,10].val = 0
x[1,a,t_a7,1,11].val = 0
x[1,a,t_a0,1,0].val = 1
x[1,a,t_a1,1,2].val = 1
x[1,a,t_a1,2,2].val = 0
x[1,a,t_a1,1,3].val = 0
x[1,a,t_a1,2,3].val = 0
x[1,a,t_a1,1,4].val = 0
x[1,a,t_a1,2,4].val = 0
x[1,a,t_a2,1,1].val = 1
x[1,a,t_a2,2,1].val = 0
x[1,a,t_a2,1,2].val = 0
x[1,a,t_a2,2,2].val = 0
x[1,a,t_a2,1,3].val = 0
x[1,a,t_a2,2,3].val = 0
x[1,a,t_a2,1,4].val = 0
x[1,a,t_a2,2,4].val = 0
x[1,a,t_a2,1,5].val = 0
x[1,a,t_a2,2,5].val = 0
x[1,a,t_a2,1,6].val = 0
x[1,a,t_a2,2,6].val = 0
x[1,a,t_a2,1,7].val = 0
x[1,a,t_a2,2,7].val = 0
x[1,a,t_a2,1,8].val = 0
x[1,a,t_a2,2,8].val = 0
x[1,a,t_a2,1,9].val = 0
x[1,a,t_a2,2,9].val = 0
x[1,a,t_a3,1,3].val = 0
x[1,a,t_a3,2,3].val = 0
x[1,a,t_a3,1,4].val = 0
x[1,a,t_a3,2,4].val = 0
x[1,a,t_a3,1,5].val = 1
x[1,a,t_a3,2,5].val = 0
x[1,a,t_a3,1,6].val = 0
x[1,a,t_a3,2,6].val = 0
x[1,a,t_a3,1,7].val = 0
x[1,a,t_a3,2,7].val = 0
x[1,a,t_a3,1,8].val = 0
x[1,a,t_a3,2,8].val = 0
x[1,a,t_a3,1,9].val = 0
x[1,a,t_a3,2,9].val = 0
x[1,a,t_a3,1,10].val = 0
x[1,a,t_a3,2,10].val = 0
```

```
x[1,a,t_a4,1,7].val = 1
x[1,a,t_a4,2,7].val = 0
x[1,a,t_a4,1,8].val = 0
x[1,a,t_a4,2,8].val = 0
x[1,a,t_a4,1,9].val = 0
x[1,a,t_a4,2,9].val = 0
x[1,a,t_a5,1,6].val = 0
x[1,a,t_a5,2,6].val = 0
x[1,a,t_a5,1,7].val = 0
x[1,a,t_a5,2,7].val = 0
x[1,a,t_a5,1,8].val = 0
x[1,a,t_a5,2,8].val = 0
x[1,a,t_a5,1,9].val = 1
x[1,a,t_a5,2,9].val = 0
x[1,a,t_a5,1,10].val = 0
x[1,a,t_a5,2,10].val = 0
x[1,a,t_a6,1,8].val = 1
x[1,a,t_a6,2,8].val = 0
x[1,a,t_a6,1,9].val = 0
x[1,a,t_a6,2,9].val = 0
x[1,a,t_a6,1,10].val = 0
x[1,a,t_a6,2,10].val = 0
BR[PM,K1].val = 2
BR[PM,K2].val = 0
BR[PM,K3].val = 5
BR[PM,K4].val = 7
Model has been successfully processed
>Exit code: 0    Time: 0.251
```

Appendix H: GLPK Code to Validate Model (V)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede
task j
set M{v in V, p in P[v],j in J[v,p]}, default {}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]}; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 1000; /* duration of tasks j of process p of
project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k
usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 3; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR{v in V, g in G, k in K[g]}, default 0; /* Amount of renewable resource k of group g*/

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]},>=0;/*Weighted tardiness cost of process p of project v*/
var BR {v in V,g in G, k in K[g]},integer;
```

```
minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */


s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 resource (k) from group h(p,v,j) for every project, process and
task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in E[v,p]..L[v,p],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,p]..L[v,p],m in M[v,p,jL]}(t)*x[v,s,jL,m,t];/*
predence relationship for processes*/

s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource k of group g   */


solve;

display z,x,BR;

data;
param projects :=1;
param resourceGroup := 1;
param timePeriods := 20;

set P[1] := a ;

set J[1,a] := t_a0 t_a1 t_a2 t_a3 t_a4 t_a5 t_a6 t_a7;

set M[1,a,t_a0] := 1;
set M[1,a,t_a1] := 1 2;
set M[1,a,t_a2] := 1 2;
set M[1,a,t_a3] := 1 2;
set M[1,a,t_a4] := 1 2;
set M[1,a,t_a5] := 1 2;
set M[1,a,t_a6] := 1 2;
set M[1,a,t_a7] := 1;

set Q[1,a,t_a1] := t_a0;
set Q[1,a,t_a2] := t_a0;
set Q[1,a,t_a3] := t_a0;
set Q[1,a,t_a5] := t_a1;
set Q[1,a,t_a4] := t_a1;
set Q[1,a,t_a6] := t_a4 t_a2;
set Q[1,a,t_a7]:= t_a5 t_a6 t_a3;
set p_N[1] := a;
set j_L[1,a] := t_a7;
set j_F[1,a] := t_a0;


set G:= PM;
set K[PM] := K1 K2 K3 K4;

param EE :=
1 a t_a0 0
1 a t_a1 2
1 a t_a2 1
1 a t_a3 3
1 a t_a4 7
1 a t_a5 6
1 a t_a6 8
1 a t_a7 9
;

param LL :=
1 a t_a0 0
1 a t_a1 4
1 a t_a2 9
1 a t_a3 10
1 a t_a4 9
1 a t_a5 10
1 a t_a6 10
1 a t_a7 11;

param d :=
1 a t_a0 1 0
1 a t_a1 1 2
1 a t_a2 1 1
1 a t_a3 1 3
1 a t_a4 1 5
1 a t_a5 1 4
1 a t_a6 1 1
1 a t_a7 1 0
1 a t_a1 2 3
1 a t_a2 2 3
1 a t_a3 2 4
1 a t_a4 2 7
1 a t_a5 2 6
1 a t_a6 2 4;

param r :=
1 a t_a1 PM K1 1 1
1 a t_a1 PM K2 1 0
1 a t_a1 PM K3 1 2
1 a t_a1 PM K4 1 1
1 a t_a1 PM K1 2 0
```

```
1 a t_a1 PM K2 2 1
1 a t_a1 PM K3 2 2
1 a t_a1 PM K4 2 1
1 a t_a2 PM K1 1 1
1 a t_a2 PM K2 1 0
1 a t_a2 PM K3 1 3
1 a t_a2 PM K4 1 2
1 a t_a2 PM K1 2 0
1 a t_a2 PM K2 2 1
1 a t_a2 PM K3 2 3
1 a t_a2 PM K4 2 2
1 a t_a3 PM K1 1 1
1 a t_a3 PM K2 1 0
1 a t_a3 PM K3 1 1
1 a t_a3 PM K4 1 4
1 a t_a3 PM K1 2 0
1 a t_a3 PM K2 2 1
1 a t_a3 PM K3 2 1
1 a t_a3 PM K4 2 4
1 a t_a4 PM K1 1 1
1 a t_a4 PM K2 1 0
1 a t_a4 PM K3 1 1
1 a t_a4 PM K4 1 3
1 a t_a4 PM K1 2 0
1 a t_a4 PM K2 2 1
1 a t_a4 PM K3 2 1
1 a t_a4 PM K4 2 3
1 a t_a5 PM K1 1 1
1 a t_a5 PM K2 1 0
1 a t_a5 PM K3 1 2
1 a t_a5 PM K4 1 2
1 a t_a5 PM K1 2 0
1 a t_a5 PM K2 2 1
1 a t_a5 PM K3 2 2
1 a t_a5 PM K4 2 2
1 a t_a6 PM K1 1 1
1 a t_a6 PM K2 1 0
1 a t_a6 PM K3 1 3
1 a t_a6 PM K4 1 4
1 a t_a6 PM K1 2 0
1 a t_a6 PM K2 2 1
1 a t_a6 PM K3 2 3
1 a t_a6 PM K4 2 4
;
/*param BR :=
1 PM K1 1
1 PM K2 2
1 PM K3 6
1 PM K4 8; */

end;
```

```
z.val = 9
x[1,a,t_a7,1,9].val = 1
x[1,a,t_a7,1,10].val = 0
x[1,a,t_a7,1,11].val = 0
x[1,a,t_a0,1,0].val = 1
x[1,a,t_a1,1,2].val = 1
x[1,a,t_a1,2,2].val = 0
x[1,a,t_a1,1,3].val = 0
x[1,a,t_a1,2,3].val = 0
x[1,a,t_a1,1,4].val = 0
x[1,a,t_a1,2,4].val = 0
x[1,a,t_a2,1,1].val = 1
x[1,a,t_a2,2,1].val = 0
x[1,a,t_a2,1,2].val = 0
x[1,a,t_a2,2,2].val = 0
x[1,a,t_a2,1,3].val = 0
x[1,a,t_a2,2,3].val = 0
x[1,a,t_a2,1,4].val = 0
x[1,a,t_a2,2,4].val = 0
x[1,a,t_a2,1,5].val = 0
x[1,a,t_a2,2,5].val = 0
x[1,a,t_a2,1,6].val = 0
x[1,a,t_a2,2,6].val = 0
x[1,a,t_a2,1,7].val = 0
x[1,a,t_a2,2,7].val = 0
x[1,a,t_a2,1,8].val = 0
x[1,a,t_a2,2,8].val = 0
x[1,a,t_a2,1,9].val = 0
x[1,a,t_a2,2,9].val = 0
x[1,a,t_a3,1,3].val = 0
x[1,a,t_a3,2,3].val = 0
x[1,a,t_a3,1,4].val = 0
x[1,a,t_a3,2,4].val = 0
x[1,a,t_a3,1,5].val = 1
x[1,a,t_a3,2,5].val = 0
x[1,a,t_a3,1,6].val = 0
x[1,a,t_a3,2,6].val = 0
x[1,a,t_a3,1,7].val = 0
x[1,a,t_a3,2,7].val = 0
x[1,a,t_a3,1,8].val = 0
x[1,a,t_a3,2,8].val = 0
x[1,a,t_a3,1,9].val = 0
x[1,a,t_a3,2,9].val = 0
x[1,a,t_a3,1,10].val = 0
x[1,a,t_a3,2,10].val = 0
x[1,a,t_a4,1,7].val = 1
x[1,a,t_a4,2,7].val = 0
x[1,a,t_a4,1,8].val = 0
x[1,a,t_a4,2,8].val = 0
x[1,a,t_a4,1,9].val = 0
```

```
x[1,a,t_a4,2,9].val = 0
x[1,a,t_a5,1,6].val = 0
x[1,a,t_a5,2,6].val = 0
x[1,a,t_a5,1,7].val = 0
x[1,a,t_a5,2,7].val = 0
x[1,a,t_a5,1,8].val = 0
x[1,a,t_a5,2,8].val = 0
x[1,a,t_a5,1,9].val = 1
x[1,a,t_a5,2,9].val = 0
x[1,a,t_a5,1,10].val = 0
x[1,a,t_a5,2,10].val = 0
x[1,a,t_a6,1,8].val = 1
x[1,a,t_a6,2,8].val = 0
x[1,a,t_a6,1,9].val = 0
x[1,a,t_a6,2,9].val = 0
x[1,a,t_a6,1,10].val = 0
x[1,a,t_a6,2,10].val = 0
BR[1,PM,K1].val = 2
BR[1,PM,K2].val = 0
BR[1,PM,K3].val = 5
BR[1,PM,K4].val = 7
Model has been successfully processed
>Exit code: 0    Time: 0.229
```

Appendix I: GLPK Code to Experiment Model (I)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups

set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede task j
set M{v in V, p in P[v],j in J[v,p]}, default {1}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]},default 0; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}, default 50; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}, default 50; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 0; /* duration of tasks j of process p of project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 0; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR{v in V, g in G, k in K[g]}, default 0; /* Amount of renewable resource k of group g*/
param R{g in G, k in K[g]}; /* total amount of renewable resource k*/

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
#var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]}>=0;/*Weighted tardiness cost of process p of project v*/
var BR {v in V,g in G, k in K[g]} integer;
```

```
#var R {g in G, k in K[g]} integer;

minimize z: sum {v in V, p in P[v]} TC1[v,p]; /* minimize the weighted tardiness of the project and
process*/
#minimize z: sum{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T}x[v,p,j,m,t];

#minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */

s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 mode for every task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in EE[v,p,jF]..LL[v,p,jF],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,s]..L[v,s],m in M[v,s,jL]}(t)*x[v,s,jL,m,t];/* predence
relationship for processes*/

#s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource k of group g   */

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V,p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= R[g,k]; /* capacity constraint
for each resource k of group g   */

#s.t. Constraint_5{v in V, p in p_N[v],j in j_L[v,p]}: TC[v] >= c[v] *( sum{m in M[v,p,j],t in E[v,p]..L[v,p]}
(t*x[v,p,j,m,t])- dd[v]); /* tardiness of project is equal to some penalty c[v] times the difference of the
finish time of the last task of the last process minus the due date of the project */

s.t. Constraint_6{v in V, p in P[v], j in j_L[v,p], g in G}: TC1[v,p] >= c1[v,p]*(sum{m in M[v,p,j], t in
EE[v,p,j]..LL[v,p,j]} (t*x[v,p,j,m,t])- ddd[v,p]); /* tardiness of process is equal to some penalty c[v,p] times
the difference of the finish time of the last task of each process minus the due date of the process */

#s.t. Constraint_7 {g in G, k in K[g]}: sum {v in V}BR[v,g,k] <= R[g,k] ;


/*table tab_data IN "CSV" "projects.csv":
V <- [project] P[V]<-[Process];*/

solve;

display z, x,BR,R;

data;
param projects :=10;
param resourceGroup := 3;
param timePeriods := 50;

# Define Processes and Orders for each Project- The decimal number indicates the order number
#Every project has a dummy starting process called "p0" with 0 duration.

set P[1] := p0   1.1      3.1      3.2;
set P[2] := p0   1.1      3.1 ;
set P[3] := p0   3.1      3.2      3.3      3.4;
set P[4] := p0   3.1;
set P[5] := p0   3.1;
set P[6] := p0   1.1;
set P[7] := p0   3.1;
set P[8] := p0   3.1;
set P[9] := p0   3.1;
set P[10] := p0  3.1      3.2;


# Define tasks for each process:
set J[1,p0] := t0;
set J[1,1.1] := t1t2      t3       t4       t5       t6       t7       t8       t9       t10;
set J[1,3.1] := t6t7      t8       t9       t10;
set J[1,3.2] := t6t7      t8       t9       t10;

set J[2,p0] := t0;
set J[2,1.1] := t1t2      t3       t4       t5       t6       t7       t8       t9       t10;
set J[2,3.1] := t6t7      t8       t9       t10;

set J[3,p0]:=t0;
set J[3,3.1]:=t6  t7      t8       t9       t10;
set J[3,3.2]:=t6  t7      t8       t9       t10;
set J[3,3.3]:=t6  t7      t8       t9       t10;
set J[3,3.4]:=t6  t7      t8       t9       t10;

set J[4,p0] :=t0;
set J[4,3.1]:= t6 t7      t8       t9       t10;

set J[5,p0] :=t0;
set J[5,3.1] := t6t7      t8       t9       t10;

set J[6,p0] :=t0;
set J[6,1.1] := t1t2      t3       t4       t5       t6       t7       t8       t9       t10;

set J[7,p0] :=t0;
set J[7,3.1] := t6t7      t8       t9       t10;

set J[8,p0] :=t0;
set J[8,3.1] := t6t7      t8       t9       t10;
set J[9,p0] :=t0;
set J[9,3.1] := t6t7      t8       t9       t10;

set J[10,p0] :=t0;
set J[10,3.1] :=t6        t7       t8       t9       t10;
set J[10,3.2] :=t6        t7       t8       t9       t10;

# Define Mode
# set all modes by default to 1

# Task Precedence
set Q[1,1.1,t2] := t1;
set Q[1,1.1,t3] := t2;
set Q[1,1.1,t4]:= t3;
set Q[1,1.1,t5] := t4;
set Q[1,1.1,t6] := t5;
set Q[1,1.1,t7]:= t6;
set Q[1,1.1,t8]:= t7;
set Q[1,1.1,t9]:= t8;
set Q[1,1.1,t10]:= t9;

set Q[1,3.1,t7]:= t6;
set Q[1,3.1,t8] := t7;
set Q[1,3.1,t9]:= t8;
set Q[1,3.1,t10]:= t9;

set Q[1,3.2,t7]:= t6;
set Q[1,3.2,t8] := t7;
set Q[1,3.2,t9]:= t8;
set Q[1,3.2,t10]:= t9;

set Q[2,1.1,t2] := t1;
set Q[2,1.1,t3] := t2;
set Q[2,1.1,t4]:= t3;
set Q[2,1.1,t5] := t4;
set Q[2,1.1,t6] := t5;
set Q[2,1.1,t7]:= t6;
set Q[2,1.1,t8]:= t7;
set Q[2,1.1,t9]:= t8;
set Q[2,1.1,t10]:= t9;

set Q[2,3.1,t7]:= t6;
set Q[2,3.1,t8] := t7;
set Q[2,3.1,t9]:= t8;
set Q[2,3.1,t10]:= t9;

set Q[3,3.1,t7]:= t6;
set Q[3,3.1,t8] := t7;
set Q[3,3.1,t9]:= t8;
set Q[3,3.1,t10]:= t9;

set Q[3,3.2,t7]:= t6;
set Q[3,3.2,t8] := t7;
set Q[3,3.2,t9]:= t8;
set Q[3,3.2,t10]:= t9;

set Q[3,3.3,t7]:= t6;
set Q[3,3.3,t8] := t7;
set Q[3,3.3,t9]:= t8;
set Q[3,3.3,t10]:= t9;

set Q[3,3.4,t7]:= t6;
set Q[3,3.4,t8] := t7;
set Q[3,3.4,t9]:= t8;
set Q[3,3.4,t10]:= t9;

set Q[4,3.1,t7]:= t6;
set Q[4,3.1,t8] := t7;
set Q[4,3.1,t9]:= t8;
set Q[4,3.1,t10]:= t9;

set Q[5,3.1,t7]:= t6;
set Q[5,3.1,t8] := t7;
set Q[5,3.1,t9]:= t8;
set Q[5,3.1,t10]:= t9;

set Q[6,1.1,t2] := t1;
set Q[6,1.1,t3] := t2;
set Q[6,1.1,t4]:= t3;
set Q[6,1.1,t5] := t4;
set Q[6,1.1,t6] := t5;
set Q[6,1.1,t7]:= t6;
set Q[6,1.1,t8]:= t7;
set Q[6,1.1,t9]:= t8;
set Q[6,1.1,t10]:= t9;

set Q[7,3.1,t7]:= t6;
set Q[7,3.1,t8] := t7;
set Q[7,3.1,t9]:= t8;
set Q[7,3.1,t10]:= t9;

set Q[8,3.1,t7]:= t6;
set Q[8,3.1,t8] := t7;
set Q[8,3.1,t9]:= t8;
set Q[8,3.1,t10]:= t9;
set Q[9,3.1,t7]:= t6;
set Q[9,3.1,t8] := t7;
set Q[9,3.1,t9]:= t8;
set Q[9,3.1,t10]:= t9;

set Q[10,3.1,t7]:= t6;
set Q[10,3.1,t8] := t7;
set Q[10,3.1,t9]:= t8;
set Q[10,3.1,t10]:= t9;

set Q[10,3.2,t7]:= t6;
set Q[10,3.2,t8] := t7;
set Q[10,3.2,t9]:= t8;
set Q[10,3.2,t10]:= t9;


# Process Precedence
set S[1,1.1] := p0;
set S[1,3.1] := 1.1;
set S[1,3.2] := 3.1;

set S[2,1.1] := p0;
set S[2,3.1] := 1.1;

set S[3,3.1] :=p0;
set S[3,3.2] :=3.1;
set S[3,3.3] :=3.2;
set S[3,3.4] :=3.3;

set S[4,3.1] :=p0;

set S[5,3.1] :=p0;

set S[6,1.1] :=p0;

set S[7,3.1] :=p0;

set S[8,3.1] :=p0;

set S[9,3.1] :=p0;

set S[10,3.1] :=p0;
set S[10,3.2] :=3.1;

# Last task of each process
set j_L[1,p0] := t0;
set j_L[1,1.1] := t10;
set j_L[1,3.1] := t10;
set j_L[1,3.2] := t10;

set j_L[2,p0] := t0;
set j_L[2,1.1] := t10;
set j_L[2,3.1] := t10;

set j_L[3,3.1] := t10;
set j_L[3,3.2] := t10;
set j_L[3,3.3] := t10;
set j_L[3,3.4] := t10;

set j_L[4,3.1] := t10;

set j_L[5,3.1] := t10;

set j_L[6,1.1] := t10;

set j_L[7,3.1] := t10;

set j_L[8,3.1] := t10;

set j_L[9,3.1] := t10;

set j_L[10,3.1] := t10;
set j_L[10,3.2] := t10;

# First task of each process
set j_F[1,p0] := t0;
set j_F[1,1.1] := t1;
set j_F[1,3.1] := t6;
set j_F[1,3.2] := t6;

set j_F[2,p0] := t0;
set j_F[2,1.1] := t1;
set j_F[2,3.1] := t6;

set j_F[3,p0] := t0;
set j_F[3,3.1] := t6;
set j_F[3,3.2] := t6;
set j_F[3,3.3] := t6;
set j_F[3,3.4] := t6;

set j_F[4,p0] := t0;
set j_F[4,3.1] := t6;

set j_F[5,p0] := t0;
set j_F[5,3.1] := t6;
set j_F[6,p0] := t0;
set j_F[6,1.1] := t1;

set j_F[7,p0] := t0;
set j_F[7,3.1] := t6;

set j_F[8,p0] := t0;
set j_F[8,3.1] := t6;

set j_F[9,p0] := t0;
set j_F[9,3.1] := t6;

set j_F[10,p0] := t0;
set j_F[10,3.1] := t6;
set j_F[10,3.2] := t6;

# The group of people
set G:= PM QC OA;

set K[PM] := PM; # The PM people
set K[OA] := OA ; #The OA people
set K[QC] := QC ; #The QC people

# k and g are equal in this example

# set the total number of each resource k
param R :=
PM PM 2
OA OA 2
QC QC 2 ;

# Set the due dates for each process
param ddd :=
1       p0      0
1       1.1     19
1       3.1     24
1       3.2     29

2       p0      0
2       1.1     19
2       3.1     24

3       p0      0
3       3.1     20
3       3.2     25
3       3.3     30
3       3.4     35
```

```
4         p0        0
4         3.1       21

5         p0        0
5         3.1       20

6         p0        0
6         1.1       23

7         p0        0
7         3.1       22

8         p0        0
8         3.1       21

9         p0        0
9         3.1       22

10        p0        0
10        3.1       23
10        3.2       28
;

# Set the early finish of each process
param E :=
1         p0        0
1         1.1       15
1         3.1       22
1         3.2       27

2         p0        0
2         1.1       16
2         3.1       22

3         p0        0
3         3.1       18
3         3.2       23
3         3.3       28
3         3.4       33

4         p0        0
4         3.1       19

5         p0        0
5         3.1       18

6         p0        0
```

```
6       1.1     19

7       p0      0
7       3.1     20

8       p0      0
8       3.1     19

9       p0      0
9       3.1     20

10      p0      0
10      3.1     21
10      3.2     26

;

# Set the late finish of each process
/*param L :=
1       p0      0
1       1.1     23
1       3.1     28
1       3.2     33
2       p0      0
2       1.1     28
2       3.1     33
3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33
4       p0      0
4       3.1     33
5       p0      0
5       3.1     33
6       p0      0
6       1.1     33
7       p0      0
7       3.1     33
8       p0      0
8       3.1     33
9       p0      0
9       3.1     23
10      p0      0
10      3.1     28
10      3.2     33;*/
```

```
# Set the early finsh of each task
param EE :=
1 p0 t0 0
1       1.1     t1      7
2       1.1     t1      7
6       1.1     t1      11

1       1.1     t2      8
2       1.1     t2      8
6       1.1     t2      12

1       1.1     t3      9
2       1.1     t3      9
6       1.1     t3      13

1       1.1     t4      10
2       1.1     t4      10
6       1.1     t4      14

1       1.1     t5      11
2       1.1     t5      11
6       1.1     t5      15

1       1.1     t6      12
1       3.1     t6      18
1       3.2     t6      23
2       1.1     t6      12
2       3.1     t6      18
3       3.1     t6      14
3       3.2     t6      19
3       3.3     t6      24
3       3.4     t6      29
4       3.1     t6      15
5       3.1     t6      14
6       1.1     t6      10
7       3.1     t6      16
8       3.1     t6      15
9       3.1     t6      16
10      3.1     t6      17
10      3.2     t6      22

1       1.1     t7      13
1       3.1     t7      19
1       3.2     t7      24
2       1.1     t7      13
2       3.1     t7      19
3       3.1     t7      15
3       3.2     t7      20
```

3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	16
5	3.1	t7	15
6	1.1	t7	11
7	3.1	t7	17
8	3.1	t7	16
9	3.1	t7	17
10	3.1	t7	18
10	3.2	t7	23

1	1.1	t8	14
1	3.1	t8	20
1	3.2	t8	25
2	1.1	t8	14
2	3.1	t8	20
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	17
5	3.1	t8	16
6	1.1	t8	12
7	3.1	t8	18
8	3.1	t8	17
9	3.1	t8	18
10	3.1	t8	19
10	3.2	t8	24

1	1.1	t9	15
1	3.1	t9	21
1	3.2	t9	26
2	1.1	t9	15
2	3.1	t9	21
3	3.1	t9	17
3	3.2	t9	22
3	3.3	t9	27
3	3.4	t9	32
4	3.1	t9	18
5	3.1	t9	17
6	1.1	t9	13
7	3.1	t9	19
8	3.1	t9	18
9	3.1	t9	19
10	3.1	t9	20
10	3.2	t9	25

1	1.1	t10	16

