Dual-Level Resource-Constrained Multi-Project Scheduling Framework for Prefabrication in Construction

by

Jing Liu

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

Doctor of Philosophy
in
Construction Engineering and Management

Department of Civil and Environmental Engineering
University of Alberta

ABSTRACT

Construction prefabrication projects, where engineered systems or components of large size and heavy weight are fabricated with limited workplace and storage areas, typically are executed in a multi-project environment. So, frequent inter-project resource transfers are not feasible and should be mitigated. Nonetheless, existing multi-project scheduling approaches give rise to extensive resource links among projects, thereby negatively impacting the stability and feasibility of resultant project schedules and increasing management difficulties in processing each project. Furthermore, the project planning and the workface operation realities are separated from each other in current practice. As a result, it is common that the project planning overestimates the actual construction productivity, while changes and variations (e.g., material logistics) during operations cannot be timely reflected to the project or general managers and present their impacts on the project's and program's schedule and cost.

This research explores a systematic dual-level resource planning framework for addressing these issues identified in conventional resource planning and project scheduling methods for multiple concurrent projects. A dual-level resource-constrained multi-project scheduling framework is proposed to provide achievable resource allocation decisions for program planning and activity scheduling for projects and operations for prefabrication projects in construction. The proposed framework is capable of (1) generating robust resource use plans for multiple concurring projects, (2) interconnecting and synchronizing schedules for various management functions, and (3) analytically evaluating the impact of inherent material logistics uncertainties on individual project schedules and costs. These advantages are illustrated and demonstrated through two literature case

studies and two actual case studies of bridge girder fabrication projects from a partner company in Edmonton, Canada.

The academic contributions of this research are identified as (1) advancement of conventional multi-project scheduling approaches by proposing a generic dual-level scheduling framework, which generates more robust schedules and integrates schedules for various management functions; (2) development of an integrated scheduling optimization model which incorporates material supplies as constraints for resource-constrained project scheduling so as to analyze the impact of material logistics uncertainties on project schedule and cost performances; and (3) the provision and definition of a new indicator (i.e., resource use robustness) for evaluating construction schedule performance.

In terms of practical contributions, the outcomes of this research would (1) provide production managers with reliable and feasible work plans at a fabrication facility, which ensure crew work continuity on individual projects, enhance resource utilization efficiency, and improve communication efficiency among project management teams; (2) create reliable program schedule, project schedule, and operation schedules, which are dynamically interconnected with each other, thereby, facilitating schedule maintenance and updating, saving the efforts for progress report among management personnel, and guiding various management functions such as evaluation of remaining fabrication capacities, prediction of project delivery performances, and execution of daily fabrication work within fabrication facilities; and (3) provide crucial decision support for practitioners to determine allowable time windows of certain critical material deliveries so as to keep the total project cost under pre-set limits and provide alternative plans in coping with disruptions and changes.

PREFACE

This thesis is an original work by Jing Liu. The study, of which this thesis is a part, has received research ethics approval from the University of Alberta Research Ethics Board, "Integrated Resource Planning Optimization Framework for Structural Steel Fabrication Industry," Pro00074621, approved on September 2, 2017. The contents of this thesis are based on three journal papers, which are reorganized for the thesis in order to streamline the logic and pertinent to the theme of this thesis.

A version of Chapter 2 has been formed by extracting the literature review section from three publications: (1) Liu, J., and Lu, M. (2018). "Robust Dual-Level Optimization Framework for Resource-Constrained Multi-Project Scheduling in Prefabrication and Modular Construction", which is under review by *Journal of Computing in Civil Engineering* on January 12, 2018; (2) Liu, J., and Lu, M. "Constraint Programming Approach to Optimizing Project Schedules Under Materials Logistics and Crew Availability Constraints", which has been published in the *Journal of Construction Engineering and Management*; and (3) "Synchronization of Various Management-Function Schedules in a Multi-Project Environment: Case Study on Bridge Girder Fabrication Projects", which is being prepared for publication in the *Journal of Construction Engineering and Management*. For the three papers, Dr. Lu was the supervisory author and was involved with problem definition and manuscript editing.

A version of Chapter 3 has been formed by extracting the Methodology section from two publications: (1) Liu, J., and Lu, M. (2018). "Robust Dual-Level Optimization Framework for Resource-Constrained Multi-Project Scheduling in Prefabrication and Modular Construction", which is under review by *Journal of Computing in Civil Engineering* on January 12, 2018 and (2)

Liu, J., and Lu, M. "Constraint Programming Approach to Optimizing Project Schedules Under Materials Logistics and Crew Availability Constraints", which has been published in the *Journal of Construction Engineering and Management*. For the two papers, Dr. Lu was the supervisory author and was involved with problem definition and manuscript editing.

A version of Chapter 4 has been formed by extracting the Case study section from one publication: Liu, J. and Lu, M. (2018). "Robust Dual-Level Optimization Framework for Resource-Constrained Multi-Project Scheduling in Prefabrication and Modular Construction," which is under review by *Journal of Computing in Civil Engineering* on January 12, 2018. Dr. Lu was the supervisory author and was involved with problem definition and manuscript editing.

A version of Chapter 5 is being prepared for publishing as Liu, J., and Lu, M. "Synchronization of Various Management-Function Schedules in a Multi-Project Environment: Case Study on Bridge Girder Fabrication Projects" *Journal of Construction Engineering and Management*. Dr. Lu was the supervisory author and was involved with problem definition and manuscript editing.

A version of Chapter 6 has been formed by extracting the Case study section from one publication: Liu, J., and Lu, M. "Constraint Programming Approach to Optimizing Project Schedules Under Materials Logistics and Crew Availability Constraints," which has been published in the *Journal of Construction Engineering and Management*. Dr. Lu was the supervisory author and was involved with problem definition and manuscript editing.

ACKNOWLEDGEMENTS

To my supervisor Dr. Ming Lu, for his support, trust, and guidance. Without his vision, help, and encouragement, this research would not have become a reality.

To my committee members: Dr. Aminah Robinson Fayek, Dr. Mohamed Al-Hussein, Dr. Amy Kim, and Dr. Pedram Mousavi for their time, comments, and contribution to this research. Special thanks to my external committee, Dr. Jesus M. de la Garza, for his constructive comments, support, and encouragement. His professional attitude really motivates me in pursing my future academic career.

To my industrial partner: Supreme Group for providing funding support, ideas, and collaborative research opportunities in making this research from thoughts to realities. To Maria Al-Hussein, Todd Collister, Spencer Allan for their valuable suggestions and comments of this research.

To my parents, my older brother Lin, and my lovely little cousin Mumu for their love, understanding, support throughout my life.

Finally, to my beloved husband Wenying for his endless support, accompany, assistance, and unconditional love.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

1 CHAPTER 1: INTRODUCTION

1.1 Problem Statement

The implementation of prefabrication and modular construction continues to gain momentum along with the adoption of advanced technologies for engineering, design, and construction. This trend has transferred significant labor-intensive work from construction sites to off-site fabrication facilities, where various types of customized construction components are prefabricated. In current practice, a typical scenario for prefabrication and modular construction can be described as follows: a specialist structural steel fabricator operates multiple concurrent projects from different clients relying on limited resources in a fabrication facility. Skilled laborers of various specialist trades are allocated to work on the fabrication of structural components for multiple projects from time to time. This leads to an unintended consequence that inter-project resource transfers are unavoidably imposed on these projects. However, extensive resource transfers among multiple projects would give rise to wasted handling efforts, undesired labor work discontinuity, disrupted learning curve effect (Hinze and Olbina 2009; Lee et al. 2015), and entail excessive communication among management teams (Hans et al. 2007). The prefabricated components are usually of large size and heavy weight. For instance, prefabricated bridge girders in the steel fabrication shop of our partner company located in Edmonton, Canada measure up to 45-meter long and weigh as much as 40,000 kg each. In order to move laborers back and forth for processing the fabrication orders from different projects, unfinished products need to be moved in and out of the fabrication shop, causing extra handling work. And the layout of the work area within the fabrication facility often needs to be reconfigured in order to accommodate unique designs on different projects. Also, transferring resources among projects lowers labor productivity. For example, welders need to spend extra time in getting familiar with drawings and specifications on different projects. Besides, project managers need to communicate with one another in order to share resources among projects and accommodate deadlines on different projects, thus significantly increasing the level of difficulty and complexity in project management and making it challenge to provide the right resources at the right time and place for executing the right project. In addition, uncertainties associated with transferring limited resources from project to project can easily disrupt the scheduled work flow. Figure 1.1 illustrates the sharing of the resource R1 on multiple projects over time in a fabrication facility. Disruptions of one project result in the ripple effect on schedules of other projects. For instance, if any delay occurs to Project 2, all subsequent activities of four concurring projects are likely to be impacted, thus causing project delays, budget overruns, and adverse client relationships. In reality, how to cope with multiple concurring projects by shuffling finite available resources in a fabrication facility remains a daunting task.

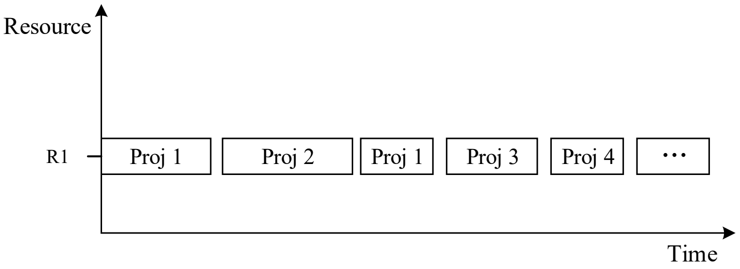


Figure 1.1: Illustration on resource sharing among different projects.

In the construction project management domain, little research has been conducted on the multi-project scheduling problem (Zhou et al. 2013). On the limited attempts, multi-project scheduling problems were first converted into a single-project problem, which can be further tackled by existing single project scheduling optimization methods (Chen and Shahandashti 2009; Siu et al. 2015; Sonmez and Uysal 2014). However, the resulting schedules give rise to extensive resource use dependencies among projects. Previous research on critical path method (CPM) and resource-constrained scheduling have mainly focused on the *intra-project* applications, which can be traced to the seminal work by Kelley Jr and Walker (1959) of formalizing the critical path method to cater for the needs of *intra-project* planning instead of *inter-project* planning. In practice, commercial scheduling software packages featuring multi-project scheduling capabilities (such as MS Project; Primavera P6) extend the use of those *intra-project* scheduling tools to multi-project scheduling applications. Nonetheless, commercial software is short of a multi-project scheduling framework for optimizing resource planning aimed at mitigating inter-project resource use dependencies.

In order to reduce undesired inter-project resource transfer, previous related research resorts to defining the unit cost for transferring a resource among different projects. Resource transfer related costs can be incorporated in the objective function for optimization (Adhau et al. 2013; Krüger and Scholl 2009; Rostami et al. 2017). Nonetheless, for construction projects including prefabrication projects, resource transfer costs not only include costs related to crew transfer and orientation time, but also costs in connection with extra material handling and labor productivity loss (e.g. unlearning curve effect) (Dozzi and AbouRizk 1993; Mastroianni and Abdelhamid 2003). For prefabrication projects in construction, it is not practical (if not impossible) to account for such costs in current job costing systems; hence, quantifying *the unit cost of transferring resource between projects* is not straightforward and demands separate research endeavor. Therefore, the development of a generic resource planning framework, which is capable of reducing inter-project resource transfers in a multi-project environment for the construction industry without the need to explicitly estimate inter-project resource transfer costs, is desired.

For the construction industry, schedules of various details are derived and maintained by different stakeholders for various management functions. In general, a master schedule summarizing project start time, delivery time, and important milestones is effective for contractor-client communication; while schedules with more operation details are intended to guide superintendents in executing daily work in the field. To standardize the representation of schedules for various management functions, CII (2004) endorsed a numeric designator system (i.e., level 0, level 1, level 2, level 3, and level 4-X). Level 0 is a single bar, which spans from start to finish and represents the schedule for a program or a total project. For program schedules, a Level 1 schedule is a combination of Level 0 schedules for each project. As the number increases, the subdivision continues from Level 2 (e.g., by areas or divisions), Level 3 (e.g., for project monitoring and control purpose), to Level 4-X (e.g., 4-9 weeks look-ahead schedules or weekly schedules). In reality, consensus on the numbering system is difficult to reach by different schedulers. To eliminate confusions caused by the numbering system, the descriptive method is also proposed for representing various schedules. By the descriptive method, program/project summary schedule, milestone schedule, project schedule, project control schedule, look ahead schedule, task lists, and supporting data are distinguished (AACE 2010). In a separate but related attempt to formalize the *Last Planner System*, four degrees of planning processes were defined for production planning and control, namely: master scheduling, phase scheduling, lookahead planning, and weekly planning (Hamzeh et al. 2008).

Schedules of finer granularity, such as weekly plans, should be consistent with master schedule and phase schedules in terms of meeting milestones for project delivery (Hamzeh et al. 2008). On the other hand, the setting of milestones in a master schedule or phase schedules needs to be aligned with more detailed schedules. To ensure the successful project delivery, it is advisable that schedules for various management functions can be derived in a “roll-up” or “roll-down” fashion based on a common source of information, as opposed to being developed as separate versions throughout a project life cycle (AACE 2010). Figure 1.2 illustrates the ideal relationship between various schedules.

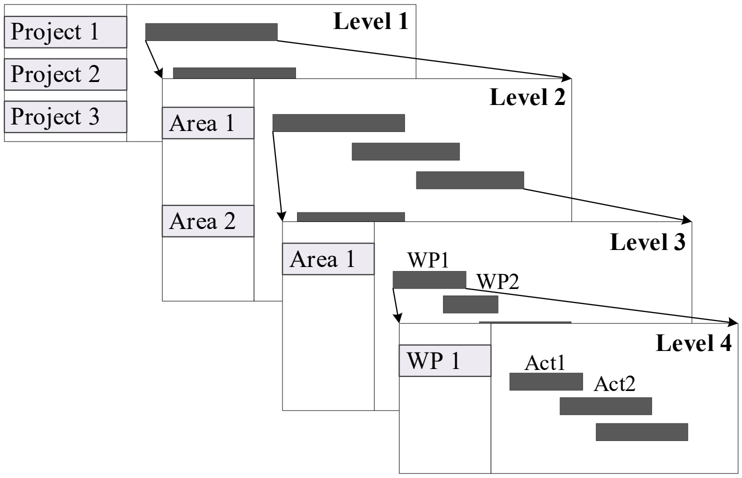


Figure 1.2: Example of the desired relationships between various schedules.

Nevertheless, schedules for various management functions could exist in isolation and the ideal state of schedule synchronization is difficult to materialize in reality. Therefore, the desired resource planning framework still needs to be capable of synchronizing schedules for various management functions.

Prefabrication and modular construction approaches also complicate the material procurement processes (Choi and Song 2014; Safa et al. 2014). Numerous constraints, such as limited production capacity of suppliers, delayed issuance of shipping permits, and unavailability of resources in the material supply chain hamper successful delivery of prefabrication projects. In the construction industry, crew operation planning and material procurement planning are managed separately. This separation also occurs in academic studies in general: project scheduling-related research and supply chain management-related research have seldom been investigated in an integrative fashion (Xu et al. 2016). However, considering the current trend of adopting off-site prefabrication and modular construction technologies on modern construction projects, it is imperative to develop an analytical methodology that factors material supply constraints into the resource-constrained scheduling optimization so as to address the impact of uncertainties in material logistics on project schedule and cost performances.

Therefore, a generic resource-constrained multi-project scheduling framework is needed to improve the resource planning practices in a multi-project environment. The following challenges are expected to be addressed by the developed framework in this study:

- Reducing inter-project resource use dependencies and providing robust resource use plans for various projects in a multi-project environment;
- Interconnecting and synchronizing schedules for various management functions;
- Incorporating the material logistics constraint into the resource-constrained project scheduling problem and analyzing the impact of its uncertainties on individual project schedules and costs.

1.2 Research Objectives

This research aims to test the following two hypotheses:

- The separation of resource planning at a fabrication facility and resource scheduling for an individual project in a multi-project environment leads to enhanced robustness of resulting resource use plans and project schedules through reducing inter-project resource transfers and seamlessly integrating schedules for various management functions. Note *robustness* herein reflects the sufficiency of the problem definition in terms of representing practical constraints in the real world.

- The implementation of the optimization algorithm to derive optimum solutions to real-world problems, which are formulated in the research, leads to improvement on current practices for resource use planning, project scheduling and control in the context of multi-project fabrication in construction.

To prove the hypotheses, a generic multi-project scheduling approach needs to be developed to enhance the resource planning practice for managing concurrent prefabrication projects. This overall objective is achieved by accomplishing the following particular sub-objectives.

- Developing a multi-project scheduling framework which is capable of reducing extensive inter-project resource transfers and synchronizing schedules for various management functions in order to improve the management efficiency and save communication efforts between project management teams.
- Proposing an analytical method to evaluate the impact of uncertainties associated with materials logistics on project schedule and cost performances.
- Developing mathematical programming models for representing problem definitions of resource planning at a fabrication facility and resource scheduling for an individual project; and allowing the implementation of the latest optimization algorithms to derive optimum solutions to real-world problems.

1.3 Research Methodology

To achieve the abovementioned objectives, this research proposes a dual-level multi-project scheduling framework to enhance the resource planning practice of multiple concurrent prefabrication projects. The proposed framework was developed based on a dual-level planning structure (Can and Ulusoy 2014; Speranza and Vercellis 1993; Yang and Sum 1993). In this structure, resource allocation and activity scheduling are separated into two levels: resources are first allocated to projects, and then activities within individual projects are scheduled subject to allocated resources availability. As a result, allocated resources are focused on one project during a certain time period, thus reducing the necessity for information sharing and potential coordination between different project managers and eliminating the management difficulty (Arora and Sachdeva 1989). Also, decisions at both levels are interconnected: at the higher level, decisions are influenced by an approximate evaluation of the future effects on the lower level; on the other hand, once higher-level decisions have been taken, they influence future decisions through the constraints incorporated into the lower level (Can and Ulusoy 2014; Speranza and Vercellis 1993; Yang and Sum 1993).

The research methodology of this thesis is shown in Figure 1.3. The literature on previous related research is first reviewed to identify gaps between academic research and practical demands in multi-project scheduling. A generic multi-project scheduling framework is proposed to address the identified gaps. One core of the proposed framework is the aggregation model that aggregates detailed activities in one project into several non-interruptible macro-activities for allocating resources, thus reducing undesired resource transfers among projects and linking schedules for various management functions. Another two key components of the proposed framework are two scheduling models: one is for allocating resources to multiple concurrent projects at the upper level, which effectively resolves the resource conflicts among various projects; the other one is for scheduling individual projects at the lower level, which provides a simulation engine for analyzing different material delivery settings to reveal its impact on project schedule and cost performances. Detailed explanation and description of the proposed framework are referred to Chapter 3. Three advantages of the proposed framework are illustrated and demonstrated in Chapter 4, Chapter 5, and Chapter 6 by conducting case studies.

Constraint programming, a methodology for solving combinatorial optimization problems, is adopted to perform optimization at both levels of the entire framework. Constraint programming integrates multiple techniques from various domains such as operation research, graph theory, and artificial intelligence, thus enhancing computational performances in terms of the solution quality and computation time (Rossi et al. 2006). It takes advantages of the techniques of constraint propagation and systematical search methods in order to find an optimal solution in an effective and efficient way (Haralick and Elliott 1980; Kumar 1992). Constraint propagation, the key idea of constraint programming, removes inconsistent variable values from the problem domain by actively using constraints. As a result, the search space is reduced before searching, thus saving computational effort (Baptiste et al. 2012). After the search space reduction, a wide range of powerful search strategies including widely-adopted and customized search algorithms (e.g., genetic algorithms, branch and bound algorithm, and tabu search) can be applied to find solutions (Baptiste et al. 2012; Hentenryck 1989). This research utilizes *IBM ILOG CPLEX Optimization Studio* for implementing constraint programming.

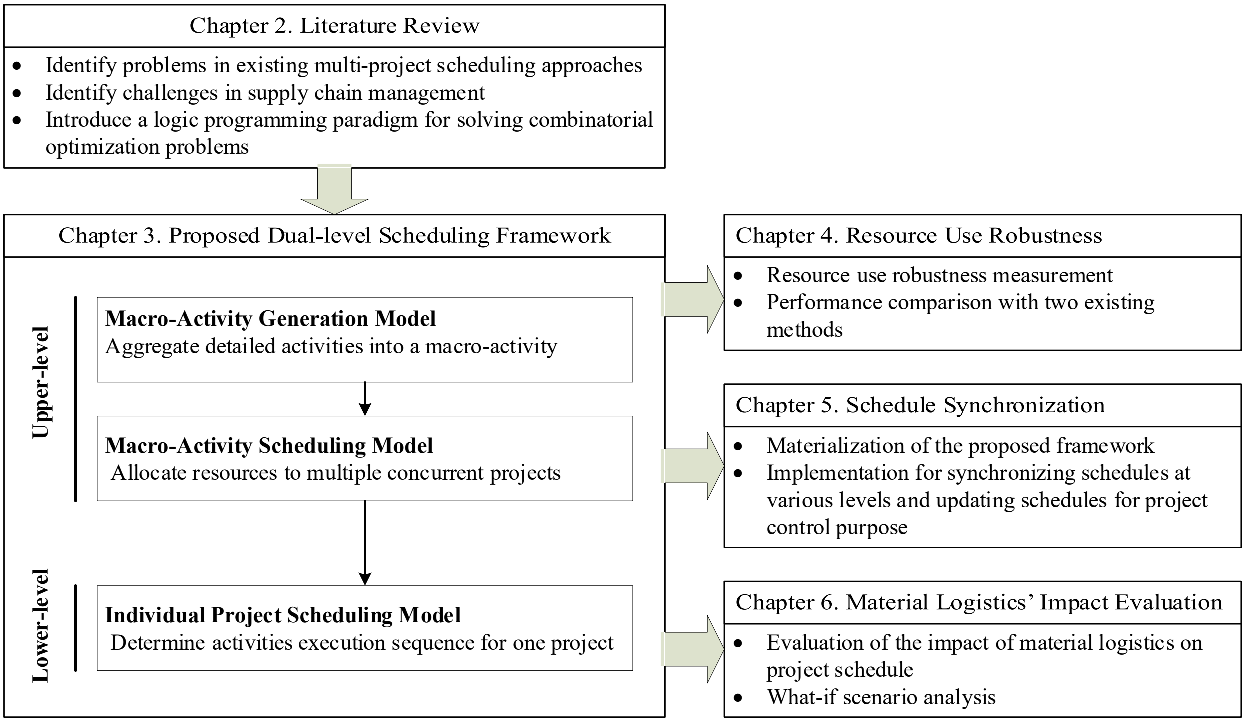


Figure 1.3: Research Methodology.

1.4 Thesis Organization

This thesis consists of seven chapters. Except for the first and last chapters, all other chapters are composed based on three journal papers produced during the author's doctoral study. To stay pertinent to the theme of the thesis, contents in three papers are reorganized into five chapters. In detail, Chapter 2 reviews previous research related to the thesis work and identifies research gaps. Chapter 3 introduces and describes the development of the dual-level multi-project scheduling framework for construction prefabrication projects. For Chapter 4, 5, and 6, each devotes to a case study with a focus on a particular practical need in the real world for illustrating and demonstrating advantages of the proposed framework in scheduling multiple concurrent prefabrication projects. Detailed contents of each chapter are listed as follows.

Chapter 2: Literature review

- Identifies two major limitations in existing multi-project scheduling approaches, namely (1) resources are frequently transferred among various projects, thereby, reducing the schedule robustness and (2) schedules for various management functions are isolated, thus leading to non-trackable project progress and delayed project delivery.
- Identifies challenges in supply chain management for prefabrication construction projects, which justify the need of incorporating material supply constraints into the resource-constrained scheduling problem. Nonetheless, this has been overlooked in previous research on resource-constrained scheduling.
- Introduces a logic programming paradigm for solving combinatorial optimization problems (e.g., scheduling problems and vehicle routing problems).

Chapter 3: Dual-level optimization framework for resource-constrained multi-project scheduling

- Proposes an analytical method for aggregating multiple activities within one project into a macro-activity.
- Develops a scheduling optimization model for allocating resources to macro-activities of multiple projects subject to limited resources at the upper level.
- Develops a scheduling optimization model for scheduling activities within one macro-activity subject to allocated resources and other constraints (e.g., precedence constraints, material logistics constraints, and deadline constraints) at the lower level.

Chapter 4: Resource use robustness of a fabrication schedule

- Defines a quantitative indicator to measure the resource use robustness of a multi-project schedule.
- Performs a case study to illustrate the applicability and feasibility of the proposed framework for mitigating inter-project resource use dependencies and enhancing schedule resource use robustness.
- Contrasts the proposed framework with two existing multi-project scheduling methodologies in order to demonstrate the advantages of the proposed framework in generating robust resource use plans for scheduling multiple concurring projects.

Chapter 5: Synchronization of various management-function schedules in a multi-project environment: case study on bridge girder fabrication projects

- Materializes the previously developed dual-level multi-project scheduling framework for scheduling multiple concurrent bridge girder fabrication projects.
- Conducts a practical case study to illustrate the applicability of the generic framework in linking and synchronizing schedules for various management functions.
- Demonstrates the schedule update process based on actual production progress for project control purpose.

Chapter 6: Optimizing project schedules under materials logistics and crew availability constraints

- Proposes a two-step analytical approach to evaluate the impact of uncertainties associated with material supplies upon the total project cost, namely: (1) developing a resource-constrained scheduling optimization model to incorporate material logistics constraints; (2)

analyzing various material delivery scenarios based on the mathematical model to characterize the impact of material supply on the project cost.

- Utilizes the developed scheduling optimization model for the lower level of the proposed framework as a simulation engine for evaluating the impact of different input settings about material logistics on the project cost through a literature case study.
- Conducts a real case study of a bridge girder fabrication project to demonstrate the applicability of the developed scheduling optimization model on solving large project networks.

Chapter 7: A summary of the research contributions, limitations, and future work.

2 CHAPTER 2: LITERATURE REVIEW

The work conducted in this thesis ultimately achieves three objectives: (1) to develop a multi-project scheduling framework for enhancing the resource use robustness of derived schedules in a multi-project environment, (2) to propose a scheduling framework which can provide a basis in linking schedules of various management functions in a multi-project environment, and (3) to provide an analytical methodology for evaluating the impact of dynamic material logistics on project schedule and cost performance. These objectives are motivated by the gaps between the needs of the current practice and previous related research. In this chapter, related studies on the following areas: multi-project scheduling (Section 2.1), supply chain management (Section 2.2), resource-constrained scheduling (Section 2.3), and constraint programming (Section 2.4) are reviewed.

2.1 Multi-Project Scheduling

Businesses in manufacturing and construction companies are generally conducted in a multi-project environment. Lova et al. (2000) conducted a survey of companies in the areas of construction, textile, computers and information systems, and public administrations; they concluded that 84% of the companies worked with multiple concurring projects, which is consistent with the findings by Payne (1995) that 90% of projects by value was executed in a multi-project environment. Although projects are managed independently, they compete for limited resources with each other, thus rendering resource allocation in a multi-project environment to be a critical decision process (Laslo and Goldberg 2008).

In previous research on project management, the multi-project scheduling problem was solved in two approaches, namely: multi-project approach and single-project approach (Lova et al. 2000). For the multi-project approach, various priority rules are usually adopted for allocating resources to multiple projects. Heuristic rules require less computational efforts, perform well for large-size projects, and can be easily integrated into commercial project management tools (Kolisch 1996). Commonly implemented heuristic rules include minimum slack rule (MINSLK), minimum latest finish time rule (MINLFT), shortest activity from shortest project first (SASP), and maximum total work content (MAXTWK) (Bock and Patterson 1990; Browning and Yassine 2010; Fendley 1968; Kim et al. 2005; Lova et al. 2000). The performance of heuristic rules is problem-dependent. The performances of different priority rules were tested in various research teams (Browning and Yassine 2010; Lova and Tormos 2001), but there is no consensus on which rule works best so far (Browning and Yassine 2010; Cohen et al. 2004). For the single-project approach, the projects are artificially linked to form a large project network with two dummy activities, namely “start” and “finish,” as shown in Figure 2.1. Then, single-project scheduling methods are employed to schedule the newly formed super-project network (Cheng et al. 2015; Sonmez and Uysal 2014). The single-project method leads to excessive negative multitasking and the presence of too many open projects. The impact of multitasking on procurement and construction supply chain has been studied by several scholars (Arbulu et al. 2002; Elfving and Tommelein 2003).

In the construction project management domain, the single-project approach is commonly adopted for solving the multi-project scheduling problem. For instance, Sonmez and Uysal (2014) proposed a hybrid algorithm integrating the backward-forward scheduling method, generic algorithms, and simulated annealing for solving resource constrained multi-project scheduling problem. Cheng et al. (2015) developed a discrete symbiotic organism search algorithm for optimizing resource leveling in the multi-project scheduling problem. Some endeavors have been made in solving the multi-project scheduling problem based on heuristic rules. For instance, Taghaddos et al. (2011) addressed the multi-project scheduling problem based on heuristic rules and developed a simulation-based auction protocol (SBAP) to solve the crane allocation problem for assembling multiple prefabricated modules in industrial construction. According to the proposed protocol, a bidding price was calculated for each agent at the start of each simulation cycle, serving as a priority index in assigning resources; then, an optimization algorithm was applied to solve the resource assignment problem aiming to minimize the total bidding price.

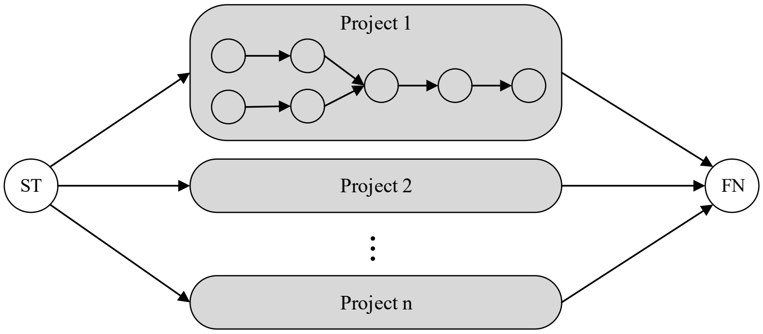


Figure 2.1: Super-project network for the single-project method.

The basic unit for requesting resources in both multi-project approach and single-project approach is the activity defined within individual projects. As a result, resources are frequently transferred among different projects, potentially disrupting the "learning curve" effect, requiring extra efforts in handling and setting up, and entailing additional communication and coordination in project execution.

In order to reduce undesired inter-project resource transfer, previous related research resorts to defining the unit cost for transferring a resource among different projects. Resource transfer related costs can be incorporated in the objective function for optimization. Krüger and Scholl (2009) incorporated resource transfer time (namely, the physical transfer time between two locations plus equipment/labor resources setup time) into a mathematical model, resulting in a priority rule-based framework to identify an optimal tradeoff between total duration and resource transfer frequency across projects. Adhau et al. (2013) employed a multi-agent system for solving the decentralized resource-constrained multi-project scheduling problem by considering resource transfer time and related cost. Rostami et al. (2017) extended the decentralized multi-project scheduling problem to determine the optimal location of the site of maintenance facilities on pipeline construction projects, aimed at minimizing project completion time and maintenance facility construction cost. It is noteworthy resource transfer times and related costs are prerequisite to formulate the above multi-project schedule optimization models.

Nonetheless, for construction projects including prefabrication projects, resource transfer costs not only include costs related to crew transfer and orientation time, but also costs in connection with extra material handling and labor productivity loss (e.g. unlearning curve effect) (Dozzi and AbouRizk 1993; Mastroianni and Abdelhamid 2003). In short, how to prepare reliable input models can present one major hurdle to applying multi-project schedule optimization models in construction as reviewed above. For prefabrication projects in construction, it is not practical (if not impossible) to account for such costs in current job costing systems; hence, quantifying *the unit cost of transferring resource between projects* is not straightforward and demands separate research endeavor.

In current practice, project schedules are initialized and maintained by individual projects to track milestones of a project in a multi-project environment (Elonen and Artto 2003). Production schedules in the fabrication facility are generally derived by a facility manager based on experience or the urgency of different activities (Yaghootkar and Gil 2012). Frequent resource transfers among multiple projects are ordinary circumstances in a multi-project environment. Also, due to the inadequate planning, schedules of various management functions remain outdated and isolated with one another. As a result, over-commitment is commonplace for most construction companies. It was reported that almost every company bids more projects than what they can handle (Elonen and Artto 2003; Payne 1995; Spuhler and Biagini 1990). For managing individual projects, there are always gaps between long-term planning and short-term planning so that it is hard to timely track project progress and identify potential delays (Hamzeh et al. 2012; Hamzeh et al. 2008). Therefore, a systematic multi-project scheduling framework is desired, which is capable of (1) integrally assessing resource capacities and allocating finite resources to multiple concurrent prefabrication projects while mitigating frequent resource transfers among projects and (2) providing a basis to link and synchronize schedules for various management functions.

2.2 Supply Chain Management in Construction

Supply chain is a concept that was originated and flourished in the manufacturing industry. However, due to particular characteristics of the construction industry, the supply chain management for the construction industry differs substantially from that for the manufacturing industry. The construction supply chain is “a network among different stakeholders within a construction project (clients, contractors, suppliers, etc.) that work together in a concerted effort to manage the flow of information, materials, services, products, and cash flow” (Xue et al. 2007). Each construction project is one-off and engages different suppliers, contractors, and clients. Thus, it is difficult for stakeholders in the construction industry to maintain the long-term, collaborative, and mutually beneficial relationships as the stakeholders in the manufacturing industry do (O'Brien et al. 2008).