```
1       3.1     t10     22
1       3.2     t10     27
2       1.1     t10     16
2       3.1     t10     22
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     19
5       3.1     t10     18
6       1.1     t10     14
7       3.1     t10     20
8       3.1     t10     19
9       3.1     t10     20
10      3.1     t10     21
10      3.2     t10     26;

# Set the  late finish of each task
/*param LL :=

1       1.1     t1      14
2       1.1     t1      19
6       1.1     t1      24

1       1.1     t2      15
2       1.1     t2      20
6       1.1     t2      25

1       1.1     t3      15
2       1.1     t3      20
6       1.1     t3      25

1       1.1     t5      18
2       1.1     t5      23
6       1.1     t5      28

1       1.1     t6      19
1       3.1     t6      24
1       3.2     t6      29
2       1.1     t6      24
2       3.1     t6      29
3       3.1     t6      14
3       3.2     t6      19
3       3.3     t6      24
3       3.4     t6      29
4       3.1     t6      29
5       3.1     t6      29
6       1.1     t6      29
```

7	3.1	t6	29
8	3.1	t6	29
9	3.1	t6	19
10	3.1	t6	24
10	3.2	t6	29

1	1.1	t7	20
1	3.1	t7	25
1	3.2	t7	30
2	1.1	t7	25
2	3.1	t7	30
3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	30
5	3.1	t7	30
6	1.1	t7	30
7	3.1	t7	30
8	3.1	t7	30
9	3.1	t7	20
10	3.1	t7	25
10	3.2	t7	30

1	1.1	t8	21
1	3.1	t8	26
1	3.2	t8	31
2	1.1	t8	26
2	3.1	t8	31
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	31
5	3.1	t8	31
6	1.1	t8	31
7	3.1	t8	31
8	3.1	t8	31
9	3.1	t8	21
10	3.1	t8	26
10	3.2	t8	31

1	1.1	t9	22
1	3.1	t9	27
1	3.2	t9	32
2	1.1	t9	27
2	3.1	t9	32

```
3       3.1     t9      17
3       3.2     t9      22
3       3.3     t9      27
3       3.4     t9      32
4       3.1     t9      32
5       3.1     t9      32
6       1.1     t9      32
7       3.1     t9      32
8       3.1     t9      32
9       3.1     t9      22
10      3.1     t9      27
10      3.2     t9      32

1       1.1     t10     23
1       3.1     t10     28
1       3.2     t10     33
2       1.1     t10     28
2       3.1     t10     33
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     33
5       3.1     t10     33
6       1.1     t10     33
7       3.1     t10     33
8       3.1     t10     33
9       3.1     t10     23
10      3.1     t10     28
10      3.2     t10     33 ;*/

# Set Durations for each task
param d :=
1       1.1     t1      1       2
2       1.1     t1      1       2
6       1.1     t1      1       2

1       1.1     t2      1       1
2       1.1     t2      1       1
6       1.1     t2      1       1

1       1.1     t3      1       1
2       1.1     t3      1       1
6       1.1     t3      1       1

1       1.1     t4      1       1
2       1.1     t4      1       1
6       1.1     t4      1       1
```

1	1.1	t5	1	1
2	1.1	t5	1	1
6	1.1	t5	1	1

1	1.1	t6	1	1
1	3.1	t6	1	1
1	3.2	t6	1	1
2	1.1	t6	1	1
2	3.1	t6	1	1
3	3.1	t6	1	1
3	3.2	t6	1	1
3	3.3	t6	1	1
3	3.4	t6	1	1
4	3.1	t6	1	1
5	3.1	t6	1	1
6	1.1	t6	1	1
7	3.1	t6	1	1
8	3.1	t6	1	1
9	3.1	t6	1	1
10	3.1	t6	1	1
10	3.2	t6	1	1

1	1.1	t7	1	1
1	3.1	t7	1	1
1	3.2	t7	1	1
2	1.1	t7	1	1
2	3.1	t7	1	1
3	3.1	t7	1	1
3	3.2	t7	1	1
3	3.3	t7	1	1
3	3.4	t7	1	1
4	3.1	t7	1	1
5	3.1	t7	1	1
6	1.1	t7	1	1
7	3.1	t7	1	1
8	3.1	t7	1	1
9	3.1	t7	1	1
10	3.1	t7	1	1
10	3.2	t7	1	1

1	1.1	t8	1	1
1	3.1	t8	1	1
1	3.2	t8	1	1
2	1.1	t8	1	1
2	3.1	t8	1	1
3	3.1	t8	1	1
3	3.2	t8	1	1

```
3         3.3       t8        1         1
3         3.4       t8        1         1
4         3.1       t8        1         1
5         3.1       t8        1         1
6         1.1       t8        1         1
7         3.1       t8        1         1
8         3.1       t8        1         1
9         3.1       t8        1         1
10        3.1       t8        1         1
10        3.2       t8        1         1

1         1.1       t9        1         1
1         3.1       t9        1         1
1         3.2       t9        1         1
2         1.1       t9        1         1
2         3.1       t9        1         1
3         3.1       t9        1         1
3         3.2       t9        1         1
3         3.3       t9        1         1
3         3.4       t9        1         1
4         3.1       t9        1         1
5         3.1       t9        1         1
6         1.1       t9        1         1
7         3.1       t9        1         1
8         3.1       t9        1         1
9         3.1       t9        1         1
10        3.1       t9        1         1
10        3.2       t9        1         1

1         1.1       t10       1         1
1         3.1       t10       1         1
1         3.2       t10       1         1
2         1.1       t10       1         1
2         3.1       t10       1         1
3         3.1       t10       1         1
3         3.2       t10       1         1
3         3.3       t10       1         1
3         3.4       t10       1         1
4         3.1       t10       1         1
5         3.1       t10       1         1
6         1.1       t10       1         1
7         3.1       t10       1         1
8         3.1       t10       1         1
9         3.1       t10       1         1
10        3.1       t10       1         1
10        3.2       t10       1         1;
```

```
# Set the  number of resources of each employee (k)- param r is equal to 1 is that (K) is used ; otherwise
0 ( by default)
# index is  V,P,J,G,K,M,1/0
param r :=
1       1.1     t1      PM      PM      1       1
2       1.1     t1      PM      PM      1       1
6       1.1     t1      PM      PM      1       1

1       1.1     t2      QC      QC      1       1
2       1.1     t2      QC      QC      1       1
6       1.1     t2      QC      QC      1       1

1       1.1     t3      OA      OA      1       1
2       1.1     t3      OA      OA      1       1
6       1.1     t3      OA      OA      1       1

1       1.1     t4      QC      QC      1       1
2       1.1     t4      QC      QC      1       1
6       1.1     t4      QC      QC      1       1

1       1.1     t5      PM      PM      1       1
2       1.1     t5      PM      PM      1       1
6       1.1     t5      PM      PM      1       1

1       1.1     t6      OA      OA      1       1
1       3.1     t6      OA      OA      1       1
1       3.2     t6      OA      OA      1       1
2       1.1     t6      OA      OA      1       1
2       3.1     t6      OA      OA      1       1
3       3.1     t6      OA      OA      1       1
3       3.2     t6      OA      OA      1       1
3       3.3     t6      OA      OA      1       1
3       3.4     t6      OA      OA      1       1
4       3.1     t6      OA      OA      1       1
5       3.1     t6      OA      OA      1       1
6       1.1     t6      OA      OA      1       1
7       3.1     t6      OA      OA      1       1
8       3.1     t6      OA      OA      1       1
9       3.1     t6      OA      OA      1       1
10      3.1     t6      OA      OA      1       1
10      3.2     t6      OA      OA      1       1

1       1.1     t7      QC      QC      1       1
1       3.1     t7      QC      QC      1       1
1       3.2     t7      QC      QC      1       1
2       1.1     t7      QC      QC      1       1
2       3.1     t7      QC      QC      1       1
3       3.1     t7      QC      QC      1       1
```

3	3.2	t7	QC	QC	1	1
3	3.3	t7	QC	QC	1	1
3	3.4	t7	QC	QC	1	1
4	3.1	t7	QC	QC	1	1
5	3.1	t7	QC	QC	1	1
6	1.1	t7	QC	QC	1	1
7	3.1	t7	QC	QC	1	1
8	3.1	t7	QC	QC	1	1
9	3.1	t7	QC	QC	1	1
10	3.1	t7	QC	QC	1	1
10	3.2	t7	QC	QC	1	1

1	1.1	t8	OA	OA	1	1
1	3.1	t8	OA	OA	1	1
1	3.2	t8	OA	OA	1	1
2	1.1	t8	OA	OA	1	1
2	3.1	t8	OA	OA	1	1
3	3.1	t8	OA	OA	1	1
3	3.2	t8	OA	OA	1	1
3	3.3	t8	OA	OA	1	1
3	3.4	t8	OA	OA	1	1
4	3.1	t8	OA	OA	1	1
5	3.1	t8	OA	OA	1	1
6	1.1	t8	OA	OA	1	1
7	3.1	t8	OA	OA	1	1
8	3.1	t8	OA	OA	1	1
9	3.1	t8	OA	OA	1	1
10	3.1	t8	OA	OA	1	1
10	3.2	t8	OA	OA	1	1

1	1.1	t9	QC	QC	1	1
1	3.1	t9	QC	QC	1	1
1	3.2	t9	QC	QC	1	1
2	1.1	t9	QC	QC	1	1
2	3.1	t9	QC	QC	1	1
3	3.1	t9	QC	QC	1	1
3	3.2	t9	QC	QC	1	1
3	3.3	t9	QC	QC	1	1
3	3.4	t9	QC	QC	1	1
4	3.1	t9	QC	QC	1	1
5	3.1	t9	QC	QC	1	1
6	1.1	t9	QC	QC	1	1
7	3.1	t9	QC	QC	1	1
8	3.1	t9	QC	QC	1	1
9	3.1	t9	QC	QC	1	1
10	3.1	t9	QC	QC	1	1
10	3.2	t9	QC	QC	1	1

```
1	1.1	t10	PM	PM	1	1
1	3.1	t10	PM	PM	1	1
1	3.2	t10	PM	PM	1	1
2	1.1	t10	PM	PM	1	1
2	3.1	t10	PM	PM	1	1
3	3.1	t10	PM	PM	1	1
3	3.2	t10	PM	PM	1	1
3	3.3	t10	PM	PM	1	1
3	3.4	t10	PM	PM	1	1
4	3.1	t10	PM	PM	1	1
5	3.1	t10	PM	PM	1	1
6	1.1	t10	PM	PM	1	1
7	3.1	t10	PM	PM	1	1
8	3.1	t10	PM	PM	1	1
9	3.1	t10	PM	PM	1	1
10	3.1	t10	PM	PM	1	1
10	3.2	t10	PM	PM	1	1

1	1.1	t10	OA	OA	1	1
1	3.1	t10	OA	OA	1	1
1	3.2	t10	OA	OA	1	1
2	1.1	t10	OA	OA	1	1
2	3.1	t10	OA	OA	1	1
3	3.1	t10	OA	OA	1	1
3	3.2	t10	OA	OA	1	1
3	3.3	t10	OA	OA	1	1
3	3.4	t10	OA	OA	1	1
4	3.1	t10	OA	OA	1	1
5	3.1	t10	OA	OA	1	1
6	1.1	t10	OA	OA	1	1
7	3.1	t10	OA	OA	1	1
8	3.1	t10	OA	OA	1	1
9	3.1	t10	OA	OA	1	1
10	3.1	t10	OA	OA	1	1
10	3.2	t10	OA	OA	1	1
;


end;
```

Appendix J: GLPK Code to Experiment Model (II)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede task j
set M{v in V, p in P[v],j in J[v,p]}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]},default 0; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}, default 50; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}, default 50; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 0; /* duration of tasks j of process p of project
v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k usage
of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 0; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
param BR{v in V, g in G, k in K[g]}, default 1; /* Amount of renewable resource k of group g*/
param R{g in G, k in K[g]}, default 0; /* total amount of renewable resource k*/

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
#var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]}>=0;/*Weighted tardiness cost of process p of project v*/
```

```
#var BR {v in V,g in G, k in K[g]} integer;
#var R {g in G, k in K[g]} integer;

minimize z: sum {v in V, p in P[v]} TC1[v,p]; /* minimize the weighted tardiness of the project and
process*/
#minimize z: sum{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T}x[v,p,j,m,t];

#minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */

s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 mode for every task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in EE[v,p,jF]..LL[v,p,jF],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,s]..L[v,s],m in M[v,s,jL]}(t)*x[v,s,jL,m,t];/* predence
relationship for processes*/

#s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource  k of group g for each project v   */

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V,p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= R[g,k]; /* capacity constraint
for each resource k of group g   */

#s.t. Constraint_5{v in V, p in p_N[v],j in j_L[v,p]}: TC[v] >= c[v] *( sum{m in M[v,p,j],t in E[v,p]..L[v,p]}
(t*x[v,p,j,m,t])- dd[v]); /* tardiness of project is equal to some penalty c[v] times the difference of the
finish time of the last task of the last process minus the due date of the project */

s.t. Constraint_6{v in V, p in P[v], j in j_L[v,p], g in G}: TC1[v,p] >= c1[v,p]*(sum{m in M[v,p,j], t in
EE[v,p,j]..LL[v,p,j]} (t*x[v,p,j,m,t])- ddd[v,p]); /* tardiness of process is equal to some penalty c[v,p] times
the difference of the finish time of the last task of each process minus the due date of the process */


solve;

display z,x,BR;

data;
param projects :=10;
param resourceGroup := 6;
param timePeriods := 50;

# Define Processes and Orders for each Project- The decimal number indicates the order number
```

```
#Every project has a dummy starting process called "p0" with 0 duration.

set P[1] := p0    1.1       3.1       3.2;
set P[2] := p0    1.1       3.1 ;
set P[3] := p0    3.1       3.2       3.3       3.4;
set P[4] := p0    3.1;
set P[5] := p0    3.1;
set P[6] := p0    1.1;
set P[7] := p0    3.1;
set P[8] := p0    3.1;
set P[9] := p0    3.1;
set P[10] := p0  3.1       3.2;


# Define tasks for each process:
set J[1,p0] := t0;
set J[1,1.1] := t1t2       t3        t4        t5        t6        t7        t8        t9        t10;
set J[1,3.1] := t6t7       t8        t9        t10;
set J[1,3.2] := t6t7       t8        t9        t10;

set J[2,p0] := t0;
set J[2,1.1] := t1t2       t3        t4        t5        t6        t7        t8        t9        t10;
set J[2,3.1] := t6t7       t8        t9        t10;

set J[3,p0]:=t0;
set J[3,3.1]:=t6  t7       t8        t9        t10;
set J[3,3.2]:=t6  t7       t8        t9        t10;
set J[3,3.3]:=t6  t7       t8        t9        t10;
set J[3,3.4]:=t6  t7       t8        t9        t10;

set J[4,p0] :=t0;
set J[4,3.1]:= t6 t7       t8        t9        t10;

set J[5,p0] :=t0;
set J[5,3.1] := t6t7       t8        t9        t10;

set J[6,p0] :=t0;
set J[6,1.1] := t1t2       t3        t4        t5        t6        t7        t8        t9        t10;

set J[7,p0] :=t0;
set J[7,3.1] := t6t7       t8        t9        t10;

set J[8,p0] :=t0;
set J[8,3.1] := t6t7       t8        t9        t10;

set J[9,p0] :=t0;
set J[9,3.1] := t6t7       t8        t9        t10;
set J[10,p0] :=t0;
set J[10,3.1] :=t6          t7          t8          t9          t10;
set J[10,3.2] :=t6          t7          t8          t9          t10;

# Define Mode
set M[1,p0,t0]:=1;
set M[1,1.1,t1]:=1;
set M[1,1.1,t2]:= 1;
set M[1,1.1,t3]:= 1;
set M[1,1.1,t4]:= 1;
set M[1,1.1,t5]:= 1;
set M[1,1.1,t6]:= 1;
set M[1,1.1,t7]:= 1;
set M[1,1.1,t8]:= 1;
set M[1,1.1,t9]:= 1;
set M[1,1.1,t10]:= 1;

set M[1,3.1,t6]:= 1;
set M[1,3.1,t7]:= 1;
set M[1,3.1,t8]:= 1;
set M[1,3.1,t9]:= 1;
set M[1,3.1,t10]:= 1;

set M[1,3.2,t6]:= 1;
set M[1,3.2,t7]:= 1;
set M[1,3.2,t8]:= 1;
set M[1,3.2,t9]:= 1;
set M[1,3.2,t10]:= 1;

set M[2,p0,t0]:=1;
set M[2,1.1,t1]:=1;
set M[2,1.1,t2]:= 1;
set M[2,1.1,t3]:= 1;
set M[2,1.1,t4]:= 1;
set M[2,1.1,t5]:= 1;
set M[2,1.1,t6]:= 1;
set M[2,1.1,t7]:= 1;
set M[2,1.1,t8]:= 1;
set M[2,1.1,t9]:= 1;
set M[2,1.1,t10]:= 1;

set M[2,3.1,t6]:= 1;
set M[2,3.1,t7]:= 1;
set M[2,3.1,t8]:= 1;
set M[2,3.1,t9]:= 1;
set M[2,3.1,t10]:= 1;

set M[3,p0,t0]:=1;
set M[3,3.1,t6]:= 1;
set M[3,3.1,t7]:= 1;
set M[3,3.1,t8]:= 1;
set M[3,3.1,t9]:= 1;
set M[3,3.1,t10]:= 1;

set M[3,3.2,t6]:= 1;
set M[3,3.2,t7]:= 1;
set M[3,3.2,t8]:= 1;
set M[3,3.2,t9]:= 1;
set M[3,3.2,t10]:= 1;

set M[3,3.3,t6]:= 1;
set M[3,3.3,t7]:= 1;
set M[3,3.3,t8]:= 1;
set M[3,3.3,t9]:= 1;
set M[3,3.3,t10]:= 1;

set M[3,3.4,t6]:= 1;
set M[3,3.4,t7]:= 1;
set M[3,3.4,t8]:= 1;
set M[3,3.4,t9]:= 1;
set M[3,3.4,t10]:= 1;

set M[4,p0,t0]:=1;
set M[4,3.1,t6]:= 1;
set M[4,3.1,t7]:= 1;
set M[4,3.1,t8]:= 1;
set M[4,3.1,t9]:= 1;
set M[4,3.1,t10]:= 1;

set M[5,p0,t0]:=1;
set M[5,3.1,t6]:= 1;
set M[5,3.1,t7]:= 1;
set M[5,3.1,t8]:= 1;
set M[5,3.1,t9]:= 1;
set M[5,3.1,t10]:= 1;

set M[6,p0,t0]:= 2;
set M[6,1.1,t1]:= 2;
set M[6,1.1,t2]:= 2;
set M[6,1.1,t3]:= 2;
set M[6,1.1,t4]:= 2;
set M[6,1.1,t5]:= 2;
set M[6,1.1,t6]:= 2;
set M[6,1.1,t7]:= 2;
set M[6,1.1,t8]:= 2;
set M[6,1.1,t9]:= 2;
set M[6,1.1,t10]:= 2;

set M[7,p0,t0]:= 2;
set M[7,3.1,t6]:= 2;
set M[7,3.1,t7]:= 2;
set M[7,3.1,t8]:= 2;
set M[7,3.1,t9]:= 2;
set M[7,3.1,t10]:= 2;

set M[8,p0,t0]:= 2;
set M[8,3.1,t6]:= 2;
set M[8,3.1,t7]:= 2;
set M[8,3.1,t8]:= 2;
set M[8,3.1,t9]:= 2;
set M[8,3.1,t10]:= 2;

set M[9,p0,t0]:= 2;
set M[9,3.1,t6]:= 2;
set M[9,3.1,t7]:= 2;
set M[9,3.1,t8]:= 2;
set M[9,3.1,t9]:= 2;
set M[9,3.1,t10]:= 2;

set M[10,p0,t0]:= 2;

set M[10,3.1,t6]:= 2;
set M[10,3.1,t7]:= 2;
set M[10,3.1,t8]:= 2;
set M[10,3.1,t9]:= 2;
set M[10,3.1,t10]:= 2;

set M[10,3.2,t6]:= 2;
set M[10,3.2,t7]:= 2;
set M[10,3.2,t8]:= 2;
set M[10,3.2,t9]:= 2;
set M[10,3.2,t10]:= 2;


# Task Precedence
set Q[1,1.1,t2] := t1;
set Q[1,1.1,t3] := t2;
set Q[1,1.1,t4]:= t3;
set Q[1,1.1,t5] := t4;
set Q[1,1.1,t6] := t5;
set Q[1,1.1,t7]:= t6;
set Q[1,1.1,t8]:= t7;
set Q[1,1.1,t9]:= t8;
set Q[1,1.1,t10]:= t9;
set Q[1,3.1,t7]:= t6;
set Q[1,3.1,t8] := t7;
set Q[1,3.1,t9]:= t8;
set Q[1,3.1,t10]:= t9;

set Q[1,3.2,t7]:= t6;
set Q[1,3.2,t8] := t7;
set Q[1,3.2,t9]:= t8;
set Q[1,3.2,t10]:= t9;

set Q[2,1.1,t2] := t1;
set Q[2,1.1,t3] := t2;
set Q[2,1.1,t4]:= t3;
set Q[2,1.1,t5] := t4;
set Q[2,1.1,t6] := t5;
set Q[2,1.1,t7]:= t6;
set Q[2,1.1,t8]:= t7;
set Q[2,1.1,t9]:= t8;
set Q[2,1.1,t10]:= t9;

set Q[2,3.1,t7]:= t6;
set Q[2,3.1,t8] := t7;
set Q[2,3.1,t9]:= t8;
set Q[2,3.1,t10]:= t9;

set Q[3,3.1,t7]:= t6;
set Q[3,3.1,t8] := t7;
set Q[3,3.1,t9]:= t8;
set Q[3,3.1,t10]:= t9;

set Q[3,3.2,t7]:= t6;
set Q[3,3.2,t8] := t7;
set Q[3,3.2,t9]:= t8;
set Q[3,3.2,t10]:= t9;

set Q[3,3.3,t7]:= t6;
set Q[3,3.3,t8] := t7;
set Q[3,3.3,t9]:= t8;
set Q[3,3.3,t10]:= t9;

set Q[3,3.4,t7]:= t6;
set Q[3,3.4,t8] := t7;
set Q[3,3.4,t9]:= t8;
set Q[3,3.4,t10]:= t9;

set Q[4,3.1,t7]:= t6;
set Q[4,3.1,t8] := t7;
set Q[4,3.1,t9]:= t8;
set Q[4,3.1,t10]:= t9;

set Q[5,3.1,t7]:= t6;
set Q[5,3.1,t8] := t7;
set Q[5,3.1,t9]:= t8;
set Q[5,3.1,t10]:= t9;

set Q[6,1.1,t2] := t1;
set Q[6,1.1,t3] := t2;
set Q[6,1.1,t4]:= t3;
set Q[6,1.1,t5] := t4;
set Q[6,1.1,t6] := t5;
set Q[6,1.1,t7]:= t6;
set Q[6,1.1,t8]:= t7;
set Q[6,1.1,t9]:= t8;
set Q[6,1.1,t10]:= t9;

set Q[7,3.1,t7]:= t6;
set Q[7,3.1,t8] := t7;
set Q[7,3.1,t9]:= t8;
set Q[7,3.1,t10]:= t9;

set Q[8,3.1,t7]:= t6;
set Q[8,3.1,t8] := t7;
set Q[8,3.1,t9]:= t8;
set Q[8,3.1,t10]:= t9;

set Q[9,3.1,t7]:= t6;
set Q[9,3.1,t8] := t7;
set Q[9,3.1,t9]:= t8;
set Q[9,3.1,t10]:= t9;

set Q[10,3.1,t7]:= t6;
set Q[10,3.1,t8] := t7;
set Q[10,3.1,t9]:= t8;
set Q[10,3.1,t10]:= t9;

set Q[10,3.2,t7]:= t6;
set Q[10,3.2,t8] := t7;
set Q[10,3.2,t9]:= t8;
set Q[10,3.2,t10]:= t9;


# Process Precedence
set S[1,1.1] := p0;
set S[1,3.1] := 1.1;
set S[1,3.2] := 3.1;
set S[2,1.1] := p0;
set S[2,3.1] := 1.1;

set S[3,3.1] :=p0;
set S[3,3.2] :=3.1;
set S[3,3.3] :=3.2;
set S[3,3.4] :=3.3;

set S[4,3.1] :=p0;

set S[5,3.1] :=p0;

set S[6,1.1] :=p0;

set S[7,3.1] :=p0;

set S[8,3.1] :=p0;

set S[9,3.1] :=p0;

set S[10,3.1] :=p0;
set S[10,3.2] :=3.1;

# Last task of each process
set j_L[1,p0] := t0;
set j_L[1,1.1] := t10;
set j_L[1,3.1] := t10;
set j_L[1,3.2] := t10;

set j_L[2,p0] := t0;
set j_L[2,1.1] := t10;
set j_L[2,3.1] := t10;

set j_L[3,3.1] := t10;
set j_L[3,3.2] := t10;
set j_L[3,3.3] := t10;
set j_L[3,3.4] := t10;

set j_L[4,3.1] := t10;

set j_L[5,3.1] := t10;

set j_L[6,1.1] := t10;

set j_L[7,3.1] := t10;

set j_L[8,3.1] := t10;
set j_L[9,3.1] := t10;

set j_L[10,3.1] := t10;
set j_L[10,3.2] := t10;

# First task of each process
set j_F[1,p0] := t0;
set j_F[1,1.1] := t1;
set j_F[1,3.1] := t6;
set j_F[1,3.2] := t6;

set j_F[2,p0] := t0;
set j_F[2,1.1] := t1;
set j_F[2,3.1] := t6;

set j_F[3,p0] := t0;
set j_F[3,3.1] := t6;
set j_F[3,3.2] := t6;
set j_F[3,3.3] := t6;
set j_F[3,3.4] := t6;

set j_F[4,p0] := t0;
set j_F[4,3.1] := t6;

set j_F[5,p0] := t0;
set j_F[5,3.1] := t6;

set j_F[6,p0] := t0;
set j_F[6,1.1] := t1;

set j_F[7,p0] := t0;
set j_F[7,3.1] := t6;

set j_F[8,p0] := t0;
set j_F[8,3.1] := t6;

set j_F[9,p0] := t0;
set j_F[9,3.1] := t6;

set j_F[10,p0] := t0;
set j_F[10,3.1] := t6;
set j_F[10,3.2] := t6;

# The group of people
set G:= PM1 PM2 QC1 QC2 OA1 OA2;

set K[PM1] := PM1; # The PM people
set K[OA1] := OA1 ; #The OA people
set K[QC1] := QC1 ; #The QC people

set K[PM2] := PM2; # The PM people
set K[OA2] := OA2 ; #The OA people
set K[QC2] := QC2 ; #The QC people

# k and g are equal in this example

# set the total number of each resource k
param R :=
PM1 PM1 1
OA1 OA1 1
QC1 QC1 1
PM2 PM2 1
OA2 OA2 1
QC2 QC2 1;


# Set the due dates for each process
param ddd :=
1       p0      0
1       1.1     19
1       3.1     24
1       3.2     29

2       p0      0
2       1.1     19
2       3.1     24

3       p0      0
3       3.1     20
3       3.2     25
3       3.3     30
3       3.4     35

4       p0      0
4       3.1     21

5       p0      0
5       3.1     20

6       p0      0
6       1.1     23

7       p0      0
7       3.1     22
```

```
8       p0      0
8       3.1     21

9       p0      0
9       3.1     22

10      p0      0
10      3.1     23
10      3.2     28
;

# Set the early finish of each process
param E :=
1       p0      0
1       1.1     15
1       3.1     22
1       3.2     27

2       p0      0
2       1.1     16
2       3.1     22

3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33

4       p0      0
4       3.1     19

5       p0      0
5       3.1     18

6       p0      0
6       1.1     19

7       p0      0
7       3.1     20

8       p0      0
8       3.1     19

9       p0      0
9       3.1     20

10      p0      0
10      3.1     21
```

```
10      3.2     26

;

# Set the late finish of each process
param L :=
1       p0      0
1       1.1     23
1       3.1     28
1       3.2     33
2       p0      0
2       1.1     28
2       3.1     33
3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33
4       p0      0
4       3.1     33
5       p0      0
5       3.1     33
6       p0      0
6       1.1     33
7       p0      0
7       3.1     33
8       p0      0
8       3.1     33
9       p0      0
9       3.1     23
10      p0      0
10      3.1     28
10      3.2     33
;

# Set the early finsh of each task
param EE :=
1 p0 t0 0
1       1.1     t1      7
2       1.1     t1      7
6       1.1     t1      11

1       1.1     t2      8
2       1.1     t2      8
6       1.1     t2      12

1       1.1     t3      9
2       1.1     t3      9
```

6	1.1	t3	13
1	1.1	t4	10
2	1.1	t4	10
6	1.1	t4	14
1	1.1	t5	11
2	1.1	t5	11
6	1.1	t5	15
1	1.1	t6	12
1	3.1	t6	18
1	3.2	t6	23
2	1.1	t6	12
2	3.1	t6	18
3	3.1	t6	14
3	3.2	t6	19
3	3.3	t6	24
3	3.4	t6	29
4	3.1	t6	15
5	3.1	t6	14
6	1.1	t6	10
7	3.1	t6	16
8	3.1	t6	15
9	3.1	t6	16
10	3.1	t6	17
10	3.2	t6	22
1	1.1	t7	13
1	3.1	t7	19
1	3.2	t7	24
2	1.1	t7	13
2	3.1	t7	19
3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	16
5	3.1	t7	15
6	1.1	t7	11
7	3.1	t7	17
8	3.1	t7	16
9	3.1	t7	17
10	3.1	t7	18
10	3.2	t7	23
1	1.1	t8	14
1	3.1	t8	20

1	3.2	t8	25
2	1.1	t8	14
2	3.1	t8	20
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	17
5	3.1	t8	16
6	1.1	t8	12
7	3.1	t8	18
8	3.1	t8	17
9	3.1	t8	18
10	3.1	t8	19
10	3.2	t8	24

1	1.1	t9	15
1	3.1	t9	21
1	3.2	t9	26
2	1.1	t9	15
2	3.1	t9	21
3	3.1	t9	17
3	3.2	t9	22
3	3.3	t9	27
3	3.4	t9	32
4	3.1	t9	18
5	3.1	t9	17
6	1.1	t9	13
7	3.1	t9	19
8	3.1	t9	18
9	3.1	t9	19
10	3.1	t9	20
10	3.2	t9	25

1	1.1	t10	16
1	3.1	t10	22
1	3.2	t10	27
2	1.1	t10	16
2	3.1	t10	22
3	3.1	t10	18
3	3.2	t10	23
3	3.3	t10	28
3	3.4	t10	33
4	3.1	t10	19
5	3.1	t10	18
6	1.1	t10	14
7	3.1	t10	20
8	3.1	t10	19

```
9       3.1     t10     20
10      3.1     t10     21
10      3.2     t10     26;

# Set the  late finish of each task
/*param LL :=

1       1.1     t1      14
2       1.1     t1      19
6       1.1     t1      24

1       1.1     t2      15
2       1.1     t2      20
6       1.1     t2      25

1       1.1     t3      15
2       1.1     t3      20
6       1.1     t3      25

1       1.1     t5      18
2       1.1     t5      23
6       1.1     t5      28

1       1.1     t6      19
1       3.1     t6      24
1       3.2     t6      29
2       1.1     t6      24
2       3.1     t6      29
3       3.1     t6      14
3       3.2     t6      19
3       3.3     t6      24
3       3.4     t6      29
4       3.1     t6      29
5       3.1     t6      29
6       1.1     t6      29
7       3.1     t6      29
8       3.1     t6      29
9       3.1     t6      19
10      3.1     t6      24
10      3.2     t6      29

1       1.1     t7      20
1       3.1     t7      25
1       3.2     t7      30
2       1.1     t7      25
2       3.1     t7      30
3       3.1     t7      15
3       3.2     t7      20
```

3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	30
5	3.1	t7	30
6	1.1	t7	30
7	3.1	t7	30
8	3.1	t7	30
9	3.1	t7	20
10	3.1	t7	25
10	3.2	t7	30

1	1.1	t8	21
1	3.1	t8	26
1	3.2	t8	31
2	1.1	t8	26
2	3.1	t8	31
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	31
5	3.1	t8	31
6	1.1	t8	31
7	3.1	t8	31
8	3.1	t8	31
9	3.1	t8	21
10	3.1	t8	26
10	3.2	t8	31

1	1.1	t9	22
1	3.1	t9	27
1	3.2	t9	32
2	1.1	t9	27
2	3.1	t9	32
3	3.1	t9	17
3	3.2	t9	22
3	3.3	t9	27
3	3.4	t9	32
4	3.1	t9	32
5	3.1	t9	32
6	1.1	t9	32
7	3.1	t9	32
8	3.1	t9	32
9	3.1	t9	22
10	3.1	t9	27
10	3.2	t9	32

```
1       1.1     t10     23
1       3.1     t10     28
1       3.2     t10     33
2       1.1     t10     28
2       3.1     t10     33
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     33
5       3.1     t10     33
6       1.1     t10     33
7       3.1     t10     33
8       3.1     t10     33
9       3.1     t10     23
10      3.1     t10     28
10      3.2     t10     33 ;*/

# Set Durations for each task
param d :=
1       1.1     t1      1       2
2       1.1     t1      1       2
6       1.1     t1      2       2

1       1.1     t2      1       1
2       1.1     t2      1       1
6       1.1     t2      2       1

1       1.1     t3      1       1
2       1.1     t3      1       1
6       1.1     t3      2       1

1       1.1     t4      1       1
2       1.1     t4      1       1
6       1.1     t4      2       1

1       1.1     t5      1       1
2       1.1     t5      1       1
6       1.1     t5      2       1

1       1.1     t6      1       1
1       3.1     t6      1       1
1       3.2     t6      1       1
2       1.1     t6      1       1
2       3.1     t6      1       1
3       3.1     t6      1       1
3       3.2     t6      1       1
3       3.3     t6      1       1
```

3	3.4	t6	1	1
4	3.1	t6	1	1
5	3.1	t6	1	1
6	1.1	t6	2	1
7	3.1	t6	2	1
8	3.1	t6	2	1
9	3.1	t6	2	1
10	3.1	t6	2	1
10	3.2	t6	2	1

1	1.1	t7	1	1
1	3.1	t7	1	1
1	3.2	t7	1	1
2	1.1	t7	1	1
2	3.1	t7	1	1
3	3.1	t7	1	1
3	3.2	t7	1	1
3	3.3	t7	1	1
3	3.4	t7	1	1
4	3.1	t7	1	1
5	3.1	t7	1	1
6	1.1	t7	2	1
7	3.1	t7	2	1
8	3.1	t7	2	1
9	3.1	t7	2	1
10	3.1	t7	2	1
10	3.2	t7	2	1

1	1.1	t8	1	1
1	3.1	t8	1	1
1	3.2	t8	1	1
2	1.1	t8	1	1
2	3.1	t8	1	1
3	3.1	t8	1	1
3	3.2	t8	1	1
3	3.3	t8	1	1
3	3.4	t8	1	1
4	3.1	t8	1	1
5	3.1	t8	1	1
6	1.1	t8	2	1
7	3.1	t8	2	1
8	3.1	t8	2	1
9	3.1	t8	2	1
10	3.1	t8	2	1
10	3.2	t8	2	1

1	1.1	t9	1	1
1	3.1	t9	1	1

```
1       3.2     t9      1       1
2       1.1     t9      1       1
2       3.1     t9      1       1
3       3.1     t9      1       1
3       3.2     t9      1       1
3       3.3     t9      1       1
3       3.4     t9      1       1
4       3.1     t9      1       1
5       3.1     t9      1       1
6       1.1     t9      2       1
7       3.1     t9      2       1
8       3.1     t9      2       1
9       3.1     t9      2       1
10      3.1     t9      2       1
10      3.2     t9      2       1

1       1.1     t10     1       1
1       3.1     t10     1       1
1       3.2     t10     1       1
2       1.1     t10     1       1
2       3.1     t10     1       1
3       3.1     t10     1       1
3       3.2     t10     1       1
3       3.3     t10     1       1
3       3.4     t10     1       1
4       3.1     t10     1       1
5       3.1     t10     1       1
6       1.1     t10     2       1
7       3.1     t10     2       1
8       3.1     t10     2       1
9       3.1     t10     2       1
10      3.1     t10     2       1
10      3.2     t10     2       1;

# Set the  number of resources of each employee (k)- param r is equal to 1 is that (K) is used ; otherwise 0 ( by default)
# index is  V,P,J,G,K,M,1/0
param r :=
1       1.1     t1      PM1     PM1     1       1
2       1.1     t1      PM1     PM1     1       1
6       1.1     t1      PM2     PM2     2       1

1       1.1     t2      QC1     QC1     1       1
2       1.1     t2      QC1     QC1     1       1
6       1.1     t2      QC2     QC2     2       1

1       1.1     t3      OA1     OA1     1       1
2       1.1     t3      OA1     OA1     1       1
```

6	1.1	t3	OA2	OA2	2	1
1	1.1	t4	QC1	QC1	1	1
2	1.1	t4	QC1	QC1	1	1
6	1.1	t4	QC2	QC2	2	1
1	1.1	t5	PM1	PM1	1	1
2	1.1	t5	PM1	PM1	1	1
6	1.1	t5	PM2	PM2	2	1
1	1.1	t6	OA1	OA1	1	1
1	3.1	t6	OA1	OA1	1	1
1	3.2	t6	OA1	OA1	1	1
2	1.1	t6	OA1	OA1	1	1
2	3.1	t6	OA1	OA1	1	1
3	3.1	t6	OA1	OA1	1	1
3	3.2	t6	OA1	OA1	1	1
3	3.3	t6	OA1	OA1	1	1
3	3.4	t6	OA1	OA1	1	1
4	3.1	t6	OA1	OA1	1	1
5	3.1	t6	OA1	OA1	1	1
6	1.1	t6	OA2	OA2	2	1
7	3.1	t6	OA2	OA2	2	1
8	3.1	t6	OA2	OA2	2	1
9	3.1	t6	OA2	OA2	2	1
10	3.1	t6	OA2	OA2	2	1
10	3.2	t6	OA2	OA2	2	1
1	1.1	t7	QC1	QC1	1	1
1	3.1	t7	QC1	QC1	1	1
1	3.2	t7	QC1	QC1	1	1
2	1.1	t7	QC1	QC1	1	1
2	3.1	t7	QC1	QC1	1	1
3	3.1	t7	QC1	QC1	1	1
3	3.2	t7	QC1	QC1	1	1
3	3.3	t7	QC1	QC1	1	1
3	3.4	t7	QC1	QC1	1	1
4	3.1	t7	QC1	QC1	1	1
5	3.1	t7	QC1	QC1	1	1
6	1.1	t7	QC2	QC2	2	1
7	3.1	t7	QC2	QC2	2	1
8	3.1	t7	QC2	QC2	2	1
9	3.1	t7	QC2	QC2	2	1
10	3.1	t7	QC2	QC2	2	1
10	3.2	t7	QC2	QC2	2	1
1	1.1	t8	OA1	OA1	1	1
1	3.1	t8	OA1	OA1	1	1

1	3.2	t8	OA1	OA1	1	1
2	1.1	t8	OA1	OA1	1	1
2	3.1	t8	OA1	OA1	1	1
3	3.1	t8	OA1	OA1	1	1
3	3.2	t8	OA1	OA1	1	1
3	3.3	t8	OA1	OA1	1	1
3	3.4	t8	OA1	OA1	1	1
4	3.1	t8	OA1	OA1	1	1
5	3.1	t8	OA1	OA1	1	1
6	1.1	t8	OA2	OA2	2	1
7	3.1	t8	OA2	OA2	2	1
8	3.1	t8	OA2	OA2	2	1
9	3.1	t8	OA2	OA2	2	1
10	3.1	t8	OA2	OA2	2	1
10	3.2	t8	OA2	OA2	2	1

1	1.1	t9	QC1	QC1	1	1
1	3.1	t9	QC1	QC1	1	1
1	3.2	t9	QC1	QC1	1	1
2	1.1	t9	QC1	QC1	1	1
2	3.1	t9	QC1	QC1	1	1
3	3.1	t9	QC1	QC1	1	1
3	3.2	t9	QC1	QC1	1	1
3	3.3	t9	QC1	QC1	1	1
3	3.4	t9	QC1	QC1	1	1
4	3.1	t9	QC1	QC1	1	1
5	3.1	t9	QC1	QC1	1	1
6	1.1	t9	QC2	QC2	2	1
7	3.1	t9	QC2	QC2	2	1
8	3.1	t9	QC2	QC2	2	1
9	3.1	t9	QC2	QC2	2	1
10	3.1	t9	QC2	QC2	2	1
10	3.2	t9	QC2	QC2	2	1

1	1.1	t10	PM1	PM1	1	1
1	3.1	t10	PM1	PM1	1	1
1	3.2	t10	PM1	PM1	1	1
2	1.1	t10	PM1	PM1	1	1
2	3.1	t10	PM1	PM1	1	1
3	3.1	t10	PM1	PM1	1	1
3	3.2	t10	PM1	PM1	1	1
3	3.3	t10	PM1	PM1	1	1
3	3.4	t10	PM1	PM1	1	1
4	3.1	t10	PM1	PM1	1	1
5	3.1	t10	PM1	PM1	1	1
6	1.1	t10	PM2	PM2	2	1
7	3.1	t10	PM2	PM2	2	1
8	3.1	t10	PM2	PM2	2	1

```
9       3.1     t10     PM2     PM2     2       1
10      3.1     t10     PM2     PM2     2       1
10      3.2     t10     PM2     PM2     2       1

1       1.1     t10     OA1     OA1     1       1
1       3.1     t10     OA1     OA1     1       1
1       3.2     t10     OA1     OA1     1       1
2       1.1     t10     OA1     OA1     1       1
2       3.1     t10     OA1     OA1     1       1
3       3.1     t10     OA1     OA1     1       1
3       3.2     t10     OA1     OA1     1       1
3       3.3     t10     OA1     OA1     1       1
3       3.4     t10     OA1     OA1     1       1
4       3.1     t10     OA1     OA1     1       1
5       3.1     t10     OA1     OA1     1       1
6       1.1     t10     OA2     OA2     2       1
7       3.1     t10     OA2     OA2     2       1
8       3.1     t10     OA2     OA2     2       1
9       3.1     t10     OA2     OA2     2       1
10      3.1     t10     OA2     OA2     2       1
10      3.2     t10     OA2     OA2     2       1
;


end;
```

Appendix K: GLPK Code to Experiment Model (III)

```
param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups

set W, default (1.. work group); # Set of project work groups
set V, default (1..projects); #Set of projects
set G /*default (1..resourceGroup)*/;# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede task j
set M{v in V, p in P[v],j in J[v,p]}, default {1,2}; # set of modes for task j of process p of project v
set MW{w in W}, default {}; # set of modes in project work group w
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]},default 0; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}, default 50; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}, default 50; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 0; /* duration of tasks j of process p of project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 0; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR{v in V, g in G, k in K[g]}, default 1; /* Amount of renewable resource k of group g*/
param R{g in G, k in K[g]}, default 0; /* Represnts total amount of resources of group g*/



#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
#var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]}>=0, integer;/*Weighted tardiness cost of process p of project v*/
```

```
#var BR {v in V,g in G, k in K[g]} integer;
#var R {g in G, k in K[g]} integer; /* Represnts total amount of resources of group g*/
var y {v in V, w in W}, binary; /* Binary variable y to assign to work group to each project*/

minimize z: sum {v in V, p in P[v]} TC1[v,p]; /* minimize the weighted tardiness of the project and
process*/
#minimize z: sum{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T}x[v,p,j,m,t];

#minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */

s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 mode for every task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in EE[v,p,jF]..LL[v,p,jF],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,s]..L[v,s],m in M[v,s,jL]}(t)*x[v,s,jL,m,t];/* predence
relationship for processes*/

#s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource  k of group g for each project v   */

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V,p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= R[g,k]; /* capacity constraint
for each resource k of group g   */

#s.t. Constraint_5{v in V, p in p_N[v],j in j_L[v,p]}: TC[v] >= c[v] *( sum{m in M[v,p,j],t in E[v,p]..L[v,p]}
(t*x[v,p,j,m,t])- dd[v]); /* tardiness of project is equal to some penalty c[v] times the difference of the
finish time of the last task of the last process minus the due date of the project */

s.t. Constraint_6{v in V, p in P[v], j in j_L[v,p], g in G}: TC1[v,p] >= c1[v,p]*(sum{m in M[v,p,j], t in
EE[v,p,j]..LL[v,p,j]} (t*x[v,p,j,m,t])- ddd[v,p]); /* tardiness of process is equal to some penalty c[v,p] times
the difference of the finish time of the last task of each process minus the due date of the process */
 s.t. Constraint_7 {v in V}:sum {w in W} y[v,w]=1; # assign a project to only one work group

 s.t. Constarint_8 {v in V,  p in P[v], j in J[v,p],w in W,m in MW[w],t in T}:x[v,p,j,m,t]<=y[v,w]; #use the
resources m from the work group that project v is part of.

solve;

display z,x,R,y;

data;
param projects :=10;
param resourceGroup := 6;
param timePeriods := 50;
param work group:=2;

# Define Processes and Orders for each Project- The decimal number indicates the order number
#Every project has a dummy starting process called "p0" with 0 duration.

set P[1] := p0    1.1       3.1       3.2;
set P[2] := p0    1.1       3.1 ;
set P[3] := p0    3.1       3.2       3.3       3.4;
set P[4] := p0    3.1;
set P[5] := p0    3.1;
set P[6] := p0    1.1;
set P[7] := p0    3.1;
set P[8] := p0    3.1;
set P[9] := p0    3.1;
set P[10] := p0   3.1       3.2;


# Define tasks for each process:
set J[1,p0] := t0;
set J[1,1.1] := t1t2       t3        t4        t5        t6        t7        t8        t9        t10;
set J[1,3.1] := t6t7       t8        t9        t10;
set J[1,3.2] := t6t7       t8        t9        t10;

set J[2,p0] := t0;
set J[2,1.1] := t1t2       t3        t4        t5        t6        t7        t8        t9        t10;
set J[2,3.1] := t6t7       t8        t9        t10;

set J[3,p0]:=t0;
set J[3,3.1]:=t6  t7       t8        t9        t10;
set J[3,3.2]:=t6  t7       t8        t9        t10;
set J[3,3.3]:=t6  t7       t8        t9        t10;
set J[3,3.4]:=t6  t7       t8        t9        t10;

set J[4,p0] :=t0;
set J[4,3.1]:= t6 t7       t8        t9        t10;

set J[5,p0] :=t0;
set J[5,3.1] := t6t7       t8        t9        t10;

set J[6,p0] :=t0;
set J[6,1.1] := t1t2       t3        t4        t5        t6        t7        t8        t9        t10;

set J[7,p0] :=t0;
set J[7,3.1] := t6t7       t8        t9        t10;

set J[8,p0] :=t0;
set J[8,3.1] := t6t7        t8        t9        t10;

set J[9,p0] :=t0;
set J[9,3.1] := t6t7        t8        t9        t10;

set J[10,p0] :=t0;
set J[10,3.1] :=t6          t7        t8        t9        t10;
set J[10,3.2] :=t6          t7        t8        t9        t10;

# Define Mode- by default give modes (1,2,3) to each task

set MW[1]:=1; # set of modes of work group 1.
set MW[2]:=2; # set of modes of work group 2


# Task Precedence
set Q[1,1.1,t2] := t1;
set Q[1,1.1,t3] := t2;
set Q[1,1.1,t4]:= t3;
set Q[1,1.1,t5] := t4;
set Q[1,1.1,t6] := t5;
set Q[1,1.1,t7]:= t6;
set Q[1,1.1,t8]:= t7;
set Q[1,1.1,t9]:= t8;
set Q[1,1.1,t10]:= t9;

set Q[1,3.1,t7]:= t6;
set Q[1,3.1,t8] := t7;
set Q[1,3.1,t9]:= t8;
set Q[1,3.1,t10]:= t9;

set Q[1,3.2,t7]:= t6;
set Q[1,3.2,t8] := t7;
set Q[1,3.2,t9]:= t8;
set Q[1,3.2,t10]:= t9;

set Q[2,1.1,t2] := t1;
set Q[2,1.1,t3] := t2;
set Q[2,1.1,t4]:= t3;
set Q[2,1.1,t5] := t4;
set Q[2,1.1,t6] := t5;
set Q[2,1.1,t7]:= t6;
set Q[2,1.1,t8]:= t7;
set Q[2,1.1,t9]:= t8;
set Q[2,1.1,t10]:= t9;

set Q[2,3.1,t7]:= t6;
set Q[2,3.1,t8] := t7;
set Q[2,3.1,t9]:= t8;
set Q[2,3.1,t10]:= t9;

set Q[3,3.1,t7]:= t6;
set Q[3,3.1,t8] := t7;
set Q[3,3.1,t9]:= t8;
set Q[3,3.1,t10]:= t9;

set Q[3,3.2,t7]:= t6;
set Q[3,3.2,t8] := t7;
set Q[3,3.2,t9]:= t8;
set Q[3,3.2,t10]:= t9;

set Q[3,3.3,t7]:= t6;
set Q[3,3.3,t8] := t7;
set Q[3,3.3,t9]:= t8;
set Q[3,3.3,t10]:= t9;

set Q[3,3.4,t7]:= t6;
set Q[3,3.4,t8] := t7;
set Q[3,3.4,t9]:= t8;
set Q[3,3.4,t10]:= t9;

set Q[4,3.1,t7]:= t6;
set Q[4,3.1,t8] := t7;
set Q[4,3.1,t9]:= t8;
set Q[4,3.1,t10]:= t9;

set Q[5,3.1,t7]:= t6;
set Q[5,3.1,t8] := t7;
set Q[5,3.1,t9]:= t8;
set Q[5,3.1,t10]:= t9;

set Q[6,1.1,t2] := t1;
set Q[6,1.1,t3] := t2;
set Q[6,1.1,t4]:= t3;
set Q[6,1.1,t5] := t4;
set Q[6,1.1,t6] := t5;
set Q[6,1.1,t7]:= t6;
set Q[6,1.1,t8]:= t7;
set Q[6,1.1,t9]:= t8;
set Q[6,1.1,t10]:= t9;

set Q[7,3.1,t7]:= t6;
set Q[7,3.1,t8] := t7;
set Q[7,3.1,t9]:= t8;
set Q[7,3.1,t10]:= t9;
set Q[8,3.1,t7]:= t6;
set Q[8,3.1,t8] := t7;
set Q[8,3.1,t9]:= t8;
set Q[8,3.1,t10]:= t9;

set Q[9,3.1,t7]:= t6;
set Q[9,3.1,t8] := t7;
set Q[9,3.1,t9]:= t8;
set Q[9,3.1,t10]:= t9;

set Q[10,3.1,t7]:= t6;
set Q[10,3.1,t8] := t7;
set Q[10,3.1,t9]:= t8;
set Q[10,3.1,t10]:= t9;

set Q[10,3.2,t7]:= t6;
set Q[10,3.2,t8] := t7;
set Q[10,3.2,t9]:= t8;
set Q[10,3.2,t10]:= t9;


# Process Precedence
set S[1,1.1] := p0;
set S[1,3.1] := 1.1;
set S[1,3.2] := 3.1;

set S[2,1.1] := p0;
set S[2,3.1] := 1.1;

set S[3,3.1] :=p0;
set S[3,3.2] :=3.1;
set S[3,3.3] :=3.2;
set S[3,3.4] :=3.3;

set S[4,3.1] :=p0;

set S[5,3.1] :=p0;

set S[6,1.1] :=p0;

set S[7,3.1] :=p0;

set S[8,3.1] :=p0;

set S[9,3.1] :=p0;

set S[10,3.1] :=p0;
set S[10,3.2] :=3.1;
```

```
# Last task of each process
set j_L[1,p0] := t0;
set j_L[1,1.1] := t10;
set j_L[1,3.1] := t10;
set j_L[1,3.2] := t10;

set j_L[2,p0] := t0;
set j_L[2,1.1] := t10;
set j_L[2,3.1] := t10;

set j_L[3,3.1] := t10;
set j_L[3,3.2] := t10;
set j_L[3,3.3] := t10;
set j_L[3,3.4] := t10;

set j_L[4,3.1] := t10;

set j_L[5,3.1] := t10;

set j_L[6,1.1] := t10;

set j_L[7,3.1] := t10;

set j_L[8,3.1] := t10;

set j_L[9,3.1] := t10;

set j_L[10,3.1] := t10;
set j_L[10,3.2] := t10;

# First task of each process
set j_F[1,p0] := t0;
set j_F[1,1.1] := t1;
set j_F[1,3.1] := t6;
set j_F[1,3.2] := t6;

set j_F[2,p0] := t0;
set j_F[2,1.1] := t1;
set j_F[2,3.1] := t6;

set j_F[3,p0] := t0;
set j_F[3,3.1] := t6;
set j_F[3,3.2] := t6;
set j_F[3,3.3] := t6;
set j_F[3,3.4] := t6;

set j_F[4,p0] := t0;
set j_F[4,3.1] := t6;

set j_F[5,p0] := t0;
set j_F[5,3.1] := t6;

set j_F[6,p0] := t0;
set j_F[6,1.1] := t1;

set j_F[7,p0] := t0;
set j_F[7,3.1] := t6;

set j_F[8,p0] := t0;
set j_F[8,3.1] := t6;

set j_F[9,p0] := t0;
set j_F[9,3.1] := t6;

set j_F[10,p0] := t0;
set j_F[10,3.1] := t6;
set j_F[10,3.2] := t6;

# The group of people
set G:= PM1 PM2 QC1 QC2 OA1 OA2 ;

# The PM people
set K[PM1] := PM1;
set K[PM2] := PM2;

#The OA people
set K[OA1] := OA1 ;
set K[OA2] := OA2 ;

#The QC people
set K[QC1] := QC1 ;
set K[QC2] := QC2 ;

# k and g are equal in this example

# set the total number of each resource k
param R :=
PM1 PM1 1
OA1 OA1 1
QC1 QC1 1

PM2 PM2 1
OA2 OA2 1
QC2 QC2 1 ;
```

```
# Set the due dates for each process
param ddd :=
1       p0      0
1       1.1     19
1       3.1     24
1       3.2     29

2       p0      0
2       1.1     19
2       3.1     24

3       p0      0
3       3.1     20
3       3.2     25
3       3.3     30
3       3.4     35

4       p0      0
4       3.1     21

5       p0      0
5       3.1     20

6       p0      0
6       1.1     23

7       p0      0
7       3.1     22

8       p0      0
8       3.1     21

9       p0      0
9       3.1     22

10      p0      0
10      3.1     23
10      3.2     28
;

# Set the early finish of each process
param E :=
1       p0      0
1       1.1     15
1       3.1     22
1       3.2     27

2       p0      0
```

```
2       1.1     16
2       3.1     22

3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33

4       p0      0
4       3.1     19

5       p0      0
5       3.1     18

6       p0      0
6       1.1     19

7       p0      0
7       3.1     20

8       p0      0
8       3.1     19

9       p0      0
9       3.1     20

10      p0      0
10      3.1     21
10      3.2     26

;

# Set the late finish of each process
/*param L :=
1       p0      0
1       1.1     50
1       3.1     50
1       3.2     50
2       p0      0
2       1.1     50
2       3.1     50
3       p0      0
3       3.1     50
3       3.2     50
3       3.3     50
3       3.4     50
4       p0      0
```

```
4         3.1       50
5         p0        0
5         3.1       50
6         p0        0
6         1.1       50
7         p0        0
7         3.1       50
8         p0        0
8         3.1       50
9         p0        0
9         3.1       50
10        p0        0
10        3.1       50
10        3.2       50
;*/

# Set the early finsh of each task
param EE :=
1 p0 t0 0
2 p0 t0 0
3 p0 t0 0
4 p0 t0 0
5 p0 t0 0
6 p0 t0 0
7 p0 t0 0
8 p0 t0 0
9 p0 t0 0
10 p0 t0 0

1         1.1       t1        7
2         1.1       t1        7
6         1.1       t1        11

1         1.1       t2        8
2         1.1       t2        8
6         1.1       t2        12

1         1.1       t3        9
2         1.1       t3        9
6         1.1       t3        13

1         1.1       t4        10
2         1.1       t4        10
6         1.1       t4        14

1         1.1       t5        11
2         1.1       t5        11
6         1.1       t5        15
```