The supply chain in construction directs the movement of all the required materials (bulk materials or engineered materials) to one construction site. Any hiccup along the supply chain would result in a negative ripple effect on ongoing and ensuing crew operations, thereby disrupting project schedules and increasing construction costs. Improper construction material management would give rise to extra inventory costs due to early deliveries or idle labor costs due to late deliveries (Said and El-Rayes 2010). Research in lean construction has aimed to streamline the material supply chain by removing all the non-value adding processes and material wastes (Arbulu and Ballard 2004). For instance, numerous studies focused on implementing the just-in-time (JIT) delivery method for delivering ready-mix concrete or rebars in the construction industry (Polat et al. 2007; Said and El-Rayes 2013). Tserng et al. (2006) proposed a decision-support model to minimize the integrated inventory cost for the steel rebar supply chain. Irizarry et al. (2013) integrated building information modeling (BIM) and geographic information systems (GIS) to provide a visualization system for monitoring statuses in the construction supply chain and minimizing the transportation cost. It is noteworthy that these studies separated supply chain management from construction crew operation schedules based on the assumption that project demand is known in terms of time and quantity and will not be adjusted subject to risks inherent in materials supply chain or logistics.

However, the broad variability in construction productivity makes it difficult to accurately predict site demand (Walsh et al. 2004). And decisions on material supply chain management and crew work planning are seamlessly interrelated in practice (Xu et al. 2016): if materials are unavailable, the construction schedules will be adjusted to minimize idling or underutilization of crew resources and meet the project due date. It leads to a changed material demand pattern on the project in comparison with the original plan. On modular construction projects, shipping large volumes of engineered materials is associated with high risks and demands elaborate planning and coordination among project stakeholders (Gosling et al. 2016; Liu et al. 2016). For instance, road use regulations are generally strictly imposed, such as limitations on the maximum height, width, and allowable travel times. Especially when engineered materials are fabricated offshore, variable factors in shipping and transportation of engineered materials (e.g., changing marine environments like waves, storms) can easily prolong the shipping time and disrupt the original construction plan (Liu et al. 2016). Recognizing the difficulty in adopting a JIT delivery method, the contractor requests the delivery of engineered materials at the earliest opportunity in order to mitigate the risks associated with late deliveries. Therefore, the integration of engineered material supply as explicit constraints is justifiable in deriving construction schedules for construction projects.

2.3 Resource-Constrained Scheduling

For scheduling a construction project, previous research has addressed resource-constrained scheduling problems (RCSP) with a particular focus on improving labor productivity, which is “a measure of the overall effectiveness of an operating system in utilizing labor, equipment, and capital to convert labor efforts into useful output (Hendrickson and Au 1989).” Examples include a two-stage genetic algorithms model to solve RCSP under the constraint of limited laborers (Chen and Weng 2009); schedule optimization with the constraint of multi-skilled crews (Arashpour et al. 2015; Liu and Wang 2012); solving RCSP with three general types of renewable resources which are recovered after finishing one activity such as labors and equipment (Zhang et al. 2006);

determining the leanest resource supply and shortest project duration with the constraint of limited craft personnel on industrial maintenance projects (Siu et al. 2015).

It is noteworthy material logistics related constraints were rarely considered explicitly in defining resource-constrained scheduling problems, as on traditional civil construction projects skilled laborers and equipment are generally deemed as the driving resources with finite limits and impose high risks on project time and cost performances. Limited attempts in considering effects of material logistics on construction schedules assume a static material supply: namely, a fixed material quantity is available from the beginning of the project (Li and Zhang 2013). Considering the high uncertainties associated with the delivery of engineered materials, an analytical model, which takes material logistics related constraints into consideration, is needed to address the impact of uncertainties inherent in materials logistics on the project schedule.

2.4 Constraint Programming

Constraint programming (CP) is a methodology for solving combinatorial optimization problems by representing them as *constraint satisfaction problems* (Baptiste and Le Pape 1995). It takes advantages of the techniques of constraint propagation and systematical search methods in order to find an optimal solution in an effective and efficient way (Haralick and Elliott 1980; Kumar 1992). CP takes advantage of the constraint propagation technique to actively cross-check constraints and discard inconsistent values of certain variables from the problem domain, thus reducing the search space and saving computational effort (Baptiste et al. 2012). Therefore, CP is a naturally appropriate methodology for solving tightly constrained problems such as scheduling, allocation, transportation, and rostering (Simonis 1996). Complementary search algorithms such as forward checking, backtracking, and maintaining arc consistency are implemented to explore possible solutions in an efficient fashion (Fromherz 2001; Rossi et al. 2006).

It has been proved that the CP approach produces better quality solutions than most of the simulated annealing methods, tabu search and all of the genetic algorithms in solving job shop scheduling problems (Baptiste and Le Pape 1995; Nuijten and Aarts 1996). Also, CP is often integrated with different optimization algorithms to improve the computation performance in terms of both computation time and solution quality. For instance, CP has been integrated with genetic algorithms (GA) to accelerate the evolution process of GA (Chiu and Hsu 2005; Nee et al. 2014). Shaw (1998) combined CP with local search for the vehicle routing problem and showed this hybrid approach was competitive against well-developed operation research (OR) metaheuristic methods, even when CP was in its infancy. Hooker et al. (1999) integrated linear programming (LP) and CP to take advantage of both the constraint propagation through CP and relaxations through LP for finding feasible and optimal solutions in a short time. Furthermore, Constraint programming has the flexibility in modeling different types of constraints, especially the ability to quickly model logical and sequential constraints. The constraints can be easily added, deleted, or modified to meet user requirements without rebuilding the model (Brailsford et al. 1999).

In regard to solving construction scheduling problems, the application of CP has been proved to be both effective and efficient. For instance, Liu and Wang (2012) used CP to enhance the computation efficiency of scheduling linear construction projects with multi-skilled crews; Tang et al. (2014) utilized the CP technique to solve a multi-objective optimization scheduling model for schedule control purpose in a railway construction project. Menesi et al. (2013) demonstrated the capability of CP to handle the time-cost trade-off problem for large-scale projects which involve thousands of activities. Therefore, CP is selected in the current research for solving the defined problem. This study utilized the optimization platform of *IBM ILOG CPLEX Optimization Studio,* which incorporates search techniques like large neighborhood search (LNS) and GA, for solving the developed scheduling optimization model.

This chapter reviews previous research on multi-project scheduling and resource-constrained scheduling. Three main limitations are identified, namely: (1) extensive inter-project resource transfers, (2) isolated schedules for various management functions, and (3) ignoring material logistics impact on fabrication schedules. In the following chapter, a dual-level multi-project scheduling framework is proposed for prefabrication in construction in order to address identified problems. Detailed mathematical formulations are also developed at both levels.

3 CHAPTER 3: DUAL-LEVEL OPTIMIZATION FRAMEWORK FOR RESOURCE-CONSTRAINED MULTI-PROJECT SCHEDULING

3.1 Introduction

Dual-level management structure is commonly adopted for managing multiple concurrent projects (Yang and Sum 1993). In the dual-level project management framework, basic scheduling units for requesting resources are different at the two levels. At the upper level, a project is the basic scheduling unit for allocating resources and determining milestones; while at the lower level, activities within the individual project are the basic unit for requesting resources subject to resources assigned and due dates determined from the upper level. Multi-project scheduling problems have also been addressed based on the dual-level management structure. For instances, Yang and Sum (1993) proposed a hierarchical two-stage decomposition model for multi-project scheduling to illustrate the interrelations among resource allocation, detailed activity scheduling, and cash flows. Can and Ulusoy (2014) developed a two-stage decomposition approach and utilized genetic algorithms to solve the multi-project scheduling problem.

The dual-level planning scheme reflects the organizational structure and the decision process in the industry (Speranza and Vercellis 1993); thus, it can be acceptable to practitioners and ready for implementation. The upper level represents the portfolio-level management which is to maximize the portfolio value by balancing different projects. "*Project portfolio is a group of projects that share and compete for the same resources and are carried out under the sponsorship or management of an organization* (Martinsuo and Lehtonen 2007)." When making decisions at the portfolio-level, project managers are expected to provide inputs regarding individual projects for setting priorities and allocating resources to different projects (Platje et al. 1994). The lower level is related to the single project management decision, by which a detailed activity schedule must be derived for each project achieving an appropriate trade-off between time, cost, and resource use (Rushton et al. 2006). Portfolio-level decisions ought to be enacted at the single-project level (Cooper et al. 1997; Cooper et al. 2000).

In line with the dual-level planning strategy, the proposed framework in this chapter tackles the multi-project scheduling problem in two phases, namely: multi-project resource allocation (i.e., upper level) and individual project scheduling (i.e., lower level). This is also in line with decision making processes in the project planning stage for a typical prefabricator in construction (such as structural steel fabricator). For the proposed framework, this process is achieved by the generation of *macro-activity*, which is the basic unit for requesting resources at the upper level. The duration and resource profile of a macro-activity are derived from the lower level optimization analysis of each project. Outcomes from the upper level analysis (i.e., start time, finish time, and allocated resources for each project) are incorporated as constraints for scheduling activities within individual project at the lower level. The multi-project resource allocation and individual project scheduling processes will be explained in detail in the ensuing section.

3.2 Dual-Level Multi-Project Scheduling Optimization Framework

The proposed dual-level framework, as depicted in Figure 3.1, is based on the dual-level management structure. At the upper level, the total available resources (e.g., labors, equipment, and space) in a fabrication facility are organized in a central resource pool. The different types of resources need to be allocated to resource pools specified to each project for a certain time period by a central resource pool manager. In doing so, the manager needs to evaluate resource requirements of each project in cooperation with each project manager at the lower level. In the following subsection "*Upper Level*," detailed explanation is presented to show how information from the lower level is utilized to derive solutions at the upper level including project milestones (i.e., project start time and finish time) and resources allocated over time for each project. At the lower level, results from the upper-level analysis (i.e., project start time, finish time, and allocated resources) become imposed constraints for further project scheduling. Essentially, the lower level optimization deals with a resource-constrained single-project scheduling problem, and the lower-level project scheduling optimization is subject to the project start time, finish time, and resources allocated over time determined in the upper-level analysis. The modeling processes of the dual-level multi-project scheduling framework are elaborated step by step as follows.

3.2.1 Upper Level

At the upper level, detailed activities in each project are first aggregated into *macro-activities* by utilizing the outcome of the lower level optimization analysis. The prefix "*macro-*" is used to distinguish the aggregated activity (i.e., a project or sub-project) from the specific activity defined within one project (Speranza and Vercellis 1993). The purpose of defining macro-activities is to remove undesired resource transfers among projects. Resources are allocated and utilized on each macro-activity in a continuous fashion. For instance, on one macro-activity in a fabrication facility, before finishing all the work on the current macro-activity, resources allocated to this macro-activity are not allowed to be transferred to other macro-activities. In practice, the macro-activity can be one project from one particular client or a collection of fabrication jobs from one project sharing similar design features and delivery milestones. Macro-activities of various granularities would lead to schedules of different cost and resource robustness.

Similar to the multi-mode resource-constrained project scheduling problem (MRCPSP), a macro-activity can have multiple alternative construction modes (i.e., macro modes) with different combinations of duration and resource requirements. In MRCPSP, activities can be executed by selecting one feasible mode characterized by pre-determined resource requirements and activity duration (Talbot 1982). Figure 3.2 gives an example of three macro modes for a macro-activity. The *x*-axis represents time, while the *y*-axis indicates the required number of resources. The function depicts the time-dependent resource use over time for a particular macro mode.

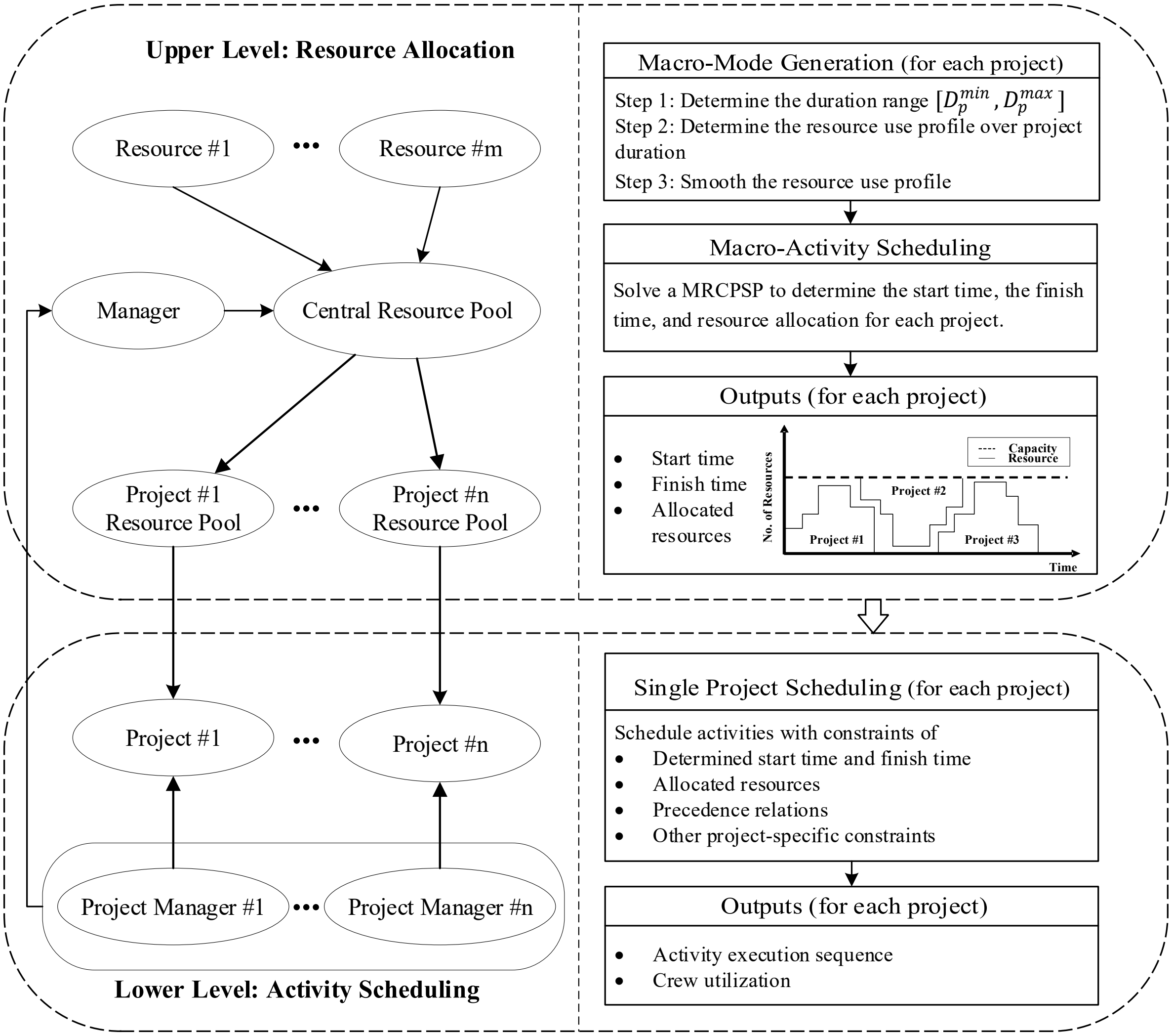


Figure 3.1: Dual-level multi-project scheduling optimization framework.

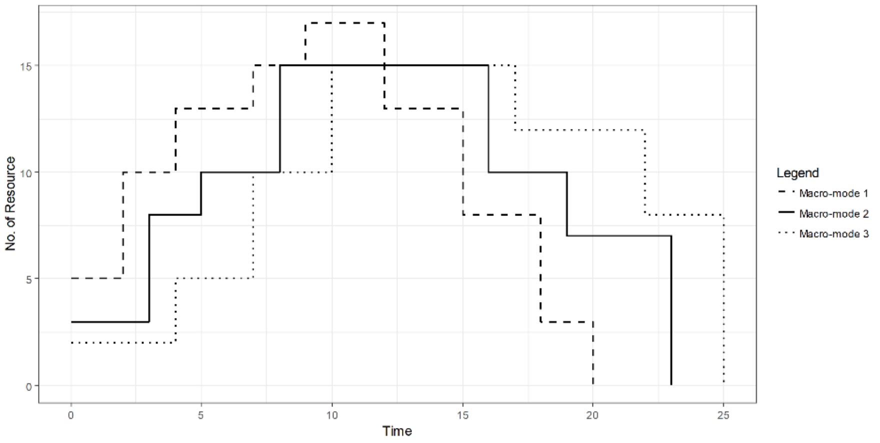


Figure 3.2: Three alternative macro modes for a macro-activity.

In order to illustrate the proposed framework, one project is treated as a unit to generate macro-activity in this chapter, which means all activities of each project are aggregated into a macro-activity. Research has been conducted on how to derive the aggregated activity for representing a project by optimizing a project schedule under the limited resource availability constraint (Can and Ulusoy 2014; Hans 2001; Neumann and Schwindt 1998; Speranza and Vercellis 1993). The algorithm proposed by Can and Ulusoy (2014) serves as the basis for generating macro-activities in the proposed framework. The detailed steps in the macro-activity generation and scheduling are illustrated as follows.

3.2.1.1 Macro-Activity Generation

In this step, multiple macro modes for a macro-activity are determined by assuming each activity has multiple alternative construction options. In the modularization and prefabrication settings, fabricators occasionally reconfigure the layout in a certain work area and modify resource use on the shop floor in an attempt to adjust the progress rate on a certain activity in processing a particular job. For instance, in the fabrication shop of our partner company located in Edmonton, Canada;

two journeymen are usually assigned to attach stiffeners to bridge girders in the bridge girder fabrication project. But one or two more journeymen are sometimes added to speed up this activity. For actual projects, historical data or operation personnel's experience can be utilized to derive alternative construction options for an activity. So, in order to cater for possible application needs identified in the real world, multi-mode is assumed to provide the flexibility in defining alternative resource use requirements on macro-activities. However, it is worth mentioning that multi-mode is not a prerequisite for generating macro-activities, but to lend certain flexibility and enhance the generalizability of the proposed framework. The detailed steps in generating macro modes for a macro-activity are presented as follows.

Step 1. Determine the duration range for each project.

The minimum duration D_p^{min} for project p is determined by scheduling this project using the fastest construction option for all activities, while the maximum duration D_p^{max} is determined by scheduling this project using the slowest construction option for all activities. The duration range for project p is $[D_p^{min}, D_p^{max}]$. In this step, the decision variables are the execution sequence and start time of each activity. Constraints include the precedence constraint and resource limit constraint. The project scheduling problem is to minimize total project duration through Constraint programming (CP).

CP is an optimization technique for solving combinatorial optimization problems by defining the objective function and representing the relations among decision variables as constraints (Baptiste and Le Pape 1995). CP takes advantage of constraint propagation and systematical search methods in order to find an optimal solution in an effective and efficient way (Haralick and Elliott 1980; Kumar 1992). Detailed theoretical fundamentals and applications are referred to Rossi et al.

(2006). *IBM ILOG CPLEX Optimization Studio 12.6.3* is utilized as the software platform for implementing CP in this research.

Step 2. Determine the resource use profile over project duration.

The resource use profile for each project is determined by solving a series of optimization models with varying deadline constraints ranging from D_p^{min} to D_p^{max}. The deadline is increased by one unit at each iteration to form the successive optimization model. In total, $D_p^{max} - D_p^{min} + 1$ resource use profiles are generated for each project. The detailed process as elaborated in the following has been implemented in *IBM ILOG CPLEX Optimization Studio 12.6.3* as part of the proposed CP based solution algorithm.

Process for determining resource use profile over the duration

```
For $deadline = D_p^{min}$ to $D_p^{max}$ by 1
    Generate one macro mode by solving the following optimization
    problem:
    Objectives:
    Minimize 1) the project cost and 2) the peak resource demand
    Constraints:
    1) all precedence relationships among activities are satisfied
    2) the number of resources in use is no greater than the total available
    resources
    3) the project duration is no greater than the $deadline$

    If (the macro mode is an inefficient one) then discard it
    Else save it as a new macro mode for this project
End for
```

For a particular optimization model, decision variables are start time and selected construction options for each activity; constraints include precedence relationships, resource limit, and project duration limit. The objective function is to minimize (1) total project cost and (2) resource use fluctuation over project duration (similar to the classic resource leveling analysis), with total project cost taking a higher priority. As such, broad fluctuations (peaks and valleys) in the resulting resource use profile are mitigated to a large degree, without compromising the goal of minimizing total cost. In addition, mitigating resource use fluctuation is also conducive to confining non-productive resource time resulting from further smoothing in the ensuing Step 3 (i.e., further fine-tuning resource use profile), as elaborated in a later section.

Worth mentioning is that the resource leveling problem has been studied extensively in project management research in arriving at a desired resource use profile, which has a gradual ascent of the number of resources to a peak followed by a gradual descent till the end of a project (El-Rayes and Jun 2009; Mattila and Abraham 1998). A number of metrics have been defined to measure the resource fluctuations, such as the sum of squares (Hegazy 1999; Son and Skibniewski 1999), release and re-hire index (El-Rayes and Jun 2009), and peak resource demand (Menesi and Hegazy 2014). In this research, the peak resource demand is used as the measure to minimize the fluctuation in the resource use profile. The function "*staticLex*" in *IBM ILOG CPLEX Optimization Studio 12.6.3* is employed to perform the multi-objective optimization. This function is able to prioritize predefined objectives in performing optimization analysis in such a way that the gain on the primary objective outweighs the loss on other objectives. For instance, the expression *staticLex(cost, fluctuation)* indicates lowering project cost is more important than avoiding resource use fluctuation. The optimizer first identifies one or more feasible solutions with the minimum project cost and then evaluates the peak resource demand of these solutions. The solution with the minimum peak resource demand is selected as the final solution. In the case of generating macro modes on a certain project, inefficient modes are eliminated while efficient ones are kept for further analysis. An inefficient mode is identified given another mode exists, of which both duration and resource cost are superior to the current one.

Step 3. Smooth the resource use profile.

Although a fabrication facility keeps a constant crew size, a gradual buildup and a gradual decline in the resource use profile minimize the chances of moving underutilized resources back and forth among projects in a multi-project environment. In this step, the leveled resource use profile obtained in Step 2 is smoothed to generate the desired shape of the resource use profile.

Figure 3.3 illustrates the process of smoothing a resource use profile. If there is more than one peak in the resource use profile generated in Step 2, the resource requirements between the first peak and the last peak should be set to the peak value. In Figure 3.3, time “t2” and “t3” are the first and last peak respectively. The two valleys between “t2” and “t3” are filled as the peak resource demand to avoid multiple peaks in the resource requirement curve. As shown in Figure 3.3, how to smooth the resource use profile before the first peak and after the last peak is different. In order to smooth the resource use profile before the first peak, it should start from the project beginning to the time of the first peak and fill the valleys by their preceding resource requirements. So the resource requirements between “t1” and “t2” are set as the resource requirements at time “t1-1”. For smoothing the resource use profile after the last peak, it should start from the project end time and move backward to the time of the last peak. The valleys in between are filled by their succeeding resource requirements. The valley from “t4” to “t5” is filled as the resource requirements at time “t5+1”.

As shown in Figure 3.3, limited resources, which are kept but idling over a certain time period (as highlighted in shaded patterns), provide the necessary “cushion” to the lower level project scheduling against unexpected disruptions and hence enhance the robustness of the generated plan.

The detailed pseudocode is illustrated as follows, which has been coded into a function by using the C# programming language in *Visual Studio 2015*.

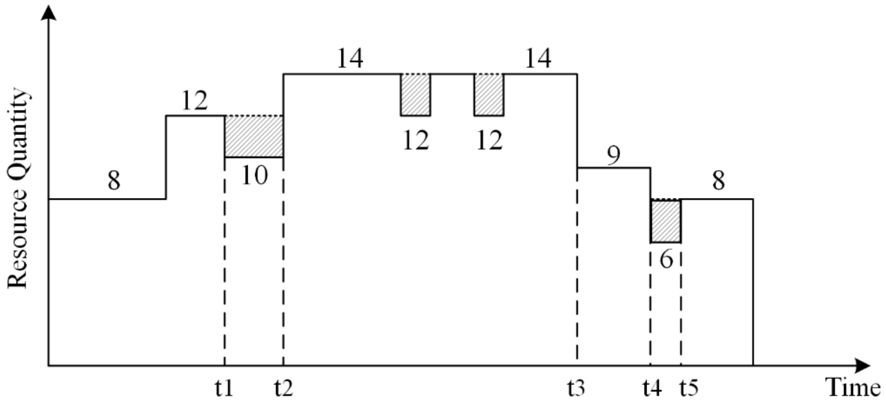


Figure 3.3: Illustration on smoothing a resource use profile.

Pseudocode for smoothing a resource use profile

```
1:  find a list $maxIndex$ of the time moment of the maximum resource use over the
    project duration
2:  Smooth the resource use profile between the first peak and the last peak if the
    original resource use profile has more than one peak
    If ($length(maxIndex) > 1$) then
        set the resource requirement for all time moments between the first index
        $maxIndex[1]$ and the last index $maxIndex[length(maxIndex)]$ in the
        list $maxIndex$ to the peak resource requirement
    else
        go to 3
    End if
3:  Smooth the resource use profile before the first peak
    For $j = 2$ to $maxIndex[1]$ by 1
        If the resource requirement at the $j$ time moment is less than the resource
        requirement at the $j - 1$ time moment, then set the resource requirement at
        the $j$ time moment equal to the resource requirement at the $j - 1$ time
        moment
    End for
4:  Smooth the resource use profile after the last peak
    For $j = Duration$ to $maxIndex[length(maxIndex)]$ by -1
```

```
        If the resource requirement at the $j$ time moment is greater than the resource requirement at the $j-1$ time moment, then set the resource requirement at the $j-1$ time moment equal to the resource requirement at the $j$ time moment
    End for
```

It is notable that macro modes for one particular macro-activity (i.e., one project) are generated based on solving the resource-constrained scheduling problem (which is a typical MRCPSP) for each project. It is a well-studied topic and various researchers have contributed to analytical solutions (Menesi et al. 2013; Sprecher and Drexl 1998; Zapata et al. 2008). As such, detailed mathematical models for the macro mode generation are not included in this chapter.

3.2.1.2 Macro-Activity Scheduling

Once macro modes for all projects are derived, macro-activities, each corresponding to a particular project, are combined into a multi-mode resource-constrained project scheduling model in order to minimize the total cost subject to total available resources. The macro-activity scheduling is a special instance of MRCPSP. In a majority of previous research on MRCPSP, the resource requirements of each activity remain constant over its duration. However, in this research, the resource requirements of macro-activities are time-dependent (as depicted by the resource use profile in Figure 3.2) instead of staying constant over the time. The problem formulation for the macro-activity scheduling is presented as follows.

$$min \text{ costs} = IC_{fab} + \sum_{p \in P} \left(C_{pv} + IC_p \right) \tag{3.1}$$

$s.t.$:

$$\sum_{p \in P} \sum_{v \in \widetilde{M}_p} \sum_{\theta = \max(0, t - \tilde{d}_{pv} + 1)}^{\min(T,t)} \omega_{pvr(t-\theta+1)} \tilde{x}_{pv\theta} \leq W_r, \quad r \in R, t \in T \tag{3.2}$$

$$\sum_{v \in \widetilde{M}_p} \sum_{t=0}^{T} \tilde{x}_{pvt} = 1, \quad p \in P \tag{3.3}$$

Variables to be determined

$\tilde{x}_{pvt} \in \{0, 1\}, p \in P, v \in \widetilde{M}_p, and \ \ t \in T$

The objective of the optimization model is to minimize total cost of all projects (i.e., Eq. (3.1)). P is the set of all projects. IC_{fab} is the indirect cost in the fabrication facility, which depends on the total duration for completing all projects. C_{pv} is the cost of macro mode v for project p. IC_p is the indirect cost of project p, which is proportional to the project duration. Constraint Eq. (3.2) is the resource availability constraint that the requirements of resource r at any time t should be no greater than the total available amount W_r. R is the set of all types of resources. $\widetilde{M}_p$ is the set of macro modes for project p. T is the total duration, which is the maximum finish time of all projects and equals to $\max_{p,v}\left(\left(t + \tilde{d}_{pv}\right)\tilde{x}_{pvt}\right)$, where t is the period index; $\tilde{x}_{pvt}$ is a binary variable that equals to 1 if project p is started at period t using macro mode v, otherwise is 0; $\tilde{d}_{pv}$ is the duration of macro mode v for project p. ω_{pvrt}, which varies with time for each macro mode, represents the required amount of resource r by project p in macro mode v at time t. Constraint Eq. (3.3) ensures that only one macro mode alternative is selected for each project.

The optimization model has been encoded in *IBM ILOG CPLEX Optimization Studio 12.6.3.* In order to model the time-dependent resource requirements ω_{pvrt}, each macro mode is broken into a number of sub-macro modes, which have constant resource requirements over their duration. In the meantime, these sub-macro modes must be continuously executed without any interruptions if the corresponding macro mode is selected. This is achieved through the hierarchical models in

IBM ILOG CPLEX CP Optimizer (CP Optimizer User's Manual 2015). After finishing the scheduling of all macro-activities, the start time, finish time, and resources allocated over time for each project are decided at the end of upper-level analysis and imposed as constraints for the lower-level scheduling.

3.2.2 Lower Level

Together with precedence relationships between activities, the number of resources allocated, start time and finish time–determined in the upper-level scheduling optimization–present constraints for the lower-level scheduling. Apart from these constraints, more project-specific constraints can be considered to schedule each project at the lower level, such as material logistics and readiness of design drawings, which were identified as common delay factors on structural steel fabrication projects. On actual projects, those project-specific constraints should be first identified. And then, each activity in the current project is scheduled subject to constraints obtained from both the upper-level (i.e., allocated resources, start time, and finish time) and the lower-level project-specific constraints (including precedence relationships and other dynamic constraints), while minimizing the total project cost or duration. In this study, material logistics is taken for an example to illustrate the project scheduling on the lower level. Specifically, one project is scheduled under material logistics and resource availability constraints. With resource supply pattern depicted in a time-dependent leveled resource use profile resulting from the upper-level resource planning analysis, the lower-level scheduling problem is essentially one special type of MRCPSP.

At the lower level, the following information is considered to derive the project schedule: (1) material inventory and material supply in terms of quantity, type, and time; (2) technical precedence relationships between activities; (3) resource availability in terms of type and quantity;

(4) the project deadline; (5) material demand of each activity in terms of quantity and type; and (6) resource demand of each activity in terms of different trades of labor and different types of equipment. This section formulates the defined problem in a mathematical programming model, ready for applying constraint programming to generate the optimal schedule in terms of the lowest crew cost plus material inventory cost.

In this section, the budgeted worker-hours (WH) and material idling time are considered in defining the objective function. After setting the unit rates for budgeted WH ($/WH) and material idling time ($/Unit-Hr) respectively, the two objectives are combined as a total cost function, expressed as Eq. (3.4).

$$\begin{aligned} \min(TC) &= \min(C_{WH} + C_{Idlem}) \\ &= \min(sum\ of\ [(time) \times (crew\ size)] \\ &\qquad + sum\ of\ [(unit\ cost) \times (inventory\ quantity) \\ &\qquad \times (idling\ time)]); \end{aligned} \tag{3.4}$$

where the first term C_{WH} is the crew cost factoring in crew delay time due to material shortage on site. In general, frequent layoff and rehiring of skilled trades working in a fabrication facility or a construction site are detrimental to productivity and morale of the deployed crews. Thus, a fabrication facility (e.g., piping spool or structural steel fabrication shops) maintains a stable size of the workforce over a certain timeframe to process the work orders from multiple clients. But the crew size allocated to a particular project may vary with time as illustrated in the upper level. So the crew cost equals to the summation of crew cost of each type over the project duration. The second item C_{Idlem} is the material inventory cost in connection with material idling time, which is calculated as the summation of the idling cost for each type of material. The idling cost of each type of material is to multiply the unit idling cost by the idle material quantity by the idling time.

Time-cost tradeoff is about selecting the normal or crash modes on activities in order to meet the deadline and achieve the lowest total project cost (i.e., total direct plus total indirect) (Chen and Weng 2009). Activity modes are feasible options of direct cost and time duration to execute particular activities. In this research, activity modes are represented in terms of the number of labor resources required and the duration of certain activities. Feasible options for relevant activities are specified as constraints in the mathematical formulation of the optimization problem. Thus, uncertainties associated with time-cost tradeoff (i.e., the selection of proper activity options leading up to the lowest total project cost) are addressed through the optimization analysis. An optimal selection of activity modes constitutes part of the optimization results. The problem formulation is given as follows.

Objective function:

$$\begin{aligned}\min(TC) &= \min(C_{WH} + C_{Idlem}) \\ &= \min\left(\sum_{r\in R}\sum_{t=0}^{T} u_r \times W_{r,t}\right. \\ &\left. + \sum_{n\in N} u_n \times \left(I_{0n} \times T + \sum_{t=0}^{T}(T-t)S_{tn} - \sum_{\theta=0}^{T}(T-\theta)\sum_{i\in V} q_{in} \sum_{j\in M_i} x_{ij\theta}\right)\right)\end{aligned} \tag{3.5}$$

As explained in Eq.(3.4), the objective function is to minimize the total cost accounting for labor cost C_{WH} and materials inventory cost C_{Idlem}. Eq. (3.5) shows the objective function in a mathematical way. T is the total project duration, which is the maximum finish time of all activities $\max_{i,j}\left((t + d_{ij})x_{ijt}\right)$ within this project, where t is period index; x_{ijt} is a binary variable that equals to 1 if activity i starts in period t using mode j, otherwise equals to 0; d_{ij} is the duration for activity i using mode j. u_r is the cost of using one laborer resource r per time unit, $W_{r,t}$ is the total available amount of laborer resource r at time t, and R is the set of laborer resources. So $\sum_{r \in R} \sum_{t=0}^{T} u_r \times W_{r,t}$ is the total laborer cost, which corresponds to the first item in Eq.(3.4). I_{0n} is the inventory amount of engineered material n at the start time ($t = 0$) of the project, u_n is the unit cost for storing one idling engineered material n per time unit. $u_n \times I_{0n} \times T$ represents the inventory cost of engineered material n if they are kept until the project's finish time. S_{tn} is the supply amount of engineered material n at time t, $u_n \times \sum_{t=0}^{T} (T - t) S_{tn}$ means the total storage cost of all supplies for engineered material n from the supply time to the project finish time. q_{in} is the required amount of engineered material n by activity i. As explained, $x_{ij\theta}$ is 1 only when activity i starts in period θ using mode j, otherwise is 0. The item $q_{in} \sum_{j \in M_i} x_{ij\theta}$ represents the material demand of activity i at time θ. $u_n \times \sum_{\theta=0}^{T} (T - \theta) \sum_{i \in V} q_{in} \sum_{j \in M_i} x_{ij\theta}$ is the total storage cost for engineered material n from the consumed time θ to the end of this project T, which does not actually happen and should be deducted from the item $u_n \times I_{0n} \times T \; + \; u_n \times \sum_{t=0}^{T} (T - t) S_{tn}$ in order to calculate the actual storage cost for engineered material n. V is the set of activities of the project. N is the set of engineered materials. And then sum up the storage cost for all types of engineered materials to get the materials inventory total cost C_{Idlem}.

$$\text{Precedence Constraint:} \quad \sum_{j \in M_k} \sum_{t=0}^{T} t x_{kjt} \geq \sum_{j \in M_i} \sum_{t=0}^{T} (t + d_{ij}) x_{ijt}, \qquad (i,k) \in P \qquad (3.6)$$

The constraint in Eq. (3.6) defines precedence relationships among various activities on the project. P is the set of precedence relationships between all activities of the project. $(i,k) \in P$ means activity i is one predecessor of activity k. The start time of activity k should be later than the finish time of activity i. $\sum_{j\in M_i}\sum_{t=0}^{T}\left(t+d_{ij}\right)x_{ijt}$ is the finish time of activity i, while $\sum_{j\in M_k}\sum_{t=0}^{T} t x_{kjt}$ is the start time of activity k. M_i and M_k are the set of crew methods of activity i and activity k.

Resource Constraint: $$\sum_{i\in V}\sum_{j\in M_i}\sum_{\theta=\max(0,t-d_{ij}+1)}^{\min(T,t)} w_{ijr}x_{ij\theta} \le W_{r,t}, \quad r\in R, t\in T \tag{3.7}$$

The constraint in Eq. (3.7) sets the limits on available laborer resources. w_{ijr} is the required amount of laborer resource r by activity i in mode j. $\sum_{j\in M_i}\sum_{\theta=\max(0,t-d_{ij}+1)}^{\min(T,t)} w_{ijr}x_{ij\theta}$ represents the laborer resource demand of activity i throughout its duration. The total requirements of laborer resource r at any time t during the project duration should be within the limit $W_{r,t}$.

Material Constraint: $$\sum_{i\in V} q_{in}\sum_{j\in M_i}\sum_{\theta=0}^{\mathrm{t}} x_{ij\theta} \le I_{0n} + \sum_{\theta=0}^{\mathrm{t}} S_{\theta n}, \quad n\in N, t\in T \tag{3.8}$$

The constraint in Eq. (3.8) sets the constraint of the material supply on the project demand at a particular time. $q_{in}\sum_{j\in M_i}\sum_{\theta=0}^{\mathrm{t}} x_{ij\theta}$ is the cumulative material demand of activity i. The accumulated consumption of engineered material n at any time t during the project duration should be within the total available amount which includes the initial inventory I_{0n} and all supplies $\sum_{\theta=0}^{\mathrm{t}} S_{\theta n}$ up to the time t.

Deadline Constraint: $$\max_{i,j}\left(\left(t+d_{ij}\right)x_{ijt}\right) \le deadline, \quad i\in V, j\in M_i \tag{3.9}$$

The constraint in Eq.(3.9) sets the limits on the project finish time. $deadline$ is the predefined deadline for this project. As explained, $\max_{i,j}\left((t+d_{ij})x_{ijt}\right)$ is the maximum finish time of all activities in this project.

Crew Method Constraint:

$$\sum_{j\in M_i}\sum_{t=0}^{T} x_{ijt} = 1, \quad i \in V \tag{3.10}$$

The constraint in Eq. (3.10) ensures that only one mode is selected for each activity. The optimization model was encoded in *IBM ILOG CPLEX Optimization Studio*.

Up to this point, resources allocated to individual projects are further assigned to detailed activities within one project. The validity of the proposed upper-level and lower-level schedule optimization models will be demonstrated through literature and practical case studies in the following chapters.

3.3 Conclusion

Previous research on multi-project scheduling problems either cannot remove the undesired resource use dependency among projects or demand accurate resource transfer costs as inputs to mitigate resource transfers across projects. To improve the multi-project resource planning practice at a prefabrication facility, a robust dual-level resource-constrained multi-project scheduling framework is proposed to allocate limited fabrication resources among multiple concurring projects in an off-site prefabrication facility; while in the meantime, mitigating frequent inter-project resource transfers and eliminating the need for quantifying the unit cost of transferring resources among projects. The proposed framework separates resource allocation to projects and activity scheduling within each project in two interconnected levels for optimization analyses. With the proposed dual-level scheduling framework, the use of finite resources is focused on one project at a time; as such, crew work continuity on individual projects is ensured and resource utilization efficiency is enhanced. During the project execution stage, the individual project manager would have full control over the allocated resources over a certain period in coping with disruptive factors (e.g., material delivery delays) on each particular project, while confining the propagation of any schedule delay to other projects. This is conducive to productivity improvement and project control. The framework can also link and synchronize schedules for various management functions so that the project planning and the workface operation realities are better aligned with each other, thus enhancing the success of delivering multiple projects. Additionally, the proposed optimization model of the lower level provides an analytical engine for simulating the impact of uncertainties in material logistics on project schedule and cost performance.

In the following chapters, the applicability and feasibility of the proposed framework in improving the resource robustness of generated schedules (Chapter 4), synchronizing schedules for various management functions (Chapter 5), and simulating the impact of different material delivery settings on the project schedule and cost performance (Chapter 6) will be demonstrated.

4 CHAPTER 4: RESOURCE USE ROBUSTNESS OF A FABRICATION SCHEDULE

4.1 Introduction

To a certain extent, a robust schedule is insusceptible to delays on non-critical activity durations caused by unexpected factors (such as resource shortage), while remaining relevant; when the schedule update is indeed required, only fine-tuning is performed instead of a major overhaul (Herroelen and Leus 2004; Zheng et al. 2013). Robust schedules are always desired in a multi-project environment where multiple stakeholders are involved under contractual obligations (Herroelen and Leus 2003). In previous research, two types of schedule robustness have been studied: solution robustness and quality robustness. *Solution robustness* means that activity start times are insensitive to changes in activity durations, while *quality robustness* indicates that the schedule performance (such as project duration and cost) is insusceptible to disruptions (Van de Vonder et al. 2005).

The solution robustness is usually measured by the weighted summation of the absolute difference between planned start times and actual start times on all activities (Leus and Herroelen 2004), as shown in Eq.(4.1). N is the total number of activities, S_i^A is the actual start time of activity i, S_i^P is the planned start time of activity i, and w_i is the weight denoting the activity disruption cost per time unit.

$$Solution\ Robustness = \sum_{i \in N} w_i \left| S_i^A - S_i^P \right| \tag{4.1}$$

A quality robust schedule means the objective value does not worsen significantly in response to schedule disruptions. The typical objective values include project duration, project cost, and project early completion bonuses or delay penalties. Commonly adopted measures of quality robustness include: (1) the expected value of the objective function such as the expected duration (Stork 2001); and (2) the probability of the solution value being no greater than a predefined threshold (e.g. the probability that the project can be finished by the deadline) (Van de Vonder et al. 2005), as shown in Eq.(4.2).

$$Quality\ Robustness = P(F_n \leq \delta) \tag{4.2}$$

where F_n is the finish time of the last activity and δ is the predefined deadline. For more measures on the solution robustness and quality robustness, the readers are referred to Van de Vonder et al. (2006).

In regard to construction scheduling, research has been conducted on attaining solution robustness (Lambrechts et al. 2008; Lgelmund and Radermacher 1983; Stork 2001), quality robustness (Fu et al. 2010; Lau et al. 2007; Zheng et al. 2013) or a trade-off between the two (Van de Vonder et al. 2006). However, robustness related to resource use has yet to be formally investigated.

Fondahl (1991), Lu and Li (2003), and Kim and de la Garza (2003) pointed out that resource based precedence relationships (i.e., resource links) play a significant part in critical path scheduling logic, as resource links specify a resource's workflow from one activity to another and would potentially disrupt activity sequencing and cause time delays. Previous research focused on inserting buffers to absorb the potential disruptions in resource flows (Herroelen and Leus 2004; Lambrechts et al. 2008; Van de Vonder et al. 2006). When it comes to a multi-project environment, the robustness of a schedule related to resource use is more significant. Inter-project resource links can quickly propagate disruptions on one project to others, adding risks and increasing difficulties in project planning and control (e.g., information sharing and work coordination among project teams). Therefore, it is justifiable to formalize an appropriate measure of resource use robustness intended to quantify the resource use dependency among multiple projects.

In this chapter, a “resource use robustness” indicator is defined to characterizes resource use dependencies among multiple concurring projects. A case study is conducted to illustrate and demonstrate the applicability and feasibility of the proposed framework in generating robust resource use plans for scheduling multiple concurring projects. Two existing multi-project scheduling methodologies (i.e., a single-project scheduling approach and an open-source multi-project scheduling platform) are employed for comparison purpose. The defined resource use robustness indicator is utilized to gauge quantitatively inter-project resource use dependencies of derived multi-project schedules.

4.2 Resource Use Robustness Measure

Extensive resource use dependencies among projects propagate delays and disruptions from one project to another. For measuring resource use dependencies among projects, a straightforward resource use robustness indicator is defined as shown in Eq. (4.3).

$$Resource\ Use\ Robustness = 1 - \frac{Resource\ Transfer\ Times}{Maximum\ Resource\ Tansfer\ Times} \tag{4.3}$$

The variable $Resource\ Transfer\ Times$ means the times of resources are transferred among multiple projects over the total duration. It is calculated as follows: when the number of resources working on one project at the current time unit decreases in comparison against the previous time unit, while at the same time, the number of resources working on any other project increases, one time of inter-project resource transfer is counted. It should be pointed out that when the required quantities of resources simultaneously increase or decrease on all the relevant projects over a certain time period, reallocation of resources to different projects are not counted as inter-project resource transfer. For instance, when scheduling two concurrent projects, Project 1 requires five more units of resources at time t than that at time $t\text{-}1$, while Project 2 also demands four more units of resources. As such, nine resource units–which have completed previous projects and remain idle– will be reassigned to the two ongoing projects but will not be counted in "resource transfer times". The denominator $Maximum\ Resource\ Tansfer\ Times$ is the maximum resource transfer times during the total time duration, which equals $Total\ Duration - 1$. The variable $Total\ Duration$ is the overall duration to complete all the relevant projects. So the resource use robustness indicator implies the percentage of scheduled time units, which is not associated with resource transfers among multiple projects over the total duration. A higher value of the resource use robustness indicator means less frequent resource transfers among projects and thus a more robust schedule.

4.3 Case Study

Two concurrent project networks consisting of twenty-five activities (i.e., Project 1) and thirty activities (i.e., Project 2) respectively are adapted from Abido and Elazouni (2010). They were previously used to illustrate the trade-off among three conflicting objectives in project scheduling analysis, i.e., the total duration of all projects, financing costs, and the required credit. Later, El-Abbasy et al. (2017) improved the multi-objective scheduling optimization problem by adding two more objectives (i.e., profit and resource fluctuations) and adapted this case study to validate the proposed model. The project networks for the two projects are shown in Figure 4.1(a) and Figure 4.1(b) respectively. The alternative construction options for each activity are listed in Table 4.1

for Project 1 and Table 4.2 for Project 2. In practice, alternative construction options can be derived from the progress tracking data in the lower level if available or from experience of production personnel. The maximum resource supply limit is 18 per day. The project indirect cost is $840/day for Project 1 and $1306/day for Project 2, respectively. To calculate the indirect cost associated with the fabrication facility, the daily rate is assumed as $3000/day to cover the operation cost (e.g., equipment) in the fabrication facility.

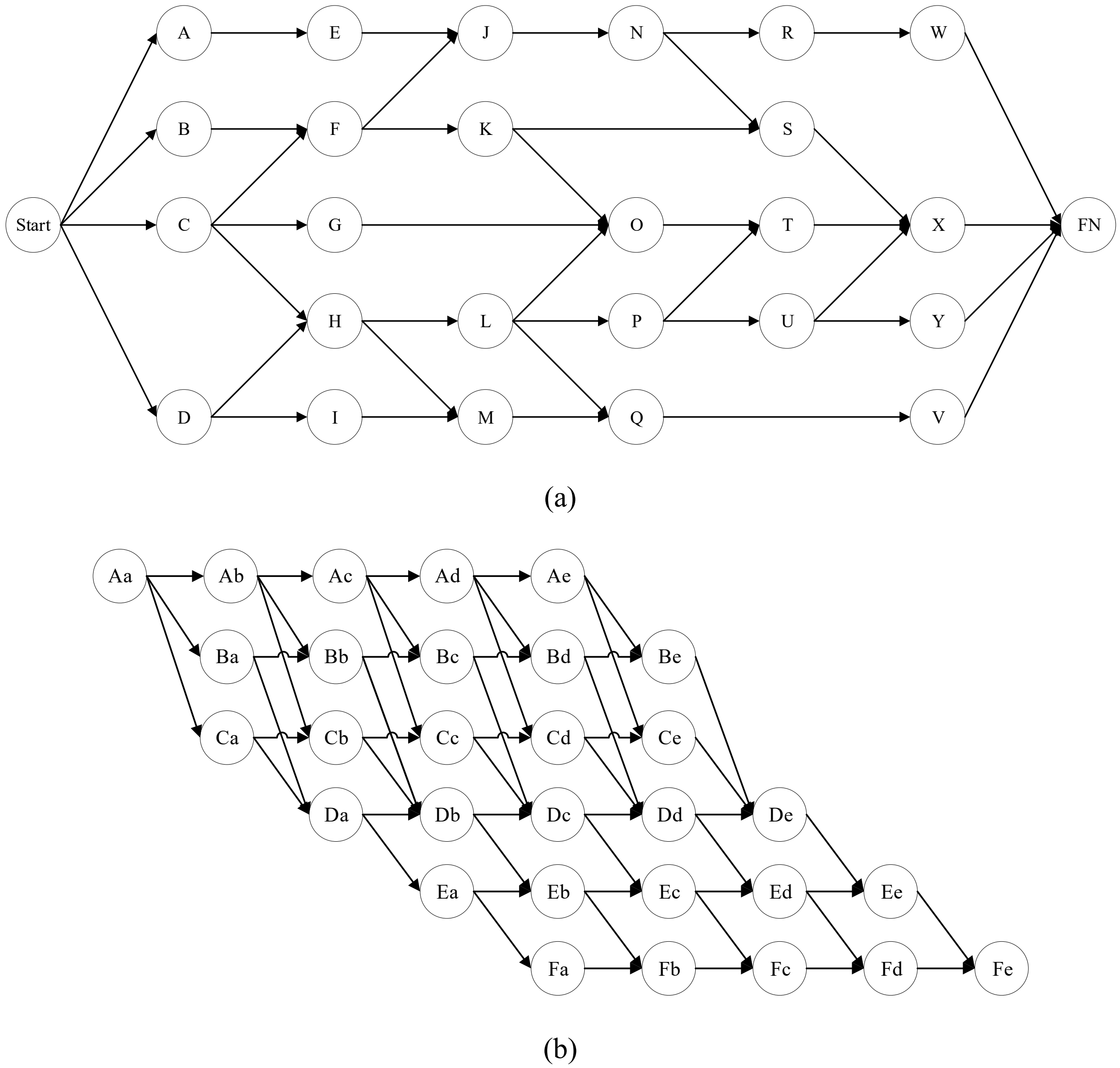


Figure 4.1: Networks for the two projects: (a) Project 1 network, (b) Project 2 network.

Table 4.1: Alternative construction options for project 1.

Activity	Option 1			Option 2		
	Duration	Daily Direct Cost	Daily Resource	Duration	Daily Direct Cost	Daily Resource
A	2	1000	3	1	2200	4
B	3	1200	2	2	1950	3
C	2	1100	1	1	2500	2
D	3	900	4	2	1500	5
E	3	1250	2	2	1950	3
F	3	1150	2	2	1850	3
G	2	1050	5	1	2400	6
H	3	950	2	1	3200	4
I	2	650	5	1	1500	6
J	5	450	1	3	1250	3
K	5	350	1	4	500	2
L	5	500	2	4	700	3
M	1	1450	5	-	-	-
N	5	400	5	3	700	7
O	5	550	5	4	750	6
P	4	500	4	3	750	5

Q	3	1350	2	2	2200	3
R	5	600	5	4	800	6
S	5	850	4	4	1200	5
T	6	700	3	4	1200	5
U	4	1200	2	3	1700	3
V	3	1850	1	2	2900	2
W	5	650	4	4	900	5
X	5	600	2	4	800	3
Y	2	100	1	1	2100	2

Table 4.2: Alternative construction options for project 2.

Activity	Option 1			Option 2		
	Duration	Daily Direct Cost	Daily Resource	Duration	Daily Direct Cost	Daily Resource
Aa : Ae	1	1700	3	-	-	-
Ba : Be	2	1500	2	1	3500	4
Ca : Ce	3	1800	4	2	3000	5
Da : De	4	1900	1	3	3600	2
Ea : Ee	3	1600	3	2	2500	4
Fa : Fe	2	2000	2	1	4200	4

4.3.1 Resource Allocation to Multiple Projects Utilizing the Proposed Framework

The four steps presented in Section 3.2.1 are followed in this case study for allocating resources to multiple concurrent projects. The macro modes for each project are first derived and then scheduled. The detailed programming code is presented in Appendix A.

Step 1. Macro-Activity Generation: determine the duration range for each project

The minimum duration for the two projects is determined by scheduling each project using Option 2, while the maximum duration for the two projects is determined by scheduling each project using Option 1. By scheduling two projects, the duration range for Project 1 is from 20 days to 27 days and the duration range for Project 2 is from 21 days to 29 days.

Step 2. Macro-Activity Generation: determine the resource use profile over project duration.

In this step, it is to determine the resource use profile over the duration. With the upper limit on duration as 20 days, the specified precedence relationships in Figure 4.1(a), the alternative construction options in Table 1, and predefined resource limit (i.e., 18 resources per day), the resource use with the least cost for Project 1 is derived as shown in Figure 4.2 (dashed line). Similarly, more resource use profile can be determined by setting the upper limit on project duration from 21 days to 27 days for Project 1 and from 21 days to 29 days for Project 2.

Step 3. Macro-Activity Generation: smooth the resource use profile

The resource use profile is smoothed to have a gradual ascent to a peak and then a gradual decent shape in this step. By following the proposed methodology to smooth the resource use profile, the first step is to find the list of the maximum resource use on each project day, which includes Day

[9, 10, 12, 13, 14, 15]. The second step is to smooth the resource use between the first peak Day 9 and the last peak Day 15. Therefore, the resource use at Day 10 is smoothed to the maximum resource use (i.e., 18 units). The second step is to smooth the resource use before the first peak Day 9. It starts from Day 2 and increases the scheduled time day by day until Day 9. If the resource use of the current day is less than that of the day before, then the resource use is set to that of the day before. For instance, the resource use on Day 4 is 11 units, which is less than the resource use (i.e., 15 units) on Day 3. So the resource use on Day 4 is adjusted to 15 units. Similar procedures are applied to other days before Day 9. The last step is to smooth the resource use profile after the last peak Day 15. It starts from the last day (i.e., Day 20) of the project and then decreases the scheduled time day by day until Day 15. If the resource use of the current day is greater than that of the day before, then the resource use of the day before is set to that of the current day. The final smoothed resource use profile is the solid line shown in Figure 4.2. Following a similar way, other macro modes can be determined for the two projects. In total, eight macro modes and nine macro modes are generated for Project 1 and Project 2. The corresponding duration and cost are listed in Table 4.3. The smoothed resource use profile for each macro mode is shown in Figure 4.3.

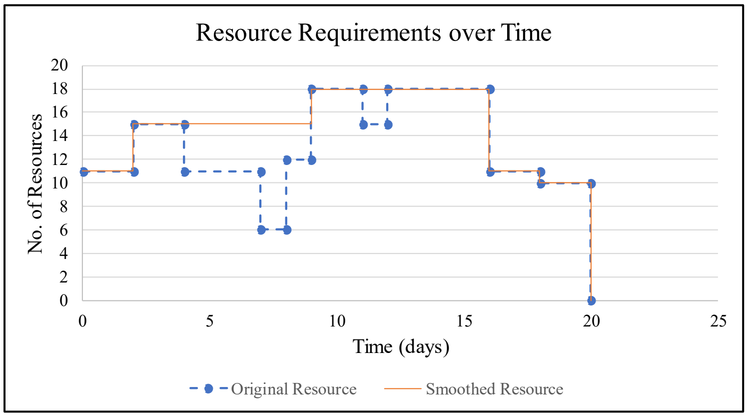
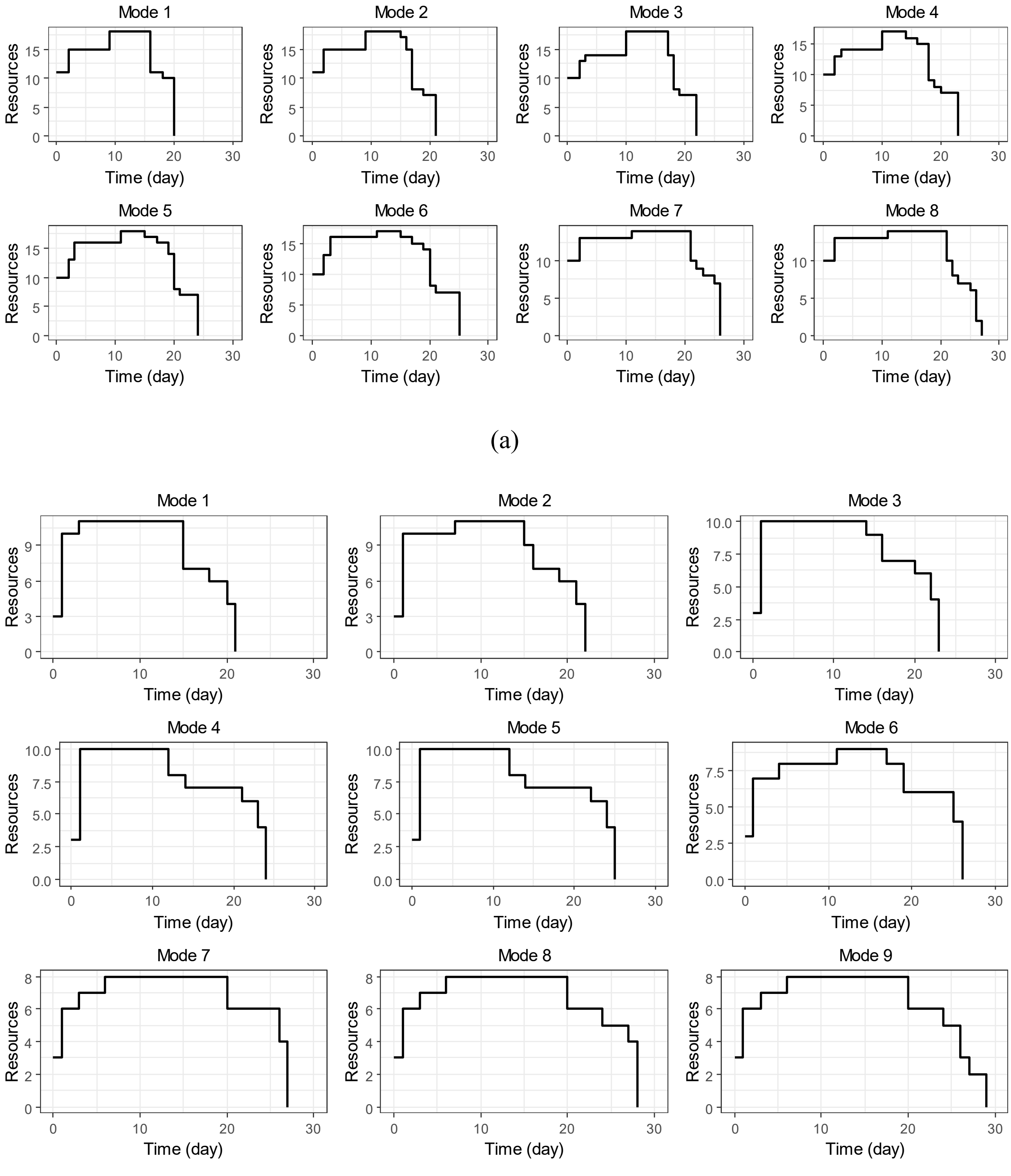


Figure 4.2: Illustration of the macro mode determination.



(a)

(b)

Figure 4.3: Macro modes for (a) Project 1, (b) Project 2.

Table 4.3: Macro modes for two projects.

Macro mode ID	Project 1		Project 2	
	Duration (days)	Cost ($)	Duration (days)	Cost ($)
1	20	75300	21	149500
2	21	73650	22	146300
3	22	72900	23	143100
4	23	72350	24	139900
5	24	71850	25	136700
6	25	71500	26	133500
7	26	71150	27	132900
8	27	70950	28	132700
9	-	-	29	132500

Step 4. Macro-Activity Scheduling

In order to better illustrate the proposed framework and the defined "resource use robustness" indicator, another project (i.e., Project 3) is added by simply duplicating Project 2. So the problem turns to schedule three concurring projects (i.e., Project 1, Project 2, and Project 3) subject to limited resources, and Project 3 has the identical input settings as Project 2. After generating macro modes for each project, the next step is to schedule the three projects by selecting one of the generated macro modes for each project. As introduced in Section 3.2.1, this is a multi-mode resource-constrained project scheduling problem (MRCPSP) consisting of three macro-activities (the first one with eight alternative macro modes, the second one with nine alternative macro modes, and the third one with nine alternative macro modes). The optimization objective is to minimize the total cost, which includes the direct cost and indirect cost, as shown in Eq. (3.1). The direct cost is determined by the selected macro mode. The cost related to each macro mode is listed in Table 4.3. The indirect cost includes two parts: the indirect cost of individual projects and indirect cost in the fabrication facility. The first part is linearly dependent on individual project durations, while the other part is proportional to the total duration of all projects.

Following the optimization model (i.e., Eq. (3.1) to Eq. (3.3)), the scheduled result is shown in Figure 4.4. *IBM ILOG CPLEX Optimization Studio 12.6.3* was utilized to solve the optimization model. It took 35.44 seconds on a desktop with a 3.2 GHz CPU and 8 GB random-access memory (RAM). The start time, finish time, selected macro mode, and allocated resources over time are determined for three projects. The total duration for completing three projects is 43 days, and the total cost is $561,590. Respectively, the macro mode 2, macro mode 3, and macro mode 7 are selected for Project 1, Project 2, and Project 3. The project duration is 21 days for Project 1, 23 days for Project 2, and 27 days for Project 3.

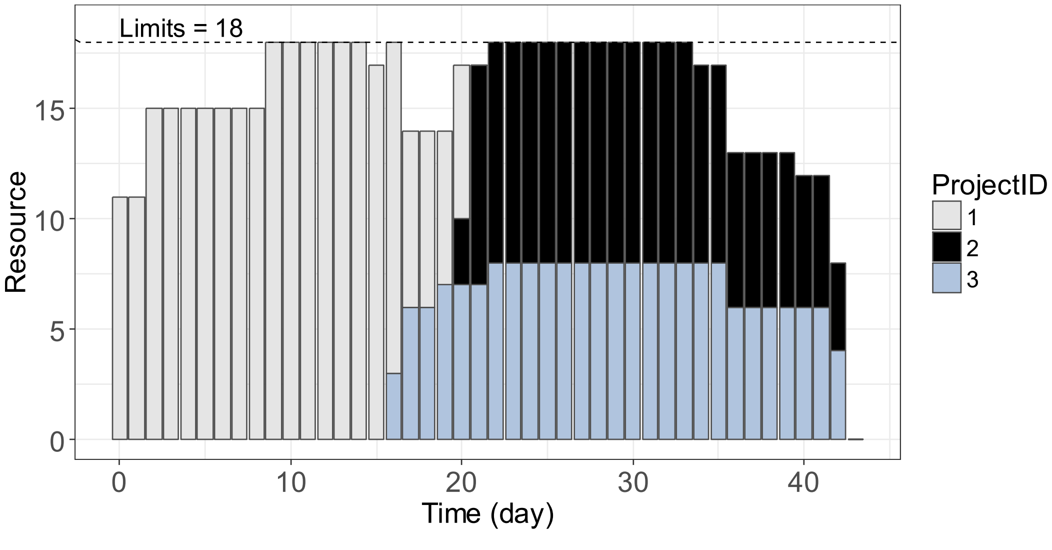


Figure 4.4: Resource allocation for three projects using the proposed method.

After determining the start time, finish time, and allocated resources for each project, the project manager is responsible for performing further activity scheduling at the lower level as per the proposed framework. Figure 4.5 gives an example of the scheduling result of Project 1 at the lower level. Each bar shows the daily resource requirement, while the dashed line indicates the resources allocated from the upper level, which are imposed as resource constraints for scheduling Project 1 at the lower level. Besides, the precedence relationships among activities are another constraint for scheduling Project 1 at the lower level. The objective is set to minimize the total project duration by selecting alternative activity construction options, as the total project cost is quite fixed after resources are allocated from the upper level. The detailed programming code in *IBM ILOG CPLEX Optimization Studio* is attached in Appendix A. As shown in Figure 4.5, project duration and the resource use profile are consistent with the definition of the corresponding macro-activity. It is notable that the smoothed resource profile at the upper level provides more than sufficient resources required at the lower level, lending some cushion of extra resources to accommodate variabilities (e.g., changes, delays, disruptions) to a certain degree during the project execution stage.

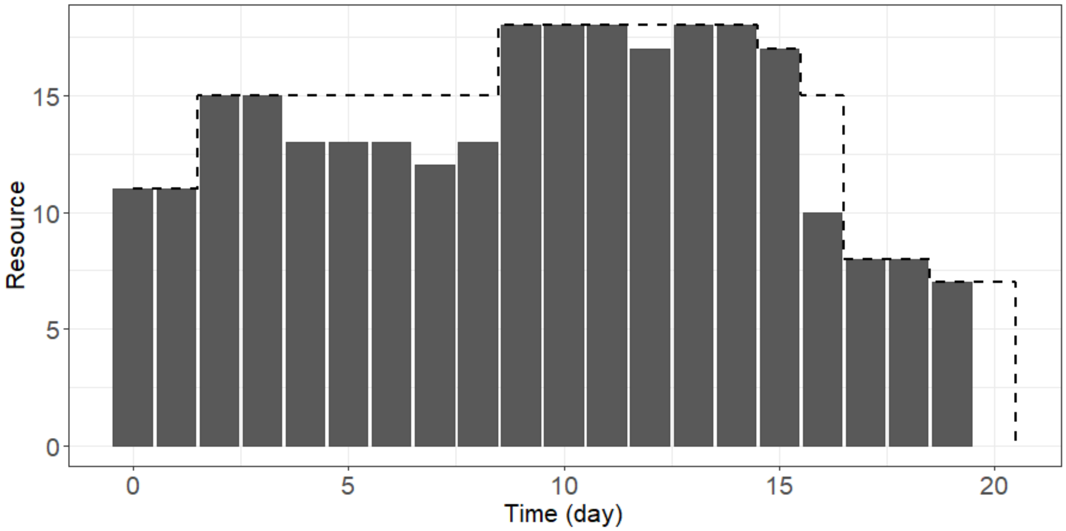


Figure 4.5: Resource allocation for Project 1 at the lower level.

4.3.2 Resource Robustness Evaluation

To compare the robustness of the resource allocation plan in terms of inter-project resource use dependencies, the derived result is compared with those given by two existing multi-project scheduling methods: (1) a single-project resource scheduling optimization approach (referred as the single-project method in the following paragraphs) and (2) an open-source optimization tool OptaPlanner (2018). The comparison of obtained results is presented as follows.

By the single-project method, the three projects are first linked to form a single project network by adding two dummy activities, namely "start" and "finish." Then the problem is transformed into a typical multi-mode resource-constrained project scheduling problem. The newly formed project network consists 85 (25+30+30) activities. The alternative construction options for each activity are listed in Table 4.1 and Table 4.2. The objective is also set as minimizing the total cost. *IBM ILOG CPLEX Optimization Studio 12.6.3* was utilized as the software platform for solving the super-project network. Constraint propagation was utilized to reduce the search space while genetic algorithm was employed to search the optimal solution. The resulting schedule is shown in Figure 4.6. The schedule has a total duration of 36 days and a total cost of $548,762. The duration for each project is the time elapsed from the start of its first activity to the end of its final activity, which is determined as 36 days, 27 days, and 30 days for Project 1, Project 2, and Project 3, respectively. Although the total duration is 7 days shorter than that derived from the proposed framework, the duration for each project is much longer, leading to higher project indirect costs.

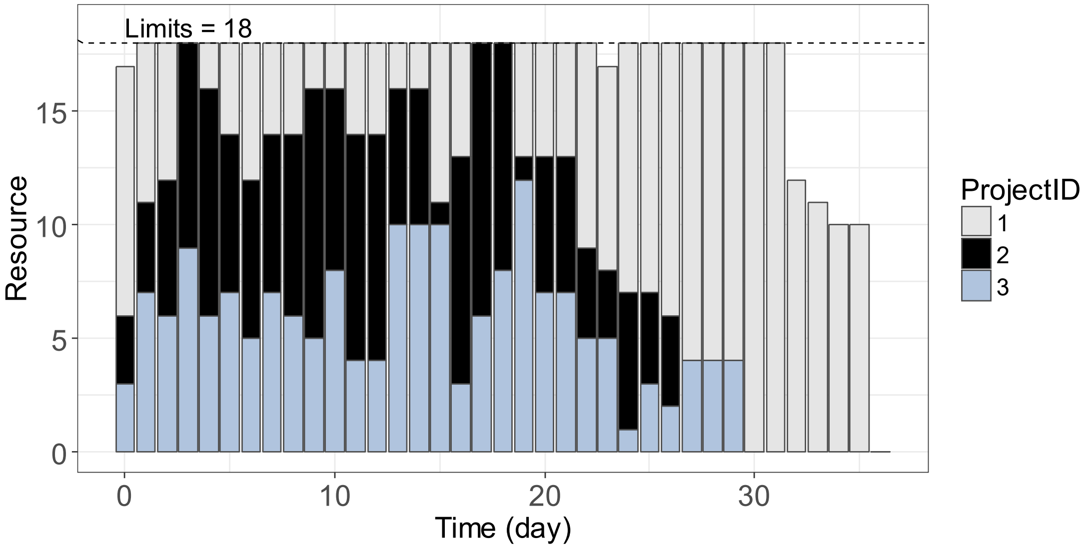


Figure 4.6: Resource allocation for three projects using the single-project method.