1	1.1	t6	12
1	3.1	t6	18
1	3.2	t6	23
2	1.1	t6	12
2	3.1	t6	18
3	3.1	t6	14
3	3.2	t6	19
3	3.3	t6	24
3	3.4	t6	29
4	3.1	t6	15
5	3.1	t6	14
6	1.1	t6	10
7	3.1	t6	16
8	3.1	t6	15
9	3.1	t6	16
10	3.1	t6	17
10	3.2	t6	22

1	1.1	t7	13
1	3.1	t7	19
1	3.2	t7	24
2	1.1	t7	13
2	3.1	t7	19
3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	16
5	3.1	t7	15
6	1.1	t7	11
7	3.1	t7	17
8	3.1	t7	16
9	3.1	t7	17
10	3.1	t7	18
10	3.2	t7	23

1	1.1	t8	14
1	3.1	t8	20
1	3.2	t8	25
2	1.1	t8	14
2	3.1	t8	20
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	17
5	3.1	t8	16

```
6       1.1     t8      12
7       3.1     t8      18
8       3.1     t8      17
9       3.1     t8      18
10      3.1     t8      19
10      3.2     t8      24

1       1.1     t9      15
1       3.1     t9      21
1       3.2     t9      26
2       1.1     t9      15
2       3.1     t9      21
3       3.1     t9      17
3       3.2     t9      22
3       3.3     t9      27
3       3.4     t9      32
4       3.1     t9      18
5       3.1     t9      17
6       1.1     t9      13
7       3.1     t9      19
8       3.1     t9      18
9       3.1     t9      19
10      3.1     t9      20
10      3.2     t9      25

1       1.1     t10     16
1       3.1     t10     22
1       3.2     t10     27
2       1.1     t10     16
2       3.1     t10     22
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     19
5       3.1     t10     18
6       1.1     t10     14
7       3.1     t10     20
8       3.1     t10     19
9       3.1     t10     20
10      3.1     t10     21
10      3.2     t10     26;

# Set the  late finish of each task
/*param LL :=
1 p0 t0 0
2 p0 t0 0
3 p0 t0 0
```

```
4 p0 t0 0
5 p0 t0 0
6 p0 t0 0
7 p0 t0 0
8 p0 t0 0
9 p0 t0 0
10 p0 t0 0

1       1.1     t1      50
2       1.1     t1      50
6       1.1     t1      50

1       1.1     t2      50
2       1.1     t2      50
6       1.1     t2      50

1       1.1     t3      50
2       1.1     t3      50
6       1.1     t3      50

1       1.1     t5      50
2       1.1     t5      50
6       1.1     t5      50

1       1.1     t6      50
1       3.1     t6      50
1       3.2     t6      50
2       1.1     t6      50
2       3.1     t6      50
3       3.1     t6      50
3       3.2     t6      50
3       3.3     t6      50
3       3.4     t6      50
4       3.1     t6      50
5       3.1     t6      50
6       1.1     t6      50
7       3.1     t6      50
8       3.1     t6      50
9       3.1     t6      50
10      3.1     t6      50
10      3.2     t6      50

1       1.1     t7      50
1       3.1     t7      50
1       3.2     t7      50
2       1.1     t7      50
2       3.1     t7      50
3       3.1     t7      50
```

3	3.2	t7	50
3	3.3	t7	50
3	3.4	t7	50
4	3.1	t7	50
5	3.1	t7	50
6	1.1	t7	50
7	3.1	t7	50
8	3.1	t7	50
9	3.1	t7	50
10	3.1	t7	50
10	3.2	t7	50

1	1.1	t8	50
1	3.1	t8	50
1	3.2	t8	50
2	1.1	t8	50
2	3.1	t8	50
3	3.1	t8	50
3	3.2	t8	50
3	3.3	t8	50
3	3.4	t8	50
4	3.1	t8	50
5	3.1	t8	50
6	1.1	t8	50
7	3.1	t8	50
8	3.1	t8	50
9	3.1	t8	50
10	3.1	t8	50
10	3.2	t8	50

1	1.1	t9	50
1	3.1	t9	50
1	3.2	t9	50
2	1.1	t9	50
2	3.1	t9	50
3	3.1	t9	50
3	3.2	t9	50
3	3.3	t9	50
3	3.4	t9	50
4	3.1	t9	50
5	3.1	t9	50
6	1.1	t9	50
7	3.1	t9	50
8	3.1	t9	50
9	3.1	t9	50
10	3.1	t9	50
10	3.2	t9	50

```
1         1.1       t10       50
1         3.1       t10       50
1         3.2       t10       50
2         1.1       t10       50
2         3.1       t10       50
3         3.1       t10       50
3         3.2       t10       50
3         3.3       t10       50
3         3.4       t10       50
4         3.1       t10       50
5         3.1       t10       50
6         1.1       t10       50
7         3.1       t10       50
8         3.1       t10       50
9         3.1       t10       50
10        3.1       t10       50
10        3.2       t10       50;*/

# Set Durations for each task
# Set Durations for each task
param d :=
1         1.1       t1        1         2
2         1.1       t1        1         2
6         1.1       t1        1         2

1         1.1       t2        1         1
2         1.1       t2        1         1
6         1.1       t2        1         1

1         1.1       t3        1         1
2         1.1       t3        1         1
6         1.1       t3        1         1

1         1.1       t4        1         1
2         1.1       t4        1         1
6         1.1       t4        1         1

1         1.1       t5        1         1
2         1.1       t5        1         1
6         1.1       t5        1         1

1         1.1       t6        1         1
1         3.1       t6        1         1
1         3.2       t6        1         1
2         1.1       t6        1         1
2         3.1       t6        1         1
3         3.1       t6        1         1
```

3	3.2	t6	1	1
3	3.3	t6	1	1
3	3.4	t6	1	1
4	3.1	t6	1	1
5	3.1	t6	1	1
6	1.1	t6	1	1
7	3.1	t6	1	1
8	3.1	t6	1	1
9	3.1	t6	1	1
10	3.1	t6	1	1
10	3.2	t6	1	1

1	1.1	t7	1	1
1	3.1	t7	1	1
1	3.2	t7	1	1
2	1.1	t7	1	1
2	3.1	t7	1	1
3	3.1	t7	1	1
3	3.2	t7	1	1
3	3.3	t7	1	1
3	3.4	t7	1	1
4	3.1	t7	1	1
5	3.1	t7	1	1
6	1.1	t7	1	1
7	3.1	t7	1	1
8	3.1	t7	1	1
9	3.1	t7	1	1
10	3.1	t7	1	1
10	3.2	t7	1	1

1	1.1	t8	1	1
1	3.1	t8	1	1
1	3.2	t8	1	1
2	1.1	t8	1	1
2	3.1	t8	1	1
3	3.1	t8	1	1
3	3.2	t8	1	1
3	3.3	t8	1	1
3	3.4	t8	1	1
4	3.1	t8	1	1
5	3.1	t8	1	1
6	1.1	t8	1	1
7	3.1	t8	1	1
8	3.1	t8	1	1
9	3.1	t8	1	1
10	3.1	t8	1	1
10	3.2	t8	1	1

1	1.1	t9	1	1
1	3.1	t9	1	1
1	3.2	t9	1	1
2	1.1	t9	1	1
2	3.1	t9	1	1
3	3.1	t9	1	1
3	3.2	t9	1	1
3	3.3	t9	1	1
3	3.4	t9	1	1
4	3.1	t9	1	1
5	3.1	t9	1	1
6	1.1	t9	1	1
7	3.1	t9	1	1
8	3.1	t9	1	1
9	3.1	t9	1	1
10	3.1	t9	1	1
10	3.2	t9	1	1

1	1.1	t10	1	1
1	3.1	t10	1	1
1	3.2	t10	1	1
2	1.1	t10	1	1
2	3.1	t10	1	1
3	3.1	t10	1	1
3	3.2	t10	1	1
3	3.3	t10	1	1
3	3.4	t10	1	1
4	3.1	t10	1	1
5	3.1	t10	1	1
6	1.1	t10	1	1
7	3.1	t10	1	1
8	3.1	t10	1	1
9	3.1	t10	1	1
10	3.1	t10	1	1
10	3.2	t10	1	1

1	1.1	t1	2	2
2	1.1	t1	2	2
6	1.1	t1	2	2

1	1.1	t2	2	1
2	1.1	t2	2	1
6	1.1	t2	2	1

1	1.1	t3	2	1
2	1.1	t3	2	1
6	1.1	t3	2	1

1	1.1	t4	2	1
2	1.1	t4	2	1
6	1.1	t4	2	1
1	1.1	t5	2	1
2	1.1	t5	2	1
6	1.1	t5	2	1
1	1.1	t6	2	1
1	3.1	t6	2	1
1	3.2	t6	2	1
2	1.1	t6	2	1
2	3.1	t6	2	1
3	3.1	t6	2	1
3	3.2	t6	2	1
3	3.3	t6	2	1
3	3.4	t6	2	1
4	3.1	t6	2	1
5	3.1	t6	2	1
6	1.1	t6	2	1
7	3.1	t6	2	1
8	3.1	t6	2	1
9	3.1	t6	2	1
10	3.1	t6	2	1
10	3.2	t6	2	1
1	1.1	t7	2	1
1	3.1	t7	2	1
1	3.2	t7	2	1
2	1.1	t7	2	1
2	3.1	t7	2	1
3	3.1	t7	2	1
3	3.2	t7	2	1
3	3.3	t7	2	1
3	3.4	t7	2	1
4	3.1	t7	2	1
5	3.1	t7	2	1
6	1.1	t7	2	1
7	3.1	t7	2	1
8	3.1	t7	2	1
9	3.1	t7	2	1
10	3.1	t7	2	1
10	3.2	t7	2	1
1	1.1	t8	2	1
1	3.1	t8	2	1
1	3.2	t8	2	1
2	1.1	t8	2	1

2	3.1	t8	2	1
3	3.1	t8	2	1
3	3.2	t8	2	1
3	3.3	t8	2	1
3	3.4	t8	2	1
4	3.1	t8	2	1
5	3.1	t8	2	1
6	1.1	t8	2	1
7	3.1	t8	2	1
8	3.1	t8	2	1
9	3.1	t8	2	1
10	3.1	t8	2	1
10	3.2	t8	2	1
1	1.1	t9	2	1
1	3.1	t9	2	1
1	3.2	t9	2	1
2	1.1	t9	2	1
2	3.1	t9	2	1
3	3.1	t9	2	1
3	3.2	t9	2	1
3	3.3	t9	2	1
3	3.4	t9	2	1
4	3.1	t9	2	1
5	3.1	t9	2	1
6	1.1	t9	2	1
7	3.1	t9	2	1
8	3.1	t9	2	1
9	3.1	t9	2	1
10	3.1	t9	2	1
10	3.2	t9	2	1
1	1.1	t10	2	1
1	3.1	t10	2	1
1	3.2	t10	2	1
2	1.1	t10	2	1
2	3.1	t10	2	1
3	3.1	t10	2	1
3	3.2	t10	2	1
3	3.3	t10	2	1
3	3.4	t10	2	1
4	3.1	t10	2	1
5	3.1	t10	2	1
6	1.1	t10	2	1
7	3.1	t10	2	1
8	3.1	t10	2	1
9	3.1	t10	2	1
10	3.1	t10	2	1

```
10      3.2     t10     2       1



;
# Set the  number of resources of each employee (k)- param r is equal to 1 is that (K) is used ; otherwise
0 ( by default)
# index is  V,P,J,G,K,M,1/0
param r :=
1       1.1     t1      PM1     PM1     1       1
2       1.1     t1      PM1     PM1     1       1
6       1.1     t1      PM1     PM1     1       1

1       1.1     t2      QC1     QC1     1       1
2       1.1     t2      QC1     QC1     1       1
6       1.1     t2      QC1     QC1     1       1

1       1.1     t3      OA1     OA1     1       1
2       1.1     t3      OA1     OA1     1       1
6       1.1     t3      OA1     OA1     1       1

1       1.1     t4      QC1     QC1     1       1
2       1.1     t4      QC1     QC1     1       1
6       1.1     t4      QC1     QC1     1       1

1       1.1     t5      PM1     PM1     1       1
2       1.1     t5      PM1     PM1     1       1
6       1.1     t5      PM1     PM1     1       1

1       1.1     t6      OA1     OA1     1       1
1       3.1     t6      OA1     OA1     1       1
1       3.2     t6      OA1     OA1     1       1
2       1.1     t6      OA1     OA1     1       1
2       3.1     t6      OA1     OA1     1       1
3       3.1     t6      OA1     OA1     1       1
3       3.2     t6      OA1     OA1     1       1
3       3.3     t6      OA1     OA1     1       1
3       3.4     t6      OA1     OA1     1       1
4       3.1     t6      OA1     OA1     1       1
5       3.1     t6      OA1     OA1     1       1
6       1.1     t6      OA1     OA1     1       1
7       3.1     t6      OA1     OA1     1       1
8       3.1     t6      OA1     OA1     1       1
9       3.1     t6      OA1     OA1     1       1
10      3.1     t6      OA1     OA1     1       1
10      3.2     t6      OA1     OA1     1       1

1       1.1     t7      QC1     QC1     1       1
```

1	3.1	t7	QC1	QC1	1	1
1	3.2	t7	QC1	QC1	1	1
2	1.1	t7	QC1	QC1	1	1
2	3.1	t7	QC1	QC1	1	1
3	3.1	t7	QC1	QC1	1	1
3	3.2	t7	QC1	QC1	1	1
3	3.3	t7	QC1	QC1	1	1
3	3.4	t7	QC1	QC1	1	1
4	3.1	t7	QC1	QC1	1	1
5	3.1	t7	QC1	QC1	1	1
6	1.1	t7	QC1	QC1	1	1
7	3.1	t7	QC1	QC1	1	1
8	3.1	t7	QC1	QC1	1	1
9	3.1	t7	QC1	QC1	1	1
10	3.1	t7	QC1	QC1	1	1
10	3.2	t7	QC1	QC1	1	1

1	1.1	t8	OA1	OA1	1	1
1	3.1	t8	OA1	OA1	1	1
1	3.2	t8	OA1	OA1	1	1
2	1.1	t8	OA1	OA1	1	1
2	3.1	t8	OA1	OA1	1	1
3	3.1	t8	OA1	OA1	1	1
3	3.2	t8	OA1	OA1	1	1
3	3.3	t8	OA1	OA1	1	1
3	3.4	t8	OA1	OA1	1	1
4	3.1	t8	OA1	OA1	1	1
5	3.1	t8	OA1	OA1	1	1
6	1.1	t8	OA1	OA1	1	1
7	3.1	t8	OA1	OA1	1	1
8	3.1	t8	OA1	OA1	1	1
9	3.1	t8	OA1	OA1	1	1
10	3.1	t8	OA1	OA1	1	1
10	3.2	t8	OA1	OA1	1	1

1	1.1	t9	QC1	QC1	1	1
1	3.1	t9	QC1	QC1	1	1
1	3.2	t9	QC1	QC1	1	1
2	1.1	t9	QC1	QC1	1	1
2	3.1	t9	QC1	QC1	1	1
3	3.1	t9	QC1	QC1	1	1
3	3.2	t9	QC1	QC1	1	1
3	3.3	t9	QC1	QC1	1	1
3	3.4	t9	QC1	QC1	1	1
4	3.1	t9	QC1	QC1	1	1
5	3.1	t9	QC1	QC1	1	1
6	1.1	t9	QC1	QC1	1	1
7	3.1	t9	QC1	QC1	1	1

8	3.1	t9	QC1	QC1	1	1
9	3.1	t9	QC1	QC1	1	1
10	3.1	t9	QC1	QC1	1	1
10	3.2	t9	QC1	QC1	1	1

1	1.1	t10	PM1	PM1	1	1
1	3.1	t10	PM1	PM1	1	1
1	3.2	t10	PM1	PM1	1	1
2	1.1	t10	PM1	PM1	1	1
2	3.1	t10	PM1	PM1	1	1
3	3.1	t10	PM1	PM1	1	1
3	3.2	t10	PM1	PM1	1	1
3	3.3	t10	PM1	PM1	1	1
3	3.4	t10	PM1	PM1	1	1
4	3.1	t10	PM1	PM1	1	1
5	3.1	t10	PM1	PM1	1	1
6	1.1	t10	PM1	PM1	1	1
7	3.1	t10	PM1	PM1	1	1
8	3.1	t10	PM1	PM1	1	1
9	3.1	t10	PM1	PM1	1	1
10	3.1	t10	PM1	PM1	1	1
10	3.2	t10	PM1	PM1	1	1

1	1.1	t10	OA1	OA1	1	1
1	3.1	t10	OA1	OA1	1	1
1	3.2	t10	OA1	OA1	1	1
2	1.1	t10	OA1	OA1	1	1
2	3.1	t10	OA1	OA1	1	1
3	3.1	t10	OA1	OA1	1	1
3	3.2	t10	OA1	OA1	1	1
3	3.3	t10	OA1	OA1	1	1
3	3.4	t10	OA1	OA1	1	1
4	3.1	t10	OA1	OA1	1	1
5	3.1	t10	OA1	OA1	1	1
6	1.1	t10	OA1	OA1	1	1
7	3.1	t10	OA1	OA1	1	1
8	3.1	t10	OA1	OA1	1	1
9	3.1	t10	OA1	OA1	1	1
10	3.1	t10	OA1	OA1	1	1
10	3.2	t10	OA1	OA1	1	1

1	1.1	t1	PM2	PM2	2	1
2	1.1	t1	PM2	PM2	2	1
6	1.1	t1	PM2	PM2	2	1

1	1.1	t2	QC2	QC2	2	1

2	1.1	t2	QC2	QC2	2	1
6	1.1	t2	QC2	QC2	2	1
1	1.1	t3	OA2	OA2	2	1
2	1.1	t3	OA2	OA2	2	1
6	1.1	t3	OA2	OA2	2	1
1	1.1	t4	QC2	QC2	2	1
2	1.1	t4	QC2	QC2	2	1
6	1.1	t4	QC2	QC2	2	1
1	1.1	t5	PM2	PM2	2	1
2	1.1	t5	PM2	PM2	2	1
6	1.1	t5	PM2	PM2	2	1
1	1.1	t6	OA2	OA2	2	1
1	3.1	t6	OA2	OA2	2	1
1	3.2	t6	OA2	OA2	2	1
2	1.1	t6	OA2	OA2	2	1
2	3.1	t6	OA2	OA2	2	1
3	3.1	t6	OA2	OA2	2	1
3	3.2	t6	OA2	OA2	2	1
3	3.3	t6	OA2	OA2	2	1
3	3.4	t6	OA2	OA2	2	1
4	3.1	t6	OA2	OA2	2	1
5	3.1	t6	OA2	OA2	2	1
6	1.1	t6	OA2	OA2	2	1
7	3.1	t6	OA2	OA2	2	1
8	3.1	t6	OA2	OA2	2	1
9	3.1	t6	OA2	OA2	2	1
10	3.1	t6	OA2	OA2	2	1
10	3.2	t6	OA2	OA2	2	1
1	1.1	t7	QC2	QC2	2	1
1	3.1	t7	QC2	QC2	2	1
1	3.2	t7	QC2	QC2	2	1
2	1.1	t7	QC2	QC2	2	1
2	3.1	t7	QC2	QC2	2	1
3	3.1	t7	QC2	QC2	2	1
3	3.2	t7	QC2	QC2	2	1
3	3.3	t7	QC2	QC2	2	1
3	3.4	t7	QC2	QC2	2	1
4	3.1	t7	QC2	QC2	2	1
5	3.1	t7	QC2	QC2	2	1
6	1.1	t7	QC2	QC2	2	1
7	3.1	t7	QC2	QC2	2	1
8	3.1	t7	QC2	QC2	2	1
9	3.1	t7	QC2	QC2	2	1