To contrast the two scheduling methods (i.e., proposed dual-level resource planning method v.s. single-project method), the daily required amount of resources for each project is plotted over the scheduled project duration, and the resource transfers among projects are highlighted with shadowed patterns in Figure 4.7. Four times of resource transfer take place during the project execution stage for the dual-level scheduling method, in contrast with twenty-three times in the case of the single-project method. The resource use robustness measure for the two methods is determined in Eq.(4.4) and Eq.(4.5), resulting in 90.5% and 34.3% for the proposed method and the single-project method, respectively. It means the resource allocation plan derived from the proposed method is the most robust in light of the least resource use dependency among projects.

$$Resource\ Use\ Robustness_{dual-level} = 1 - \frac{Resource\ Transfer\ Times}{Maximum\ Resource\ Tansfer\ Times}$$

$$= 1 - \frac{4}{43 - 1} = 90.5\% \tag{4.4}$$

$$Resource\ Use\ Robustness_{single-project} = 1 - \frac{Resource\ Transfer\ Times}{Maximum\ Resource\ Tansfer\ Times}$$

$$= 1 - \frac{23}{36 - 1} = 34.3\% \tag{4.5}$$

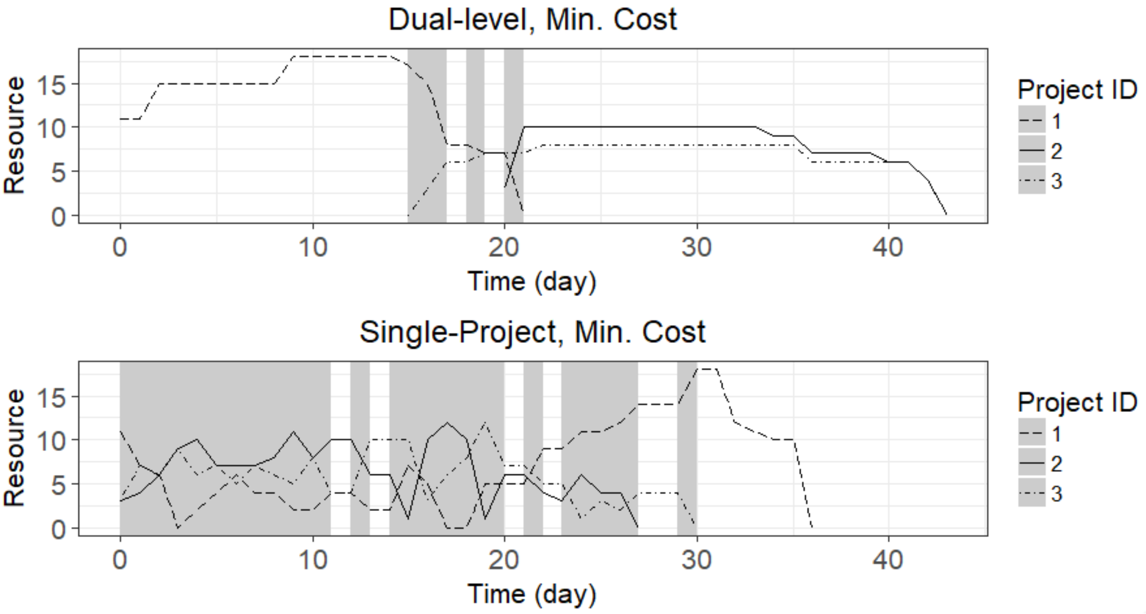


Figure 4.7: Daily required resources per project by proposed method and single-project method with the objective function of minimizing total cost.

OptaPlanner is an open-source and lightweight, embeddable planning engine developed by Red Hat, a well-known multinational software company. *OptaPlanner* employs sophisticated optimization heuristics and metaheuristics, such as Tabu search, simulated annealing, and late acceptance for optimizing planning problems. In this section, the “Project job scheduling” function of *OptaPlanner*, which is designed for the multi-mode resource-constrained multi-project scheduling problem, is utilized for cross-checking with the proposed methodology. The optimization objective is to minimize the individual project duration and total duration. By defining the project network and alternative construction modes for three projects in *OptaPlanner*, the resulting resource allocation is presented in Figure 4.8. The total duration is 39 days, and the total cost is $580,142. In detail, the duration of Project 1, Project 2, and Project 3 is 39 days, 24 days, and 28 days, respectively.

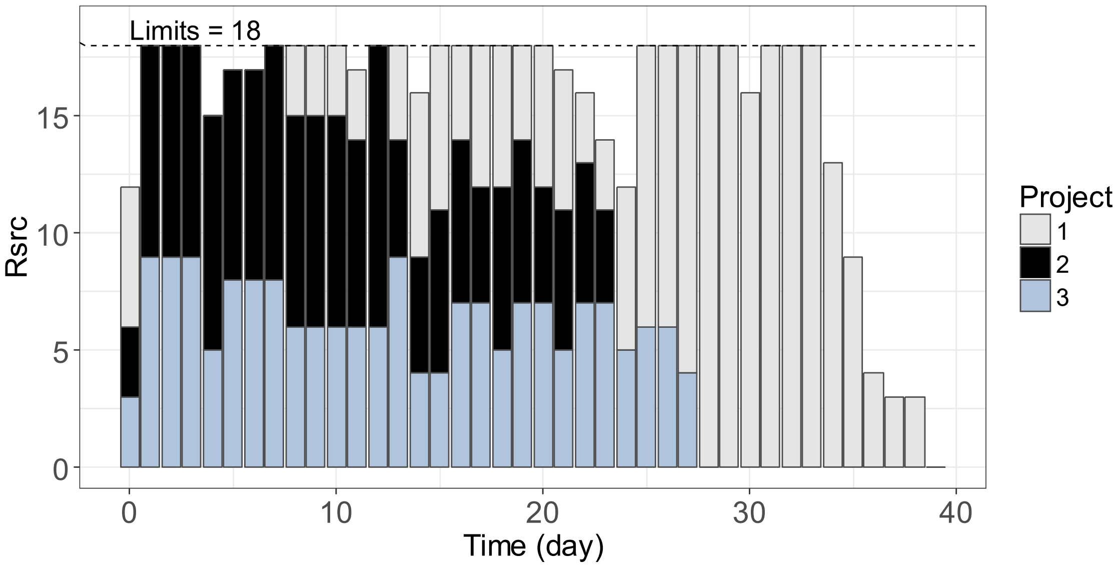


Figure 4.8: Resource allocation for three projects using *OptaPlanner*.

To ensure consistency, the proposed method and the single-project method are modified to minimize individual project duration and the total duration as well. The resulting resource allocation results after modifications are shown in Figure 4.9 and Figure 4.10. The daily required amount of resources per project is plotted over the scheduled project duration, and the resource transfers among projects are also highlighted with shadowed patterns in Figure 4.11.

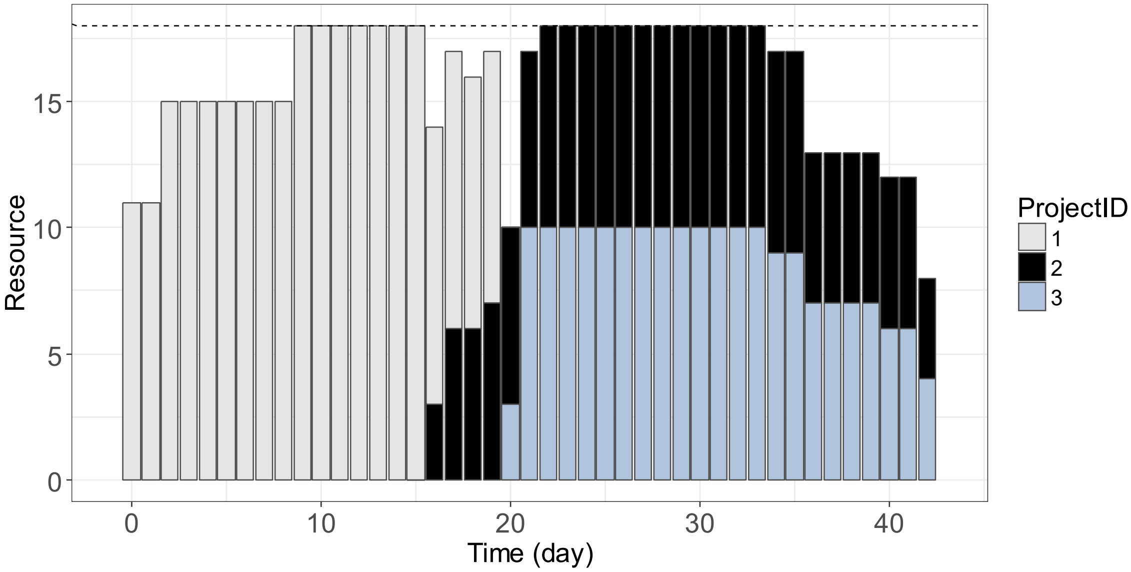


Figure 4.9: Resource allocation for three projects using the proposed method with the objective function of minimizing individual project duration and total duration.

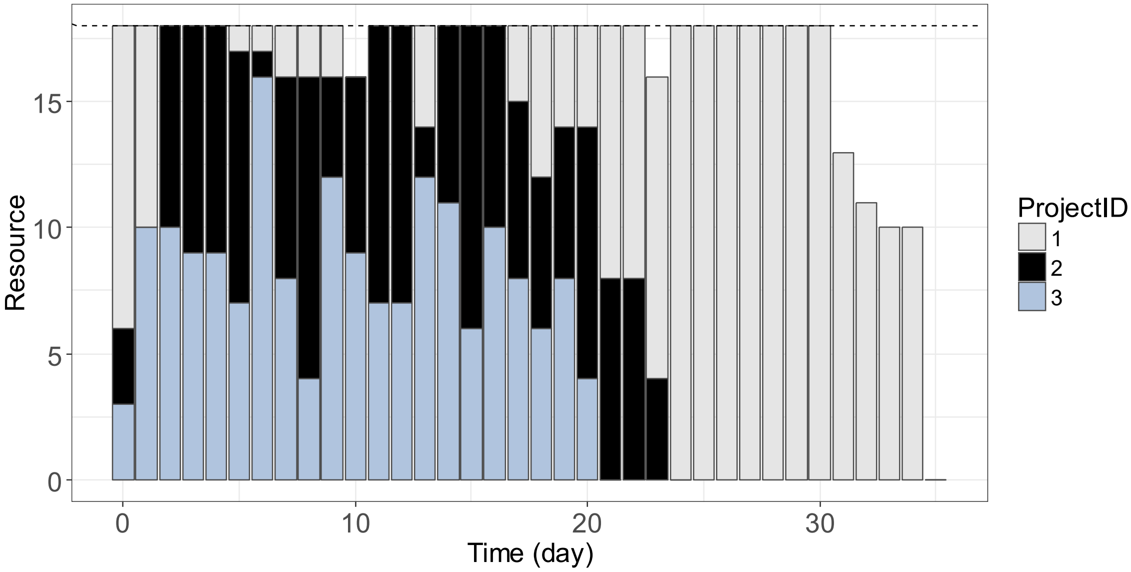


Figure 4.10: Resource allocation for three projects using the single-project method with the objective function of minimizing individual project duration and total duration.

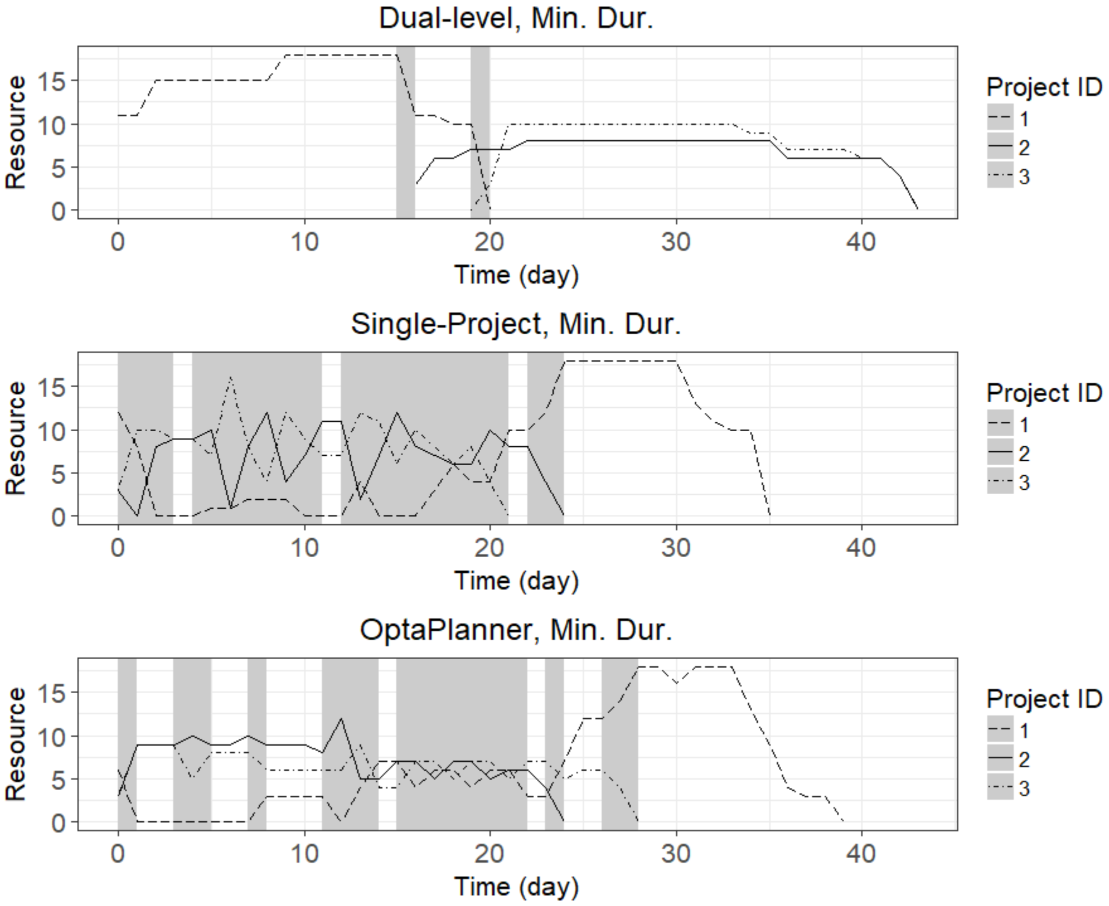


Figure 4.11: Comparison of daily required resources by three different scheduling methods.

Accordingly, the times of inter-project resource transfers are counted, and resource use robustness measures for corresponding schedules are updated, as summarized in Table 4.4.

Table 4.4: Comparison of different scheduling methods.

		Dual-level, Min. Cost	Single-project Min. Cost	OptaPlanner Min. Dur.	Dual-level Min. Dur.	Single-Project Min. Dur.
Total Duration		43	36	39	43	35
Total Cost		$561,590	$548,762	$580,142	$562,400	$571,720
Resource Use Robustness		90.5%	34.3%	56.4%	95.2%	41.2%
Individual Project Duration	Proj. 1	21	36	39	20	35
	Proj. 2	23	27	24	27	24
	Proj. 3	27	30	28	23	21

From Table 4.4, it is noteworthy that setting different objective functions would exert impacts on the resulting total duration, cost, individual project duration, and the resource use robustness measure. It is observed that minimizing total cost leads to more inter-project resource transfers than minimizing individual project duration and total duration. This can be explained by the fact that minimizing individual project duration and total duration leads to mitigating the concurrency of different projects, thus reducing the times of inter-project resource transfers.

Subject to the same objective function (i.e., minimizing duration of each project and total duration), the proposed dual-level scheduling methodology provides schedules with the fewest instances of inter-project resource transfer (over 90% resource use robustness measure). It means the resource allocation plan derived from the proposed method is the most feasible in light of the least resource use dependency among projects. This would lead to much less effort in materials handling, workstation setup, coordination and communication in the execution of different projects in practice. Also, the proposed dual-level scheduling method leads to one resource working continuously to complete one project before being transferred to another. During the project execution stage, the shop superintendent would have full control over the allocated resources over a certain period to cope with any disruptive factors associated with each project and confine the propagation of any schedule delay to other projects. In contrast, the other two methods give rise to increased need for resource transfer among different projects over the scheduled period. Although the single-project method or *OptaPlanner* generates schedules with shorter total duration and lower total cost, chances are much higher to incur extra overhead costs because resources are frequently transferred between the three projects. The optimized schedules would also more likely be disrupted and become irrelevant during the project execution stage.

In this chapter, *project* is defined as a basic unit to generate macro-activities for the upper-level scheduling analysis. For practical applications, it would be more effective to break down one project into several macro-activities each being a sub-project with distinctive common features (such as dimensions, complexity, quality specifications, and delivery dates). With the current version of the proposed framework, how to break down *project* into *macro-activity* in various granularities can be readily evaluated by the resulting cost function and the resource robustness indicator. In collaboration with our partner company the proposed methodology had been validated and implemented in scheduling five bridge fabrication projects consisting of a total of 120 plate girders. Plate girder fabrication of the five bridges lasted for more than one year. Plate girders are generally I-beams arranged in parallel girder lines in the bridge engineering design, which provide longitudinal supports for bridges. Along each girder line, multiple girders are connected to achieve the as-designed length of the bridge span. By analyzing alternatives, *girder line* had been identified as the sub-project in defining macro-activities for scheduling shop fabrication over the five bridges. The senior schedulers had confirmed that the derived schedules to be practically feasible.

4.4 Conclusion

Current scheduling methods for planning shared resource use on multiple projects give rise to extensive resource links among projects, thereby reducing the robustness of the resultant project schedule. To improve the multi-project resource planning practice at a prefabrication facility, a robust dual-level resource-constrained multi-project scheduling framework has been proposed and presented in Chapter 3. In this chapter, the applicability of the proposed framework in mitigating inter-project resource use dependencies and enhancing schedule resource robustness is demonstrated by conducting a case study. A resource use robustness indicator is defined to quantify resource use dependencies among multiple concurring projects. The proposed framework is contrasted with two existing multi-project scheduling methodologies in generating robust resource use plans for scheduling multiple concurring projects using the same case. By adding the practical constraint of “reducing inter-project resource transfer”, the resulting schedule of the proposed framework increases project time and cost. This is expected and justifiable. The deliverables of this research will provide production managers or project managers at a fabrication facility with reliable and feasible work plans, ensuring crew work continuity on individual projects, and enhancing resource utilization efficiency. The framework can also serve as an analytical tool to guide the planning process of establishing appropriate project breakdown structures through trial and error on scenarios of various macro-activity granularity, leading to improved schedule performances (e.g., schedule resource robustness, total duration, and budget).

Although the proposed framework can effectively reduce inter-project resource use dependencies, additional measures, such as the solution robustness, quality robustness, project duration, and cost can be integrated into a cost function together with resource robustness to comprehensively evaluate the proposed framework in the future work. The case study in this chapter treats all the laborers working in the fabrication facility as one type of finite resource. To deal with multiple types of resources, resources need to be distinguished into different types with corresponding availability limits in the mathematical formulation model, along with specifying requirements for each type of resources for each activity. In the following chapter, an actual case study, which involves multiple types of resources (e.g., different trades of laborers and workspace), will be conducted to further demonstrate the practical applicability of the proposed framework in scheduling multiple concurrent prefabrication projects and linking schedules for various management functions.

5 CHAPTER 5: SYNCHRONIZATION OF VARIOUS MANAGEMENT-FUNCTION SCHEDULES IN A MULTI-PROJECT ENVIRONMENT: CASE STUDY ON BRIDGE GIRDER FABRICATION PROJECTS

5.1 Introduction

Bridge girder fabrication, which epitomizes prefabrication and modular construction approaches, is commonly operated in a multi-project environment, where multiple fabrication projects from various clients are executed at the same time with limited fabrication resources (e.g., skilled labors and workspaces) in a fabrication facility. *Project Management Institute (PMI)* defines "*program*" as "*a group of related projects managed in a coordinated way to obtain benefits and control not available from managing them individually* (Wideman 2004)." In a multi-project environment, a program manager is concerned with how to effectively allocate limited resources across projects in order to realize program goals; in the meantime, the operations manager at a fabrication facility is responsible for delivering each individual project subject to milestones and budget constraints. However, schedules for different management functions are isolated in current practice mainly due to the lack of (1) a unifying work breakdown structure (WBS), (2) a systematic and integrated scheduling approach, and (3) a ready-to-use schedule synchronization tool. At present, in a multi-project environment schedules are initialized and maintained separately for tracking milestones on individual projects, rather than by an integrated program (Elonen and Artto 2003). In the fabrication facility operations are planned by a facility manager based on experiences and executed in a mode analogous to "fire extinguishing" (Yaghootkar and Gil 2012). The piecemeal planning process leads to resources conflicts among projects (Laslo and Goldberg 2008; Payne 1995), disputes among project management teams (Flyvbjerg 2014; Platje et al. 1994), and failure in delivering project milestones (Hamzeh et al. 2012).

In reality, a multi-project scheduling approach is desired to generate schedules by synchronizing various management functions based on a unified WBS, thereby enhancing communication efficiency and project management performance. In particular, this paper is aimed to improve the multi-project scheduling performance for bridge girder fabrication projects by (1) developing a unified WBS that serves various project management functions such as estimating, planning, and scheduling, (2) synchronizing schedules for various management functions through implementing a dual-level multi-project scheduling framework, and (3) finally developing a multi-project scheduling tool to facilitate the implementation of the dual-level scheduling framework in practice. In collaboration with our industry partner in Edmonton, Canada, three bridge girder fabrication projects from the real world are adapted into a demonstration case. In the next section, background information about schedules of different details and the newly proposed dual-level multi-project scheduling framework is provided. Next, the framework implementation is presented step by step. The bridge girders fabrication process is introduced prior to presenting the case study on planning and scheduling three bridge girder fabrication projects. Finally, a discussion section highlights the potential advantages of the proposed new framework and conclusions are drawn.

5.2 Background

5.2.1 Schedules for Various Management Functions

In general, a master schedule summarizing project start time, delivery time, and important milestones is effective for contractor-client communication; while schedules with more operation details are intended to guide superintendents to execute daily work in the field. To standardize the representation of schedules for various management functions, CII (2004) endorsed a numeric designator system (i.e., level 0, level 1, level 2, level 3, and level 4-X). Level 0 is a single bar, which spans from start to finish and represents the schedule for a program or a total project. For program schedules, a Level 1 schedule is a combination of Level 0 schedules for each individual project. As the number increases, the subdivision continues from Level 2 (e.g., by areas or divisions), Level 3 (e.g., for project monitoring and control purpose), to Level 4-X (e.g., 4-9 weeks look-ahead schedules or weekly schedules). In reality, consensus on the numbering system is difficult to reach by different schedulers. To eliminate confusions caused by the numbering system, the descriptive method is also proposed for representing various schedules. By the descriptive method, program/project summary schedule, milestone schedule, project schedule, project control schedule, look ahead schedule, task lists, and supporting data are distinguished (AACE 2010). In a separate but related attempt to formalize the *Last Planner System*, four degrees of planning processes were defined for production planning and control, namely: master scheduling, phase scheduling, lookahead planning, and weekly planning (Hamzeh et al. 2008).

Schedules of finer granularity, such as weekly plans, should be consistent with master schedules and phase schedules in terms of meeting milestones for project delivery (Hamzeh et al. 2008). On the other hand, the setting of milestones in a master schedule or phase schedules needs to be aligned with more detailed schedules. To ensure the successful project delivery, it is advisable that schedules for various management functions should be derived in a “roll-up” or “roll-down” fashion based on a common source of information, as opposed to being developed as separate versions throughout a project life cycle (AACE 2010). Figure 5.1 illustrates the ideal relationship between various schedules. Nevertheless, schedules for various management functions generally exist in isolation and the ideal state of schedule synchronization is difficult to materialize in reality.

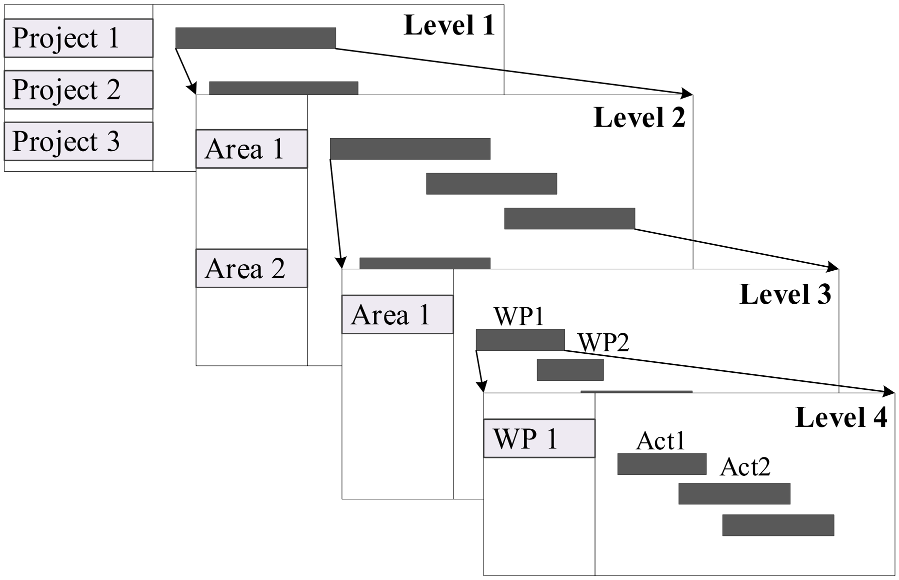


Figure 5.1: Example of the desired relationships between various schedules.

5.2.2 Dual-Level Multi-Project Scheduling Approaches

The dual-level planning structure is originally formalized by researchers in the Operational Research (OR) domain, which has been proven effective to manage resources in a multi-project environment (Can and Ulusoy 2014; Speranza and Vercellis 1993; Yang and Sum 1993). Compared to multi-project resource allocation and scheduling research in the construction engineering and management (CEM) domain, the main characteristic of the dual-level planning structure is that resources are allocated to each individual project for a certain period, as opposed to being allocated to specific activities. As such, the dual-level planning methodology reduces frequent resource transfers from one project to another during schedule execution.

To facilitate its implementation in CEM domain, Liu and Lu (2018) adapted the OR approach into a dual-level resource planning framework for scheduling multiple prefabrication projects. The proposed framework contains two levels of analysis, namely: 1) the upper-level for determining milestones and allocating resources to projects and 2) the lower-level for performing detailed activity scheduling within one project subject to milestones and resources determined at the upper level, plus other project-specific constraints. It is worth mentioning that the "*dual-level*" defined in the newly proposed multi-project scheduling framework differs from the "*levels of schedules*" defined by CII. The upper level in the "dual-level" framework refers to project planning analyses and decisions, including breaking down project into activities, identifying precedence relationships among activities, determining time and resource use requirements on activities, establishing deliverable milestones, and devising a workable plan for multiple projects (Andersen et al. 2009; PMI 2000). The lower level in the "dual-level" framework refers to workface planning, which is intended to organize and deliver all the necessary elements before craft people start specific work in the field (CII 2013). So, the outputs of the upper-level analysis include program schedule (Level 1) and project schedule (Level 2) as per CII classification. Meanwhile, the lower-level analysis generally results in operation schedule (Level 4). To avoid confusion, hereafter the word "*level*" is used exclusively to represent the two levels of analyses in the proposed dual-level framework, namely: project planning at the upper level and workface planning at the lower level.

To achieve the linkage between the two levels and to align with current practice, a new work breakdown unit called "*macro-activity*" is defined to represent a larger scope of work from one project. On real-world projects, the appropriate scope of macro-activities needs to be determined in establishing work breakdown structure (WBS). Each macro-activity, which consists of a group of detailed activities at the lower level, acts as the basic work unit for requesting and utilizing resources; interrupting a macro-activity by preempting allocated resources would result in extra costs (e.g., resource transfer cost, productivity loss due to unlearning curve effect) and hence is not allowed in scheduling. The duration and resource requirements of the aggregated macro-activity are determined by scheduling component activities at the lower level subject to constraints such as available resources.

Upon generating macro-activities from the multiple projects being planned, the next step is to determine the start or finish milestones and allocate available resources for each macro-activity by solving a resource-constrained scheduling problem. In this step, macro-activity is the basic unit for resource allocation instead of activities at the lower level. Precedence relationships among macro-activities and resource availability limits are the two sets of constraints imposed for sequencing macro-activities over time. In general, two objectives are defined in scheduling macro-activities; they are (1) minimizing the total duration of the entire set of projects being scheduled and (2) minimizing differences between preset finish times (i.e. project delivery milestones) and scheduled finish times on each project. Subject to the allocated resources and determined milestones resulting from macro-activity scheduling at the upper level, the lower-level activity scheduling problem turns into the classic resource-constrained single project scheduling problem. The elaboration and detailed formulation of macro-activity aggregation model, macro-activity scheduling model, and lower-level activity scheduling model have been explained in Chapter 3. The remainder of this chapter focuses on methodology implementation in a case study.

5.3 Methodology Implementation

In this section, the case study of scheduling three bridge girder fabrication projects is implemented following the dual-level multi-project scheduling framework, as depicted in Figure 5.2. The first step is to define macro-activities from each project by analyzing three types of project data; they are (1) work breakdown structure (WBS), (2) engineering design drawings, and (3) historical productivity data. Investigating the WBS of bridge girder fabrication projects is conducive to determining the scope of each macro-activity. Engineering design drawings are analyzed to take off the work content for each macro-activity. Analyzing historical productivity data is instrumental in determining resource requirements and time duration on each macro-activity. The second step is to schedule the generated macro-activities by allocating limited resources, resulting in start or completion milestones on each macro-activity. The last step is to derive schedules for various management functions including program schedule, project schedule, and operation schedule.

Figure 5.2 also shows the relationships between the three schedules in accordance with numeric and descriptive schedule levels classified by CII/AACE. First, a program schedule, which includes start time, finish time, and span of each project, corresponds to the Level 1 schedule. It is intended to support the decision making in terms of organization’s business strategies, such as bidding new projects. Also, project schedules, corresponding to Level 2 schedule, depict the sequences in the execution of all the macro-activities contained in one project and assist in managing important milestones so to meet clients’ deadlines. Besides, operation schedules (i.e., Level 4 schedule) are developed to guide the superintendent of a fabrication shop in planning the execution of daily operations in the fabrication facility for the near future (e.g., 4-6 weeks) and monitoring work progress. The three schedules need to be synchronized in such a way that if any change occurs to the operation schedules, the impact would be immediately reflected in project and program schedules.

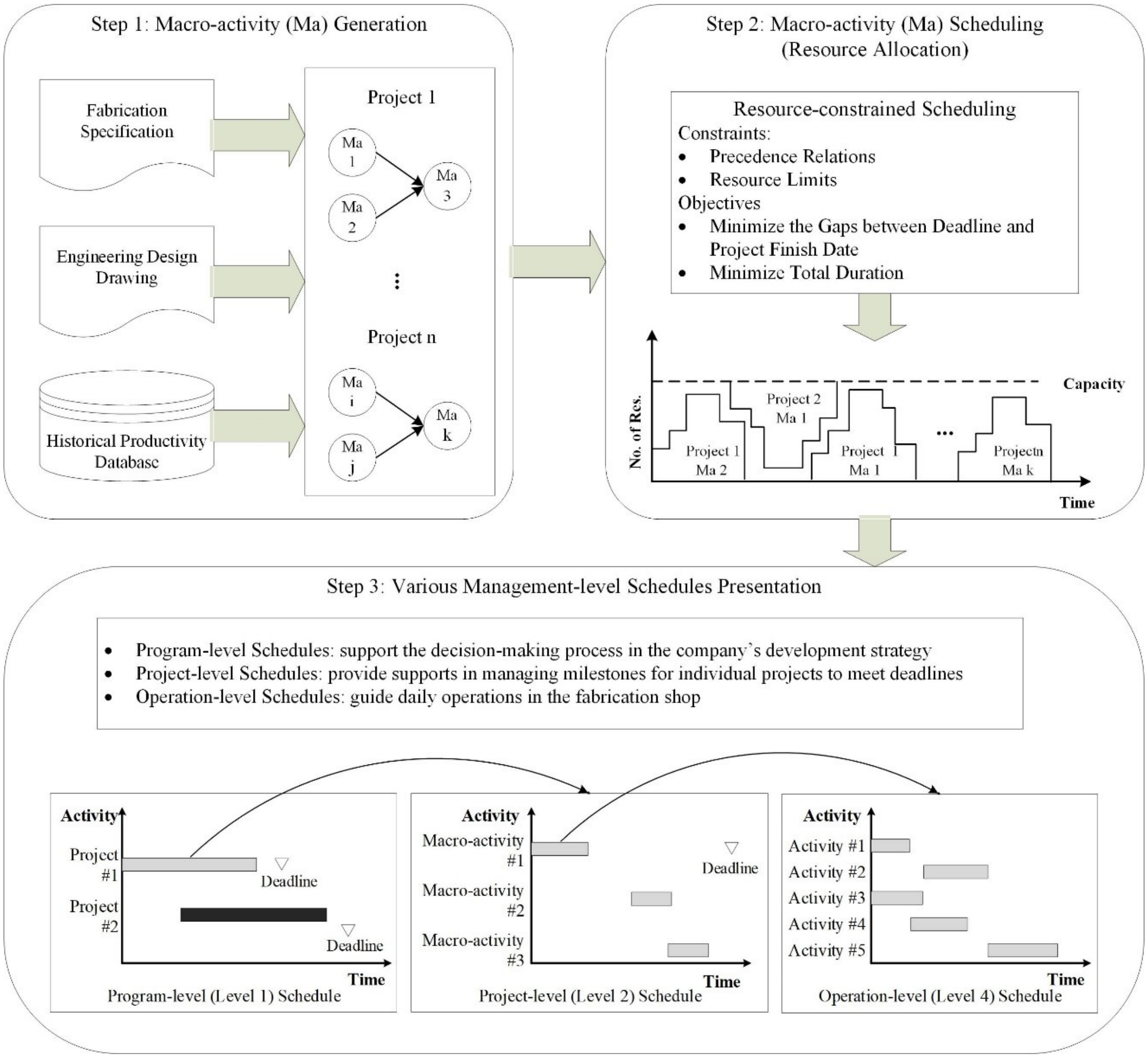


Figure 5.2: Research methodology flow chart.

In the following sections, the current practice in fabricating bridge girders is first introduced to provide a general background for the case study. Next, three real-world bridge girder fabrication projects to be processed by a fabrication facility are utilized to illustrate (1) how to determine the scope of a macro-activity for employing the proposed framework in scheduling multiple concurrent projects and (2) how to generate schedules for various management functions and how those schedules interrelate and interact.

5.4 Bridge Girder Fabrication Practice

In the fabrication shop, raw steel plates go through a series of operations that include cutting, drilling, fitting, welding, sandblasting, painting, and other surface finishing work. Figure 5.3 depicts a finished plate girder with key features annotated. A plate girder, which consists of one web, two flanges (the top flange and the bottom flange), stiffeners, and shear studs, is widely used in steel superstructures (Krause 2015). The web mainly provides shear strength, flanges provide bending strength, while stiffeners can provide shear buckling resistance, bearing force, and flexure resistance depending on the stiffener types (e.g., intermediate transverse stiffeners, bearing stiffeners, and longitudinal stiffeners). Shear studs ensure shear connections between steel and concrete and prevent relative motions in both horizontal and vertical directions.

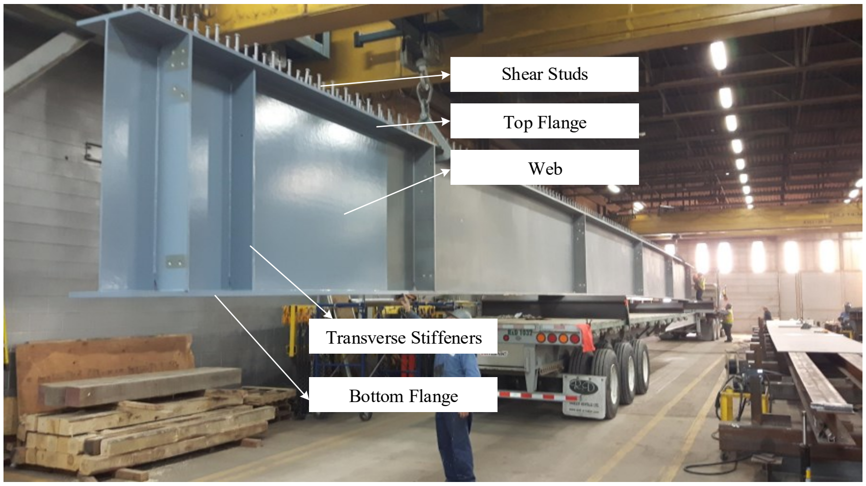


Figure 5.3: Finished plate girder for shipping.

Plate girders are usually arranged in parallel *girder lines* in the bridge engineering design. Along each girder line, multiple individual girders are bolted together on the construction site to form a continuous girder line with the as-designed length of the bridge span. The typical fabrication process of one plate girder is illustrated in Figure 5.4.

After shop drawings and all the required materials are ready, it starts with shop detailing on raw flat plates based on engineering design drawings, including pre-blast, cut, and drilling. Then, webs and flanges are made from these cut plates through straightening and splicing processes. For all the connections (e.g., splicing flanges, splicing webs, and assembling flanges and web), tack welds are applied as temporary connections to hold components in position before final welding is performed. After the preparation of webs and flanges is done, one web and two flanges are assembled into a girder by tack welds. In this step, specific machinery (e.g., overhead cranes and squeezer) is utilized to lift, handle, and fix the web and flanges. Flanges need to be fitted tightly to the web with no gap. Once the web and flanges are assembled, final welding is performed to permanently connect web and flanges. Next, stiffeners and studs are attached to the assembled girder based on the engineering drawings. Upon finishing this step, one girder is formed to shape. The following step is to drill holes for field splicing so that two adjacent girders in the same girder line can be connected by bolting in on-site installation. At this step, the two adjacent girders are first aligned in the fabrication shop. Drilling is then performed on the girder splice end, flange splice plates, and web splice plates, followed by sandblasting, painting, and other surface finishing work. The fabricated plate girders are inspected prior to being shipped to the site for installation.

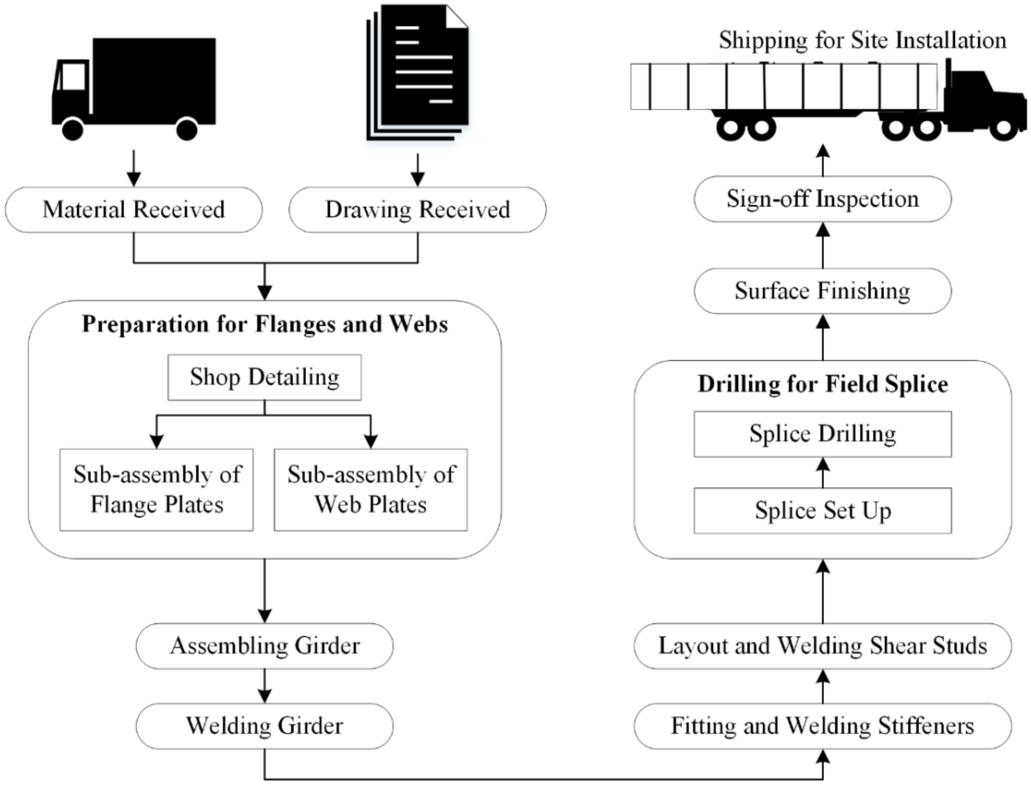


Figure 5.4: The fabrication process for plate girders.

As for drilling for field splicing, one fabricated girder needs to sit inside the fabrication shop waiting for the next girder to be ready. However, if the next girder cannot be completed soon, say, within one week, the finished girder needs to be moved out of the shop in order to make room for fabricating subsequent girders. The girder will only be moved back to the shop for drilling at a later time when the other girder is ready. Therefore, girders on one girder line are generally fabricated continuously to avoid extra handling. This logical constraint in girder fabrication provides the basis for defining the scope of macro-activities on multiple bridge girder fabrication projects.

5.5 Description of Three Bridge Fabrication Projects

The prefabricated bridge girders are of more than ten meters long and one meter wide, so workspace in the fabrication facility is identified as one critical resource for project scheduling. Although the fabrication process for a plate girder is straightforward as shown in Figure 5.4, different fabrication shops may have a different layout of workstations, depending on size of the individual facility. Figure 5.5 shows the work sequence along workstations in the fabrication facility studied. Station 1, Station 2, and Station 3 are for preparing flanges; Station 4 is for preparing webs. After flanges and webs are ready, the ensuing step is to assemble girder (at Station 5), weld girder (at Station 6), attach stiffeners and studs (at Station 7), drill for field splice (at Station 8), and sandblast (at Station 9).

In this case study, fabrication of plate girders on three bridges (i.e., Bridge 1, Bridge 2, and Bridge 3) is planned and scheduled. The layouts for the three bridges are shown in Figure 5.6. Each line segment separated by "*Field Splice*" is a girder. "*Field Splice*" is the bolted connection between individual girders. Each horizontal line is a girder line. In total, Bridge 1 consists of 15 girders which are grouped into three girder lines (i.e., G1, G2, and G3); Bridge 2 is made of 20 girders grouped into four girder lines (i.e., G1, G2, G3, and G4); and Bridge 3 consists of 6 girders grouped into three girder lines (i.e., G1, G2, and G3). The identification for each girder is denoted by the identification number of the girder line and the column number. For instance, G1A denotes the girder on girder line "G1" in Column "A."

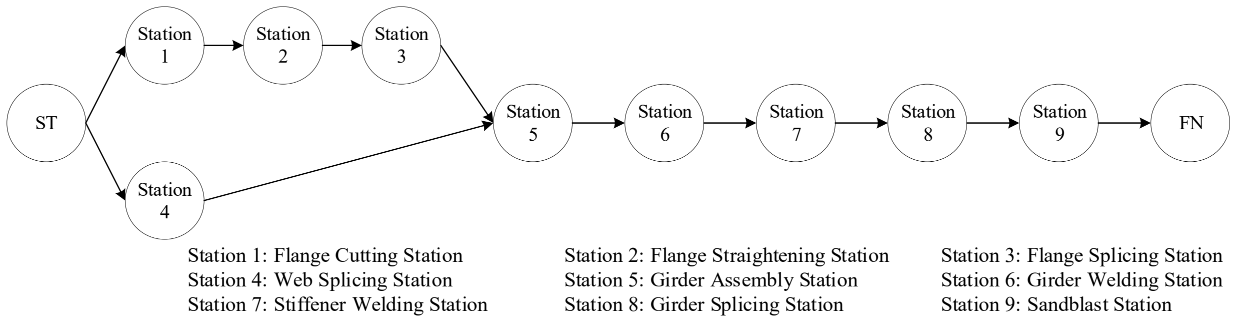


Figure 5.5: Working process for fabricating one plate girder in the partner company.

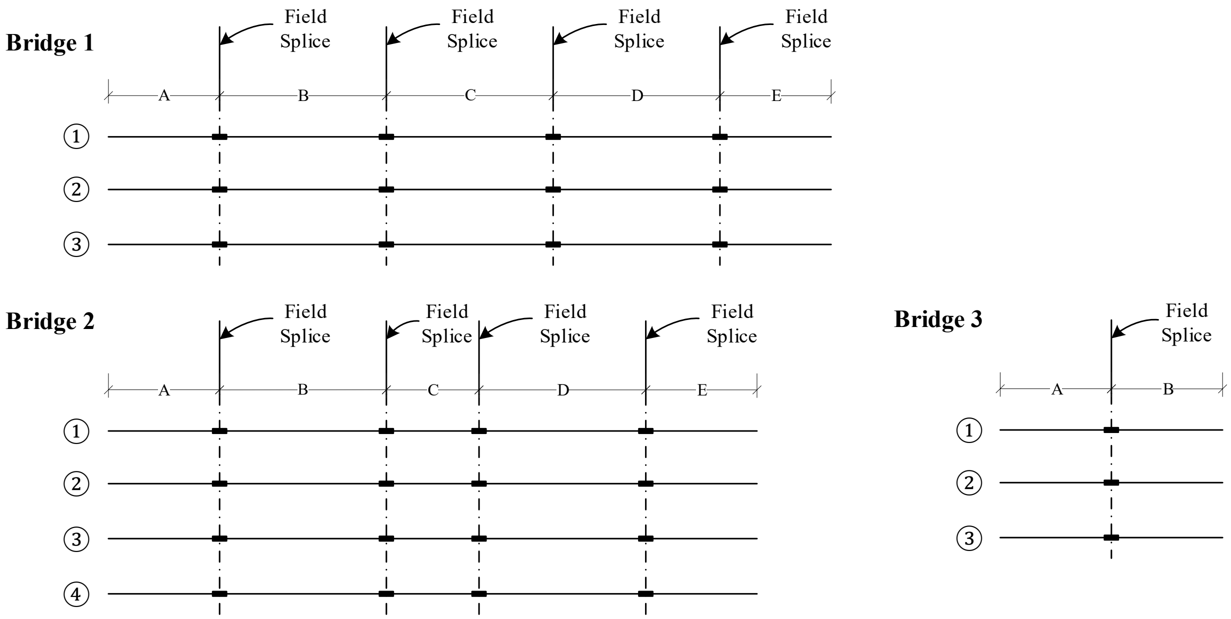
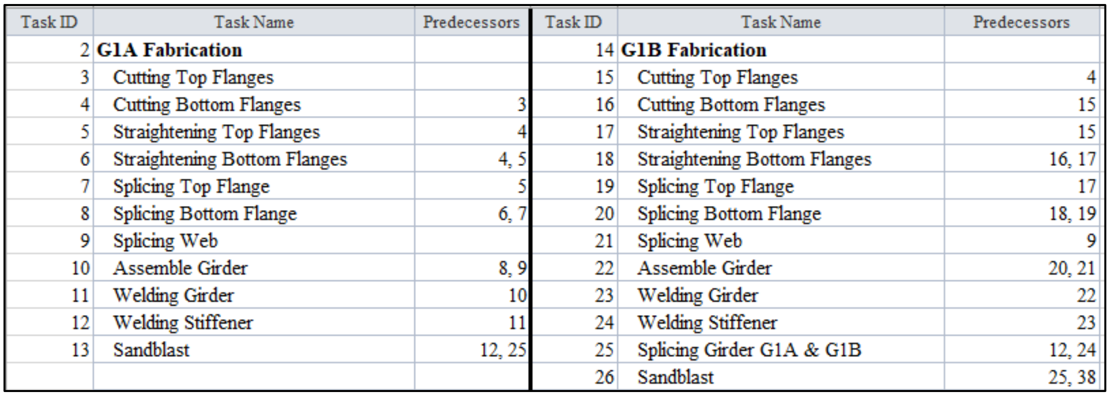


Figure 5.6: Girder layouts for the three bridges.

Each girder needs to go through the fabrication process articulated by workstation-based activities, as shown in Figure 5.5. Workstation-based fabrication activities are listed in Figure 5.7 for girder G1A and G1B of Bridge 1, respectively. As there are two flanges on one girder, one activity is defined for the top flange and the other for the bottom flange. For instance, the activity *cutting flanges* is separated as *cutting top flanges* and *cutting bottom flanges* in Figure 5.7. The precedence relationships are also included in Figure 5.7, which is consistent with those shown in Figure 5.5.

In a similar way, workstation-based fabrication activities are defined for all the girders of the three bridges.

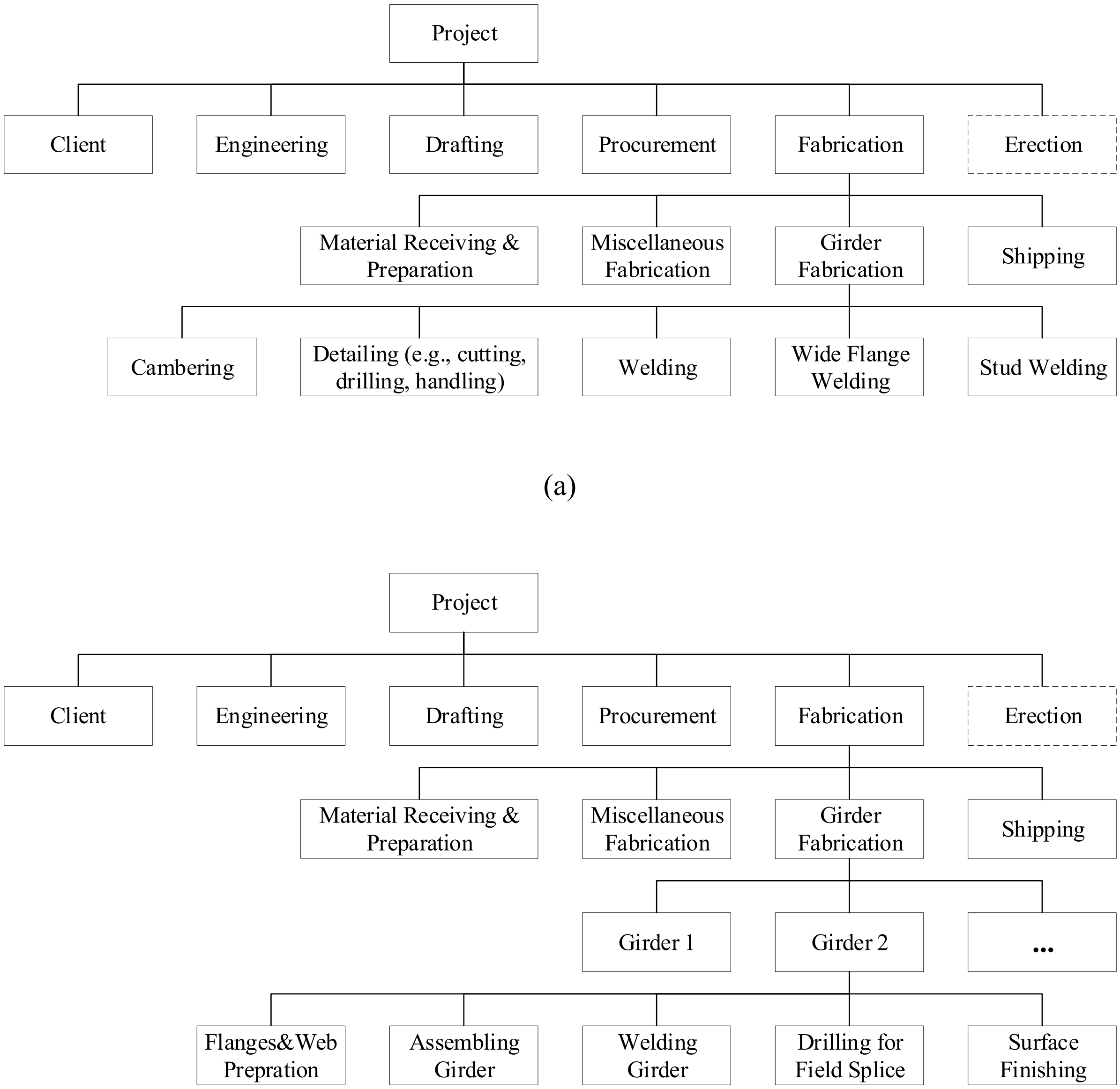


Task ID	Task Name	Predecessors	Task ID	Task Name	Predecessors
2	**G1A Fabrication**		14	**G1B Fabrication**	
3	Cutting Top Flanges		15	Cutting Top Flanges	4
4	Cutting Bottom Flanges	3	16	Cutting Bottom Flanges	15
5	Straightening Top Flanges	4	17	Straightening Top Flanges	15
6	Straightening Bottom Flanges	4, 5	18	Straightening Bottom Flanges	16, 17
7	Splicing Top Flange	5	19	Splicing Top Flange	17
8	Splicing Bottom Flange	6, 7	20	Splicing Bottom Flange	18, 19
9	Splicing Web		21	Splicing Web	9
10	Assemble Girder	8, 9	22	Assemble Girder	20, 21
11	Welding Girder	10	23	Welding Girder	22
12	Welding Stiffener	11	24	Welding Stiffener	23
13	Sandblast	12, 25	25	Splicing Girder G1A & G1B	12, 24
			26	Sandblast	25, 38

Figure 5.7: Illustration of workstation-based fabrication activities.

5.6 Macro-Activity Generation from Individual Project

In order to define and generate macro-activities, three types of information (i.e., work breakdown structure, engineering design drawing, and historical production database) are investigated. In practice, two separate work breakdown structure (WBS) are available for estimating and scheduling purposes. Figure 5.8 contrasts the two types of WBS in the particular context of the fabrication process. It can be seen that the first three branches of the two WBSs are consistent with each other. Divergence arises from the fourth branch. The estimating WBS is expanded according to different trades of laborers, as the estimating WBS is developed mainly to account for labor cost; while the scheduling WBS is further expanded by girders and then by main milestones for progress tracking



(a)

(b)

Figure 5.8: Two types of WBS in the partner company: (a) estimation, (b) scheduling.

To facilitate macro-activity definition, a new WBS is proposed by integrating the two existing ones in Figure 5.9, where the first three branches are omitted to avoid repetition. Considering the fabrication practice that girders on the same girder line in one bridge are fabricated continuously, one more layer (i.e., girder line) is added to the WBS. Then the WBS is first expanded by each individual girder and further expanded by each workstation and trades of laborers.

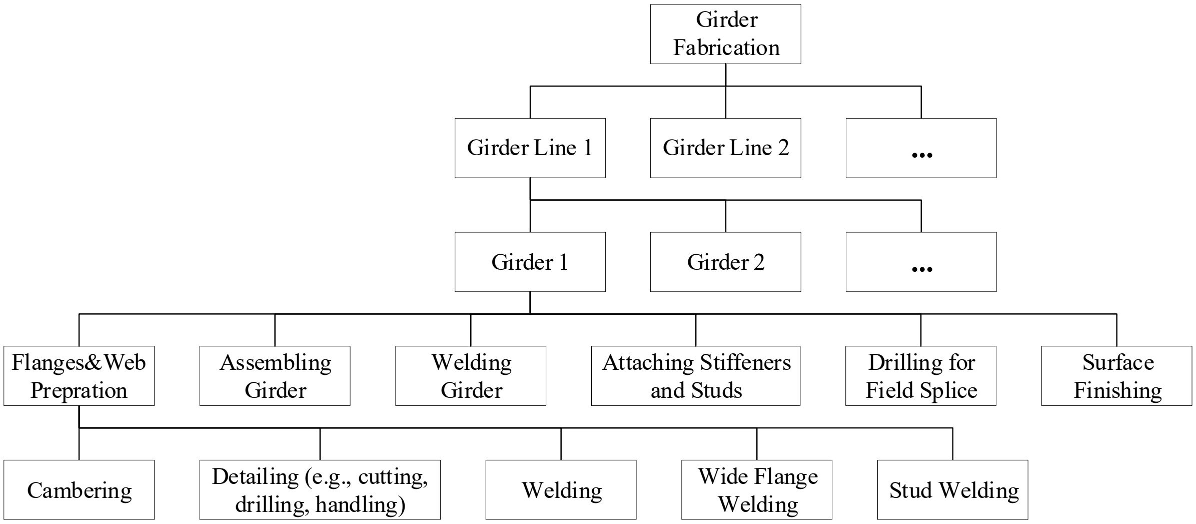


Figure 5.9: Proposed WBS for macro-activity definition.

Based on the new WBS, each macro-activity is corresponding to a particular girder line. Specifically, Bridge 1 has three macro-activities, Bridge 2 has four macro-activities, and Bridge 3 has three macro-activities. As girders on the girder lines G1 and G3 of Bridge 1 are of the same type and design layout, the two girder lines thus are aggregated to one macro-activity (named *Bridge1 M1*). Similarly, G2 of Bridge 1 is transformed to macro-activity *Bridge1 M2*; G1 and G4 of Bridge 2 are aggregated into macro-activity *Bridge2 M1*; G2 and G3 of Bridge 2 are represented by macro-activity *Bridge2 M2*; the macro-activity denoting G1 and G3 of Bridge 3 is called *Bridge3 M1*; G2 of Bridge 3 is defined as macro-activity *Bridge3 M2*. The correspondence relationship between girder lines and macro-activities in the present case study is tabulated in Table 5.1.

Table 5.1: Correspondence relationship between girder lines and defined macro-activities.

Bridge	Girder Line	Macro-activity Name
Bridge 1	G1, G3	Bridge1 M1
	G2	Bridge1 M2
Bridge 2	G1, G4	Bridge2 M1
	G2, G3	Bridge2 M2
Bridge 3	G1, G3	Bridge3 M1
	G2	Bridge3 M2

To determine duration and resource requirements of workstation-based fabrication activities (Figure 5.7), engineering design drawings of the three bridges and historical productivity data from the partner company were investigated. The design features of a girder were considered, including the web thickness, girder length, girder depth, shape (e.g., skewed or not, kinked or not), and the number and type of stiffeners attached to the girder. For instance, a girder with the length of 15 meters can be fabricated directly from raw plates, the length of which is greater than 15 meters; while for an overlong girder (e.g., 40 meters), two raw plates are required to be cut and spliced to reach the design length. So the above two girders are classified into two distinct girder types for determining resource use and time requirements of relevant fabrication activities. Given girders of the same length, the quantity and type of stiffeners attached on the girder also impact activity time in fabrication. For instance, as illustrated in Figure 5.10, a bridge consists of four girder lines (as represented in parallel straight lines), with the splice connections between adjacent girders marked as short solid bars. Note in Figure 5.10, *exterior girder lines* are contrasted against *interior girder lines* in Figure 5.10; along each girder line, *exterior girders* at the ends and *interior girders* in the middle are also distinguished. Intermediate diaphragm stiffeners are designed to connect two adjacent girders each belonging to different girder lines. As Figure 5.10 shows, a girder on the exterior girder line only requires the welding of intermediate diaphragm stiffeners on one side of the web in order to connect with the girder on interior girder line. On the other hand, a girder on the interior girder line needs the welding of intermediate diaphragm stiffeners on two sides of its web for connecting with adjacent girders.

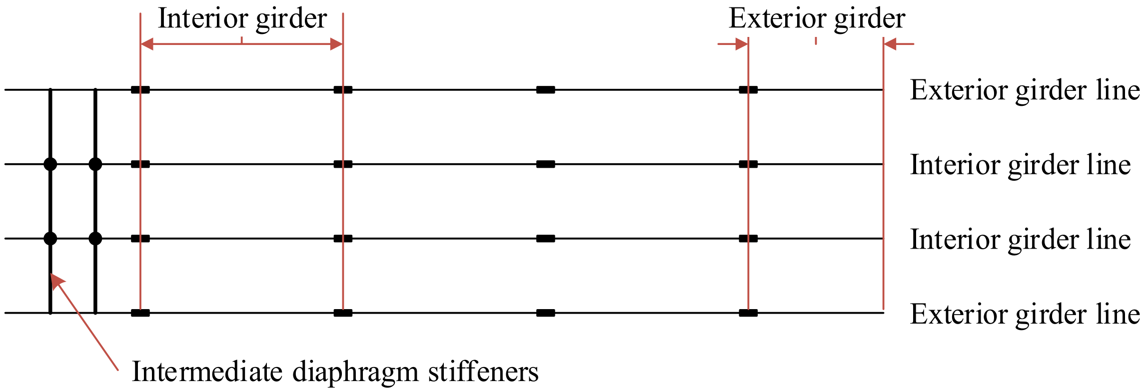


Figure 5.10: Example of intermediate diaphragm stiffener design.

By analyzing the design information, six types of girders (i.e., type 1 to 6) are defined, as listed in Table 5.2. Each girder of the three bridges is categorized into one of the six girder types, also given in Table 5.2.

Table 5.2: Features of girder types and classification of girders in the three bridge projects.

Girder Type	Description	Length (m)	Girders		
			Bridge 1	Bridge 2	Bridge 3
Type 1	Exterior girder line, exterior girder	45.9	G1A, G1E, G3A, G3E	G1A, G1E, G4A, G4E	G1A, G1B, G3A, G3B
Type 2	Interior girder line, exterior girder	45.9	G2A, G2E	G2A, G2E, G3A, G3E	G2A, G2B
Type 3	Exterior girder line, interior girder	44.5	G1B, G1C, G1D, G3B, G3C, G3D	G1B, G1D, G4B, G4D	-
Type 4	Interior girder line, interior girder	44.5	G2B, G2C, G2D	G2B, G2D, G3B, G3D	-
Type 5	Exterior girder line, exterior girder	20.5	-	G1C, G4C	-
Type 6	Interior girder line, interior girder	20.5	-	G2C, G3C	-

The most likely value for processing each type of girder at each station is derived based on historical data and utilized in this case study. The labor requirements over different time periods at each workstation are listed in Table 5.3, Table 5.4, and Table 5.5 based on the girder type. It is noteworthy labor requirements and duration to process a particular girder at some workstations can be time-dependent. Taking the fabrication work in Station 1 for girder type 1 for example, the total duration is 8 hours, it requires 2 journeymen in the first three hours (0~3 hour), and then 1 journeyman for the last five hours (3~8 hour). When performing operation scheduling, these two time periods are separated into two consecutive activities, each having a constant resource requirement over certain time duration. For instance, Station 1 it has two activities in fabricating on girder type 1: one activity lasts for 3 hours and requires 2 journeymen; while the other one lasts for 5 hours and requires 1 journeyman. The second activity starts right after the first activity is finished.

Table 5.3: Labor requirements at each workstation for girder type 1, 2, 3 and 4.

Workstation	Start Time (hour)	End Time (hour)	Labor Type	Labor Quantity
Station 1	0	3	Journeyman	2
	3	8	Journeyman	1
Station 2	0	8	Journeyman	2
Station 3	0	17	Journeyman	1
	17	18	Journeyman	4
Station 4	0	12.5	Journeyman	2
	12.5	15.25	Journeyman	4
Station 5	0	10	Journeyman	4
	10	20.25	Journeyman	2
Station 6	0	11	Journeyman	2
Station 8	0	13	Journeyman	1
	13	13.75	Journeyman	2
Station 9	0	10.5	Sandblast-man	2

Table 5.4: Labor requirements at each workstation for girder type 5 and girder type 6.

Workstation	Start Time (hour)	End Time (hour)	Labor Type	Labor Quantity
Station 1	0	1.5	Journeyman	2
	1.5	4	Journeyman	1
Station 2	0	3	Journeyman	2
Station 3	0	1.25	Journeyman	1
	1.25	2.75	Journeyman	4
Station 4	0	8.5	Journeyman	2
	8.5	10.5	Journeyman	4
Station 5	0	10	Journeyman	4
	10	20.25	Journeyman	2
Station 6	0	11	Journeyman	2
Station 8	0	13	Journeyman	1
	13	13.75	Journeyman	2
Station 9	0	10.5	Sandblast-man	2

Table 5.5: Labor requirements of stiffener welding station (i.e., Station 7) for each type of girder.

Girder Type	Start Time (hour)	End Time (hour)	Journeyman Requirements
Type 1	0	54.1	2
Type 2	0	64.85	2
Type 3	0	41.2	2
Type 4	0	51.95	2
Type 5	0	21.4	2
Type 6	0	27.85	2

In the current case, two particular trades of labor resources (i.e., journeyman and sandblast-man) are identified as the limited resources shared by projects. The studied fabrication shop owns semi-automated welding tools. It is observed that journeymen work on every type of fabrication tasks (e.g., cutting, drilling, fitting, and welding) in the fabrication shop. Owing to the large size of bridge girders sandblast is conducted outside the fabrication shop and by separated laborers (referred to as *sandblast-man*). The fabrication shop runs on a 5-day-per-week calendar and rotates two shifts (total 16 hours per day). Besides labor resources, each workstation is treated as a resource with limited space capacity to handle one girder at one time. The available limit on each type of resources in fabrication operations is listed in Table 5.6.

Table 5.6: Resource limits in the studied fabrication shop.

Resource Type	Limits
Station 1	1
Station 2	1
Station 3	1
Station 4	1
Station 5	1
Station 6	2
Station 7	2
Station 8	2
Station 9	3
Journeyman	8
Sandblast-man	8

The next step is to generate the time-dependent resource requirements for defined macro-activities by utilizing the prepared input data. Detailed workstation-based fabrication activities for each girder line are similar to those shown in Figure 5.7. The resource requirements and duration for each activity are specific to the girder type, as listed in Table 5.3, Table 5.4, and Table 5.5. To generate macro-activities, a resource-constrained scheduling optimization problem is solved. The resource availability given in Table 5.6 is imposed as one type of constraint to generate the macro-activity for one girder line. The other constraint is activity precedence relationships (as defined in Figure 5.7). Decision variables are start times for all fabrication activities of girders on the same girder line. The optimization objective is to minimize (1) total project duration and (2) resource use fluctuation over project duration with the former taking a higher priority. Constraint programming, which is widely utilized to solve construction scheduling problems (Liu and Lu

2018; Liu and Wang 2012; Menesi and Hegazy 2014), is adopted as the optimization algorithm to generate macro-activities. The detailed formulation of macro-activity generation has been given in Section 3.2.1.

By scheduling the detailed fabrication activities on one girder line, the corresponding macro-activity is generated. Figure 5.11 presents the resulting fabrication schedule for the macro-activity of girder line G1 of Bridge 3.

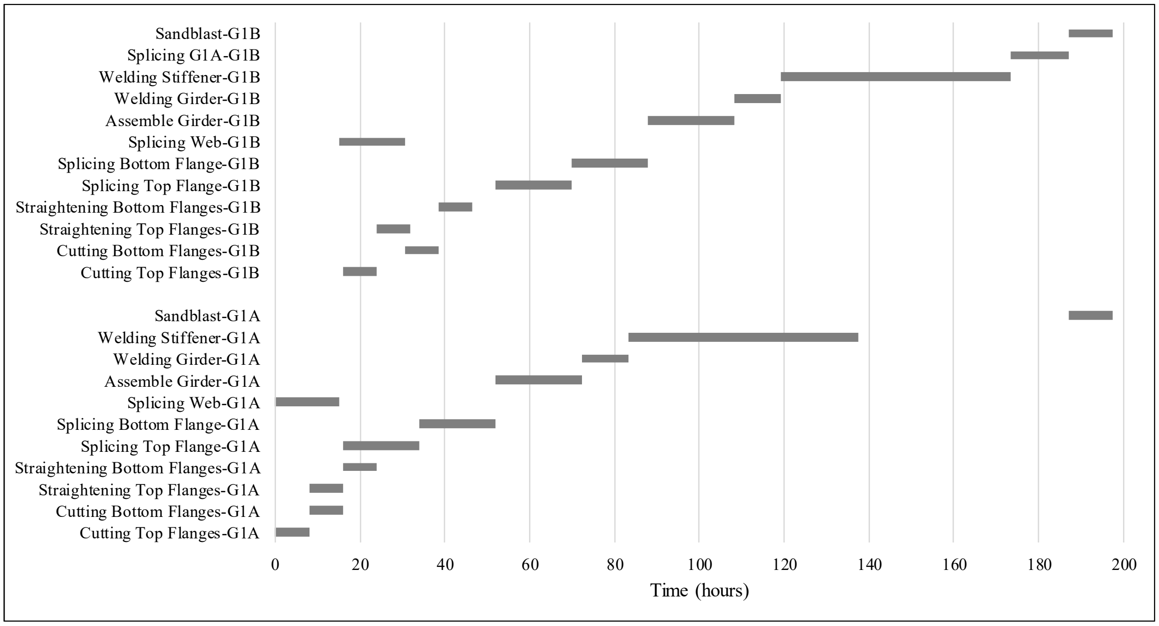


Figure 5.11: Fabrication schedule for girder line G1 of Bridge 3.