10	3.1	t7	QC2	QC2	2	1
10	3.2	t7	QC2	QC2	2	1

1	1.1	t8	OA2	OA2	2	1
1	3.1	t8	OA2	OA2	2	1
1	3.2	t8	OA2	OA2	2	1
2	1.1	t8	OA2	OA2	2	1
2	3.1	t8	OA2	OA2	2	1
3	3.1	t8	OA2	OA2	2	1
3	3.2	t8	OA2	OA2	2	1
3	3.3	t8	OA2	OA2	2	1
3	3.4	t8	OA2	OA2	2	1
4	3.1	t8	OA2	OA2	2	1
5	3.1	t8	OA2	OA2	2	1
6	1.1	t8	OA2	OA2	2	1
7	3.1	t8	OA2	OA2	2	1
8	3.1	t8	OA2	OA2	2	1
9	3.1	t8	OA2	OA2	2	1
10	3.1	t8	OA2	OA2	2	1
10	3.2	t8	OA2	OA2	2	1

1	1.1	t9	QC2	QC2	2	1
1	3.1	t9	QC2	QC2	2	1
1	3.2	t9	QC2	QC2	2	1
2	1.1	t9	QC2	QC2	2	1
2	3.1	t9	QC2	QC2	2	1
3	3.1	t9	QC2	QC2	2	1
3	3.2	t9	QC2	QC2	2	1
3	3.3	t9	QC2	QC2	2	1
3	3.4	t9	QC2	QC2	2	1
4	3.1	t9	QC2	QC2	2	1
5	3.1	t9	QC2	QC2	2	1
6	1.1	t9	QC2	QC2	2	1
7	3.1	t9	QC2	QC2	2	1
8	3.1	t9	QC2	QC2	2	1
9	3.1	t9	QC2	QC2	2	1
10	3.1	t9	QC2	QC2	2	1
10	3.2	t9	QC2	QC2	2	1

1	1.1	t10	PM2	PM2	2	1
1	3.1	t10	PM2	PM2	2	1
1	3.2	t10	PM2	PM2	2	1
2	1.1	t10	PM2	PM2	2	1
2	3.1	t10	PM2	PM2	2	1
3	3.1	t10	PM2	PM2	2	1
3	3.2	t10	PM2	PM2	2	1
3	3.3	t10	PM2	PM2	2	1
3	3.4	t10	PM2	PM2	2	1

```
4       3.1     t10     PM2     PM2     2       1
5       3.1     t10     PM2     PM2     2       1
6       1.1     t10     PM2     PM2     2       1
7       3.1     t10     PM2     PM2     2       1
8       3.1     t10     PM2     PM2     2       1
9       3.1     t10     PM2     PM2     2       1
10      3.1     t10     PM2     PM2     2       1
10      3.2     t10     PM2     PM2     2       1

1       1.1     t10     OA2     OA2     2       1
1       3.1     t10     OA2     OA2     2       1
1       3.2     t10     OA2     OA2     2       1
2       1.1     t10     OA2     OA2     2       1
2       3.1     t10     OA2     OA2     2       1
3       3.1     t10     OA2     OA2     2       1
3       3.2     t10     OA2     OA2     2       1
3       3.3     t10     OA2     OA2     2       1
3       3.4     t10     OA2     OA2     2       1
4       3.1     t10     OA2     OA2     2       1
5       3.1     t10     OA2     OA2     2       1
6       1.1     t10     OA2     OA2     2       1
7       3.1     t10     OA2     OA2     2       1
8       3.1     t10     OA2     OA2     2       1
9       3.1     t10     OA2     OA2     2       1
10      3.1     t10     OA2     OA2     2       1
10      3.2     t10     OA2     OA2     2       1;


end;
```

Appendix L: GLPK Code to Experiment Model (IV)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede task j
set M{v in V, p in P[v],j in J[v,p]}, default {1}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]},default 0; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}, default 50; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 0; /* duration of tasks j of process p of project
v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k usage
of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 0; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR{v in V, g in G, k in K[g]}, default 0; /* Amount of renewable resource k of group g*/
#param R{g in G, k in K[g]}, default 0; /* total amount of renewable resource k*/

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
#var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
```

```
var TC1{v in V, p in P[v]}>=0;/*Weighted tardiness cost of process p of project v*/
#var BR {v in V,g in G, k in K[g]} integer;
var R {g in G, k in K[g]} integer;

minimize z: sum {v in V, p in P[v]} TC1[v,p]; /* minimize the weighted tardiness of the project and
process*/
#minimize z: sum{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T}x[v,p,j,m,t];

#minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */

s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 mode for every task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in EE[v,p,jF]..LL[v,p,jF],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,s]..L[v,s],m in M[v,s,jL]}(t)*x[v,s,jL,m,t];/* predence
relationship for processes*/

#s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource k of group g   */

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V,p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= R[g,k]; /* capacity constraint
for each resource k of group g   */

#s.t. Constraint_5{v in V, p in p_N[v],j in j_L[v,p]}: TC[v] >= c[v] *( sum{m in M[v,p,j],t in E[v,p]..L[v,p]}
(t*x[v,p,j,m,t])- dd[v]); /* tardiness of project is equal to some penalty c[v] times the difference of the
finish time of the last task of the last process minus the due date of the project */

s.t. Constraint_6{v in V, p in P[v], j in j_L[v,p], g in G}: TC1[v,p] >= c1[v,p]*(sum{m in M[v,p,j], t in
EE[v,p,j]..LL[v,p,j]} (t*x[v,p,j,m,t])- ddd[v,p]); /* tardiness of process is equal to some penalty c[v,p] times
the difference of the finish time of the last task of each process minus the due date of the process */

#s.t. Constraint_7 {g in G, k in K[g]}: sum {v in V}BR[v,g,k] <= R[g,k] ;


/*table tab_data IN "CSV" "projects.csv":
V <- [project] P[V]<-[Process];*/

solve;

display z, x,R;
```

```
data;
param projects :=10;
param resourceGroup := 3;
param timePeriods := 50;

# Define Processes and Orders for each Project- The decimal number indicates the order number
#Every project has a dummy starting process called "p0" with 0 duration.

set P[1] := p0   1.1      3.1      3.2;
set P[2] := p0   1.1      3.1 ;
set P[3] := p0   3.1      3.2      3.3      3.4;
set P[4] := p0   3.1;
set P[5] := p0   3.1;
set P[6] := p0   1.1;
set P[7] := p0   3.1;
set P[8] := p0   3.1;
set P[9] := p0   3.1;
set P[10] := p0  3.1      3.2;


# Define tasks for each process:
set J[1,p0] := t0;
set J[1,1.1] := t1t2      t3       t4       t5       t6       t7       t8       t9       t10;
set J[1,3.1] := t6t7      t8       t9       t10;
set J[1,3.2] := t6t7      t8       t9       t10;

set J[2,p0] := t0;
set J[2,1.1] := t1t2      t3       t4       t5       t6       t7       t8       t9       t10;
set J[2,3.1] := t6t7      t8       t9       t10;

set J[3,p0]:=t0;
set J[3,3.1]:=t6  t7      t8       t9       t10;
set J[3,3.2]:=t6  t7      t8       t9       t10;
set J[3,3.3]:=t6  t7      t8       t9       t10;
set J[3,3.4]:=t6  t7      t8       t9       t10;

set J[4,p0] :=t0;
set J[4,3.1]:= t6 t7      t8       t9       t10;

set J[5,p0] :=t0;
set J[5,3.1] := t6t7      t8       t9       t10;

set J[6,p0] :=t0;
set J[6,1.1] := t1t2      t3       t4       t5       t6       t7       t8       t9       t10;

set J[7,p0] :=t0;
set J[7,3.1] := t6t7      t8       t9       t10;
set J[8,p0] :=t0;
set J[8,3.1] := t6t7        t8        t9        t10;

set J[9,p0] :=t0;
set J[9,3.1] := t6t7        t8        t9        t10;

set J[10,p0] :=t0;
set J[10,3.1] :=t6          t7        t8        t9        t10;
set J[10,3.2] :=t6          t7        t8        t9        t10;

# Define Mode
# set all modes by default to 1

# Task Precedence
set Q[1,1.1,t2] := t1;
set Q[1,1.1,t3] := t2;
set Q[1,1.1,t4]:= t3;
set Q[1,1.1,t5] := t4;
set Q[1,1.1,t6] := t5;
set Q[1,1.1,t7]:= t6;
set Q[1,1.1,t8]:= t7;
set Q[1,1.1,t9]:= t8;
set Q[1,1.1,t10]:= t9;

set Q[1,3.1,t7]:= t6;
set Q[1,3.1,t8] := t7;
set Q[1,3.1,t9]:= t8;
set Q[1,3.1,t10]:= t9;

set Q[1,3.2,t7]:= t6;
set Q[1,3.2,t8] := t7;
set Q[1,3.2,t9]:= t8;
set Q[1,3.2,t10]:= t9;

set Q[2,1.1,t2] := t1;
set Q[2,1.1,t3] := t2;
set Q[2,1.1,t4]:= t3;
set Q[2,1.1,t5] := t4;
set Q[2,1.1,t6] := t5;
set Q[2,1.1,t7]:= t6;
set Q[2,1.1,t8]:= t7;
set Q[2,1.1,t9]:= t8;
set Q[2,1.1,t10]:= t9;

set Q[2,3.1,t7]:= t6;
set Q[2,3.1,t8] := t7;
set Q[2,3.1,t9]:= t8;
set Q[2,3.1,t10]:= t9;
set Q[3,3.1,t7]:= t6;
set Q[3,3.1,t8] := t7;
set Q[3,3.1,t9]:= t8;
set Q[3,3.1,t10]:= t9;

set Q[3,3.2,t7]:= t6;
set Q[3,3.2,t8] := t7;
set Q[3,3.2,t9]:= t8;
set Q[3,3.2,t10]:= t9;

set Q[3,3.3,t7]:= t6;
set Q[3,3.3,t8] := t7;
set Q[3,3.3,t9]:= t8;
set Q[3,3.3,t10]:= t9;

set Q[3,3.4,t7]:= t6;
set Q[3,3.4,t8] := t7;
set Q[3,3.4,t9]:= t8;
set Q[3,3.4,t10]:= t9;

set Q[4,3.1,t7]:= t6;
set Q[4,3.1,t8] := t7;
set Q[4,3.1,t9]:= t8;
set Q[4,3.1,t10]:= t9;

set Q[5,3.1,t7]:= t6;
set Q[5,3.1,t8] := t7;
set Q[5,3.1,t9]:= t8;
set Q[5,3.1,t10]:= t9;

set Q[6,1.1,t2] := t1;
set Q[6,1.1,t3] := t2;
set Q[6,1.1,t4]:= t3;
set Q[6,1.1,t5] := t4;
set Q[6,1.1,t6] := t5;
set Q[6,1.1,t7]:= t6;
set Q[6,1.1,t8]:= t7;
set Q[6,1.1,t9]:= t8;
set Q[6,1.1,t10]:= t9;

set Q[7,3.1,t7]:= t6;
set Q[7,3.1,t8] := t7;
set Q[7,3.1,t9]:= t8;
set Q[7,3.1,t10]:= t9;

set Q[8,3.1,t7]:= t6;
set Q[8,3.1,t8] := t7;
set Q[8,3.1,t9]:= t8;
set Q[8,3.1,t10]:= t9;

set Q[9,3.1,t7]:= t6;
set Q[9,3.1,t8] := t7;
set Q[9,3.1,t9]:= t8;
set Q[9,3.1,t10]:= t9;

set Q[10,3.1,t7]:= t6;
set Q[10,3.1,t8] := t7;
set Q[10,3.1,t9]:= t8;
set Q[10,3.1,t10]:= t9;

set Q[10,3.2,t7]:= t6;
set Q[10,3.2,t8] := t7;
set Q[10,3.2,t9]:= t8;
set Q[10,3.2,t10]:= t9;


# Process Precedence
set S[1,1.1] := p0;
set S[1,3.1] := 1.1;
set S[1,3.2] := 3.1;

set S[2,1.1] := p0;
set S[2,3.1] := 1.1;

set S[3,3.1] :=p0;
set S[3,3.2] :=3.1;
set S[3,3.3] :=3.2;
set S[3,3.4] :=3.3;

set S[4,3.1] :=p0;

set S[5,3.1] :=p0;

set S[6,1.1] :=p0;

set S[7,3.1] :=p0;

set S[8,3.1] :=p0;

set S[9,3.1] :=p0;

set S[10,3.1] :=p0;
set S[10,3.2] :=3.1;

# Last task of each process
```

```
set j_L[1,p0] := t0;
set j_L[1,1.1] := t10;
set j_L[1,3.1] := t10;
set j_L[1,3.2] := t10;

set j_L[2,p0] := t0;
set j_L[2,1.1] := t10;
set j_L[2,3.1] := t10;

set j_L[3,3.1] := t10;
set j_L[3,3.2] := t10;
set j_L[3,3.3] := t10;
set j_L[3,3.4] := t10;

set j_L[4,3.1] := t10;

set j_L[5,3.1] := t10;

set j_L[6,1.1] := t10;

set j_L[7,3.1] := t10;

set j_L[8,3.1] := t10;

set j_L[9,3.1] := t10;

set j_L[10,3.1] := t10;
set j_L[10,3.2] := t10;

# First task of each process
set j_F[1,p0] := t0;
set j_F[1,1.1] := t1;
set j_F[1,3.1] := t6;
set j_F[1,3.2] := t6;

set j_F[2,p0] := t0;
set j_F[2,1.1] := t1;
set j_F[2,3.1] := t6;

set j_F[3,p0] := t0;
set j_F[3,3.1] := t6;
set j_F[3,3.2] := t6;
set j_F[3,3.3] := t6;
set j_F[3,3.4] := t6;

set j_F[4,p0] := t0;
set j_F[4,3.1] := t6;
set j_F[5,p0] := t0;
set j_F[5,3.1] := t6;

set j_F[6,p0] := t0;
set j_F[6,1.1] := t1;

set j_F[7,p0] := t0;
set j_F[7,3.1] := t6;

set j_F[8,p0] := t0;
set j_F[8,3.1] := t6;

set j_F[9,p0] := t0;
set j_F[9,3.1] := t6;

set j_F[10,p0] := t0;
set j_F[10,3.1] := t6;
set j_F[10,3.2] := t6;

# The group of people
set G:= PM QC OA;

set K[PM] := PM; # The PM people
set K[OA] := OA ; #The OA people
set K[QC] := QC ; #The QC people

# k and g are equal in this example

# set the total number of each resource k
/*param R :=
PM PM 2
OA OA 2
QC QC 2 ;*/

# Set the due dates for each process
param ddd :=
1       p0      0
1       1.1     19
1       3.1     24
1       3.2     29

2       p0      0
2       1.1     19
2       3.1     24

3       p0      0
3       3.1     20
3       3.2     25
```

```
3       3.3     30
3       3.4     35

4       p0      0
4       3.1     21

5       p0      0
5       3.1     20

6       p0      0
6       1.1     23

7       p0      0
7       3.1     22

8       p0      0
8       3.1     21

9       p0      0
9       3.1     22

10      p0      0
10      3.1     23
10      3.2     28
;

# Set the early finish of each process
param E :=
1       p0      0
1       1.1     15
1       3.1     22
1       3.2     27

2       p0      0
2       1.1     16
2       3.1     22

3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33

4       p0      0
4       3.1     19

5       p0      0
5       3.1     18
```

```
6       p0      0
6       1.1     19

7       p0      0
7       3.1     20

8       p0      0
8       3.1     19

9       p0      0
9       3.1     20

10      p0      0
10      3.1     21
10      3.2     26

;

# Set the late finish of each process
param L :=
1       p0      0
1       1.1     23
1       3.1     28
1       3.2     33
2       p0      0
2       1.1     28
2       3.1     33
3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33
4       p0      0
4       3.1     33
5       p0      0
5       3.1     33
6       p0      0
6       1.1     33
7       p0      0
7       3.1     33
8       p0      0
8       3.1     33
9       p0      0
9       3.1     23
10      p0      0
10      3.1     28
10      3.2     33
```

```
;

# Set the early finsh of each task
param EE :=
1 p0 t0 0
1       1.1     t1      7
2       1.1     t1      7
6       1.1     t1      11

1       1.1     t2      8
2       1.1     t2      8
6       1.1     t2      12

1       1.1     t3      9
2       1.1     t3      9
6       1.1     t3      13

1       1.1     t4      10
2       1.1     t4      10
6       1.1     t4      14

1       1.1     t5      11
2       1.1     t5      11
6       1.1     t5      15

1       1.1     t6      12
1       3.1     t6      18
1       3.2     t6      23
2       1.1     t6      12
2       3.1     t6      18
3       3.1     t6      14
3       3.2     t6      19
3       3.3     t6      24
3       3.4     t6      29
4       3.1     t6      15
5       3.1     t6      14
6       1.1     t6      10
7       3.1     t6      16
8       3.1     t6      15
9       3.1     t6      16
10      3.1     t6      17
10      3.2     t6      22

1       1.1     t7      13
1       3.1     t7      19
1       3.2     t7      24
2       1.1     t7      13
2       3.1     t7      19
```

3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	16
5	3.1	t7	15
6	1.1	t7	11
7	3.1	t7	17
8	3.1	t7	16
9	3.1	t7	17
10	3.1	t7	18
10	3.2	t7	23

1	1.1	t8	14
1	3.1	t8	20
1	3.2	t8	25
2	1.1	t8	14
2	3.1	t8	20
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	17
5	3.1	t8	16
6	1.1	t8	12
7	3.1	t8	18
8	3.1	t8	17
9	3.1	t8	18
10	3.1	t8	19
10	3.2	t8	24

1	1.1	t9	15
1	3.1	t9	21
1	3.2	t9	26
2	1.1	t9	15
2	3.1	t9	21
3	3.1	t9	17
3	3.2	t9	22
3	3.3	t9	27
3	3.4	t9	32
4	3.1	t9	18
5	3.1	t9	17
6	1.1	t9	13
7	3.1	t9	19
8	3.1	t9	18
9	3.1	t9	19
10	3.1	t9	20
10	3.2	t9	25

```
1       1.1     t10     16
1       3.1     t10     22
1       3.2     t10     27
2       1.1     t10     16
2       3.1     t10     22
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     19
5       3.1     t10     18
6       1.1     t10     14
7       3.1     t10     20
8       3.1     t10     19
9       3.1     t10     20
10      3.1     t10     21
10      3.2     t10     26;

# Set the  late finish of each task
#param LL :=

1       1.1     t1      14
2       1.1     t1      19
6       1.1     t1      24

1       1.1     t2      15
2       1.1     t2      20
6       1.1     t2      25

1       1.1     t3      15
2       1.1     t3      20
6       1.1     t3      25

1       1.1     t5      18
2       1.1     t5      23
6       1.1     t5      28

1       1.1     t6      19
1       3.1     t6      24
1       3.2     t6      29
2       1.1     t6      24
2       3.1     t6      29
3       3.1     t6      14
3       3.2     t6      19
3       3.3     t6      24
3       3.4     t6      29
4       3.1     t6      29
```

5	3.1	t6	29
6	1.1	t6	29
7	3.1	t6	29
8	3.1	t6	29
9	3.1	t6	19
10	3.1	t6	24
10	3.2	t6	29

1	1.1	t7	20
1	3.1	t7	25
1	3.2	t7	30
2	1.1	t7	25
2	3.1	t7	30
3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	30
5	3.1	t7	30
6	1.1	t7	30
7	3.1	t7	30
8	3.1	t7	30
9	3.1	t7	20
10	3.1	t7	25
10	3.2	t7	30

1	1.1	t8	21
1	3.1	t8	26
1	3.2	t8	31
2	1.1	t8	26
2	3.1	t8	31
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	31
5	3.1	t8	31
6	1.1	t8	31
7	3.1	t8	31
8	3.1	t8	31
9	3.1	t8	21
10	3.1	t8	26
10	3.2	t8	31

1	1.1	t9	22
1	3.1	t9	27
1	3.2	t9	32

```
2       1.1     t9      27
2       3.1     t9      32
3       3.1     t9      17
3       3.2     t9      22
3       3.3     t9      27
3       3.4     t9      32
4       3.1     t9      32
5       3.1     t9      32
6       1.1     t9      32
7       3.1     t9      32
8       3.1     t9      32
9       3.1     t9      22
10      3.1     t9      27
10      3.2     t9      32

1       1.1     t10     23
1       3.1     t10     28
1       3.2     t10     33
2       1.1     t10     28
2       3.1     t10     33
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     33
5       3.1     t10     33
6       1.1     t10     33
7       3.1     t10     33
8       3.1     t10     33
9       3.1     t10     23
10      3.1     t10     28
10      3.2     t10     33 ;

# Set Durations for each task
param d :=
1       1.1     t1      1       2
2       1.1     t1      1       2
6       1.1     t1      1       2

1       1.1     t2      1       1
2       1.1     t2      1       1
6       1.1     t2      1       1

1       1.1     t3      1       1
2       1.1     t3      1       1
6       1.1     t3      1       1

1       1.1     t4      1       1
```

2	1.1	t4	1	1
6	1.1	t4	1	1
1	1.1	t5	1	1
2	1.1	t5	1	1
6	1.1	t5	1	1
1	1.1	t6	1	1
1	3.1	t6	1	1
1	3.2	t6	1	1
2	1.1	t6	1	1
2	3.1	t6	1	1
3	3.1	t6	1	1
3	3.2	t6	1	1
3	3.3	t6	1	1
3	3.4	t6	1	1
4	3.1	t6	1	1
5	3.1	t6	1	1
6	1.1	t6	1	1
7	3.1	t6	1	1
8	3.1	t6	1	1
9	3.1	t6	1	1
10	3.1	t6	1	1
10	3.2	t6	1	1
1	1.1	t7	1	1
1	3.1	t7	1	1
1	3.2	t7	1	1
2	1.1	t7	1	1
2	3.1	t7	1	1
3	3.1	t7	1	1
3	3.2	t7	1	1
3	3.3	t7	1	1
3	3.4	t7	1	1
4	3.1	t7	1	1
5	3.1	t7	1	1
6	1.1	t7	1	1
7	3.1	t7	1	1
8	3.1	t7	1	1
9	3.1	t7	1	1
10	3.1	t7	1	1
10	3.2	t7	1	1
1	1.1	t8	1	1
1	3.1	t8	1	1
1	3.2	t8	1	1
2	1.1	t8	1	1
2	3.1	t8	1	1

```
3       3.1     t8      1       1
3       3.2     t8      1       1
3       3.3     t8      1       1
3       3.4     t8      1       1
4       3.1     t8      1       1
5       3.1     t8      1       1
6       1.1     t8      1       1
7       3.1     t8      1       1
8       3.1     t8      1       1
9       3.1     t8      1       1
10      3.1     t8      1       1
10      3.2     t8      1       1

1       1.1     t9      1       1
1       3.1     t9      1       1
1       3.2     t9      1       1
2       1.1     t9      1       1
2       3.1     t9      1       1
3       3.1     t9      1       1
3       3.2     t9      1       1
3       3.3     t9      1       1
3       3.4     t9      1       1
4       3.1     t9      1       1
5       3.1     t9      1       1
6       1.1     t9      1       1
7       3.1     t9      1       1
8       3.1     t9      1       1
9       3.1     t9      1       1
10      3.1     t9      1       1
10      3.2     t9      1       1

1       1.1     t10     1       1
1       3.1     t10     1       1
1       3.2     t10     1       1
2       1.1     t10     1       1
2       3.1     t10     1       1
3       3.1     t10     1       1
3       3.2     t10     1       1
3       3.3     t10     1       1
3       3.4     t10     1       1
4       3.1     t10     1       1
5       3.1     t10     1       1
6       1.1     t10     1       1
7       3.1     t10     1       1
8       3.1     t10     1       1
9       3.1     t10     1       1
10      3.1     t10     1       1
10      3.2     t10     1       1;
```

```
# Set the  number of resources of each employee (k)- param r is equal to 1 is that (K) is used ; otherwise
0 ( by default)
# index is  V,P,J,G,K,M,1/0
param r :=
1       1.1     t1      PM      PM      1       1
2       1.1     t1      PM      PM      1       1
6       1.1     t1      PM      PM      1       1

1       1.1     t2      QC      QC      1       1
2       1.1     t2      QC      QC      1       1
6       1.1     t2      QC      QC      1       1

1       1.1     t3      OA      OA      1       1
2       1.1     t3      OA      OA      1       1
6       1.1     t3      OA      OA      1       1

1       1.1     t4      QC      QC      1       1
2       1.1     t4      QC      QC      1       1
6       1.1     t4      QC      QC      1       1

1       1.1     t5      PM      PM      1       1
2       1.1     t5      PM      PM      1       1
6       1.1     t5      PM      PM      1       1

1       1.1     t6      OA      OA      1       1
1       3.1     t6      OA      OA      1       1
1       3.2     t6      OA      OA      1       1
2       1.1     t6      OA      OA      1       1
2       3.1     t6      OA      OA      1       1
3       3.1     t6      OA      OA      1       1
3       3.2     t6      OA      OA      1       1
3       3.3     t6      OA      OA      1       1
3       3.4     t6      OA      OA      1       1
4       3.1     t6      OA      OA      1       1
5       3.1     t6      OA      OA      1       1
6       1.1     t6      OA      OA      1       1
7       3.1     t6      OA      OA      1       1
8       3.1     t6      OA      OA      1       1
9       3.1     t6      OA      OA      1       1
10      3.1     t6      OA      OA      1       1
10      3.2     t6      OA      OA      1       1