Accordingly, the time-dependent resource requirements of the aggregated macro-activity *Bridge3 M1* is generated. The resource requirements on macro activities are further smoothed by following algorithms described in Section 3.2.1. Figure 5.12 shows the resource use profiles for journeymen and sandblast-men on macro-activity “Bridge 3 M1”. The horizontal axis represents the time in hours; the vertical axis is the required number of journeyman/sandblast-man.

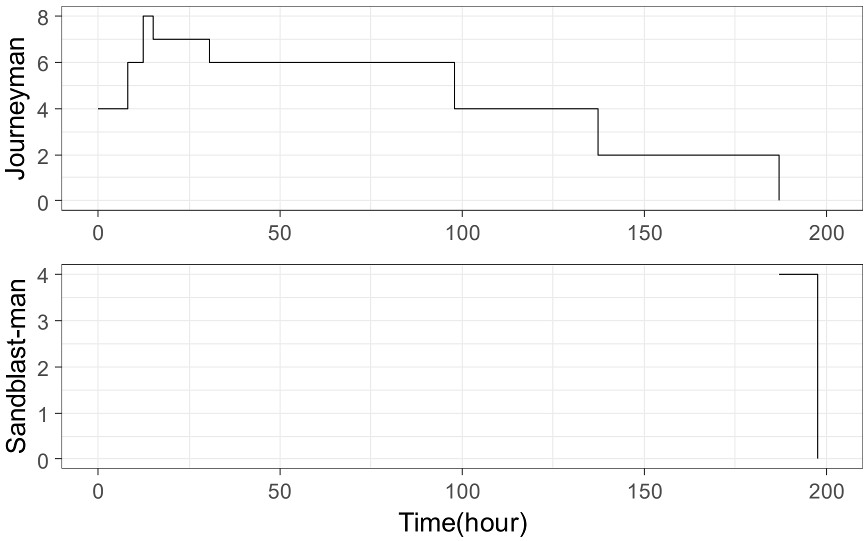


Figure 5.12: Time-dependent labor requirements of macro-activity "*Bridge 3 M1*".

Figure 5.13 and Figure 5.14 present the derived time-dependent requirements of journeyman and sandblast-man respectively for all the aggregated macro-activities in the current case study. Each block in Figure 5.13 and Figure 5.14 corresponds to a macro-activity. For instance, the block "*Bridge1 M1*" represents the defined macro-activity *Bridge1 M1* for girder lines G1 and G3. Each macro-activity can be expanded to a detailed fabrication activity schedule, similar to the one as illustrated in Figure 5.11.

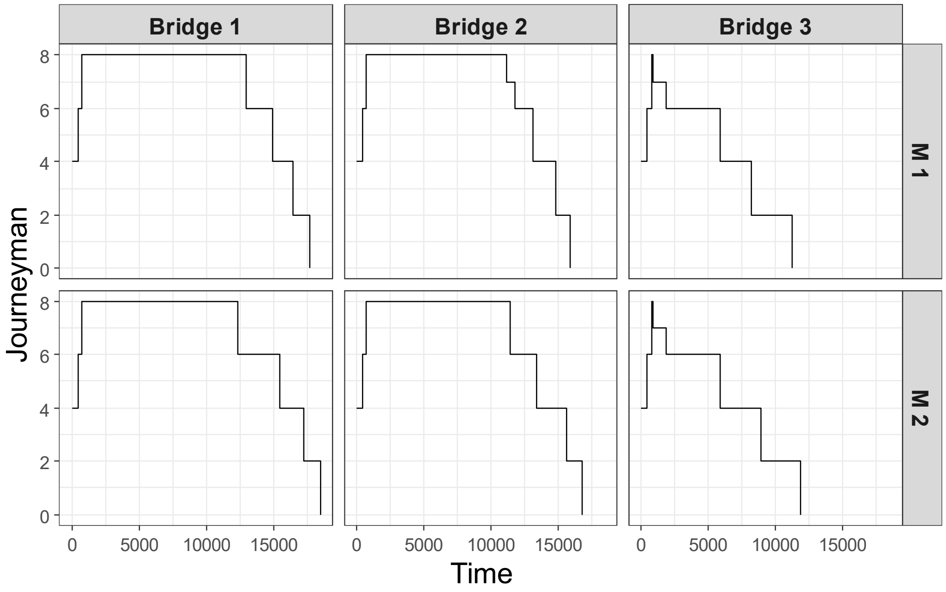


Figure 5.13: Time-dependent requirements of journeymen.

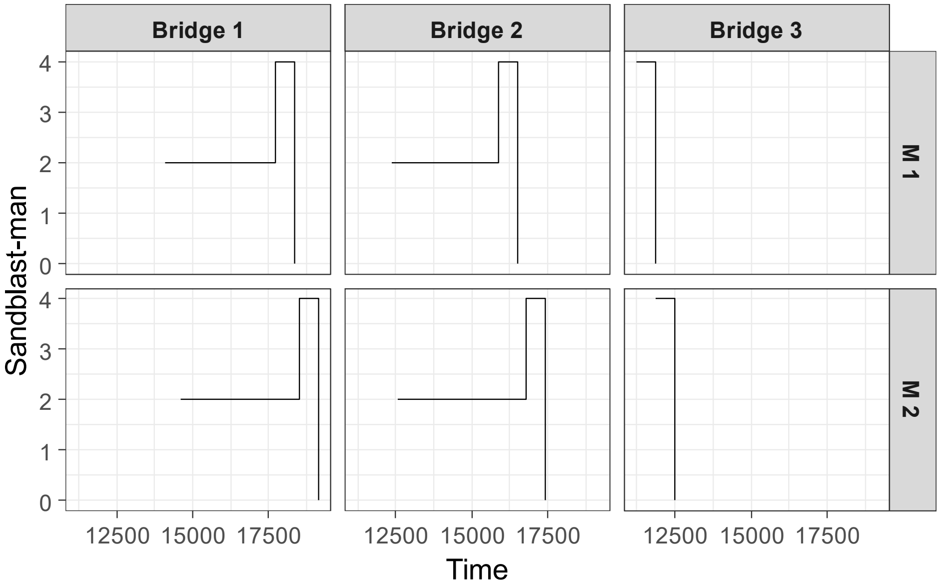


Figure 5.14: Time-dependent requirements of sandblast-men.

5.7 Resource Allocation to Macro-Activities

After generating macro-activities for all the three bridges, the whole set of macro-activities are scheduled for allocating resources available in the fabrication shop to each macro-activity. The girder fabrication is subject to the required-at-site (RAS) dates for installation which are specified by the relevant client in consideration of the bridge site installation schedule. In the current case, the fabrication deadline is the end of the 28th week for Bridge 1, the end of the 32nd week for Bridge 2, and the 7th week for Bridge 3, respectively. The overall objective is to minimize the total project cost, which includes direct cost and indirect cost. Note, the direct cost (i.e., laborer cost) is not a variable in the current case, as only one macro-mode is applicable to each macro-activity and the direct cost is to add up the labor cost of each macro-activity. As such, only indirect costs are considered in defining the cost function in the case study, aimed to optimally allocate resources among the three bridges. Three components are considered in total indirect cost, they are: (1) project indirect cost which is proportional to the duration of an individual project, (2) shop indirect cost which is proportional to the total duration of fabricating all the three bridges, and (3) penalties which is proportional to the delay between RAS deadlines and scheduled finish times on respective projects. Based on the actual rates provided by the partner company, the unit costs are scaled as 1:1:100 on the three cost components in Eq. (5.1). The ratio can be updated when applied to other projects based on actual costs. The objective function is shown as Eq.(5.1).

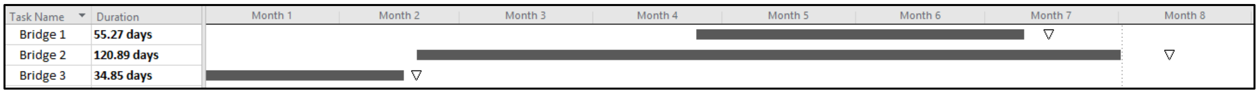
$$\min \text{cost} = \min\left(1 \times Dur_{fab} + \sum_{b\epsilon B} (1 \times Dur_b + 100 \times Delay_b)\right) \tag{5.1}$$

where B is the set of all the bridges being processed. Dur_{fab} is the total duration for completing the three bridges. Dur_b is the duration of Bridge b. $Delay_b$ is the differences between the RAS deadline and the scheduled finish time. If Bridge b is finished before the deadline, then $Delay_b$ equals to 0; early completion bonus or penalty costs are ignored in this case.

It is notable that there are no technologically constrained precedence relationships between macro-activities for fabrication. The total resource availability is the only constraint imposed for sequencing macro-activities. The start time and finish time of each macro-activity are determined by scheduling all the macro-activities subject to total available resources in the fabrication shop.

5.8 Derivation of Schedules for Various Management Functions

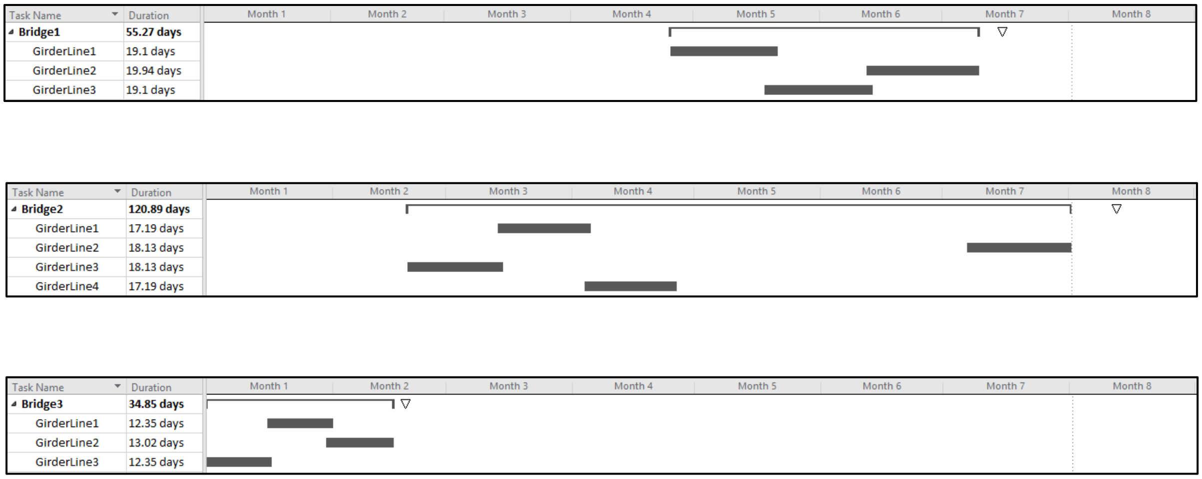
To facilitate the implementation of the proposed framework and the presentation of schedules for various management functions, an MS Project add-on has been developed in collaboration with the partner company for managing multiple concurrent bridge girder fabrication projects. The derived scheduling outcome contains schedules of various details in support of different management functions. In this paper, three schedules are developed and synchronized, namely, program schedule, project schedule, and operation schedule. From the program schedule perspective, Gantt chart showing the start time, finish time, and span for each bridge is provided, as shown in Figure 5.15. The wedge indicates the deadline for a particular bridge. The program schedule provides necessary information for communicating with clients on project delivery dates and assisting in formulating the organization’s business planning strategies, for example, supporting the decision-making process in bidding new projects by integrally accessing resource availability in the current fabrication facility. In this case, the fabrication shop will be ready to work on new projects after Day 157 (i.e., the end of Month 7), when Bridge 2 is finished. In case any new projects is considered for bidding, it can be placed into the project bucket so as to simulate the impact on completion times of the ongoing projects and determine the practically feasible deadline for delivering the newly added project.



Task Name	Duration
Bridge 1	55.27 days
Bridge 2	120.89 days
Bridge 3	34.85 days

Figure 5.15: Program schedule for the three bridges.

In addition, project schedules (as shown in Figure 5.16), which represent the start time and span of each girder line (i.e., macro-activity), are capable of providing decision support for managing milestones of each project in meeting clients' needs. The resulting schedule information can be also useful to arrange for girder shipment to site. Furthermore, it can be seen from Figure 5.16 that resources in the fabrication shop are focused on the processing of one girder line at a time without shifting back and forth among different girder lines in the fabrication process. This significantly reduces excessive interference effect in connection with handling multiple concurring projects while also avoiding the occurrence of crowding on the shop floor. The effectiveness of the proposed framework in mitigating resource dependencies among projects has been thoroughly discussed and elaborately demonstrated in Chapter 4.



Task Name	Duration
Bridge1	55.27 days
GirderLine1	19.1 days
GirderLine2	19.94 days
GirderLine3	19.1 days

Task Name	Duration
Bridge2	120.89 days
GirderLine1	17.19 days
GirderLine2	18.13 days
GirderLine3	18.13 days
GirderLine4	17.19 days

Task Name	Duration
Bridge3	34.85 days
GirderLine1	12.35 days
GirderLine2	13.02 days
GirderLine3	12.35 days

Figure 5.16: Project schedule for the three bridges.

Besides schedules in support of program and project management functions, schedules to guide shop floor operations can also be derived. Figure 5.17 presents the weekly schedule for various workstations (i.e., flange cutting station, flange straightening station, flange splicing station, web splicing station, girder assembly station, and girder welding station) for the first week. It shows that crews work on the fabrication of the third girder line of Bridge 3 in this week. The start time, finish time, and flow of girders going through each workstation are depicted in the weekly schedule, assisting shop superintendent in work planning. Similarly, the monthly schedules for each workstation can be generated to assist in identifying critical constraints and removing production bottlenecks in advance.

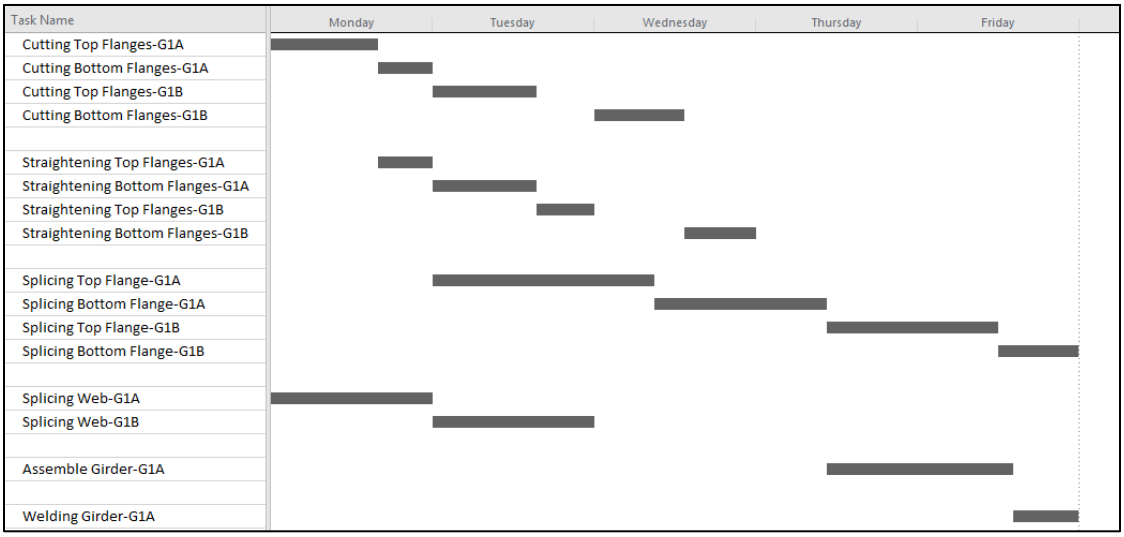


Figure 5.17: Operation schedule for the first week.

In reality, lack of effective communication and the fragmented nature of current scheduling practice make it difficult for superintendents to make effective and efficient production schedules at the workface of the fabrication facility in line with goals and constraints in project and program schedules (Han et al. 2007; Shokri et al. 2015). With the newly proposed scheduling framework and the computer tool, the operation schedule derived from the proposed framework is seamlessly interconnected with project and program schedules. Given any variations or changes in the actual operation process, it is straightforward to evaluate their impact on project and overall program objectives. For the current case study, an application scenario is simulated as follows to illustrate how to update the derived schedules based on actual progress.

5.8.1 Application Scenario

On Thursday of the first week, the superintendent performs the routine progress checking in the morning and identifies that the activity "Assemble Girder-G1A" will be delayed for 2 days due to equipment maintenance at the girder assembly station. Figure 5.18 shows both the updated and original production schedule for the first week. The black bars represent the original production schedule, while the grey bars denote the updated one. It can be seen the duration of activity "Assemble Girder-G1A" is prolonged and the subsequent activity "Welding girder-G1A" will be delayed.

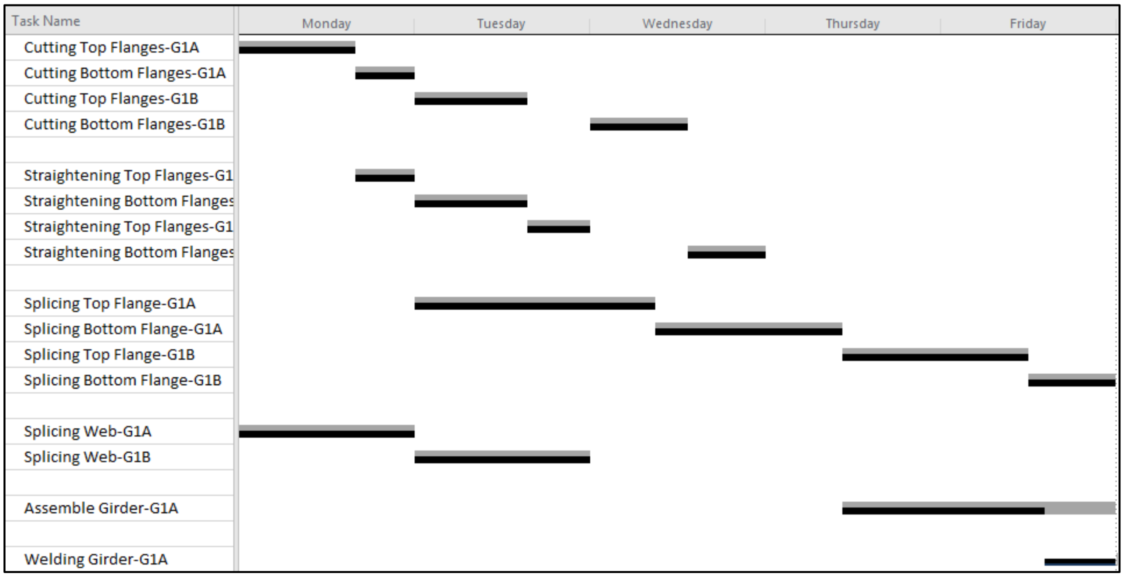
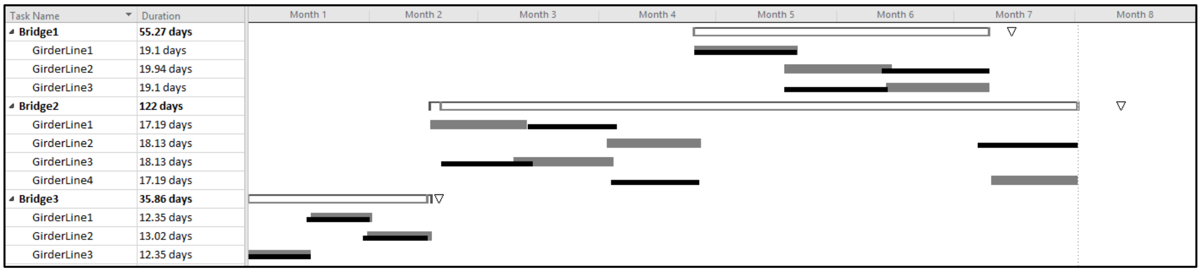


Figure 5.18: Updated operation schedule for the first week.

Once the shop superintendent confirms the actual progress, the computer tool is employed to re-allocate resources to each macro-activity (i.e., girder line) in order to mitigate the negative impact of the progress delay on project delivery date. The fabrication work planned for the coming four weeks is kept unchanged as the preparation work already has started. So the fabrication sequences for Bridge 3 girder line G3 and G1 are fixed as per the original schedule. Only the fabrication sequences for the remaining girder lines that have yet to start are adjusted in updating the fabrication schedule. It is also assumed when updating the fabrication schedule, productivity on remaining activities is not influenced by the progress delay and remains the same as that in the original schedule.

The optimization functions of the proposed scheduling framework have been fully automated in *IBM ILOG CPLEX Optimization Studio* and further integrated with the MS Project add-on. On a desktop with a 3.2 GHz CPU and 8 GB random-access memory (RAM), the schedule was re-optimized in 316 seconds CPU time. Figure 5.19 presents the updated Gantt chart for the three bridges. The grey bars show the updated fabrication sequences. Compared to the original schedule indicated by the black bars, the fabrication sequence of some girder lines (e.g., Bridge 2 girder line 1 and 3) is adjusted in order to mitigate the effect of the progress delay. After the adjustment, Bridge 1 and Bridge 2 are still finished at the same time (i.e., 133440th min and 145926th min) as in the original plans. Only the finish time of Bridge 3 is marginally delayed by 13.8 hours (i.e., from 34428th mins to 33600th min).



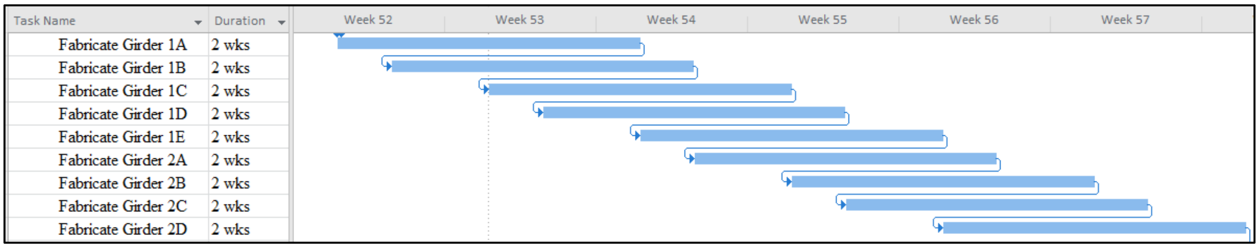
Task Name	Duration
Bridge1	**55.27 days**
GirderLine1	19.1 days
GirderLine2	19.94 days
GirderLine3	19.1 days
Bridge2	**122 days**
GirderLine1	17.19 days
GirderLine2	18.13 days
GirderLine3	18.13 days
GirderLine4	17.19 days
Bridge3	**35.86 days**
GirderLine1	12.35 days
GirderLine2	13.02 days
GirderLine3	12.35 days

Figure 5.19: Updated Gantt chart for the three bridges.

The imposed delay factor in actual progress for this demonstration is trivial, exerting limited impact on the individual project schedule and overall program schedule. Nonetheless, on actual projects in the real world, production delays are cumulative over time. If the impacts of these small deviations are not mitigated in time, they would soon spiral into severe schedule slippage and cost overrun. With the proposed framework and the developed computer tool, the impact of the delay in actual progress can be analyzed just in time and the schedules can be updated quickly. In a similar way, if the deadline for a certain project changes (i.e. RAS), the impact on schedules for other management functions can be readily analyzed and mitigated by optimization, significantly facilitating information sharing, improving communication efficiency, and enhancing project management performance

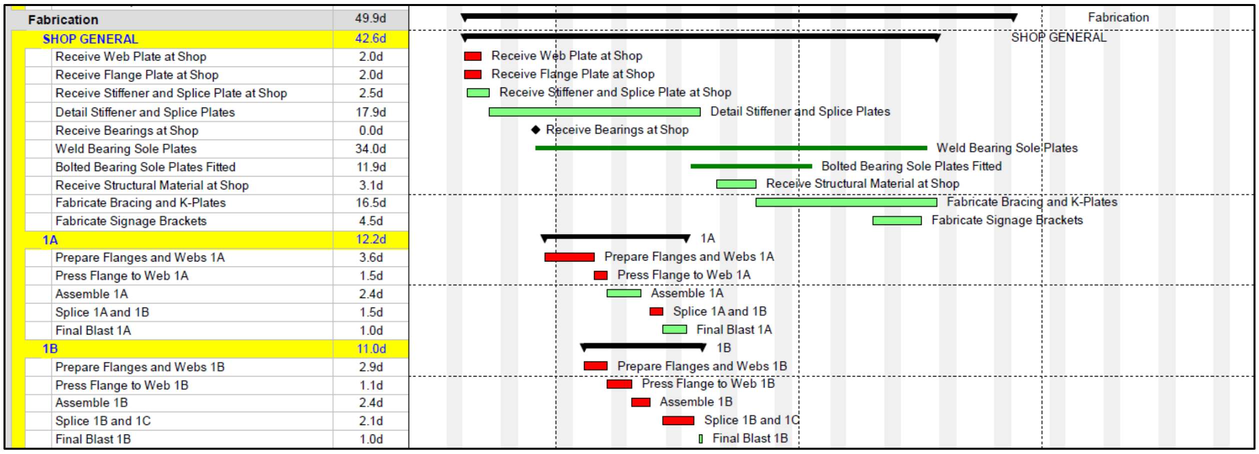
5.9 Discussion

Given the same projects, schedules for different management purposes are developed and maintained by different function managers in the partner company. As illustrated in Figure 5.20 and Figure 5.21, the execution plan prepared in the bidding stage is developed by a project manager in MS Project, while more detailed fabrication schedules for project control are established by a senior scheduler in Primavera P6. As a result, schedules are saved in separated files in different format and information does not flow seamlessly between different management functions. To exchange information and achieve synchronization, intensive communication is required and tedious manual efforts in schedule maintenance are necessary. In reality, schedules will become obsolete soon and hence are not taken seriously. In consequence, much of the scheduling effort is deemed wasteful and non-value-adding.



Task Name	Duration
Fabricate Girder 1A	2 wks
Fabricate Girder 1B	2 wks
Fabricate Girder 1C	2 wks
Fabricate Girder 1D	2 wks
Fabricate Girder 1E	2 wks
Fabricate Girder 2A	2 wks
Fabricate Girder 2B	2 wks
Fabricate Girder 2C	2 wks
Fabricate Girder 2D	2 wks

Figure 5.20: Execution plan included in bidding documents.



Fabrication	49.9d
SHOP GENERAL	42.6d
Receive Web Plate at Shop	2.0d
Receive Flange Plate at Shop	2.0d
Receive Stiffener and Splice Plate at Shop	2.5d
Detail Stiffener and Splice Plates	17.9d
Receive Bearings at Shop	0.0d
Weld Bearing Sole Plates	34.0d
Bolted Bearing Sole Plates Fitted	11.9d
Receive Structural Material at Shop	3.1d
Fabricate Bracing and K-Plates	16.5d
Fabricate Signage Brackets	4.5d
1A	12.2d
Prepare Flanges and Webs 1A	3.6d
Press Flange to Web 1A	1.5d
Assemble 1A	2.4d
Splice 1A and 1B	1.5d
Final Blast 1A	1.0d
1B	11.0d
Prepare Flanges and Webs 1B	2.9d
Press Flange to Web 1B	1.1d
Assemble 1B	2.4d
Splice 1B and 1C	2.1d
Final Blast 1B	1.0d

Figure 5.21: Schedule developed for project control.

The dual-level multi-project scheduling framework illustrated in the presented case study is proven to be effective to synchronize schedules of various details for different management functions and avoid significant schedule slippage and cost overrun in project execution. With the MS Project add-on developed to facilitate information sharing and schedule optimization, actual progress in the fabrication facility can be taken into account in plan adjustment and schedule updating on a timely basis.

5.10 Conclusion

In this chapter, an application of the proposed dual-level multi-project scheduling framework is presented to improve the synchronization among schedules for different management functions in managing multiple concurring projects at a fabrication facility. In collaboration with the partner company, a new WBS is first developed by integrating two existing WBSs and macro-activities are defined. The work content in association with girder lines of a bridge is determined in scoping a macro-activity. After allocating resources to defined macro-activities, three types of schedules (i.e., program, project, operation schedule) are derived. An integrated program schedule including all projects is instrumental in resolving resource conflicts among projects and providing support for decision-making in the company's business development strategies, such as bidding new projects. Schedules for individual projects are also available to provide support in managing project milestones. Operation schedules are generated to guide work planning in the fabrication facility. Furthermore, schedules for different management functions are linked with one another through computer automation in the case study. The optimization functions of the proposed scheduling framework have been fully automated in *IBM ILOG CPLEX Optimization Studio* and further integrated with the MS Project add-on, which has been adopted in the partner company for scheduling bridge girder fabrication projects. The individual project objectives are better aligned with the company strategic goals, while actual operation progress in the fabrication shop can be tracked and their impact on individual project milestones and business strategic goals can be analyzed in a timely fashion.

6 CHAPTER 6: OPTIMIZING PROJECT SCHEDULES UNDER MATERIALS LOGISTICS AND CREW AVAILABILITY CONSTRAINTS

6.1 Introduction

The material management performance is a crucial factor in determining project delivery success (Tserng et al. 2006). Poor material management increased the project duration by 50% to 130% (Thomas and Sanvido 2000) and was identified as one of the common factors that accounted for lower productivity (Thomas et al. 2005; Yi and Chan 2013). It was reported that about 30% of labor productivity could be lost due to the shortage of materials when they were needed (Caldas et al. 2014).

Construction-applied materials, which remain as part of the facility being built after they are handled and placed in the field, are categorized as *permanent resources* in project management (Tatum 2012). Two types of construction-applied materials can be distinguished, namely: *bulk materials* and *engineered materials.* The bulk materials are usually manufactured in large quantities, which are commonly available in stocks of a supplier and can be procured in a short lead time; examples include rebars in stock lengths, masonry concrete blocks, and ready-mix concrete. In contrast, engineered materials are prefabricated in specialized fabrication facilities according to custom designs catering for specific project needs; the fabrication process requires special engineering expertise, resources, and a relatively long lead time, as exemplified by the fabrication of structural steel beams, process pipe spools, and special equipment (Tatum 2012).

The current practice is to procure all types of materials based on predicted delivery times as per a predefined project baseline schedule. Suppliers are expected to deliver materials to meet construction project schedules, however, the timely delivery is always affected by many factors, such as availability of special materials, suppliers' production resource planning, and logistics-related delays (Ala-Risku and Kärkkäinen 2006; Angkiriwang et al. 2014; Horvath 2004; Patil et al. 2012). Especially for engineered materials, material procurement and logistics processes are more complicated (Choi and Song 2014; Safa et al. 2014), subject to numerous constraints such as limited production capacity of suppliers, delayed issuance of shipping permits, and unavailability of handling and delivery resources in the material supply chain. In the construction industry, crew operation planning and material procurement planning are managed separately. This division also occurs in academic studies in general: project scheduling-related research and supply chain management-related research have seldom been investigated in an integrative fashion (Xu et al. 2016). However, considering the current trend of adopting off-site prefabrication and modular construction technologies on modern construction projects (Haas 2000; O'Connor et al. 2014), it is imperative to develop an analytical methodology that factors material supply constraints into the resource-constrained scheduling optimization so as to address the impact of uncertainties in material deliveries on project schedule and cost performance.

Previous research on the resource-constrained scheduling optimization has coped with complicated constraints on limited crew resources. The resulting schedule features the optimal crew configuration and work sequence leading to a lower project budget and shorter project duration. Nonetheless, the underlying assumption of the just-in-time delivery of materials may not hold in real-world projects. Therefore, a gap is identified in the body of knowledge related to scheduling: how to analytically evaluate the impact of uncertainties associated with the delivery and inventory of materials in the resource-constrained scheduling optimization. Hence, this chapter aims to address the impact of uncertainties inherent in materials logistics on the project schedule in two steps, as shown in Figure 6.1. The first step is to develop a mathematical model to minimize the total project cost, subject to constraints relevant to project schedules including crew limits, technical precedence relationships, predefined deadline, multiple activity modes, and material logistics. The second step is to take advantage of the valid optimization model for evaluating the impact of different input settings on material logistics on the project cost. In particular, the impact of uncertain material deliveries upon the project schedule is assessed in an analytical way. The lower level of the proposed dual-level framework (Section 3.2.2) provides the desired mathematical model for analyzing different material delivery settings to reveal its impact on the project schedule. In this chapter, the delivery date of a particular material is singled out as the risk factor of interest to illustrate the application of the proposed approach by two case studies.

The remainder of this chapter is organized as follows. The mathematical model presented in Section 3.2.2 is first illustrated with a sample project and validated through an independent simulation model. After the mathematical model is validated, different material delivery dates for one critical material are simulated by iteratively executing the mathematical model, resulting in the characterization of their impact on the total project cost. Further, a bridge girder fabrication project is presented for demonstrating the applicability of the mathematical model on large-scale project networks. At last, contributions and future extensions of this research are presented in a discussion section before conclusions are drawn.