1       1.1     t7      QC      QC      1       1
1       3.1     t7      QC      QC      1       1
1       3.2     t7      QC      QC      1       1
2       1.1     t7      QC      QC      1       1
2       3.1     t7      QC      QC      1       1
```

3	3.1	t7	QC	QC	1	1
3	3.2	t7	QC	QC	1	1
3	3.3	t7	QC	QC	1	1
3	3.4	t7	QC	QC	1	1
4	3.1	t7	QC	QC	1	1
5	3.1	t7	QC	QC	1	1
6	1.1	t7	QC	QC	1	1
7	3.1	t7	QC	QC	1	1
8	3.1	t7	QC	QC	1	1
9	3.1	t7	QC	QC	1	1
10	3.1	t7	QC	QC	1	1
10	3.2	t7	QC	QC	1	1

1	1.1	t8	OA	OA	1	1
1	3.1	t8	OA	OA	1	1
1	3.2	t8	OA	OA	1	1
2	1.1	t8	OA	OA	1	1
2	3.1	t8	OA	OA	1	1
3	3.1	t8	OA	OA	1	1
3	3.2	t8	OA	OA	1	1
3	3.3	t8	OA	OA	1	1
3	3.4	t8	OA	OA	1	1
4	3.1	t8	OA	OA	1	1
5	3.1	t8	OA	OA	1	1
6	1.1	t8	OA	OA	1	1
7	3.1	t8	OA	OA	1	1
8	3.1	t8	OA	OA	1	1
9	3.1	t8	OA	OA	1	1
10	3.1	t8	OA	OA	1	1
10	3.2	t8	OA	OA	1	1

1	1.1	t9	QC	QC	1	1
1	3.1	t9	QC	QC	1	1
1	3.2	t9	QC	QC	1	1
2	1.1	t9	QC	QC	1	1
2	3.1	t9	QC	QC	1	1
3	3.1	t9	QC	QC	1	1
3	3.2	t9	QC	QC	1	1
3	3.3	t9	QC	QC	1	1
3	3.4	t9	QC	QC	1	1
4	3.1	t9	QC	QC	1	1
5	3.1	t9	QC	QC	1	1
6	1.1	t9	QC	QC	1	1
7	3.1	t9	QC	QC	1	1
8	3.1	t9	QC	QC	1	1
9	3.1	t9	QC	QC	1	1
10	3.1	t9	QC	QC	1	1
10	3.2	t9	QC	QC	1	1

```
1       1.1     t10     PM      PM      1       1
1       3.1     t10     PM      PM      1       1
1       3.2     t10     PM      PM      1       1
2       1.1     t10     PM      PM      1       1
2       3.1     t10     PM      PM      1       1
3       3.1     t10     PM      PM      1       1
3       3.2     t10     PM      PM      1       1
3       3.3     t10     PM      PM      1       1
3       3.4     t10     PM      PM      1       1
4       3.1     t10     PM      PM      1       1
5       3.1     t10     PM      PM      1       1
6       1.1     t10     PM      PM      1       1
7       3.1     t10     PM      PM      1       1
8       3.1     t10     PM      PM      1       1
9       3.1     t10     PM      PM      1       1
10      3.1     t10     PM      PM      1       1
10      3.2     t10     PM      PM      1       1

1       1.1     t10     OA      OA      1       1
1       3.1     t10     OA      OA      1       1
1       3.2     t10     OA      OA      1       1
2       1.1     t10     OA      OA      1       1
2       3.1     t10     OA      OA      1       1
3       3.1     t10     OA      OA      1       1
3       3.2     t10     OA      OA      1       1
3       3.3     t10     OA      OA      1       1
3       3.4     t10     OA      OA      1       1
4       3.1     t10     OA      OA      1       1
5       3.1     t10     OA      OA      1       1
6       1.1     t10     OA      OA      1       1
7       3.1     t10     OA      OA      1       1
8       3.1     t10     OA      OA      1       1
9       3.1     t10     OA      OA      1       1
10      3.1     t10     OA      OA      1       1
10      3.2     t10     OA      OA      1       1
;


end;
```

Appendix M: GLPK Code to Experiment Model (V)

```
# Written By Emad Sadoon
# Last modified 15 September 2018

param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups


set V, default (1..projects); #Set of projects
set G, default (1..resourceGroup);# set of resource groups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede task j
set M{v in V, p in P[v],j in J[v,p]}; # set of modes for task j of process p of project v
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]},default 0; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}, default 50; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}, default 50; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 0; /* duration of tasks j of process p of project
v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k usage
of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 0; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
param BR{v in V, g in G, k in K[g]}, default 1; /* Amount of renewable resource k of group g*/
#param R{g in G, k in K[g]}, default 0; /* total amount of renewable resource k*/

#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
#var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
```

```
var TC1{v in V, p in P[v]}>=0;/*Weighted tardiness cost of process p of project v*/
#var BR {v in V,g in G, k in K[g]} integer;
var R {g in G, k in K[g]} integer;

minimize z: sum {v in V, p in P[v]} TC1[v,p]; /* minimize the weighted tardiness of the project and
process*/
#minimize z: sum{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T}x[v,p,j,m,t];

#minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */

s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 mode for every task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in EE[v,p,jF]..LL[v,p,jF],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,s]..L[v,s],m in M[v,s,jL]}(t)*x[v,s,jL,m,t];/* predence
relationship for processes*/

#s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource  k of group g for each project v   */

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V,p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= R[g,k]; /* capacity constraint
for each resource k of group g   */

#s.t. Constraint_5{v in V, p in p_N[v],j in j_L[v,p]}: TC[v] >= c[v] *( sum{m in M[v,p,j],t in E[v,p]..L[v,p]}
(t*x[v,p,j,m,t])- dd[v]); /* tardiness of project is equal to some penalty c[v] times the difference of the
finish time of the last task of the last process minus the due date of the project */

s.t. Constraint_6{v in V, p in P[v], j in j_L[v,p], g in G}: TC1[v,p] >= c1[v,p]*(sum{m in M[v,p,j], t in
EE[v,p,j]..LL[v,p,j]} (t*x[v,p,j,m,t])- ddd[v,p]); /* tardiness of process is equal to some penalty c[v,p] times
the difference of the finish time of the last task of each process minus the due date of the process */


solve;

display z,x,R;

data;
param projects :=10;
param resourceGroup := 6;
param timePeriods := 50;
```

```
# Define Processes and Orders for each Project- The decimal number indicates the order number
#Every project has a dummy starting process called "p0" with 0 duration.

set P[1] := p0    1.1       3.1       3.2;
set P[2] := p0    1.1       3.1 ;
set P[3] := p0    3.1       3.2       3.3       3.4;
set P[4] := p0    3.1;
set P[5] := p0    3.1;
set P[6] := p0    1.1;
set P[7] := p0    3.1;
set P[8] := p0    3.1;
set P[9] := p0    3.1;
set P[10] := p0   3.1       3.2;


# Define tasks for each process:
set J[1,p0] := t0;
set J[1,1.1] := t1t2      t3        t4        t5        t6        t7        t8        t9        t10;
set J[1,3.1] := t6t7      t8        t9        t10;
set J[1,3.2] := t6t7      t8        t9        t10;

set J[2,p0] := t0;
set J[2,1.1] := t1t2      t3        t4        t5        t6        t7        t8        t9        t10;
set J[2,3.1] := t6t7      t8        t9        t10;

set J[3,p0]:=t0;
set J[3,3.1]:=t6  t7      t8        t9        t10;
set J[3,3.2]:=t6  t7      t8        t9        t10;
set J[3,3.3]:=t6  t7      t8        t9        t10;
set J[3,3.4]:=t6  t7      t8        t9        t10;

set J[4,p0] :=t0;
set J[4,3.1]:= t6 t7      t8        t9        t10;

set J[5,p0] :=t0;
set J[5,3.1] := t6t7      t8        t9        t10;

set J[6,p0] :=t0;
set J[6,1.1] := t1t2      t3        t4        t5        t6        t7        t8        t9        t10;

set J[7,p0] :=t0;
set J[7,3.1] := t6t7      t8        t9        t10;

set J[8,p0] :=t0;
set J[8,3.1] := t6t7      t8        t9        t10;

set J[9,p0] :=t0;
set J[9,3.1] := t6t7      t8        t9        t10;
set J[10,p0] :=t0;
set J[10,3.1] :=t6          t7        t8        t9        t10;
set J[10,3.2] :=t6          t7        t8        t9        t10;

# Define Mode
set M[1,p0,t0]:=1;
set M[1,1.1,t1]:=1;
set M[1,1.1,t2]:= 1;
set M[1,1.1,t3]:= 1;
set M[1,1.1,t4]:= 1;
set M[1,1.1,t5]:= 1;
set M[1,1.1,t6]:= 1;
set M[1,1.1,t7]:= 1;
set M[1,1.1,t8]:= 1;
set M[1,1.1,t9]:= 1;
set M[1,1.1,t10]:= 1;

set M[1,3.1,t6]:= 1;
set M[1,3.1,t7]:= 1;
set M[1,3.1,t8]:= 1;
set M[1,3.1,t9]:= 1;
set M[1,3.1,t10]:= 1;

set M[1,3.2,t6]:= 1;
set M[1,3.2,t7]:= 1;
set M[1,3.2,t8]:= 1;
set M[1,3.2,t9]:= 1;
set M[1,3.2,t10]:= 1;

set M[2,p0,t0]:=1;
set M[2,1.1,t1]:=1;
set M[2,1.1,t2]:= 1;
set M[2,1.1,t3]:= 1;
set M[2,1.1,t4]:= 1;
set M[2,1.1,t5]:= 1;
set M[2,1.1,t6]:= 1;
set M[2,1.1,t7]:= 1;
set M[2,1.1,t8]:= 1;
set M[2,1.1,t9]:= 1;
set M[2,1.1,t10]:= 1;

set M[2,3.1,t6]:= 1;
set M[2,3.1,t7]:= 1;
set M[2,3.1,t8]:= 1;
set M[2,3.1,t9]:= 1;
set M[2,3.1,t10]:= 1;
set M[3,p0,t0]:=1;
set M[3,3.1,t6]:= 1;
set M[3,3.1,t7]:= 1;
set M[3,3.1,t8]:= 1;
set M[3,3.1,t9]:= 1;
set M[3,3.1,t10]:= 1;

set M[3,3.2,t6]:= 1;
set M[3,3.2,t7]:= 1;
set M[3,3.2,t8]:= 1;
set M[3,3.2,t9]:= 1;
set M[3,3.2,t10]:= 1;

set M[3,3.3,t6]:= 1;
set M[3,3.3,t7]:= 1;
set M[3,3.3,t8]:= 1;
set M[3,3.3,t9]:= 1;
set M[3,3.3,t10]:= 1;

set M[3,3.4,t6]:= 1;
set M[3,3.4,t7]:= 1;
set M[3,3.4,t8]:= 1;
set M[3,3.4,t9]:= 1;
set M[3,3.4,t10]:= 1;

set M[4,p0,t0]:=1;
set M[4,3.1,t6]:= 1;
set M[4,3.1,t7]:= 1;
set M[4,3.1,t8]:= 1;
set M[4,3.1,t9]:= 1;
set M[4,3.1,t10]:= 1;

set M[5,p0,t0]:=1;
set M[5,3.1,t6]:= 1;
set M[5,3.1,t7]:= 1;
set M[5,3.1,t8]:= 1;
set M[5,3.1,t9]:= 1;
set M[5,3.1,t10]:= 1;

set M[6,p0,t0]:= 2;
set M[6,1.1,t1]:= 2;
set M[6,1.1,t2]:= 2;
set M[6,1.1,t3]:= 2;
set M[6,1.1,t4]:= 2;
set M[6,1.1,t5]:= 2;
set M[6,1.1,t6]:= 2;
set M[6,1.1,t7]:= 2;
set M[6,1.1,t8]:= 2;
set M[6,1.1,t9]:= 2;
set M[6,1.1,t10]:= 2;

set M[7,p0,t0]:= 2;
set M[7,3.1,t6]:= 2;
set M[7,3.1,t7]:= 2;
set M[7,3.1,t8]:= 2;
set M[7,3.1,t9]:= 2;
set M[7,3.1,t10]:= 2;

set M[8,p0,t0]:= 2;
set M[8,3.1,t6]:= 2;
set M[8,3.1,t7]:= 2;
set M[8,3.1,t8]:= 2;
set M[8,3.1,t9]:= 2;
set M[8,3.1,t10]:= 2;

set M[9,p0,t0]:= 2;
set M[9,3.1,t6]:= 2;
set M[9,3.1,t7]:= 2;
set M[9,3.1,t8]:= 2;
set M[9,3.1,t9]:= 2;
set M[9,3.1,t10]:= 2;

set M[10,p0,t0]:= 2;

set M[10,3.1,t6]:= 2;
set M[10,3.1,t7]:= 2;
set M[10,3.1,t8]:= 2;
set M[10,3.1,t9]:= 2;
set M[10,3.1,t10]:= 2;

set M[10,3.2,t6]:= 2;
set M[10,3.2,t7]:= 2;
set M[10,3.2,t8]:= 2;
set M[10,3.2,t9]:= 2;
set M[10,3.2,t10]:= 2;


# Task Precedence
set Q[1,1.1,t2] := t1;
set Q[1,1.1,t3] := t2;
set Q[1,1.1,t4]:= t3;
set Q[1,1.1,t5] := t4;
set Q[1,1.1,t6] := t5;
set Q[1,1.1,t7]:= t6;
set Q[1,1.1,t8]:= t7;
set Q[1,1.1,t9]:= t8;
set Q[1,1.1,t10]:= t9;

set Q[1,3.1,t7]:= t6;
set Q[1,3.1,t8] := t7;
set Q[1,3.1,t9]:= t8;
set Q[1,3.1,t10]:= t9;

set Q[1,3.2,t7]:= t6;
set Q[1,3.2,t8] := t7;
set Q[1,3.2,t9]:= t8;
set Q[1,3.2,t10]:= t9;

set Q[2,1.1,t2] := t1;
set Q[2,1.1,t3] := t2;
set Q[2,1.1,t4]:= t3;
set Q[2,1.1,t5] := t4;
set Q[2,1.1,t6] := t5;
set Q[2,1.1,t7]:= t6;
set Q[2,1.1,t8]:= t7;
set Q[2,1.1,t9]:= t8;
set Q[2,1.1,t10]:= t9;

set Q[2,3.1,t7]:= t6;
set Q[2,3.1,t8] := t7;
set Q[2,3.1,t9]:= t8;
set Q[2,3.1,t10]:= t9;

set Q[3,3.1,t7]:= t6;
set Q[3,3.1,t8] := t7;
set Q[3,3.1,t9]:= t8;
set Q[3,3.1,t10]:= t9;

set Q[3,3.2,t7]:= t6;
set Q[3,3.2,t8] := t7;
set Q[3,3.2,t9]:= t8;
set Q[3,3.2,t10]:= t9;

set Q[3,3.3,t7]:= t6;
set Q[3,3.3,t8] := t7;
set Q[3,3.3,t9]:= t8;
set Q[3,3.3,t10]:= t9;

set Q[3,3.4,t7]:= t6;
set Q[3,3.4,t8] := t7;
set Q[3,3.4,t9]:= t8;
set Q[3,3.4,t10]:= t9;

set Q[4,3.1,t7]:= t6;
set Q[4,3.1,t8] := t7;
set Q[4,3.1,t9]:= t8;
set Q[4,3.1,t10]:= t9;

set Q[5,3.1,t7]:= t6;
set Q[5,3.1,t8] := t7;
set Q[5,3.1,t9]:= t8;
set Q[5,3.1,t10]:= t9;

set Q[6,1.1,t2] := t1;
set Q[6,1.1,t3] := t2;
set Q[6,1.1,t4]:= t3;
set Q[6,1.1,t5] := t4;
set Q[6,1.1,t6] := t5;
set Q[6,1.1,t7]:= t6;
set Q[6,1.1,t8]:= t7;
set Q[6,1.1,t9]:= t8;
set Q[6,1.1,t10]:= t9;

set Q[7,3.1,t7]:= t6;
set Q[7,3.1,t8] := t7;
set Q[7,3.1,t9]:= t8;
set Q[7,3.1,t10]:= t9;

set Q[8,3.1,t7]:= t6;
set Q[8,3.1,t8] := t7;
set Q[8,3.1,t9]:= t8;
set Q[8,3.1,t10]:= t9;

set Q[9,3.1,t7]:= t6;
set Q[9,3.1,t8] := t7;
set Q[9,3.1,t9]:= t8;
set Q[9,3.1,t10]:= t9;

set Q[10,3.1,t7]:= t6;
set Q[10,3.1,t8] := t7;
set Q[10,3.1,t9]:= t8;
set Q[10,3.1,t10]:= t9;

set Q[10,3.2,t7]:= t6;
set Q[10,3.2,t8] := t7;
set Q[10,3.2,t9]:= t8;
set Q[10,3.2,t10]:= t9;


# Process Precedence
set S[1,1.1] := p0;
set S[1,3.1] := 1.1;
set S[1,3.2] := 3.1;

set S[2,1.1] := p0;
set S[2,3.1] := 1.1;

set S[3,3.1] :=p0;
set S[3,3.2] :=3.1;
set S[3,3.3] :=3.2;
set S[3,3.4] :=3.3;

set S[4,3.1] :=p0;

set S[5,3.1] :=p0;

set S[6,1.1] :=p0;

set S[7,3.1] :=p0;

set S[8,3.1] :=p0;

set S[9,3.1] :=p0;

set S[10,3.1] :=p0;
set S[10,3.2] :=3.1;

# Last task of each process
set j_L[1,p0] := t0;
set j_L[1,1.1] := t10;
set j_L[1,3.1] := t10;
set j_L[1,3.2] := t10;

set j_L[2,p0] := t0;
set j_L[2,1.1] := t10;
set j_L[2,3.1] := t10;

set j_L[3,3.1] := t10;
set j_L[3,3.2] := t10;
set j_L[3,3.3] := t10;
set j_L[3,3.4] := t10;

set j_L[4,3.1] := t10;

set j_L[5,3.1] := t10;

set j_L[6,1.1] := t10;

set j_L[7,3.1] := t10;
set j_L[8,3.1] := t10;

set j_L[9,3.1] := t10;

set j_L[10,3.1] := t10;
set j_L[10,3.2] := t10;

# First task of each process
set j_F[1,p0] := t0;
set j_F[1,1.1] := t1;
set j_F[1,3.1] := t6;
set j_F[1,3.2] := t6;

set j_F[2,p0] := t0;
set j_F[2,1.1] := t1;
set j_F[2,3.1] := t6;

set j_F[3,p0] := t0;
set j_F[3,3.1] := t6;
set j_F[3,3.2] := t6;
set j_F[3,3.3] := t6;
set j_F[3,3.4] := t6;

set j_F[4,p0] := t0;
set j_F[4,3.1] := t6;

set j_F[5,p0] := t0;
set j_F[5,3.1] := t6;

set j_F[6,p0] := t0;
set j_F[6,1.1] := t1;

set j_F[7,p0] := t0;
set j_F[7,3.1] := t6;

set j_F[8,p0] := t0;
set j_F[8,3.1] := t6;

set j_F[9,p0] := t0;
set j_F[9,3.1] := t6;

set j_F[10,p0] := t0;
set j_F[10,3.1] := t6;
set j_F[10,3.2] := t6;

# The group of people
set G:= PM1 PM2 QC1 QC2 OA1 OA2;
set K[PM1] := PM1; # The PM people
set K[OA1] := OA1 ; #The OA people
set K[QC1] := QC1 ; #The QC people

set K[PM2] := PM2; # The PM people
set K[OA2] := OA2 ; #The OA people
set K[QC2] := QC2 ; #The QC people

# k and g are equal in this example

# set the total number of each resource k
/*param R :=
PM1 PM1 1
OA1 OA1 1
QC1 QC1 1
PM2 PM2 1
OA2 OA2 1
QC2 QC2 1;*/


# Set the due dates for each process
param ddd :=
1        p0       0
1        1.1      19
1        3.1      24
1        3.2      29

2        p0       0
2        1.1      19
2        3.1      24

3        p0       0
3        3.1      20
3        3.2      25
3        3.3      30
3        3.4      35

4        p0       0
4        3.1      21

5        p0       0
5        3.1      20

6        p0       0
6        1.1      23

7        p0       0
7        3.1      22
```

```
8       p0      0
8       3.1     21

9       p0      0
9       3.1     22

10      p0      0
10      3.1     23
10      3.2     28
;

# Set the early finish of each process
param E :=
1       p0      0
1       1.1     15
1       3.1     22
1       3.2     27

2       p0      0
2       1.1     16
2       3.1     22

3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33

4       p0      0
4       3.1     19

5       p0      0
5       3.1     18

6       p0      0
6       1.1     19

7       p0      0
7       3.1     20

8       p0      0
8       3.1     19

9       p0      0
9       3.1     20

10      p0      0
```

```
10      3.1     21
10      3.2     26

;

# Set the late finish of each process
param L :=
1       p0      0
1       1.1     23
1       3.1     28
1       3.2     33
2       p0      0
2       1.1     28
2       3.1     33
3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33
4       p0      0
4       3.1     33
5       p0      0
5       3.1     33
6       p0      0
6       1.1     33
7       p0      0
7       3.1     33
8       p0      0
8       3.1     33
9       p0      0
9       3.1     23
10      p0      0
10      3.1     28
10      3.2     33
;

# Set the early finsh of each task
param EE :=
1 p0 t0 0
1       1.1     t1      7
2       1.1     t1      7
6       1.1     t1      11

1       1.1     t2      8
2       1.1     t2      8
6       1.1     t2      12

1       1.1     t3      9
```

2	1.1	t3	9
6	1.1	t3	13

1	1.1	t4	10
2	1.1	t4	10
6	1.1	t4	14

1	1.1	t5	11
2	1.1	t5	11
6	1.1	t5	15

1	1.1	t6	12
1	3.1	t6	18
1	3.2	t6	23
2	1.1	t6	12
2	3.1	t6	18
3	3.1	t6	14
3	3.2	t6	19
3	3.3	t6	24
3	3.4	t6	29
4	3.1	t6	15
5	3.1	t6	14
6	1.1	t6	10
7	3.1	t6	16
8	3.1	t6	15
9	3.1	t6	16
10	3.1	t6	17
10	3.2	t6	22

1	1.1	t7	13
1	3.1	t7	19
1	3.2	t7	24
2	1.1	t7	13
2	3.1	t7	19
3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	16
5	3.1	t7	15
6	1.1	t7	11
7	3.1	t7	17
8	3.1	t7	16
9	3.1	t7	17
10	3.1	t7	18
10	3.2	t7	23

1	1.1	t8	14

1	3.1	t8	20
1	3.2	t8	25
2	1.1	t8	14
2	3.1	t8	20
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	17
5	3.1	t8	16
6	1.1	t8	12
7	3.1	t8	18
8	3.1	t8	17
9	3.1	t8	18
10	3.1	t8	19
10	3.2	t8	24

1	1.1	t9	15
1	3.1	t9	21
1	3.2	t9	26
2	1.1	t9	15
2	3.1	t9	21
3	3.1	t9	17
3	3.2	t9	22
3	3.3	t9	27
3	3.4	t9	32
4	3.1	t9	18
5	3.1	t9	17
6	1.1	t9	13
7	3.1	t9	19
8	3.1	t9	18
9	3.1	t9	19
10	3.1	t9	20
10	3.2	t9	25

1	1.1	t10	16
1	3.1	t10	22
1	3.2	t10	27
2	1.1	t10	16
2	3.1	t10	22
3	3.1	t10	18
3	3.2	t10	23
3	3.3	t10	28
3	3.4	t10	33
4	3.1	t10	19
5	3.1	t10	18
6	1.1	t10	14
7	3.1	t10	20

```
8       3.1     t10     19
9       3.1     t10     20
10      3.1     t10     21
10      3.2     t10     26;

# Set the  late finish of each task
/*param LL :=

1       1.1     t1      14
2       1.1     t1      19
6       1.1     t1      24

1       1.1     t2      15
2       1.1     t2      20
6       1.1     t2      25

1       1.1     t3      15
2       1.1     t3      20
6       1.1     t3      25

1       1.1     t5      18
2       1.1     t5      23
6       1.1     t5      28

1       1.1     t6      19
1       3.1     t6      24
1       3.2     t6      29
2       1.1     t6      24
2       3.1     t6      29
3       3.1     t6      14
3       3.2     t6      19
3       3.3     t6      24
3       3.4     t6      29
4       3.1     t6      29
5       3.1     t6      29
6       1.1     t6      29
7       3.1     t6      29
8       3.1     t6      29
9       3.1     t6      19
10      3.1     t6      24
10      3.2     t6      29

1       1.1     t7      20
1       3.1     t7      25
1       3.2     t7      30
2       1.1     t7      25
2       3.1     t7      30
3       3.1     t7      15
```

3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	30
5	3.1	t7	30
6	1.1	t7	30
7	3.1	t7	30
8	3.1	t7	30
9	3.1	t7	20
10	3.1	t7	25
10	3.2	t7	30

1	1.1	t8	21
1	3.1	t8	26
1	3.2	t8	31
2	1.1	t8	26
2	3.1	t8	31
3	3.1	t8	16
3	3.2	t8	21
3	3.3	t8	26
3	3.4	t8	31
4	3.1	t8	31
5	3.1	t8	31
6	1.1	t8	31
7	3.1	t8	31
8	3.1	t8	31
9	3.1	t8	21
10	3.1	t8	26
10	3.2	t8	31

1	1.1	t9	22
1	3.1	t9	27
1	3.2	t9	32
2	1.1	t9	27
2	3.1	t9	32
3	3.1	t9	17
3	3.2	t9	22
3	3.3	t9	27
3	3.4	t9	32
4	3.1	t9	32
5	3.1	t9	32
6	1.1	t9	32
7	3.1	t9	32
8	3.1	t9	32
9	3.1	t9	22
10	3.1	t9	27
10	3.2	t9	32

```
1       1.1     t10     23
1       3.1     t10     28
1       3.2     t10     33
2       1.1     t10     28
2       3.1     t10     33
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     33
5       3.1     t10     33
6       1.1     t10     33
7       3.1     t10     33
8       3.1     t10     33
9       3.1     t10     23
10      3.1     t10     28
10      3.2     t10     33 ;*/

# Set Durations for each task
param d :=
1       1.1     t1      1       2
2       1.1     t1      1       2
6       1.1     t1      2       2

1       1.1     t2      1       1
2       1.1     t2      1       1
6       1.1     t2      2       1

1       1.1     t3      1       1
2       1.1     t3      1       1
6       1.1     t3      2       1

1       1.1     t4      1       1
2       1.1     t4      1       1
6       1.1     t4      2       1

1       1.1     t5      1       1
2       1.1     t5      1       1
6       1.1     t5      2       1

1       1.1     t6      1       1
1       3.1     t6      1       1
1       3.2     t6      1       1
2       1.1     t6      1       1
2       3.1     t6      1       1
3       3.1     t6      1       1
3       3.2     t6      1       1
```

3	3.3	t6	1	1
3	3.4	t6	1	1
4	3.1	t6	1	1
5	3.1	t6	1	1
6	1.1	t6	2	1
7	3.1	t6	2	1
8	3.1	t6	2	1
9	3.1	t6	2	1
10	3.1	t6	2	1
10	3.2	t6	2	1

1	1.1	t7	1	1
1	3.1	t7	1	1
1	3.2	t7	1	1
2	1.1	t7	1	1
2	3.1	t7	1	1
3	3.1	t7	1	1
3	3.2	t7	1	1
3	3.3	t7	1	1
3	3.4	t7	1	1
4	3.1	t7	1	1
5	3.1	t7	1	1
6	1.1	t7	2	1
7	3.1	t7	2	1
8	3.1	t7	2	1
9	3.1	t7	2	1
10	3.1	t7	2	1
10	3.2	t7	2	1

1	1.1	t8	1	1
1	3.1	t8	1	1
1	3.2	t8	1	1
2	1.1	t8	1	1
2	3.1	t8	1	1
3	3.1	t8	1	1
3	3.2	t8	1	1
3	3.3	t8	1	1
3	3.4	t8	1	1
4	3.1	t8	1	1
5	3.1	t8	1	1
6	1.1	t8	2	1
7	3.1	t8	2	1
8	3.1	t8	2	1
9	3.1	t8	2	1
10	3.1	t8	2	1
10	3.2	t8	2	1

1	1.1	t9	1	1

```
1       3.1     t9      1       1
1       3.2     t9      1       1
2       1.1     t9      1       1
2       3.1     t9      1       1
3       3.1     t9      1       1
3       3.2     t9      1       1
3       3.3     t9      1       1
3       3.4     t9      1       1
4       3.1     t9      1       1
5       3.1     t9      1       1
6       1.1     t9      2       1
7       3.1     t9      2       1
8       3.1     t9      2       1
9       3.1     t9      2       1
10      3.1     t9      2       1
10      3.2     t9      2       1

1       1.1     t10     1       1
1       3.1     t10     1       1
1       3.2     t10     1       1
2       1.1     t10     1       1
2       3.1     t10     1       1
3       3.1     t10     1       1
3       3.2     t10     1       1
3       3.3     t10     1       1
3       3.4     t10     1       1
4       3.1     t10     1       1
5       3.1     t10     1       1
6       1.1     t10     2       1
7       3.1     t10     2       1
8       3.1     t10     2       1
9       3.1     t10     2       1
10      3.1     t10     2       1
10      3.2     t10     2       1;

# Set the  number of resources of each employee (k)- param r is equal to 1 is that (K) is used ; otherwise
0 ( by default)
# index is  V,P,J,G,K,M,1/0
param r :=
1       1.1     t1      PM1     PM1     1       1
2       1.1     t1      PM1     PM1     1       1
6       1.1     t1      PM2     PM2     2       1

1       1.1     t2      QC1     QC1     1       1
2       1.1     t2      QC1     QC1     1       1
6       1.1     t2      QC2     QC2     2       1

1       1.1     t3      OA1     OA1     1       1
```

2	1.1	t3	OA1	OA1	1	1
6	1.1	t3	OA2	OA2	2	1
1	1.1	t4	QC1	QC1	1	1
2	1.1	t4	QC1	QC1	1	1
6	1.1	t4	QC2	QC2	2	1
1	1.1	t5	PM1	PM1	1	1
2	1.1	t5	PM1	PM1	1	1
6	1.1	t5	PM2	PM2	2	1
1	1.1	t6	OA1	OA1	1	1
1	3.1	t6	OA1	OA1	1	1
1	3.2	t6	OA1	OA1	1	1
2	1.1	t6	OA1	OA1	1	1
2	3.1	t6	OA1	OA1	1	1
3	3.1	t6	OA1	OA1	1	1
3	3.2	t6	OA1	OA1	1	1
3	3.3	t6	OA1	OA1	1	1
3	3.4	t6	OA1	OA1	1	1
4	3.1	t6	OA1	OA1	1	1
5	3.1	t6	OA1	OA1	1	1
6	1.1	t6	OA2	OA2	2	1
7	3.1	t6	OA2	OA2	2	1
8	3.1	t6	OA2	OA2	2	1
9	3.1	t6	OA2	OA2	2	1
10	3.1	t6	OA2	OA2	2	1
10	3.2	t6	OA2	OA2	2	1
1	1.1	t7	QC1	QC1	1	1
1	3.1	t7	QC1	QC1	1	1
1	3.2	t7	QC1	QC1	1	1
2	1.1	t7	QC1	QC1	1	1
2	3.1	t7	QC1	QC1	1	1
3	3.1	t7	QC1	QC1	1	1
3	3.2	t7	QC1	QC1	1	1
3	3.3	t7	QC1	QC1	1	1
3	3.4	t7	QC1	QC1	1	1
4	3.1	t7	QC1	QC1	1	1
5	3.1	t7	QC1	QC1	1	1
6	1.1	t7	QC2	QC2	2	1
7	3.1	t7	QC2	QC2	2	1
8	3.1	t7	QC2	QC2	2	1
9	3.1	t7	QC2	QC2	2	1
10	3.1	t7	QC2	QC2	2	1
10	3.2	t7	QC2	QC2	2	1
1	1.1	t8	OA1	OA1	1	1

1	3.1	t8	OA1	OA1	1	1
1	3.2	t8	OA1	OA1	1	1
2	1.1	t8	OA1	OA1	1	1
2	3.1	t8	OA1	OA1	1	1
3	3.1	t8	OA1	OA1	1	1
3	3.2	t8	OA1	OA1	1	1
3	3.3	t8	OA1	OA1	1	1
3	3.4	t8	OA1	OA1	1	1
4	3.1	t8	OA1	OA1	1	1
5	3.1	t8	OA1	OA1	1	1
6	1.1	t8	OA2	OA2	2	1
7	3.1	t8	OA2	OA2	2	1
8	3.1	t8	OA2	OA2	2	1
9	3.1	t8	OA2	OA2	2	1
10	3.1	t8	OA2	OA2	2	1
10	3.2	t8	OA2	OA2	2	1
1	1.1	t9	QC1	QC1	1	1
1	3.1	t9	QC1	QC1	1	1
1	3.2	t9	QC1	QC1	1	1
2	1.1	t9	QC1	QC1	1	1
2	3.1	t9	QC1	QC1	1	1
3	3.1	t9	QC1	QC1	1	1
3	3.2	t9	QC1	QC1	1	1
3	3.3	t9	QC1	QC1	1	1
3	3.4	t9	QC1	QC1	1	1
4	3.1	t9	QC1	QC1	1	1
5	3.1	t9	QC1	QC1	1	1
6	1.1	t9	QC2	QC2	2	1
7	3.1	t9	QC2	QC2	2	1
8	3.1	t9	QC2	QC2	2	1
9	3.1	t9	QC2	QC2	2	1
10	3.1	t9	QC2	QC2	2	1
10	3.2	t9	QC2	QC2	2	1
1	1.1	t10	PM1	PM1	1	1
1	3.1	t10	PM1	PM1	1	1
1	3.2	t10	PM1	PM1	1	1
2	1.1	t10	PM1	PM1	1	1
2	3.1	t10	PM1	PM1	1	1
3	3.1	t10	PM1	PM1	1	1
3	3.2	t10	PM1	PM1	1	1
3	3.3	t10	PM1	PM1	1	1
3	3.4	t10	PM1	PM1	1	1
4	3.1	t10	PM1	PM1	1	1
5	3.1	t10	PM1	PM1	1	1
6	1.1	t10	PM2	PM2	2	1
7	3.1	t10	PM2	PM2	2	1

```
8       3.1     t10     PM2     PM2     2       1
9       3.1     t10     PM2     PM2     2       1
10      3.1     t10     PM2     PM2     2       1
10      3.2     t10     PM2     PM2     2       1

1       1.1     t10     OA1     OA1     1       1
1       3.1     t10     OA1     OA1     1       1
1       3.2     t10     OA1     OA1     1       1
2       1.1     t10     OA1     OA1     1       1
2       3.1     t10     OA1     OA1     1       1
3       3.1     t10     OA1     OA1     1       1
3       3.2     t10     OA1     OA1     1       1
3       3.3     t10     OA1     OA1     1       1
3       3.4     t10     OA1     OA1     1       1
4       3.1     t10     OA1     OA1     1       1
5       3.1     t10     OA1     OA1     1       1
6       1.1     t10     OA2     OA2     2       1
7       3.1     t10     OA2     OA2     2       1
8       3.1     t10     OA2     OA2     2       1
9       3.1     t10     OA2     OA2     2       1
10      3.1     t10     OA2     OA2     2       1
10      3.2     t10     OA2     OA2     2       1
;


end;
```

Appendix N: GLPK Code to Experiment Model (VI)

```
# Written By Emad Sadoon
# Last modified 03 November 2018


param projects; #Number of projects
param work group; #Number of project work groups
param resourceGroup; #Number of projects
param timePeriods; #Number of project work groups

set W, default (1.. work group); # Set of project work groups
set V, default (1..projects); #Set of projects
set G /*default (1..resourceGroup)*/;# set of resourcegroups
set T, default (0..timePeriods);# set of time periods.

set P{v in V}, default {};# set of processes of project v
set J{v in V, p in P[v]}, default {}; #set of tasks of process p of project v
set p_N{v in V} within P[v], default {};#   last process (indexed with N)  of project v
set j_L{v in V, p in P[v]} within J[v,p], default{}; #last task (indexed with L)  of process p of project v
set j_F{v in V, p in P[v]} within J[v,p], default{}; #first task (indexed with F) of process p of project v
set S{v in V, p in P[v]} default {};# S[v,p] is a subset of processes that immediately precede process p
set Q{v in V, p in P[v], j in J[v,p]}, default {}; # Q[v,p,j] is a subset of tasks that immediately precede task j
set M{v in V, p in P[v],j in J[v,p]}, default {1,2}; # set of modes for task j of process p of project v
set MW{w in W}, default {}; # set of modes in project work group w
set K{g in G}, default {}; #set of renewal resources for group g


#Model Parameters
param EE{v in V, p in P[v], j in J[v,p]},default 0; /*earliest finish time of task j of process p of project v*/
param LL{v in V, p in P[v], j in J[v,p]}, default 50; /*latest finish time of task  j of process p of project v*/
param E{v in V, p in P[v]}; /*earliest finish time of process p of project v*/
param L{v in V, p in P[v]}, default 50; /* latest finish time of process p of project v*/
param d{v in V, p in P[v], j in J[v,p], m in M[v,p,j]}, default 0; /* duration of tasks j of process p of project v, operating by resource k of group g*/
param r{v in V, p in P[v], j in J[v,p],g in G,k in K[g], m in M[v,p,j]}, default 0;/* renewable resource k usage of activity j of project v, operating on mode m */
param dd{v in V}; /*assigned due date for project v*/
param ddd{v in V, p in P[v]}, default 0; /*due date for the  process p of project v*/
param c{v in V}; /*relative weight of project v*/
param c1{v in V, p in P[v]}, default 1; /*relative weight of project v of process p*/
param b{g in G, k in K[g]}; /* unit cost of renewable resource k belonging to group g*/
param td, default 1000; /* total resource budget*/
#param BR{v in V, g in G, k in K[g]}, default 1; /* Amount of renewable resource k of group g*/
#param R{g in G, k in K[g]}, default 0; /* Represnts total amount of resources of group g*/
```

```
#Model Variables
var x{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T},binary;
#var TC{v in V}, >=0;/*Weighted tardiness cost of project v*/
var TC1{v in V, p in P[v]}>=0, integer;/*Weighted tardiness cost of process p of project v*/
#var BR {v in V,g in G, k in K[g]} integer;
var R {g in G, k in K[g]} integer; /* Represnts total amount of resources of group g*/
var y {v in V, w in W}, binary; /* Binary variable y to assign to work group to each project*/

minimize z: sum {v in V, p in P[v]} TC1[v,p]; /* minimize the weighted tardiness of the project and
process*/
#minimize z: sum{v in V, p in P[v], j in J[v,p], m in M[v,p,j], t in T}x[v,p,j,m,t];

#minimize z: sum {v in V, p in P[v],j in j_L[v,p],m in M[v,p,j], t in EE[v,p,j]..LL[v,p,j]}t*x[v,p,j,m,t]; /*
minimize project completion time */

s.t. constraint_1{v in V, p in P[v], j in J[v,p]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]} x[v,p,j,m,t] =
1;#1;/*Assign only 1 finish time (t) and 1 mode for every task */

s.t. constraint_2{v in V, p in P[v], j in J[v,p],q in Q[v,p,j]}: sum{t in EE[v,p,j]..LL[v,p,j],m in M[v,p,j]}(t-
d[v,p,j,m])*x[v,p,j,m,t] >= sum{t in EE[v,p,q]..LL[v,p,q],m in M[v,p,q]}(t)*x[v,p,q,m,t];/* precedence
relationsip for tasks*/

s.t. constraint_3{v in V, p in P[v],s in S[v,p],jF in j_F[v,p], jL in j_L[v,s]}: sum{t in EE[v,p,jF]..LL[v,p,jF],m in
M[v,p,jF]}(t-d[v,p,jF,m])*x[v,p,jF,m,t] >= sum{t in E[v,s]..L[v,s],m in M[v,s,jL]}(t)*x[v,s,jL,m,t];/* predence
relationship for processes*/

#s.t. constraint_4{v in V,g in G, k in K[g],t in T}: sum{p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= BR[v,g,k]; /* capacity
constraint for each resource  k of group g for each project v   */

s.t. constraint_4{g in G, k in K[g],t in T}: sum{v in V,p in P[v],j in J[v,p],m in M[v,p,j],q in
max(t,EE[v,p,j])..min(t + d[v,p,j,m]-1,LL[v,p,j]) }r[v,p,j,g,k,m]*x[v,p,j,m,q]<= R[g,k]; /* capacity constraint
for each resource k of group g   */

#s.t. Constraint_5{v in V, p in p_N[v],j in j_L[v,p]}: TC[v] >= c[v] *( sum{m in M[v,p,j],t in E[v,p]..L[v,p]}
(t*x[v,p,j,m,t])- dd[v]); /* tardiness of project is equal to some penalty c[v] times the difference of the
finish time of the last task of the last process minus the due date of the project */

s.t. Constraint_6{v in V, p in P[v], j in j_L[v,p], g in G}: TC1[v,p] >= c1[v,p]*(sum{m in M[v,p,j], t in
EE[v,p,j]..LL[v,p,j]} (t*x[v,p,j,m,t])- ddd[v,p]); /* tardiness of process is equal to some penalty c[v,p] times
the difference of the finish time of the last task of each process minus the due date of the process */
 s.t. Constraint_7 {v in V}:sum {w in W} y[v,w]=1; # assign a project to only one work group

 s.t. Constarint_8 {v in V,  p in P[v], j in J[v,p],w in W,m in MW[w],t in T}:x[v,p,j,m,t]<=y[v,w]; #use the
resources m from the work group that project v is part of.

solve;
```

```
display z,x,R,y;

data;
param projects :=10;
param resourceGroup := 6;
param timePeriods := 50;
param work group:=2;

# Define Processes and Orders for each Project- The decimal number indicates the order number
#Every project has a dummy starting process called "p0" with 0 duration.

set P[1] := p0    1.1       3.1       3.2;
set P[2] := p0    1.1       3.1 ;
set P[3] := p0    3.1       3.2       3.3       3.4;
set P[4] := p0    3.1;
set P[5] := p0    3.1;
set P[6] := p0    1.1;
set P[7] := p0    3.1;
set P[8] := p0    3.1;
set P[9] := p0    3.1;
set P[10] := p0   3.1       3.2;


# Define tasks for each process:
set J[1,p0] := t0;
set J[1,1.1] := t1t2      t3        t4        t5        t6        t7        t8        t9        t10;
set J[1,3.1] := t6t7      t8        t9        t10;
set J[1,3.2] := t6t7      t8        t9        t10;

set J[2,p0] := t0;
set J[2,1.1] := t1t2      t3        t4        t5        t6        t7        t8        t9        t10;
set J[2,3.1] := t6t7      t8        t9        t10;

set J[3,p0]:=t0;
set J[3,3.1]:=t6  t7      t8        t9        t10;
set J[3,3.2]:=t6  t7      t8        t9        t10;
set J[3,3.3]:=t6  t7      t8        t9        t10;
set J[3,3.4]:=t6  t7      t8        t9        t10;

set J[4,p0] :=t0;
set J[4,3.1]:= t6 t7      t8        t9        t10;

set J[5,p0] :=t0;
set J[5,3.1] := t6t7      t8        t9        t10;

set J[6,p0] :=t0;
set J[6,1.1] := t1t2      t3        t4        t5        t6        t7        t8        t9        t10;
set J[7,p0] :=t0;
set J[7,3.1] := t6t7        t8        t9        t10;

set J[8,p0] :=t0;
set J[8,3.1] := t6t7        t8        t9        t10;

set J[9,p0] :=t0;
set J[9,3.1] := t6t7        t8        t9        t10;

set J[10,p0] :=t0;
set J[10,3.1] :=t6          t7        t8        t9        t10;
set J[10,3.2] :=t6          t7        t8        t9        t10;

# Define Mode- by default give modes (1,2,3) to each task

set MW[1]:=1; # set of modes of work group 1.
set MW[2]:=2; # set of modes of work group 2


# Task Precedence
set Q[1,1.1,t2] := t1;
set Q[1,1.1,t3] := t2;
set Q[1,1.1,t4]:= t3;
set Q[1,1.1,t5] := t4;
set Q[1,1.1,t6] := t5;
set Q[1,1.1,t7]:= t6;
set Q[1,1.1,t8]:= t7;
set Q[1,1.1,t9]:= t8;
set Q[1,1.1,t10]:= t9;

set Q[1,3.1,t7]:= t6;
set Q[1,3.1,t8] := t7;
set Q[1,3.1,t9]:= t8;
set Q[1,3.1,t10]:= t9;

set Q[1,3.2,t7]:= t6;
set Q[1,3.2,t8] := t7;
set Q[1,3.2,t9]:= t8;
set Q[1,3.2,t10]:= t9;

set Q[2,1.1,t2] := t1;
set Q[2,1.1,t3] := t2;
set Q[2,1.1,t4]:= t3;
set Q[2,1.1,t5] := t4;
set Q[2,1.1,t6] := t5;
set Q[2,1.1,t7]:= t6;
set Q[2,1.1,t8]:= t7;
set Q[2,1.1,t9]:= t8;
set Q[2,1.1,t10]:= t9;

set Q[2,3.1,t7]:= t6;
set Q[2,3.1,t8] := t7;
set Q[2,3.1,t9]:= t8;
set Q[2,3.1,t10]:= t9;

set Q[3,3.1,t7]:= t6;
set Q[3,3.1,t8] := t7;
set Q[3,3.1,t9]:= t8;
set Q[3,3.1,t10]:= t9;

set Q[3,3.2,t7]:= t6;
set Q[3,3.2,t8] := t7;
set Q[3,3.2,t9]:= t8;
set Q[3,3.2,t10]:= t9;

set Q[3,3.3,t7]:= t6;
set Q[3,3.3,t8] := t7;
set Q[3,3.3,t9]:= t8;
set Q[3,3.3,t10]:= t9;

set Q[3,3.4,t7]:= t6;
set Q[3,3.4,t8] := t7;
set Q[3,3.4,t9]:= t8;
set Q[3,3.4,t10]:= t9;

set Q[4,3.1,t7]:= t6;
set Q[4,3.1,t8] := t7;
set Q[4,3.1,t9]:= t8;
set Q[4,3.1,t10]:= t9;

set Q[5,3.1,t7]:= t6;
set Q[5,3.1,t8] := t7;
set Q[5,3.1,t9]:= t8;
set Q[5,3.1,t10]:= t9;

set Q[6,1.1,t2] := t1;
set Q[6,1.1,t3] := t2;
set Q[6,1.1,t4]:= t3;
set Q[6,1.1,t5] := t4;
set Q[6,1.1,t6] := t5;
set Q[6,1.1,t7]:= t6;
set Q[6,1.1,t8]:= t7;
set Q[6,1.1,t9]:= t8;
set Q[6,1.1,t10]:= t9;

set Q[7,3.1,t7]:= t6;
set Q[7,3.1,t8] := t7;
set Q[7,3.1,t9]:= t8;
set Q[7,3.1,t10]:= t9;

set Q[8,3.1,t7]:= t6;
set Q[8,3.1,t8] := t7;
set Q[8,3.1,t9]:= t8;
set Q[8,3.1,t10]:= t9;

set Q[9,3.1,t7]:= t6;
set Q[9,3.1,t8] := t7;
set Q[9,3.1,t9]:= t8;
set Q[9,3.1,t10]:= t9;

set Q[10,3.1,t7]:= t6;
set Q[10,3.1,t8] := t7;
set Q[10,3.1,t9]:= t8;
set Q[10,3.1,t10]:= t9;

set Q[10,3.2,t7]:= t6;
set Q[10,3.2,t8] := t7;
set Q[10,3.2,t9]:= t8;
set Q[10,3.2,t10]:= t9;


# Process Precedence
set S[1,1.1] := p0;
set S[1,3.1] := 1.1;
set S[1,3.2] := 3.1;

set S[2,1.1] := p0;
set S[2,3.1] := 1.1;

set S[3,3.1] :=p0;
set S[3,3.2] :=3.1;
set S[3,3.3] :=3.2;
set S[3,3.4] :=3.3;

set S[4,3.1] :=p0;

set S[5,3.1] :=p0;

set S[6,1.1] :=p0;

set S[7,3.1] :=p0;

set S[8,3.1] :=p0;
set S[9,3.1] :=p0;

set S[10,3.1] :=p0;
set S[10,3.2] :=3.1;

# Last task of each process
set j_L[1,p0] := t0;
set j_L[1,1.1] := t10;
set j_L[1,3.1] := t10;
set j_L[1,3.2] := t10;

set j_L[2,p0] := t0;
set j_L[2,1.1] := t10;
set j_L[2,3.1] := t10;

set j_L[3,3.1] := t10;
set j_L[3,3.2] := t10;
set j_L[3,3.3] := t10;
set j_L[3,3.4] := t10;

set j_L[4,3.1] := t10;

set j_L[5,3.1] := t10;

set j_L[6,1.1] := t10;

set j_L[7,3.1] := t10;

set j_L[8,3.1] := t10;

set j_L[9,3.1] := t10;

set j_L[10,3.1] := t10;
set j_L[10,3.2] := t10;

# First task of each process
set j_F[1,p0] := t0;
set j_F[1,1.1] := t1;
set j_F[1,3.1] := t6;
set j_F[1,3.2] := t6;

set j_F[2,p0] := t0;
set j_F[2,1.1] := t1;
set j_F[2,3.1] := t6;

set j_F[3,p0] := t0;
set j_F[3,3.1] := t6;
set j_F[3,3.2] := t6;
set j_F[3,3.3] := t6;
set j_F[3,3.4] := t6;

set j_F[4,p0] := t0;
set j_F[4,3.1] := t6;

set j_F[5,p0] := t0;
set j_F[5,3.1] := t6;

set j_F[6,p0] := t0;
set j_F[6,1.1] := t1;

set j_F[7,p0] := t0;
set j_F[7,3.1] := t6;

set j_F[8,p0] := t0;
set j_F[8,3.1] := t6;

set j_F[9,p0] := t0;
set j_F[9,3.1] := t6;

set j_F[10,p0] := t0;
set j_F[10,3.1] := t6;
set j_F[10,3.2] := t6;

# The group of people
set G:= PM1 PM2 QC1 QC2 OA1 OA2 ;

# The PM people
set K[PM1] := PM1;
set K[PM2] := PM2;

#The OA people
set K[OA1] := OA1 ;
set K[OA2] := OA2 ;

#The QC people
set K[QC1] := QC1 ;
set K[QC2] := QC2 ;

# k and g are equal in this example

# set the total number of each resource k
/*param R :=
PM1 PM1 1
OA1 OA1 1
QC1 QC1 1
```

```
PM2 PM2 1
OA2 OA2 1
QC2 QC2 1 ;*/

# Set the due dates for each process
param ddd :=
1       p0      0
1       1.1     19
1       3.1     24
1       3.2     29

2       p0      0
2       1.1     19
2       3.1     24

3       p0      0
3       3.1     20
3       3.2     25
3       3.3     30
3       3.4     35