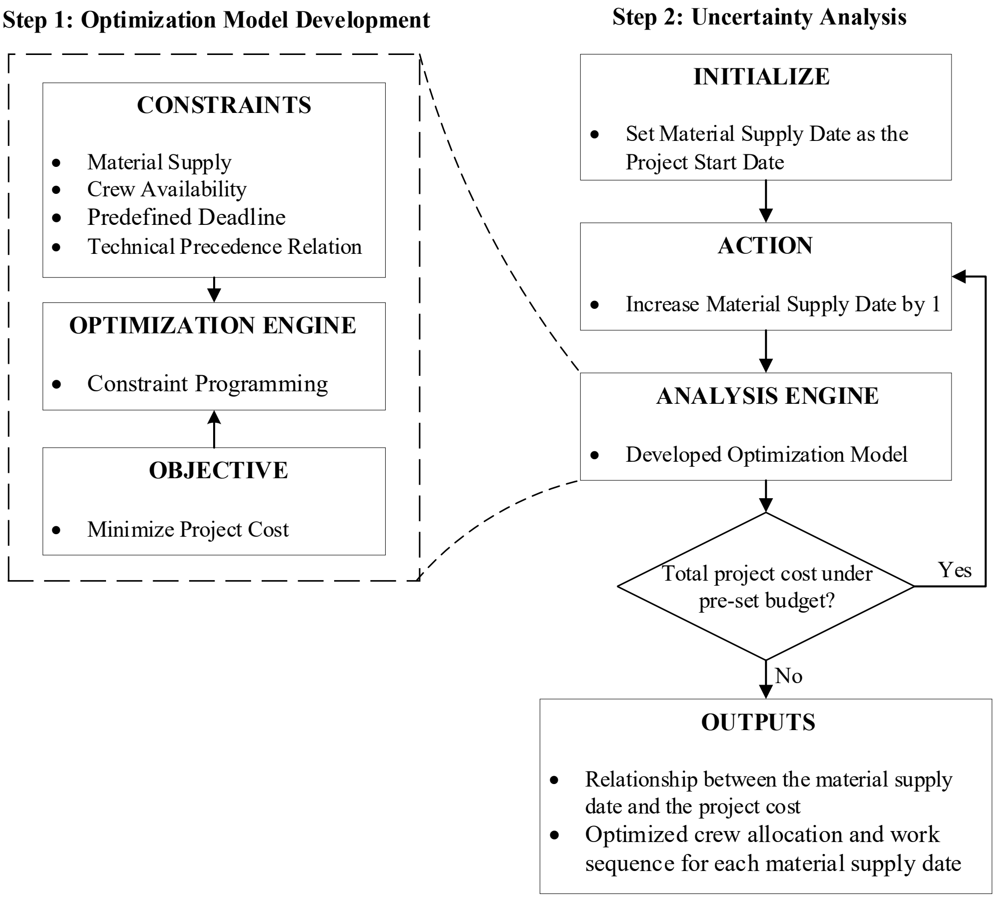


Figure 6.1: Research methodology flow chart.

6.2 Case Study: Ten-Activity Sample Project

The data for the case project including activity precedence relationships (Table 6.1), crew methods (i.e., modes) for each activity (Table 6.2), and resource limits per day (i.e., 30 workers per day) is taken from Chen and Weng (2009). In order to illustrate the definition of engineered material involved fabrication project scheduling problems, material types and quantities required by each activity are added to the case problem, as listed in Table 6.1. In this case, it is assumed that three types of engineered materials M1, M2, and M3 are required. The project indicates a strict 60-day deadline; a laborer costs $50 per day, idling material costs $2 per unit per day for M1, $2 for M2, and $3 for M3 respectively. A baseline schedule with the least laborer cost is developed without considering the material supply information, as shown in Figure 6.2. The baseline schedule has a 52-day project duration. The selected crew method for each activity is also indicated by the enclosed value next to the activity ID in Figure 6.2. For instance, “1 (method 1)” represents that on Activity 1, Crew Method 1 is selected.

Table 6.1: Predecessor and required material for each activity.

Activity ID	**Predecessor**	**Material**	
		Type	Quantity
1	-	(M1, M2)	(2, 1)
2	-	(M2)	(2)
3	-	(M1, M3)	(1, 3)
4	1	(M1)	(4)
5	2	(M1, M2, M3)	(2, 5, 3)
6	3	(M2, M3)	(5, 4)
7	3	(M2)	(2)
8	5, 6	(M2)	(7)
9	7	(M3)	(3)
10	4, 8, 9	-	-

Table 6.2: Construction methods of each activity.

Activity ID	**Crew Method 1**		**Crew Method 2**		**Crew Method 3**		**Crew Method 4**	
	Duration (days)	Labor	Duration (days)	Labor	Duration (days)	Labor	Duration (days)	Labor
1	5	15	-	-	-	-	-	-
2	4	16	6	10	8	7	9	6
3	6	13	8	9	10	7	-	-
4	12	16	15	10	18	8	-	-
5	22	18	24	16	26	14	28	12
6	14	20	18	15	24	8	-	-
7	9	17	10	14	-	-	-	-
8	14	7	15	6	16	4	-	-
9	15	5	18	4	20	3	-	-
10	3	4	5	2	-	-	-	-

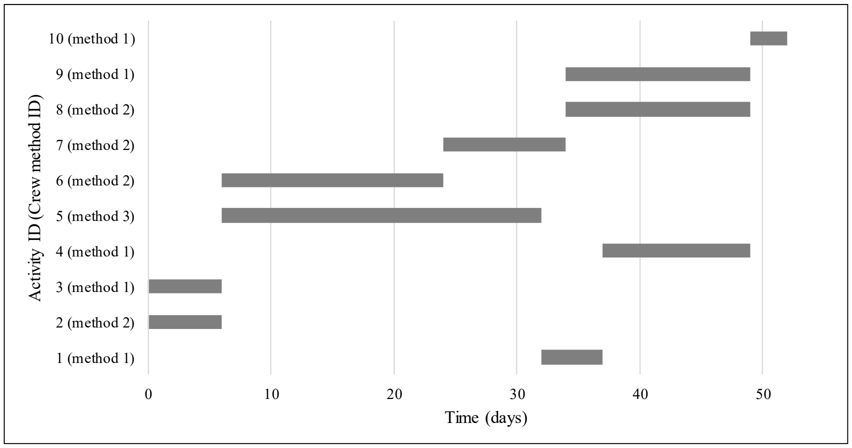


Figure 6.2: Baseline schedule for the project.

Subsequently, detailed material supply is determined based on the material demand of the baseline schedule. 5 units M1, 15 units M2, and 10 units M3 are in stock at the start of the project so to ensure a smooth project start. 4 units M1, 7 units M2, and 3 units M3 are to be delivered by respective fabricators and arrive on Day 30 which caters to the remaining activities in the project baseline schedule. Figure 6.3 shows the relationship between the as-scheduled material demand vs. the as-planned material supply for three types of materials respectively, indicating a proper match between material supply and material demand as per the baseline schedule.

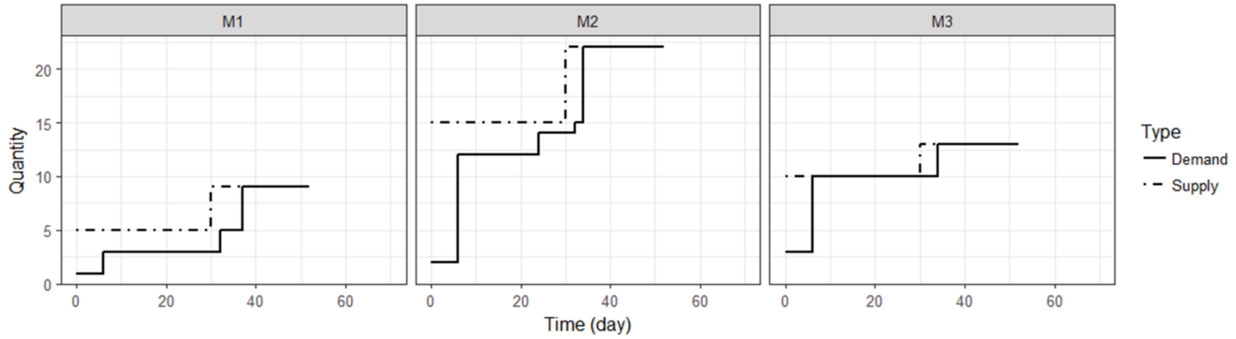


Figure 6.3: Planned cumulative material supply and cumulative material demand chart.

In this case, the total cost as per Eq.(3.5) is calculated as follows.

The labor cost:

$$C_{WH} = T \times \sum_{r} u_r \times W_r = 52 \text{ days} \times \$50/(work \cdot day) \times 30\, workers \ = \$78{,}000$$

Taking M1 for example to calculate the material inventory cost: M1 has 5 units in the inventory on site at the beginning of the project, so $I_{01} = 5$. The inventory cost related to the initial inventory from the project start to the project end $I_{01} \times T$ equals to $5\, units \times 52\, days = 260\, units \cdot days$; M1 has one additional supply of 4 units at Day 30, so the corresponding storage cost from the supply date to the project end is:

$$\sum_{t=0}^{T}(T-t)S_{t1} = (52\,days - 30\,days) \times 4\,units = 88\,units \cdot days$$

For the material consumption, Activity 1, Activity 3, Activity 4, and Activity 5 require 2 units, 1 unit, 4 units, and 2 units of M1 respectively. And Activity 1, Activity 3, Activity 4 and Activity 5 start at Day 32, Day 0, Day 37, and Day 6 respectively; thus, the inventory cost which does not materialize is:

$$\begin{aligned}\sum_{\theta=0}^{T}(T-\theta)\sum_{i\in V} q_{i1}\sum_{j\in M_i} x_{ij\theta} \\ &= (52\,days - 32\,days) \times 2\,units + (52\,days - 0\,days) \times 1\,units \\ &+ (52\,days - 37\,days) \times 4\,units + (52\,days - 6\,days) \times 2\,units \\ &= 244\,units \cdot days\end{aligned}$$

Therefore, the inventory cost related to M1 equals to the cost of the initial inventory adding that of the additional supply (at Day 30) and subtracting the inventory cost related to the consumed materials.

$$\begin{aligned}C_{Idlem} = u_1 \times \left(I_{01} \times T + \sum_{t=0}^{T}(T-t)S_{t1} - \sum_{\theta=0}^{T}(T-\theta)\sum_{i\in V} q_{i1}\sum_{j\in M_i} x_{ij\theta} \right) \\ &= \$2/(units \\ &\cdot days) \times (260\,units \cdot days + 88\,units \cdot days - 244\,units \cdot days) = \$208\end{aligned}$$

Similarly, the material inventory cost related to M2 and M3 is calculated as \$620 and \$162 respectively. The total cost as defined in Eq.(3.5) related to the baseline schedule is

$$TC = C_{WH} + C_{Idlem} = \$78{,}000 + \$208 + \$620 + \$162 = \$78{,}990$$

Soon, the material procurement manager updates on the shipment of 7 units of M2 to be delayed from Day 30 to Day 40 due to the extra time required for confirming the shipping permit, while the shipments of M1 and M3 stick to the original plan. Figure 6.4 shows the resulting changes in the cumulative material supply and the cumulative material demand based on the baseline schedule. It is obvious that there is a material shortage on M2 from Day 34 to Day 40 (the grayed portion on M2 in Figure 6.4); in consequence, Activity 8 cannot be started as per the baseline schedule. Therefore, it is necessary to update the plan in order to bring material demand in line with the updated information on material supply.

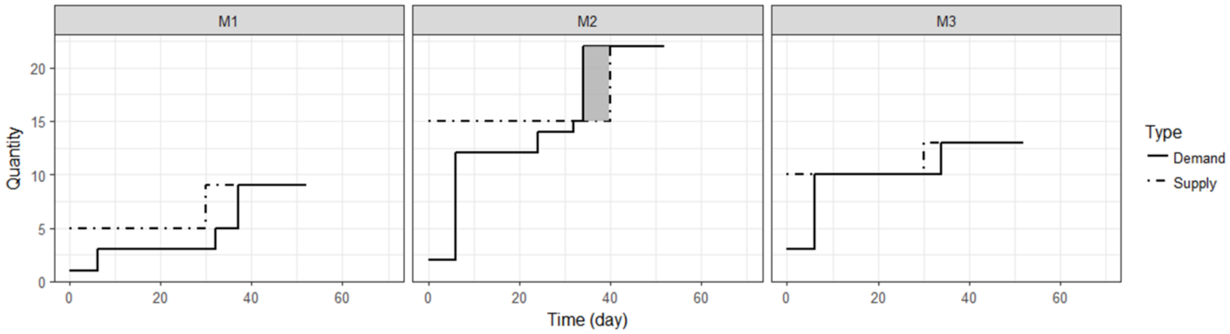


Figure 6.4: Actual cumulative material supply and cumulative material demand chart.

6.2.1 Project Planning Scenario

In this scenario, all activities' execution time can be adjusted to match material demand with material supply. The limit of the crew resource remains $W = (30)$, the initial inventory amounts for three types of engineered materials remain $I_0 = (5, 15, 10)$. The supply date for 7 units M2 is updated to Day 40, while the supply dates for 4 units M1 and 3 units M3 remain unchanged on Day 30, so $S_{30,1} = 4, S_{40,2} = 7, S_{30,3} = 3$. The unit cost of crew resource remains $u_r = (50)$ per laborer every day. The unit cost of idling engineered material is $u_n = (2, 2, 3)$ per unit per day for M1, M2, M3, respectively. Table 6.1 defines the activity set V, the predecessor set P, and the required amount q_{in} for each engineered material n for activity i. Table 6.2 defines the crew method set M_i with the duration D_{ij} and crew requirement w_{ijr} for each mode j of activity i.

Entering all the updated inputs to the proposed optimization model as presented in Eq. (3.5) - (3.10), an alternative plan is derived as shown in Figure 6.5. The project duration is extended to 57 days due to the delay of M2 supply. Compared to the baseline schedule, the execution sequence of some activities (e.g., Activity 1 & Activity 4) is adjusted, while the crew methods on particular activities differ from the default options in the baseline schedule. For example, the crew method of Activity 8 is changed from Method 2 (15 days, 6 laborers) to Method 1 (14 days, 7 laborers). In consequence, the material demand pattern is updated and matched up with the current material supply pattern, leading to no material shortage on the three types of materials along the project duration, as shown in Figure 6.6. Meanwhile, the total cost as defined in Eq. (3.5) changes to \$86,010, increasing by 8.89% against the baseline schedule.

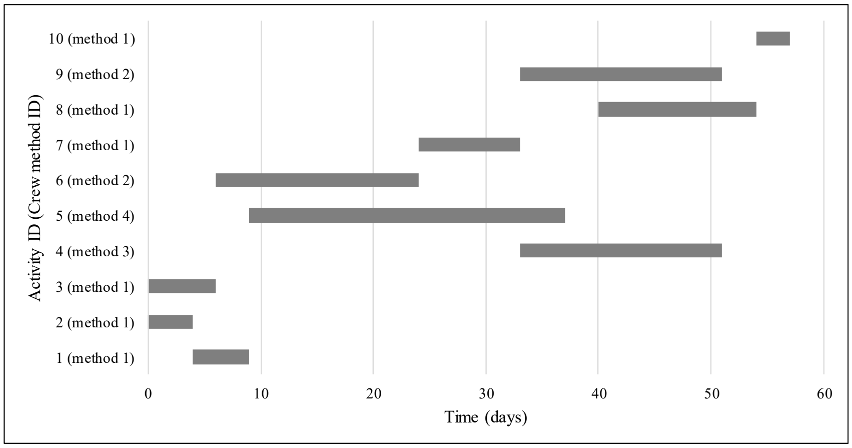


Figure 6.5: Updated schedule from project planning perspective when material M2 is delayed to Day 40.

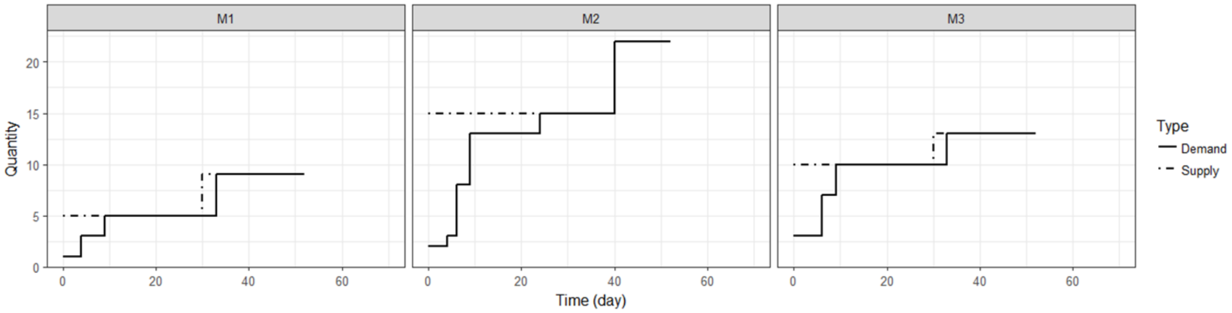


Figure 6.6: Material supply-demand chart: project planning perspective.

6.2.2 Project Control Scenario

In this application scenario, the material delivery delay takes place and is reported during project execution. The crew methods and starting times for those activities, which have already started, should be kept unchanged. Assuming the same information on material delays as given in the previous scenario is confirmed on Day 15; the start times and crew methods on ongoing activities (i.e. Activity 2, Activity 3, Activity 5 and Activity 6) remain the same as in the baseline schedule. The proposed optimization model is applied to re-plan the execution sequence of the remaining activities in the project in order to rebalance material demand and material supply. The optimized schedule result is updated, as shown in Figure 6.7. It is noteworthy the crew method of Activity 8 is adjusted, while the execution sequence for Activity 1 and Activity 4 differs from the baseline schedule. The optimized schedule is with 57-day project duration, which remains the same as the previous scenario. Nonetheless, the total cost defined as in Eq. (3.5) increases by 0.20% to $86,181 against the previous scenario and by 9.10% against the baseline schedule as a result of incurring extra material inventory cost. Figure 6.8 also shows the updated material supply-demand patterns over project duration for this scenario. It is noted the material demand line is always under the material supply line across three different material types.

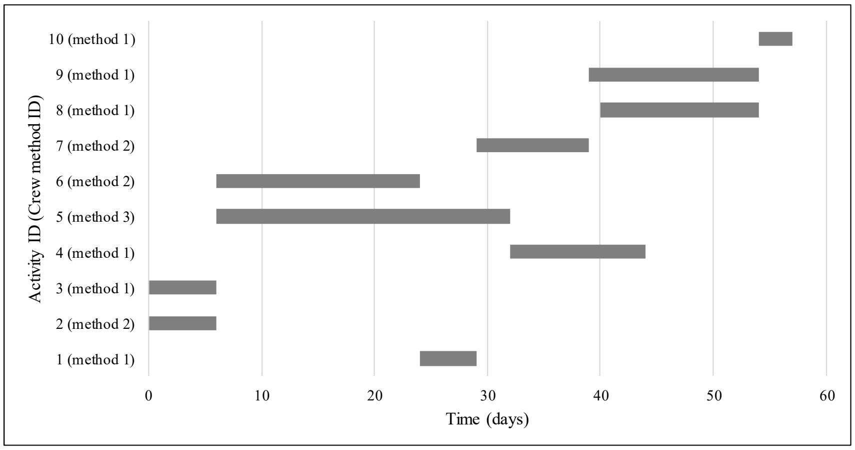


Figure 6.7: Updated schedule from project control perspective when material M2 is delayed to Day 40.

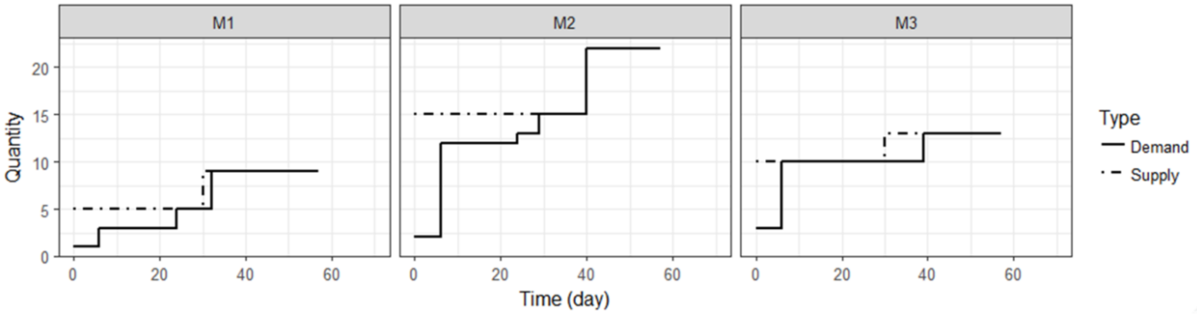


Figure 6.8: Material supply-demand chart: project control perspective.

6.2.3 Cross-Validation by Simulation

As a benchmark problem with the known optimum solution is not available for the defined problem, it is difficult to verify the optimality of the solution given by the proposed optimization model. Also, for construction projects, the feasibility of one schedule is more important than the optimality. Therefore, this section focuses on the validation of the feasibility of the solution obtained in the first case study. In the following case study, the feasibility and validity of the generated schedule by the proposed approach were confirmed through the face validation of experts.

In this section, a simulation model was independently developed based on the simplified discrete event simulation approach (SDESA) (Lu 2003) in order to validate the feasibility of the optimization result. SDESA simply serves as a virtual platform for executing the optimized project schedule on the case study. Interested readers can refer to Lu and Wong (2005) and Lu et al. (2008) for more details on modeling techniques, computer platform, and practical applications of SDESA.

SDESA distinguishes reusable resources and disposable resources. The reusable resource is used to model commonly seen crew resource such as laborers and equipment, while disposable resources are utilized to represent intermediate products or command units that are generated by one activity and required by another. Disposable resources can be used to effectively model the interdependent relationships among various activities or processes.

On the project planning scenario, the resulting crew method of each activity in terms of the duration and the required number of laborers were specified in the SDESA simulation model. Figure 6.9 shows the activity bar chart for the project planning scenario schedule generated by SDESA, which is identical to the optimized schedule obtained from solving the mathematical formulation (as shown in Figure 6.5). Further, the activity bar chart in Figure 6.9 is elaborated in Figure 6.10 to show the detailed laborer resource allocation on each activity. The simulation results represent the schedule resulting from constraint programming optimization in an intuitive, transparent fashion. All the precedence relationships, daily labor resource limit, and daily material availability limits are satisfied. As such, the feasibility of the optimization result is cross-validated. In a similar way, the feasibility of the optimized schedule obtained in the project control scenario is cross-validated but not presented herein due to the size limit.

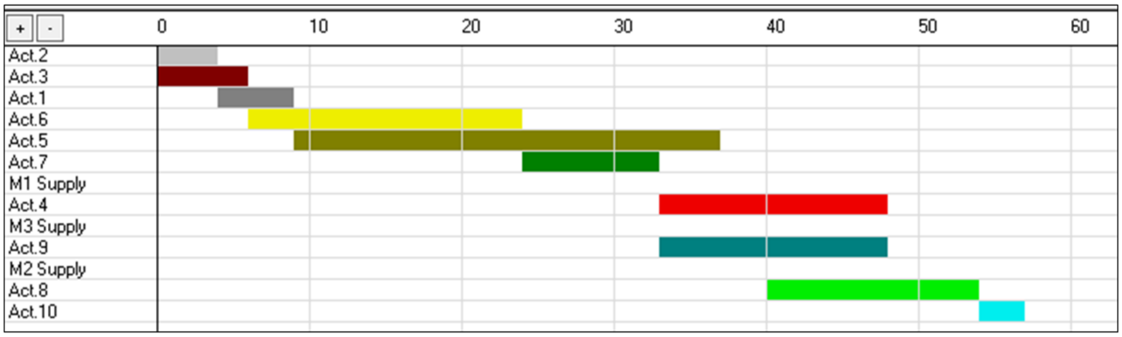


Figure 6.9: Simulated schedule by SDESA for the project planning scenario.

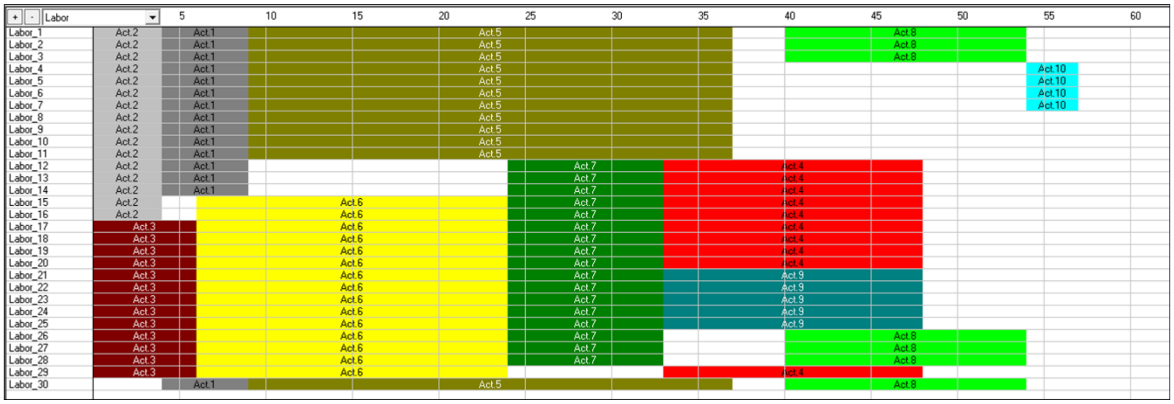


Figure 6.10: Resource allocation chart in SDESA for the project planning scenario.

6.2.4 Budget-Constrained Material Delivery Time Window

Based on the optimization model, the impact of the delivery date for M2 on the total project cost is analyzed to derive the relationship between the material delivery date and the total cost, while the delivery dates for M1 and M3 are fixed on Day 30. The delivery dates for M2 are postponed from the beginning of the project until the total project duration reaches the deadline Day 60. The optimization model was executed for each input setting of the project. The derived relationship between the delivery date for M2 and the total project cost is shown in Figure 6.11. Each point represents a particular optimal solution, which has a detailed schedule for the corresponding input setting. The optimal project duration in connection with each delivery date of M2 is distinguished with points in different shapes. The relationship between the project duration and M2's delivery dates is shown in Figure 6.12. As illustrated in Figure 6.12, deliveries before Day 35 has no impact on the project duration. If only considering the project duration constraint, M2 has 35 days float in keeping the project total duration as 52 days. When considering a pre-set budget, the budget-constrained material delivery time window for M2 can be determined from Figure 6.11. All points under the pre-set budget are allowable delivery times for M2. For instance, given the maximum project budget of $87,000, the allowable delivery time window for M2 ranges from Day 0 to 40. It can provide decision support for project managers for determining the material receiving time to make the total cost under control.

Figure 6.11: Relationship between the delivery date of M2 and the project total cost with negligible material inventory cost.

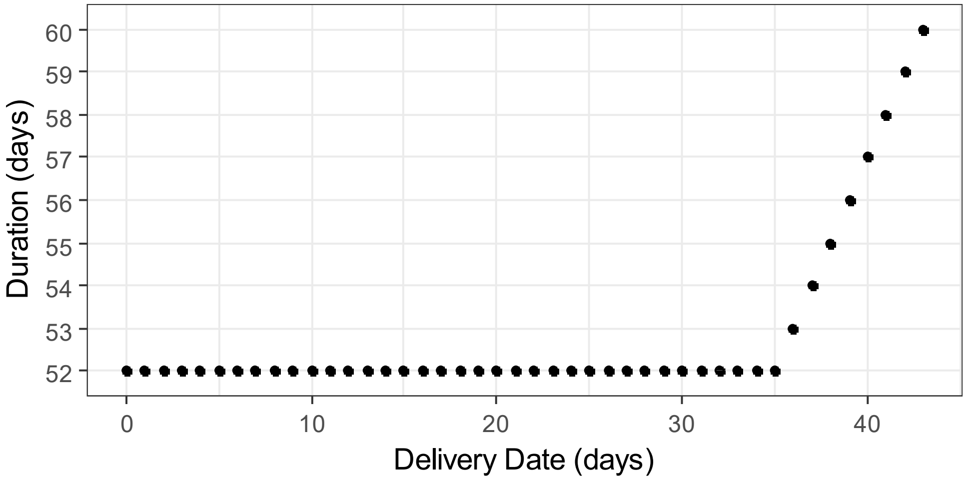


Figure 6.12: Relationship between the delivery date of M2 and the project total cost with negligible material inventory cost.

For this case study, the material inventory cost is relatively low compared to the labor cost. Therefore, any cost saving due to material inventory is limited in the delivery time window from Day 0 to Day 35, which leads to the small slope on the left side of the optimal delivery time (i.e. Day 35). But for projects whose material inventory cost is comparable to or higher than the labor cost, the shape of the relationships between M2 delivery date and the total cost will change. If the material inventory cost for M1, M2, and M3 is changed to (40, 40, 50) per unit per day which is comparable to the labor cost ($50 per labor per day), the relationships between delivery date of M2 and the project total cost is shown in Figure 6.13. Similarly, if the material inventory cost is much higher than the labor cost, such as (800, 800, 1000) for (M1, M2, M3) per unit per day respectively, the derived curve between M2 delivery date and the total cost is presented in Figure 6.14. After setting the budget limit, the budget-constrained material delivery time window can be determined for each case. The relative ratio between labor rate and material rate changes not only the shape of the relationship between the delivery date of M2 and the resulting total cost, but also the material delivery time windows. It is obvious that for projects with low material inventory costs, delivering materials as early as possible is advisable; while for projects with significant material inventory cost, delivering materials earlier than needed would lead to undesired increments of the total cost. For applying this proposed optimization model in practice, the actual unit rates on material storage and labor usage can be utilized to simulate the relationship between material delivery times and the total project cost and determine the allowable delivery time window corresponding to a certain budget limit.

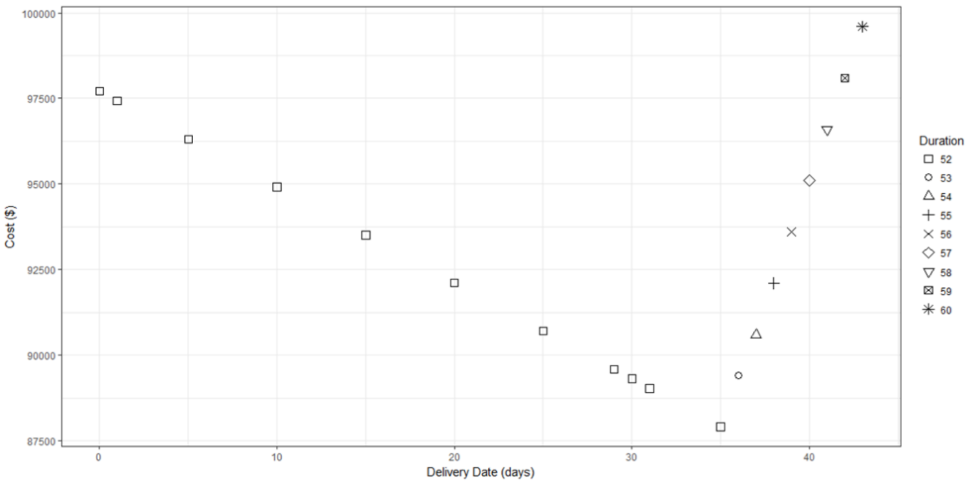


Figure 6.13: Relationship between the delivery date of M2 and the project total cost with comparable material inventory cost.

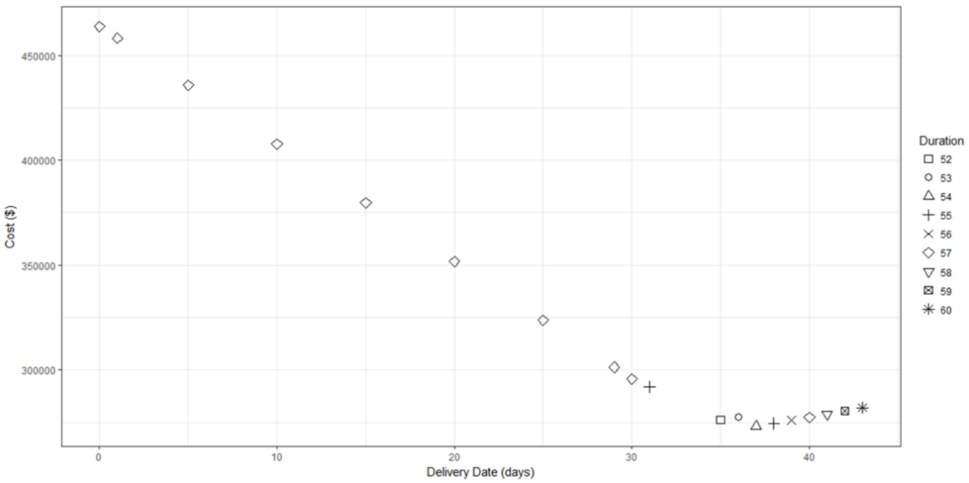


Figure 6.14: Relationship between the delivery date of M2 and the project total cost with significant material inventory cost.

6.3 Case Study: Bridge Girder Fabrication Project

This case study aims to show the applicability of the proposed optimization model in scheduling projects of practical size. This case study is based on the project data on fabricating plate girders from a fabrication shop located in Alberta, Canada. Plate girders, each consisting of one web, two flanges, many stiffeners, and shear studs, are fabricated from raw steel plates in an off-site fabrication facility. One fabrication shop has multiple workstations (e.g., flanges preparation station, webs preparation station, girder assemble station, girder welding station, stiffener and studs welding station, girder splicing station, and sandblast station); a series of operations are performed in fabrication shops, including cutting, drilling, fitting, welding, sandblasting, painting, and other surface finishing (Song and AbouRizk 2008). Figure 6.15 shows the typical process for fabricating one plate girder, which specifies the technical precedence relationships. The description of each activity involved resources, and duration are listed in Table 6.3. Note: in this case study, the time unit for activity duration is *minute*, which is the basic time unit for scheduling. But for other cases, the time unit can be hour or day or week.

Five types of resources (i.e., JM: Journeyman, JMW: Journeyman welder, CR: Crane, DR: Drill machine, CM: cutting machine.) are involved in this case study. In the shop, there are 12 JM, 10 JMW, 4 CR, 4 DR, and 2 CM available. Three bridges are fabricated in this shop. One bridge has 15 girders, one is made of 18 girders, and another one has 6 girders. Each girder of different bridges is custom-designed.