4       p0      0
4       3.1     21

5       p0      0
5       3.1     20

6       p0      0
6       1.1     23

7       p0      0
7       3.1     22

8       p0      0
8       3.1     21

9       p0      0
9       3.1     22

10      p0      0
10      3.1     23
10      3.2     28
;

# Set the early finish of each process
param E :=
1       p0      0
1       1.1     15
```

```
1       3.1     22
1       3.2     27

2       p0      0
2       1.1     16
2       3.1     22

3       p0      0
3       3.1     18
3       3.2     23
3       3.3     28
3       3.4     33

4       p0      0
4       3.1     19

5       p0      0
5       3.1     18

6       p0      0
6       1.1     19

7       p0      0
7       3.1     20

8       p0      0
8       3.1     19

9       p0      0
9       3.1     20

10      p0      0
10      3.1     21
10      3.2     26

;

# Set the late finish of each process
/*param L :=
1       p0      0
1       1.1     50
1       3.1     50
1       3.2     50
2       p0      0
2       1.1     50
2       3.1     50
3       p0      0
3       3.1     50
```

```
3       3.2     50
3       3.3     50
3       3.4     50
4       p0      0
4       3.1     50
5       p0      0
5       3.1     50
6       p0      0
6       1.1     50
7       p0      0
7       3.1     50
8       p0      0
8       3.1     50
9       p0      0
9       3.1     50
10      p0      0
10      3.1     50
10      3.2     50
;*/

# Set the early finsh of each task
param EE :=
1 p0 t0 0
2 p0 t0 0
3 p0 t0 0
4 p0 t0 0
5 p0 t0 0
6 p0 t0 0
7 p0 t0 0
8 p0 t0 0
9 p0 t0 0
10 p0 t0 0

1       1.1     t1      7
2       1.1     t1      7
6       1.1     t1      11

1       1.1     t2      8
2       1.1     t2      8
6       1.1     t2      12

1       1.1     t3      9
2       1.1     t3      9
6       1.1     t3      13

1       1.1     t4      10
2       1.1     t4      10
6       1.1     t4      14
```

1	1.1	t5	11
2	1.1	t5	11
6	1.1	t5	15

1	1.1	t6	12
1	3.1	t6	18
1	3.2	t6	23
2	1.1	t6	12
2	3.1	t6	18
3	3.1	t6	14
3	3.2	t6	19
3	3.3	t6	24
3	3.4	t6	29
4	3.1	t6	15
5	3.1	t6	14
6	1.1	t6	10
7	3.1	t6	16
8	3.1	t6	15
9	3.1	t6	16
10	3.1	t6	17
10	3.2	t6	22

1	1.1	t7	13
1	3.1	t7	19
1	3.2	t7	24
2	1.1	t7	13
2	3.1	t7	19
3	3.1	t7	15
3	3.2	t7	20
3	3.3	t7	25
3	3.4	t7	30
4	3.1	t7	16
5	3.1	t7	15
6	1.1	t7	11
7	3.1	t7	17
8	3.1	t7	16
9	3.1	t7	17
10	3.1	t7	18
10	3.2	t7	23

1	1.1	t8	14
1	3.1	t8	20
1	3.2	t8	25
2	1.1	t8	14
2	3.1	t8	20
3	3.1	t8	16
3	3.2	t8	21

```
3       3.3     t8      26
3       3.4     t8      31
4       3.1     t8      17
5       3.1     t8      16
6       1.1     t8      12
7       3.1     t8      18
8       3.1     t8      17
9       3.1     t8      18
10      3.1     t8      19
10      3.2     t8      24

1       1.1     t9      15
1       3.1     t9      21
1       3.2     t9      26
2       1.1     t9      15
2       3.1     t9      21
3       3.1     t9      17
3       3.2     t9      22
3       3.3     t9      27
3       3.4     t9      32
4       3.1     t9      18
5       3.1     t9      17
6       1.1     t9      13
7       3.1     t9      19
8       3.1     t9      18
9       3.1     t9      19
10      3.1     t9      20
10      3.2     t9      25

1       1.1     t10     16
1       3.1     t10     22
1       3.2     t10     27
2       1.1     t10     16
2       3.1     t10     22
3       3.1     t10     18
3       3.2     t10     23
3       3.3     t10     28
3       3.4     t10     33
4       3.1     t10     19
5       3.1     t10     18
6       1.1     t10     14
7       3.1     t10     20
8       3.1     t10     19
9       3.1     t10     20
10      3.1     t10     21
10      3.2     t10     26;

# Set the  late finish of each task
```

```
/*param LL :=
1 p0 t0 0
2 p0 t0 0
3 p0 t0 0
4 p0 t0 0
5 p0 t0 0
6 p0 t0 0
7 p0 t0 0
8 p0 t0 0
9 p0 t0 0
10 p0 t0 0

1       1.1     t1      50
2       1.1     t1      50
6       1.1     t1      50

1       1.1     t2      50
2       1.1     t2      50
6       1.1     t2      50

1       1.1     t3      50
2       1.1     t3      50
6       1.1     t3      50

1       1.1     t5      50
2       1.1     t5      50
6       1.1     t5      50

1       1.1     t6      50
1       3.1     t6      50
1       3.2     t6      50
2       1.1     t6      50
2       3.1     t6      50
3       3.1     t6      50
3       3.2     t6      50
3       3.3     t6      50
3       3.4     t6      50
4       3.1     t6      50
5       3.1     t6      50
6       1.1     t6      50
7       3.1     t6      50
8       3.1     t6      50
9       3.1     t6      50
10      3.1     t6      50
10      3.2     t6      50

1       1.1     t7      50
1       3.1     t7      50
```

1	3.2	t7	50
2	1.1	t7	50
2	3.1	t7	50
3	3.1	t7	50
3	3.2	t7	50
3	3.3	t7	50
3	3.4	t7	50
4	3.1	t7	50
5	3.1	t7	50
6	1.1	t7	50
7	3.1	t7	50
8	3.1	t7	50
9	3.1	t7	50
10	3.1	t7	50
10	3.2	t7	50

1	1.1	t8	50
1	3.1	t8	50
1	3.2	t8	50
2	1.1	t8	50
2	3.1	t8	50
3	3.1	t8	50
3	3.2	t8	50
3	3.3	t8	50
3	3.4	t8	50
4	3.1	t8	50
5	3.1	t8	50
6	1.1	t8	50
7	3.1	t8	50
8	3.1	t8	50
9	3.1	t8	50
10	3.1	t8	50
10	3.2	t8	50

1	1.1	t9	50
1	3.1	t9	50
1	3.2	t9	50
2	1.1	t9	50
2	3.1	t9	50
3	3.1	t9	50
3	3.2	t9	50
3	3.3	t9	50
3	3.4	t9	50
4	3.1	t9	50
5	3.1	t9	50
6	1.1	t9	50
7	3.1	t9	50

```
8       3.1     t9      50
9       3.1     t9      50
10      3.1     t9      50
10      3.2     t9      50

1       1.1     t10     50
1       3.1     t10     50
1       3.2     t10     50
2       1.1     t10     50
2       3.1     t10     50
3       3.1     t10     50
3       3.2     t10     50
3       3.3     t10     50
3       3.4     t10     50
4       3.1     t10     50
5       3.1     t10     50
6       1.1     t10     50
7       3.1     t10     50
8       3.1     t10     50
9       3.1     t10     50
10      3.1     t10     50
10      3.2     t10     50;*/

# Set Durations for each task
# Set Durations for each task
param d :=
1       1.1     t1      1       2
2       1.1     t1      1       2
6       1.1     t1      1       2

1       1.1     t2      1       1
2       1.1     t2      1       1
6       1.1     t2      1       1

1       1.1     t3      1       1
2       1.1     t3      1       1
6       1.1     t3      1       1

1       1.1     t4      1       1
2       1.1     t4      1       1
6       1.1     t4      1       1

1       1.1     t5      1       1
2       1.1     t5      1       1
6       1.1     t5      1       1

1       1.1     t6      1       1
1       3.1     t6      1       1
```

1	3.2	t6	1	1
2	1.1	t6	1	1
2	3.1	t6	1	1
3	3.1	t6	1	1
3	3.2	t6	1	1
3	3.3	t6	1	1
3	3.4	t6	1	1
4	3.1	t6	1	1
5	3.1	t6	1	1
6	1.1	t6	1	1
7	3.1	t6	1	1
8	3.1	t6	1	1
9	3.1	t6	1	1
10	3.1	t6	1	1
10	3.2	t6	1	1

1	1.1	t7	1	1
1	3.1	t7	1	1
1	3.2	t7	1	1
2	1.1	t7	1	1
2	3.1	t7	1	1
3	3.1	t7	1	1
3	3.2	t7	1	1
3	3.3	t7	1	1
3	3.4	t7	1	1
4	3.1	t7	1	1
5	3.1	t7	1	1
6	1.1	t7	1	1
7	3.1	t7	1	1
8	3.1	t7	1	1
9	3.1	t7	1	1
10	3.1	t7	1	1
10	3.2	t7	1	1

1	1.1	t8	1	1
1	3.1	t8	1	1
1	3.2	t8	1	1
2	1.1	t8	1	1
2	3.1	t8	1	1
3	3.1	t8	1	1
3	3.2	t8	1	1
3	3.3	t8	1	1
3	3.4	t8	1	1
4	3.1	t8	1	1
5	3.1	t8	1	1
6	1.1	t8	1	1
7	3.1	t8	1	1
8	3.1	t8	1	1

9	3.1	t8	1	1
10	3.1	t8	1	1
10	3.2	t8	1	1
1	1.1	t9	1	1
1	3.1	t9	1	1
1	3.2	t9	1	1
2	1.1	t9	1	1
2	3.1	t9	1	1
3	3.1	t9	1	1
3	3.2	t9	1	1
3	3.3	t9	1	1
3	3.4	t9	1	1
4	3.1	t9	1	1
5	3.1	t9	1	1
6	1.1	t9	1	1
7	3.1	t9	1	1
8	3.1	t9	1	1
9	3.1	t9	1	1
10	3.1	t9	1	1
10	3.2	t9	1	1
1	1.1	t10	1	1
1	3.1	t10	1	1
1	3.2	t10	1	1
2	1.1	t10	1	1
2	3.1	t10	1	1
3	3.1	t10	1	1
3	3.2	t10	1	1
3	3.3	t10	1	1
3	3.4	t10	1	1
4	3.1	t10	1	1
5	3.1	t10	1	1
6	1.1	t10	1	1
7	3.1	t10	1	1
8	3.1	t10	1	1
9	3.1	t10	1	1
10	3.1	t10	1	1
10	3.2	t10	1	1
1	1.1	t1	2	2
2	1.1	t1	2	2
6	1.1	t1	2	2
1	1.1	t2	2	1
2	1.1	t2	2	1
6	1.1	t2	2	1

1	1.1	t3	2	1
2	1.1	t3	2	1
6	1.1	t3	2	1
1	1.1	t4	2	1
2	1.1	t4	2	1
6	1.1	t4	2	1
1	1.1	t5	2	1
2	1.1	t5	2	1
6	1.1	t5	2	1
1	1.1	t6	2	1
1	3.1	t6	2	1
1	3.2	t6	2	1
2	1.1	t6	2	1
2	3.1	t6	2	1
3	3.1	t6	2	1
3	3.2	t6	2	1
3	3.3	t6	2	1
3	3.4	t6	2	1
4	3.1	t6	2	1
5	3.1	t6	2	1
6	1.1	t6	2	1
7	3.1	t6	2	1
8	3.1	t6	2	1
9	3.1	t6	2	1
10	3.1	t6	2	1
10	3.2	t6	2	1
1	1.1	t7	2	1
1	3.1	t7	2	1
1	3.2	t7	2	1
2	1.1	t7	2	1
2	3.1	t7	2	1
3	3.1	t7	2	1
3	3.2	t7	2	1
3	3.3	t7	2	1
3	3.4	t7	2	1
4	3.1	t7	2	1
5	3.1	t7	2	1
6	1.1	t7	2	1
7	3.1	t7	2	1
8	3.1	t7	2	1
9	3.1	t7	2	1
10	3.1	t7	2	1
10	3.2	t7	2	1

1	1.1	t8	2	1
1	3.1	t8	2	1
1	3.2	t8	2	1
2	1.1	t8	2	1
2	3.1	t8	2	1
3	3.1	t8	2	1
3	3.2	t8	2	1
3	3.3	t8	2	1
3	3.4	t8	2	1
4	3.1	t8	2	1
5	3.1	t8	2	1
6	1.1	t8	2	1
7	3.1	t8	2	1
8	3.1	t8	2	1
9	3.1	t8	2	1
10	3.1	t8	2	1
10	3.2	t8	2	1

1	1.1	t9	2	1
1	3.1	t9	2	1
1	3.2	t9	2	1
2	1.1	t9	2	1
2	3.1	t9	2	1
3	3.1	t9	2	1
3	3.2	t9	2	1
3	3.3	t9	2	1
3	3.4	t9	2	1
4	3.1	t9	2	1
5	3.1	t9	2	1
6	1.1	t9	2	1
7	3.1	t9	2	1
8	3.1	t9	2	1
9	3.1	t9	2	1
10	3.1	t9	2	1
10	3.2	t9	2	1

1	1.1	t10	2	1
1	3.1	t10	2	1
1	3.2	t10	2	1
2	1.1	t10	2	1
2	3.1	t10	2	1
3	3.1	t10	2	1
3	3.2	t10	2	1
3	3.3	t10	2	1
3	3.4	t10	2	1
4	3.1	t10	2	1
5	3.1	t10	2	1
6	1.1	t10	2	1

```
7         3.1       t10       2         1
8         3.1       t10       2         1
9         3.1       t10       2         1
10        3.1       t10       2         1
10        3.2       t10       2         1



;
# Set the  number of resources of each employee (k)- param r is equal to 1 is that (K) is used ; otherwise
0 ( by default)
# index is  V,P,J,G,K,M,1/0
param r :=
1         1.1       t1        PM1       PM1       1         1
2         1.1       t1        PM1       PM1       1         1
6         1.1       t1        PM1       PM1       1         1

1         1.1       t2        QC1       QC1       1         1
2         1.1       t2        QC1       QC1       1         1
6         1.1       t2        QC1       QC1       1         1

1         1.1       t3        OA1       OA1       1         1
2         1.1       t3        OA1       OA1       1         1
6         1.1       t3        OA1       OA1       1         1

1         1.1       t4        QC1       QC1       1         1
2         1.1       t4        QC1       QC1       1         1
6         1.1       t4        QC1       QC1       1         1

1         1.1       t5        PM1       PM1       1         1
2         1.1       t5        PM1       PM1       1         1
6         1.1       t5        PM1       PM1       1         1

1         1.1       t6        OA1       OA1       1         1
1         3.1       t6        OA1       OA1       1         1
1         3.2       t6        OA1       OA1       1         1
2         1.1       t6        OA1       OA1       1         1
2         3.1       t6        OA1       OA1       1         1
3         3.1       t6        OA1       OA1       1         1
3         3.2       t6        OA1       OA1       1         1
3         3.3       t6        OA1       OA1       1         1
3         3.4       t6        OA1       OA1       1         1
4         3.1       t6        OA1       OA1       1         1
5         3.1       t6        OA1       OA1       1         1
6         1.1       t6        OA1       OA1       1         1
7         3.1       t6        OA1       OA1       1         1
8         3.1       t6        OA1       OA1       1         1
9         3.1       t6        OA1       OA1       1         1
```

10	3.1	t6	OA1	OA1	1	1
10	3.2	t6	OA1	OA1	1	1
1	1.1	t7	QC1	QC1	1	1
1	3.1	t7	QC1	QC1	1	1
1	3.2	t7	QC1	QC1	1	1
2	1.1	t7	QC1	QC1	1	1
2	3.1	t7	QC1	QC1	1	1
3	3.1	t7	QC1	QC1	1	1
3	3.2	t7	QC1	QC1	1	1
3	3.3	t7	QC1	QC1	1	1
3	3.4	t7	QC1	QC1	1	1
4	3.1	t7	QC1	QC1	1	1
5	3.1	t7	QC1	QC1	1	1
6	1.1	t7	QC1	QC1	1	1
7	3.1	t7	QC1	QC1	1	1
8	3.1	t7	QC1	QC1	1	1
9	3.1	t7	QC1	QC1	1	1
10	3.1	t7	QC1	QC1	1	1
10	3.2	t7	QC1	QC1	1	1
1	1.1	t8	OA1	OA1	1	1
1	3.1	t8	OA1	OA1	1	1
1	3.2	t8	OA1	OA1	1	1
2	1.1	t8	OA1	OA1	1	1
2	3.1	t8	OA1	OA1	1	1
3	3.1	t8	OA1	OA1	1	1
3	3.2	t8	OA1	OA1	1	1
3	3.3	t8	OA1	OA1	1	1
3	3.4	t8	OA1	OA1	1	1
4	3.1	t8	OA1	OA1	1	1
5	3.1	t8	OA1	OA1	1	1
6	1.1	t8	OA1	OA1	1	1
7	3.1	t8	OA1	OA1	1	1
8	3.1	t8	OA1	OA1	1	1
9	3.1	t8	OA1	OA1	1	1
10	3.1	t8	OA1	OA1	1	1
10	3.2	t8	OA1	OA1	1	1
1	1.1	t9	QC1	QC1	1	1
1	3.1	t9	QC1	QC1	1	1
1	3.2	t9	QC1	QC1	1	1
2	1.1	t9	QC1	QC1	1	1
2	3.1	t9	QC1	QC1	1	1
3	3.1	t9	QC1	QC1	1	1
3	3.2	t9	QC1	QC1	1	1
3	3.3	t9	QC1	QC1	1	1
3	3.4	t9	QC1	QC1	1	1

4	3.1	t9	QC1	QC1	1	1
5	3.1	t9	QC1	QC1	1	1
6	1.1	t9	QC1	QC1	1	1
7	3.1	t9	QC1	QC1	1	1
8	3.1	t9	QC1	QC1	1	1
9	3.1	t9	QC1	QC1	1	1
10	3.1	t9	QC1	QC1	1	1
10	3.2	t9	QC1	QC1	1	1

1	1.1	t10	PM1	PM1	1	1
1	3.1	t10	PM1	PM1	1	1
1	3.2	t10	PM1	PM1	1	1
2	1.1	t10	PM1	PM1	1	1
2	3.1	t10	PM1	PM1	1	1
3	3.1	t10	PM1	PM1	1	1
3	3.2	t10	PM1	PM1	1	1
3	3.3	t10	PM1	PM1	1	1
3	3.4	t10	PM1	PM1	1	1
4	3.1	t10	PM1	PM1	1	1
5	3.1	t10	PM1	PM1	1	1
6	1.1	t10	PM1	PM1	1	1
7	3.1	t10	PM1	PM1	1	1
8	3.1	t10	PM1	PM1	1	1
9	3.1	t10	PM1	PM1	1	1
10	3.1	t10	PM1	PM1	1	1
10	3.2	t10	PM1	PM1	1	1

1	1.1	t10	OA1	OA1	1	1
1	3.1	t10	OA1	OA1	1	1
1	3.2	t10	OA1	OA1	1	1
2	1.1	t10	OA1	OA1	1	1
2	3.1	t10	OA1	OA1	1	1
3	3.1	t10	OA1	OA1	1	1
3	3.2	t10	OA1	OA1	1	1
3	3.3	t10	OA1	OA1	1	1
3	3.4	t10	OA1	OA1	1	1
4	3.1	t10	OA1	OA1	1	1
5	3.1	t10	OA1	OA1	1	1
6	1.1	t10	OA1	OA1	1	1
7	3.1	t10	OA1	OA1	1	1
8	3.1	t10	OA1	OA1	1	1
9	3.1	t10	OA1	OA1	1	1
10	3.1	t10	OA1	OA1	1	1
10	3.2	t10	OA1	OA1	1	1

1	1.1	t1	PM2	PM2	2	1

2	1.1	t1	PM2	PM2	2	1
6	1.1	t1	PM2	PM2	2	1

1	1.1	t2	QC2	QC2	2	1
2	1.1	t2	QC2	QC2	2	1
6	1.1	t2	QC2	QC2	2	1

1	1.1	t3	OA2	OA2	2	1
2	1.1	t3	OA2	OA2	2	1
6	1.1	t3	OA2	OA2	2	1

1	1.1	t4	QC2	QC2	2	1
2	1.1	t4	QC2	QC2	2	1
6	1.1	t4	QC2	QC2	2	1

1	1.1	t5	PM2	PM2	2	1
2	1.1	t5	PM2	PM2	2	1
6	1.1	t5	PM2	PM2	2	1

1	1.1	t6	OA2	OA2	2	1
1	3.1	t6	OA2	OA2	2	1
1	3.2	t6	OA2	OA2	2	1
2	1.1	t6	OA2	OA2	2	1
2	3.1	t6	OA2	OA2	2	1
3	3.1	t6	OA2	OA2	2	1
3	3.2	t6	OA2	OA2	2	1
3	3.3	t6	OA2	OA2	2	1
3	3.4	t6	OA2	OA2	2	1
4	3.1	t6	OA2	OA2	2	1
5	3.1	t6	OA2	OA2	2	1
6	1.1	t6	OA2	OA2	2	1
7	3.1	t6	OA2	OA2	2	1
8	3.1	t6	OA2	OA2	2	1
9	3.1	t6	OA2	OA2	2	1
10	3.1	t6	OA2	OA2	2	1
10	3.2	t6	OA2	OA2	2	1

1	1.1	t7	QC2	QC2	2	1
1	3.1	t7	QC2	QC2	2	1
1	3.2	t7	QC2	QC2	2	1
2	1.1	t7	QC2	QC2	2	1
2	3.1	t7	QC2	QC2	2	1
3	3.1	t7	QC2	QC2	2	1
3	3.2	t7	QC2	QC2	2	1
3	3.3	t7	QC2	QC2	2	1
3	3.4	t7	QC2	QC2	2	1
4	3.1	t7	QC2	QC2	2	1
5	3.1	t7	QC2	QC2	2	1

6	1.1	t7	QC2	QC2	2	1
7	3.1	t7	QC2	QC2	2	1
8	3.1	t7	QC2	QC2	2	1
9	3.1	t7	QC2	QC2	2	1
10	3.1	t7	QC2	QC2	2	1
10	3.2	t7	QC2	QC2	2	1

1	1.1	t8	OA2	OA2	2	1
1	3.1	t8	OA2	OA2	2	1
1	3.2	t8	OA2	OA2	2	1
2	1.1	t8	OA2	OA2	2	1
2	3.1	t8	OA2	OA2	2	1
3	3.1	t8	OA2	OA2	2	1
3	3.2	t8	OA2	OA2	2	1
3	3.3	t8	OA2	OA2	2	1
3	3.4	t8	OA2	OA2	2	1
4	3.1	t8	OA2	OA2	2	1
5	3.1	t8	OA2	OA2	2	1
6	1.1	t8	OA2	OA2	2	1
7	3.1	t8	OA2	OA2	2	1
8	3.1	t8	OA2	OA2	2	1
9	3.1	t8	OA2	OA2	2	1
10	3.1	t8	OA2	OA2	2	1
10	3.2	t8	OA2	OA2	2	1

1	1.1	t9	QC2	QC2	2	1
1	3.1	t9	QC2	QC2	2	1
1	3.2	t9	QC2	QC2	2	1
2	1.1	t9	QC2	QC2	2	1
2	3.1	t9	QC2	QC2	2	1
3	3.1	t9	QC2	QC2	2	1
3	3.2	t9	QC2	QC2	2	1
3	3.3	t9	QC2	QC2	2	1
3	3.4	t9	QC2	QC2	2	1
4	3.1	t9	QC2	QC2	2	1
5	3.1	t9	QC2	QC2	2	1
6	1.1	t9	QC2	QC2	2	1
7	3.1	t9	QC2	QC2	2	1
8	3.1	t9	QC2	QC2	2	1
9	3.1	t9	QC2	QC2	2	1
10	3.1	t9	QC2	QC2	2	1
10	3.2	t9	QC2	QC2	2	1

1	1.1	t10	PM2	PM2	2	1
1	3.1	t10	PM2	PM2	2	1
1	3.2	t10	PM2	PM2	2	1
2	1.1	t10	PM2	PM2	2	1
2	3.1	t10	PM2	PM2	2	1

```
3       3.1     t10     PM2     PM2     2       1
3       3.2     t10     PM2     PM2     2       1
3       3.3     t10     PM2     PM2     2       1
3       3.4     t10     PM2     PM2     2       1
4       3.1     t10     PM2     PM2     2       1
5       3.1     t10     PM2     PM2     2       1
6       1.1     t10     PM2     PM2     2       1
7       3.1     t10     PM2     PM2     2       1
8       3.1     t10     PM2     PM2     2       1
9       3.1     t10     PM2     PM2     2       1
10      3.1     t10     PM2     PM2     2       1
10      3.2     t10     PM2     PM2     2       1

1       1.1     t10     OA2     OA2     2       1
1       3.1     t10     OA2     OA2     2       1
1       3.2     t10     OA2     OA2     2       1
2       1.1     t10     OA2     OA2     2       1
2       3.1     t10     OA2     OA2     2       1
3       3.1     t10     OA2     OA2     2       1
3       3.2     t10     OA2     OA2     2       1
3       3.3     t10     OA2     OA2     2       1
3       3.4     t10     OA2     OA2     2       1
4       3.1     t10     OA2     OA2     2       1
5       3.1     t10     OA2     OA2     2       1
6       1.1     t10     OA2     OA2     2       1
7       3.1     t10     OA2     OA2     2       1
8       3.1     t10     OA2     OA2     2       1
9       3.1     t10     OA2     OA2     2       1
10      3.1     t10     OA2     OA2     2       1
10      3.2     t10     OA2     OA2     2       1;



end;
```