The materials for fabricating girders include steel plates, studs, k-plate, gussets, etc., which are pre-ordered from retailers or specific mills. For special materials, the mill certificate data and results of impact tests are required for review and acceptance. In this case study, only major materials required in those girder fabrication projects are considered, as the impact of their timely deliveries on crew utilization and the total project schedule is significant. For the three bridges, the type and required quantity of materials were extracted from the purchasing order of

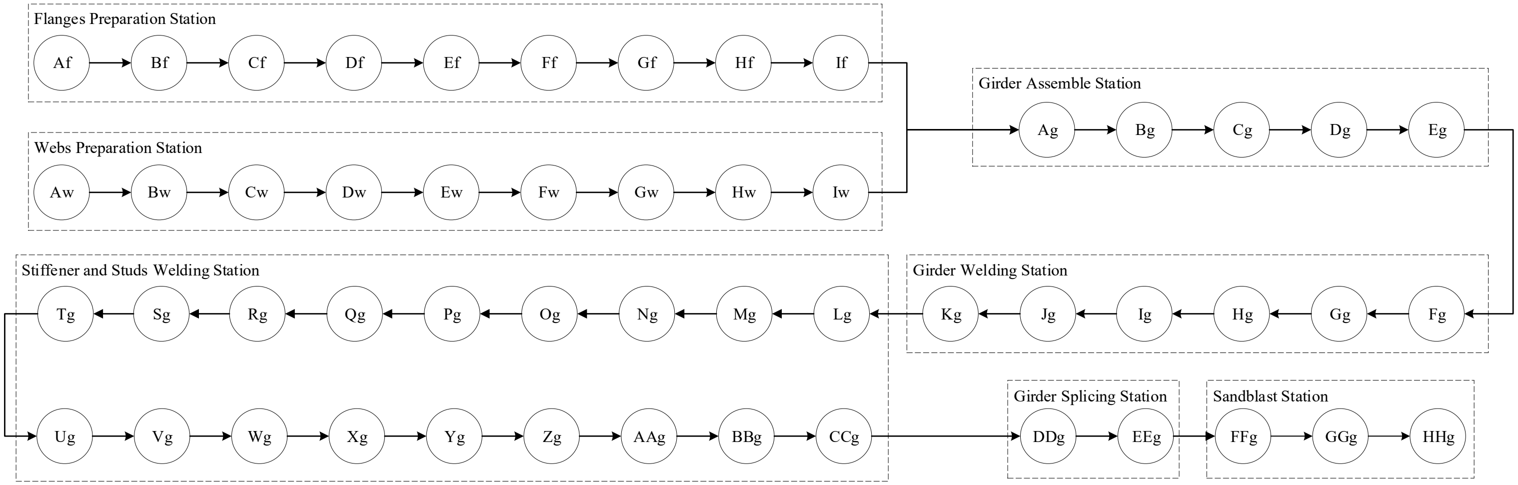


Figure 6.15: Process for fabricating one plate girder.

Table 6.3: Detailed description of activities for fabricating girders.

Work Station	Activity ID	Activity Name	Required Resources	Duration (min)
Flanges Preparation Station	Af	Pre-blast plates for flanges	1 JM	120
	Bf	Layout and cut	1 JM + 1 CM	90
	Cf	Straighten	1 JM	45
	Df	Drill flange ends	1 JM + 1DR	60
	Ef	Splice flanges of side 1	1 JMW	240
	Ff	Turning	1 JM + 2 CR	20
	Gf	Splice flanges of side 2	1 JMW	240
	Hf	Scarf cut	1 JM	60

	If	Stack flanges	1 JM	20
Webs Preparation Station	Aw	Set-up plates for webs	1 JM	15
	Bw	Layout camber	1 JM	90
	Cw	Cut web splice	1 JM + 1 CM	60
	Dw	Weld web splice side 1	1 JMW	120
	Ew	Blast	1 JM	120
	Fw	Turn web	1 JM + 2CR	45
	Gw	Weld web splice side 2	1 JMW	120
	Hw	Blast	1 JM	120
	Iw	Handling to assemble station	1 JM + 2CR	20
Girder Assemble Station	Ag	Layout camber	1 JM	150
	Bg	Cut web	1 JM + 1 CM	360
	Cg	Press Flanges	1 JM	120
	Dg	Tack flanges to web	1 JMW	600
	Eg	Handling to girder weld station	1 JM + 2CR	45
Girder Welding Station	Fg	Grinding tacks of side 1	3 JM	240
	Gg	Weld side 1	1 JMW	300
	Hg	Turning	2 JM + 2CR	30
	Ig	Grinding tacks of side 2	3 JM	240
	Jg	Weld side 2	1 JMW	300
	Kg	Handling to stiffener station	2 JM + 2CR	30
	Lg	Layout stiffeners for side 1	1 JM	60

Stiffener and Studs Welding Station	Mg	Fitting bearing and intermediate stiffeners for side 1	1 JM	360
	Ng	Turning	2 CR + 2JM	30
	Og	Layout stiffeners for side 2	1 JM	60
	Pg	Drill webs for gussets, etc.	1 JM + 1 DR	240
	Qg	Fitting bearing and intermediate stiffeners for side 2	1 JM	360
	Rg	Studs layout	1 JM	120
	Sg	Weld stiffeners to web for side 1	1 JMW	600
	Tg	Turning	2 CR + 2JM	30
	Ug	Weld stiffeners to flanges for side 1	2 JMW	300
	Vg	Turning	2 CR + 2JM	30
	Wg	Weld stiffeners to web for side 2	1 JMW	600
	Xg	Turning	2 CR + 2JM	30
	Yg	Weld stiffeners to flanges for side 2	2 JMW	300
	Zg	Checking	1 JM	30
	AAg	Shoot studs	1 JM	480
	BBg	Test and cleaning	1 JM	60
	CCg	Handling to field splice station	2 CR + 2JM	30
Girder Splicing Station	DDg	Drill web and flanges	1 JM + 1 DR	480
	EEg	Match splice plates with flanges and web	1 JM	60
Sandblast Station	FFg	Final blast	1 JM	600
	GGg	Final checks	1 JM	60
	HHg	Final Dress	1 JM	960

Table 6.4: Required material information for each girder.

Bridge Name	Girder Name	Activity ID	Material Name	Quantity
BR1	GR1 - GR15	Af	PL50MM	11891 kg
		Aw	PL20MM	9362 kg
		Mg	PL20MM	7781 kg
		Qg	PL20MM	7781 kg
		AAg	Studs	435
		EEg	PL16MM	3651 kg
BR2	GR1 - GR18	Af	PL45MM	27094 kg
		Aw	PL30MM	10282 kg
		Mg	PL20MM	9338 kg
		Qg	PL20MM	9338 kg
		AAg	Studs	659
		EEg	PL16MM	8763 kg
BR3	GR1 - GR6	Af	PL50MM	17100 kg
		Aw	PL30MM	9493 kg
		Mg	PL20MM	6225 kg
		Qg	PL20MM	6225 kg
		AAg	Studs	324
		EEg	PL16MM	4674 kg

the partner company and are listed in Table 6.4. Note that the format PLXXMM means the plate with the thickness of XX millimeter. In practical, different girders of one bridge may require different quantities of one material or different types of materials, but in this case study, due to the size limit of the paper, it was assumed that all girders of one bridge require the same type and the same quantity of materials.

Material inventory and supply information were extracted from the company's material management system. At the beginning of the projects, there are 180.482-ton PL50MM, 337.309-ton PL45MM, 132.014-ton PL30MM, 542.327-ton PL20MM, 110.568-ton PL16MM, and 8792-unit studs. There are two material deliveries to the shop during the execution of projects, one delivers at the third week after starting the projects: 54.793-ton PL50MM, 55.720-ton PL45MM, 70.621-ton PL30MM, 136.487-ton PL20MM, 52.693-ton PL16MM, and 4985-unit studs; the other delivers at the sixth week after starting the projects: 45.690-ton PL50MM, 94.663-ton PL45MM, 39.399-ton PL30MM, 105.914-ton PL20MM, 77.282-ton PL16MM, and 6554-unit studs. The labor cost and material inventory cost are scaled based on actual data: \$50/day per resource, \$1/day per 10-ton for plates, and \$1/day per thousand studs. As the materials are stacked on the open space outside the fabrication shop for the partner company, the material inventory cost is relatively low compared to the laborer cost.

For this case study, each girder has 52 fabrication activities, so for three bridges, there are over two thousand activities in total $(15 + 18 + 6) \times 52 = 2028$. Table 6.3 defines the activity set V with the duration D_i and resource requirement w_{ir} for each girder i. In this case study, each activity only has one crew method specified; thus, no alternative crew method constraints apply. Figure 6.15 presents the predecessor set P for each girder, Table 6.4 lists the required material type

n and amount q_{in} for girders of different bridges. Five types of resources (i.e., JM, JMW, CR, DR, CM) and six types of materials (i.e., PL50MM, PL45MM, PL30MM, PL20MM, PL16MM, studs) are involved. The limits for resources (JM, JMW, CR, DR, CM) are $W = (12, 10, 4, 4, 2)$; the unit costs of resources are $u_r = (50, 50, 50, 50, 50)$ per resource per day. The initial inventory amounts for materials (PL50MM, PL45MM, PL30MM, PL20MM, PL16MM, studs) are $I_0 = (180.482\ ton,\ 337.309\ ton,\ 132.014\ ton,\ 542.327\ ton,\ 110.568\ ton,\ 8792\ units)$, unit material inventory costs are $u_n = \$(0.1, 0.1, 0.1, 0.1, 0.1, 0.001)$ per ton for plates and per unit for studs every day. There are two batches of materials to be delivered: One is to occur at the third week $S_{8100} = (54.793\ ton, 55.720\ ton, 70.621\ ton, 136.487\ ton, 52.693\ ton, 4985\ units)$; the other supply is planned at the sixth week $S_{16200} = (45.690\ ton,\ 94.663\ ton,\ 39.399\ ton,\ 105.914\ ton,\ 77.282\ ton,\ 6554\ \ units)$. In total, there are 4056 variables for this case study.

The project and material information are formulated into the model Eq. (3.5) – Eq. (3.10) to determine the optimal fabrication sequence. The optimization model was solved in *IBM ILOG CPLEX Optimization Studio 12.6.3* on a desktop with a 3.2 GHz CPU and 8 GB random-access memory (RAM). An optimal execution plan was produced in 78 minutes CPU time with the total cost of $802,986 and the total fabrication duration of 49 days. The results were face-validated by two experts from the partner company. One is a production manager with more than 30 years' experience in bridge construction; the other one is a scheduler with more than 50 years' experience in scheduling heavy construction projects. Based on their experiences, the fabrication of these three bridges as defined in the scope of the current case study would be completed in 12 weeks (60 working days) and would cost around $1,000,000. The cost and the duration provided by the two experts are 25% and 23% higher than those derived from the proposed approach. The subject experts confirmed that if the material supply plan and crew operations schedule had been managed in an integrative and more efficient way, around 20% improvement on project duration and cost performances would have been achieved. This means if optimization is utilized properly, there are immediate opportunities for at least 20% cost savings in reality. This also validates that the results derived from the optimization model are in line with the experts' expectation. The feasibility of the detailed schedule was further confirmed by subject experts. In this way, the proposed optimization model is proven sufficient and efficient to plan material logistics and crew operations on projects of practical size.

6.4 Discussion

Engineered material availability is a predominant uncertainty for construction scheduling. Although material supply patterns are generally stipulated in the contract, project delays or budget overruns in connection with materials logistics (e.g., late deliveries or early arrivals) are commonplace in the real world. Analytical decision support, which utilizes optimization and simulation computing, is desired to assist in the specification of material supply patterns and to evaluate the impact of uncertainties associated with material supplies on the project schedule. This chapter proposes a two-step analytical approach to evaluate the impact of such uncertainties upon the total project cost, namely: (1) developing a resource-constrained scheduling optimization model to incorporate material logistics constraints; (2) analyzing various material delivery scenarios based on the mathematical model to characterize the impact of material supply on the project cost. For practical applications, the proposed approach can provide an optimized schedule for each particular setting of material logistics and crew availability. The derived alternative plans– along with the observed relationship between the material delivery time and the total project cost– will better prepare project managers in coping with disruptions and changes. It is worth mentioning that the current research has set a particular focus on analyzing how the material delivery time impacts the project schedule performance. Nonetheless, by taking more scheduling constraints (e.g., engineering drawing availability and limited workspace) into consideration, the proposed methodology can be readily enhanced to analyze the impact of other factors on the project schedule performance in its further extensions.

The proposed methodology is developed in the application context of a fabrication facility. The crew size and crew productivity are assumed to be deterministic due to the relatively well-controlled work environment in a fabrication facility. However, if these factors are associated with considerable uncertainties and have a significant consequence on project schedules and costs, the proposed methodology can be adapted in the future extension to take these factors as risk variables and characterize their impact on project duration and cost – independently or along with material logistics.

In addition, a pre-set project deadline is assumed as a hard constraint in the proposed methodology. In reality, however, delaying project completion time without significantly increasing project cost can be an alternate solution in coping with risks in material logistics. Hence, another future direction in extending the current optimization model is to incorporate the predefined deadline into the objective function by factoring in early-completion bonuses and late penalties.

6.5 Conclusion

The availability of finite labor and equipment resources is recognized as the high-risk factor in construction planning. Nonetheless, modern construction projects resort to off-site prefabrication and modular construction technologies for better quality and productivity benefits. The supply chain of materials becomes complicated and risky, presenting itself as a critical constraint in planning crew operations in the field. Therefore, it calls for incorporating material supply logistics as explicit constraints in deriving project schedules and analyzing its impacts on project schedule performances.

In this chapter, the optimization model presented in Section 3.2.2, is utilized to simulate the impact of different material delivery settings on the project schedule. How to cope with the variable material delivery times by using the optimization model is illustrated from both project planning and project control perspective based on a sample case study. An independent simulation model was built on the same case problem in order to shed light on model validity and practical feasibility of obtained optimal solutions. Once validated, the optimization model provides the analytical engine for assessing the impact of different material delivery times on the project cost. The relationship between the material delivery date and the total project cost is revealed in a quantitative, visual way based on the same case. The derived budget-constrained material delivery time window can provide crucial decision support for practitioners to determine the allowable time window in the delivery of certain critical materials so as to keep the total project cost under preset limits. Furthermore, in close collaboration with our industry partner, another case study using a bridge steel girder fabrication project of practical size was conducted to demonstrate the applicability of the proposed optimization model on large project networks.

7 CHAPTER 7: CONCLUSION

7.1 Research Conclusions

This research problem is aimed to address the issue in planning and scheduling multiple prefabrication projects, where several engineered systems or components of large size and heavy weight are fabricated in a fabrication facility with limited workplace and storage areas. As a result, frequent inter-project resource transfers are not feasible and should be mitigated. Nonetheless, current scheduling methods for planning shared resource use on multiple projects give rise to extensive resource links among projects which disrupt the “learning curve” effect, require extra efforts in handling and setting up work, and entail additional communication and coordination in project execution or rely on defining the unit cost for transferring a resource among different projects to reduce inter-project resource transfer. For prefabrication in construction, the unit cost of transferring resources between projects is not straightforward to quantify and demands separate research endeavor. As such, schedules derived by professional project management software, which ignore resource work continuity, are insufficient to practice and become useless. The main objective of this research is to improve the multi-project scheduling practice by introducing a dual-level multi-project scheduling framework that can provide reliable and achievable resource plans for multiple concurrent projects. In collaboration with a major structural steel fabricator located in Edmonton, Canada, successful case studies based on practices and projects in the real world have been conducted to demonstrate the feasibility and applicability of the proposed approach.

This research has addressed three questions in connection with the multi-project scheduling problem: (1) how to effectively mitigate undesired, frequent resource transfers among multiple projects in order to enhance the schedule robustness; (2) how to link and synchronize schedules for various management functions so as to eliminate misalignments between resource planning and project scheduling; and (3) how to analytically evaluate the impact of uncertainties in time-dependent material logistics on individual project schedules and costs. In conducting the research, four main steps were taken: first, a dual-level framework for optimally allocating resources to multiple concurrent projects was proposed and the mathematical problem formulations for both levels were provided (Chapter 3). Second, the feasibility of the proposed framework in reducing inter-project resource transfers was demonstrated through a case study, and a resource use robustness indicator was defined to measure the frequency of inter-project resource transfers for a derived schedule (Chapter 4). Third, the applicability of the proposed framework in interconnecting and synchronizing schedules for various management functions was illustrated by a case study based on real-world bridge girder fabrication projects (Chapter 5). Finally, the scheduling optimization model at the lower level of the proposed framework was utilized to evaluate the impact of different input settings of material logistics on project schedule and cost performances (Chapter 6).

This research is applicable to an industry where various projects with distinct designs are executed at the same location and at the same time competing for limited resources (e.g., skilled laborers in a prefabrication facility). The involved resources are scarce, and more time is needed for handling, setup, and relearning specifications if they are transferred between projects with different designs. A similar application scenario is pipe rack module assembly in a module yard. For applying the proposed framework to other types of projects, two key aspects need to be identified, namely, the way to define WBS including macro-activities and limited resources identification.

7.2 Academic Contributions

The research outcomes contribute to the following academic areas:

- Advancement of multi-project scheduling approaches by proposing a generic dual-level scheduling framework. By separating resource planning at a fabrication facility and resource scheduling for an individual project into two interconnected levels, intensive inter-project resource transfers can be reduced, thus providing more robust construction schedules. Also, schedules for various management functions can be seamlessly linked through the two interconnected levels, thus facilitating information sharing and improving project management practice.
- Development of an integrated scheduling optimization model for simulating the impact of material delivery dates on project schedule and cost performances. Material supply information, which is generally overlooked or considered separately in previous research on construction scheduling, is incorporated as an additional constraint in the newly developed resource-constrained scheduling optimization model.
- Provision and definition of a new schedule performance indicator (i.e., resource use robustness) for evaluating construction schedules. Cost, duration, and resource fluctuations are commonly adopted indicators to evaluate schedule performance in the construction domain. This research proposes a new indicator for assessing schedule performance from the perspective of inter-project resource transfers, which is important for a multi-project environment.

7.3 Industrial Contributions

The deliverables of this research contribute to advancing practice in the construction prefabrication industry from the following perspectives:

- Providing production managers with reliable and feasible project schedules and work plans at a fabrication facility, ensuring crew work continuity on individual projects, enhancing resource utilization efficiency, and improving communication efficiency at different levels of project management.
- Creating reliable program schedule, project schedule, and operation schedules, which are dynamically interconnected to each other, thereby, facilitating schedule maintenance and updating and saving the efforts for progress report among management personnel. These schedules also provide guidance on various management functions, such as evaluation of remaining fabrication capacities, prediction of project delivery performances, and execution of daily fabrication work within fabrication facilities.
- Providing crucial decision support for practitioners to determine the allowable time window of certain critical material deliveries so as to keep the total project cost under pre-set limits. Alternative plans are also available to better prepare project managers in coping with disruptions and changes.
- Developing an MS Project add-on, which implements the proposed framework to schedule multiple concurrent bridge girder fabrication projects for a structural steel fabricator in Edmonton, Canada.

7.4 Research Limitations and Future Work

Although the research findings in above chapters support the developed approaches, certain limitations of this research should be noted and explored. In order to address these limitations, related future research is identified and recommended as well.

- In the real world, individual projects are generally constrained by client specified start times, which can be incorporated as an additional hard constraint in the mathematical model for resource allocation in order to extend the proposed scheduling framework.
- The current research develops the deterministic framework to provide a sufficient yet analytical basis to the ill-structured scheduling problem being studied. In the future, the framework can be extended by defining stochastic models to incorporate uncertainties (e.g., rework and change orders) and variations. For instance, statistical distributions can be fitted to model activity duration; uncertainties inherent in resource requirements can be considered when generating macro-activities.
- In the future, it will be imperative to address how the dynamic changes at the lower level impact the decisions at the upper level and how to update upper level schedules based on actual progress at the lower level in a timely fashion. Ultimately, a closed feedback loop will be formed in the dual-level optimization framework; as such, dynamic changes at the lower level can be continuously factored in the upper level optimization and their impacts are refreshed on decisions at the upper level (e.g., resource use conflicts and project delays). This is especially relevant when the current research scope extends from project planning and scheduling to more dynamic project control.

- The setup cost, material handling cost, and labor productivity loss due to unlearning curve effect in connection with resource transfers can be further researched in order to define resource transfer costs in a quantitatively reliable, practically feasible way, so as to incorporate resource use robustness measure into the total cost function for optimization.
- A particular focus has been set on analyzing how the material delivery time impacts project schedule and cost performances in this research. Nonetheless, by taking more scheduling constraints (e.g., engineering drawing availability) into consideration, the proposed methodology in Chapter 6 can be readily enhanced to analyze the impact of other factors on project schedule and cost performances in the further extensions.
- The case on a bridge girder fabrication project has demonstrated the applicability of the developed lower-level optimization model on scheduling project networks with more than 2,000 activities. With the increase of number and size of projects, the developed upper-level and lower-level optimization models are likely become intractable by existing algorithms. Systematic simulation tests are recommended for future research to investigate the scalability of the developed optimization models in scheduling a larger number of projects and more complex project networks.

REFERENCES

AACE (2010). *Schedule Levels of Detail - As Applied in Engineering, Procurement, and Construction*, R. Ed., ed., AACE international, Morgantown, WV, 37R-06.

Abido, M., and Elazouni, A. M. (2010). "Multiobjective evolutionary finance-based scheduling: Entire projects' portfolio." *J. Comput. Civ. Eng.*, 25(1), 85-97.

Adhau, S., Mittal, M., and Mittal, A. (2013). "A multi-agent system for decentralized multi-project scheduling with resource transfers." *International journal of production economics*, 146(2), 646-661.

Ala-Risku, T., and Kärkkäinen, M. (2006). "Material delivery problems in construction projects: A possible solution." *International Journal of Production Economics*, 104(1), 19-29.

Andersen, E. S., Grude, K. V., and Haug, T. (2009). *Goal directed project management: effective techniques and strategies*, Kogan Page Publishers.

Angkiriwang, R., Pujawan, I. N., and Santosa, B. (2014). "Managing uncertainty through supply chain flexibility: reactive vs. proactive approaches." *Production & Manufacturing Research*, 2(1), 50-70.

Arashpour, M., Wakefield, R., Blismas, N., and Minas, J. (2015). "Optimization of process integration and multi-skilled resource utilization in off-site construction." *Automation in Construction*, 50, 72-80.

Arbulu, R., and Ballard, G. (2004). "Lean supply systems in construction." *International Group for Lean Construction. Conpenhague, Dinamarca*.

Arbulu, R. J., Tommelein, I. D., Walsh, K. D., and Hershauer, J. C. "CEPM 3: contributors to lead time in construction supply chains: case of pipe supports used in power plants."

Proc., Proceedings of the 34th conference on Winter simulation: exploring new frontiers, Winter Simulation Conference, 1745-1751.

Arora, R., and Sachdeva, R. (1989). "Distributed simulation of resource constrained project scheduling." *Computers & operations research*, 16(4), 295-304.

Baptiste, P., and Le Pape, C. (1995). "A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling." *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI),* Montreal, Canada, 600-606.

Baptiste, P., and Le Pape, C. "A theoretical and experimental comparison of constraint propagation techniques for disjunctive scheduling." *Proc., IJCAI (1)*, 600-606.

Baptiste, P., Le Pape, C., and Nuijten, W. (2012). *Constraint-based scheduling: applying constraint programming to scheduling problems*, Springer Science & Business Media.

Bock, D. B., and Patterson, J. H. (1990). "A comparison of due date setting, resource assignment, and job preemption heuristics for the multiproject scheduling problem." *Decision Sciences*, 21(2), 387-402.

Brailsford, S. C., Potts, C. N., and Smith, B. M. (1999). "Constraint satisfaction problems: Algorithms and applications." *European Journal of Operational Research*, 119(3), 557-581.

Browning, T. R., and Yassine, A. A. (2010). "Resource-constrained multi-project scheduling: Priority rule performance revisited." *International Journal of Production Economics*, 126(2), 212-228.

Caldas, C. H., Kim, J.-Y., Haas, C. T., Goodrum, P. M., and Zhang, D. (2014). "Method to assess the level of implementation of productivity practices on industrial projects." *J. Constr. Eng. Manage.*, 141(1), 04014061.

Can, A., and Ulusoy, G. (2014). "Multi-project scheduling with two-stage decomposition." *Annals of Operations Research*, 217(1), 95-116.

Chen, P.-H., and Shahandashti, S. M. (2009). "Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints." *Automation in Construction*, 18(4), 434-443.

Chen, P.-H., and Weng, H. (2009). "A two-phase GA model for resource-constrained project scheduling." *Automation in Construction*, 18(4), 485-498.

Cheng, M.-Y., Prayogo, D., and Tran, D.-H. (2015). "Optimizing multiple-resources leveling in multiple projects using discrete symbiotic organisms search." *J. Comput. Civ. Eng.*, 30(3), 04015036.

Chiu, C., and Hsu, P.-L. (2005). "A constraint-based genetic algorithm approach for mining classification rules." *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(2), 205-220.

Choi, J., and Song, H. (2014). "Evaluation of the modular method for industrial plant construction projects." *International Journal of Construction Management*, 14(3), 171-180.

CII (2004). *Project Control for Construction*, Rev.Ed., ed., Construction Industry Institute, Austin, Tex., RS6-5.

CII (2013). "Enhanced Work Packaging: Design through Work Face Execution (Best Practice)." Construction Industry Institute, Austin, Tex.

Cohen, I., Mandelbaum, A., and Shtub, A. (2004). "Multi-project scheduling and control: A process-based comparative study of the critical chain methodology and some alternatives." *Project Management Journal*, 35, 39-49.

Cooper, R. G., Edgett, S. J., and Kleinschmidt, E. J. (1997). "Portfolio management in new product development: Lessons from the leaders—I." *Research-Technology Management*, 40(5), 16-28.

Cooper, R. G., Edgett, S. J., and Kleinschmidt, E. J. (2000). "New problems, new solutions: making portfolio management more effective." *Research-Technology Management*, 43(2), 18-33.

CP Optimizer User's Manual (2015). "IBM ILOG CPLEX Optimization Studio CP Optimizer User's Manual." C. I. Corp., ed., 131.

Dozzi, S., and AbouRizk, S. (1993). "Human factors and productivity improvement." *Productivity in construction*, 17-26.

El-Abbasy, M. S., Elazouni, A., and Zayed, T. (2017). "Generic Scheduling Optimization Model for Multiple Construction Projects." *J. Comput. Civ. Eng.*, 31(4), 04017003.

El-Rayes, K., and Jun, D. H. (2009). "Optimizing resource leveling in construction projects." *J. Constr. Eng. Manage.*, 135(11), 1172-1180.

Elfving, J., and Tommelein, I. (2003). "Impact of Multitasking and Merge Bias on Procurement of Complex Equipment." Proc. Winter Sim. Conf.(WSC03), New Orleans, LA."

Elonen, S., and Artto, K. A. (2003). "Problems in managing internal development projects in multi-project environments." *International Journal of Project Management*, 21(6), 395-402.

Fendley, L. G. (1968). "Toward Development Of A Complete Multiproject Scheduling System." *Journal of Industrial Engineering*, 19(10), 505.

Flyvbjerg, B. (2014). "What you should know about megaprojects and why: An overview." *Project management journal*, 45(2), 6-19.

Fondahl, J. W. (1991). "The Development of the Construction Engineer: Past Progress and Future Problems." *J. Constr. Eng. Manage.*, 117(3), 380-392.

Fromherz, M. P. "Constraint-based scheduling." *Proc., American Control Conference*, IEEE, 3231-3244.

Fu, N., Varakantham, P., and Lau, H. C. "Towards finding robust execution strategies for RCPSP/max with durational uncertainty." *Proc., the Twentieth International Conference on Automated Planning and Scheduling*, 73-80.

Gosling, J., Pero, M., Schoenwitz, M., Towill, D., and Cigolini, R. (2016). "Defining and categorizing modules in building projects: an international perspective." *Journal of Construction Engineering and Management*, 142(11), 04016062.

Haas, C. T. (2000). *Prefabrication and preassembly trends and effects on the construction workforce*, Center for Construction Industry Studies.

Hamzeh, F., Ballard, G., and Tommelein, I. D. (2012). "Rethinking Lookahead Planning to Optimize Construction Workflow." *Lean Construction Journal*.

Hamzeh, F. R., Ballard, G., and Tommelein, I. D. "Improving construction workflow-the connective role of lookahead planning." *Proc., Proceedings for the 16th annual conference of the International Group for Lean Construction*, 635-646.

Han, S. H., Kim, D. Y., and Kim, H. (2007). "Predicting profit performance for selecting candidate international construction projects." *J. Constr. Eng. Manage.*, 133(6), 425-436.

Hans, E. W. (2001). *Resource loading by branch-and-price techniques*, Twente University Press, The Netherlands.

Hans, E. W., Herroelen, W., Leus, R., and Wullink, G. (2007). "A hierarchical approach to multi-project planning under uncertainty." *Omega-The International Journal of Management Science*, 35(5), 563-577.

Haralick, R. M., and Elliott, G. L. (1980). "Increasing tree search efficiency for constraint satisfaction problems." *Artificial intelligence*, 14(3), 263-313.

Hegazy, T. (1999). "Optimization of resource allocation and leveling using genetic algorithms." *J. Constr. Eng. Manage.*, 125(3), 167-175.

Hendrickson, C., and Au, T. (1989). *Project management for construction: Fundamental concepts for owners, engineers, architects, and builders*, Chris Hendrickson.

Hentenryck, P. V. (1989). *Constraint satisfaction in logic programming*, MIT Press.

Herroelen, W., and Leus, R. (2003). "Solution and quality robust project scheduling: a methodological framework." *Research report 0315*, Department of Applied Economics, K.U. Leuven.

Herroelen, W., and Leus, R. (2004). "The construction of stable project baseline schedules." *European Journal of Operational Research*, 156(3), 550-565.

Herroelen, W., and Leus, R. (2004). "Robust and reactive project scheduling: a review and classification of procedures." *International Journal of Production Research*, 42(8), 1599-1620.

Hinze, J., and Olbina, S. (2009). "Empirical analysis of the learning curve principle in prestressed concrete piles." *J. Constr. Eng. Manage.*, 135(5), 425-431.

Hooker, J. N., Ottosson, G., Thorsteinsson, E. S., and Kim, H.-J. (1999). "On integrating constraint propagation and linear programming for combinatorial optimization."

Horvath, A. (2004). "Construction materials and the environment." *Annu. Rev. Environ. Resour.*, 29, 181-204.

Irizarry, J., Karan, E. P., and Jalaei, F. (2013). "Integrating BIM and GIS to improve the visual monitoring of construction supply chain management." *Automation in Construction*, 31, 241-254.

Kelley Jr, J. E., and Walker, M. R. "Critical-path planning and scheduling." *Proc., Proceedings of the eastern joint computer conference*, ACM, 160-173.

Kim, K., and de la Garza, J. M. (2003). "Phantom float." *Journal of Construction Engineering and Management*, 129(5).

Kim, K., Yun, Y., Yoon, J., Gen, M., and Yamazaki, G. (2005). "Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling." *Computers in industry*, 56(2), 143-160.

Kolisch, R. (1996). "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation." *European Journal of Operational Research*, 90(2), 320-333.

Krause, S. (2015). "Steel Bridge Design Handbook: Steel Bridge Fabrication."

Krüger, D., and Scholl, A. (2009). "A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times." *European Journal of Operational Research*, 197(2), 492-508.

Kumar, V. (1992). "Algorithms for constraint-satisfaction problems: A survey." *AI magazine*, 13(1), 32.

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

Laslo, Z., and Goldberg, A. I. (2008). "Resource allocation under uncertainty in a multi-project matrix environment: Is organizational conflict inevitable?" *International Journal of Project Management*, 26(8), 773-788.

Lau, H. C., Xiao, F., and Ou, T. (2007). "Robust local search and its application to generating robust schedules." *Proc., the Seventeenth International Conference on Automated Planning and Scheduling*, AAAI.

Lee, B., Lee, H.-S., Park, M., and Kim, H. (2015). "Influence factors of learning-curve effect in high-rise building constructions." *J. Constr. Eng. Manage.*, 141(8), 04015019.

Leus, R., and Herroelen, W. (2004). "Stability and resource allocation in project planning." *IIE transactions*, 36(7), 667-682.

Lgelmund, G., and Radermacher, F. J. (1983). "Algorithmic approaches to preselective strategies for stochastic scheduling problems." *Networks*, 13(1), 29-48.

Li, H., and Zhang, H. (2013). "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints." *Automation in construction*, 35, 431-438.

Liu, J., and Lu, M. (2018). "Constraint programming approach to optimizing project schedules under materials logistics and crew availability constraints." *J. Constr. Eng. Manage.*, 144(7), 04018049.

Liu, J., and Lu, M. (2018). "Robust Dual-Level Optimization Framework For Resource-Constrained Multi-Project Scheduling " *J. Comput. Civ. Eng. (in review)*.

Liu, J., Siu, M.-F. F., and Lu, M. "Modular construction system simulation incorporating off-shore fabrication and multi-mode transportation." *Proc., Winter Simulation Conference (WSC), 2016*, IEEE, 3269-3280.

Liu, S.-S., and Wang, C.-J. (2012). "Optimizing linear project scheduling with multi-skilled crews." *automation in construction*, 24, 16-23.

Lova, A., Maroto, C., and Tormos, P. (2000). "A multicriteria heuristic method to improve resource allocation in multiproject scheduling." *European Journal of Operational Research*, 127(2), 408-424.

Lova, A., and Tormos, P. (2001). "Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling." *Annals of Operations Research*, 102(1-4), 263-286.

Lu, M. (2003). "Simplified discrete-event simulation approach for construction simulation." *Journal of Construction Engineering and Management*, 129(5), 537-546.

Lu, M., Lam, H.-C., and Dai, F. (2008). "Resource-constrained critical path analysis based on discrete event simulation and particle swarm optimization." *Automation in construction*, 17(6), 670-681.

Lu, M., and Li, H. (2003). "Resource-activity critical-path method for construction planning." *J. Constr. Eng. Manage.*, 129(4), 412-420.

Lu, M., and Wong, L.-C. "Comparing PROMODEL and SDESA in modeling construction operations." *Proc., Simulation Conference, 2005 Proceedings of the Winter*, IEEE, 9 pp.

Martinsuo, M., and Lehtonen, P. (2007). "Role of single-project management in achieving portfolio management efficiency." *International journal of project management*, 25(1), 56-65.

Mastroianni, R., and Abdelhamid, T. "The challenge: The impetus for change to lean project delivery." *Proc., Proc., IGLC-11, 11th Conf. of Int. Group for Lean Construction*, 418-426.

Mattila, K. G., and Abraham, D. M. (1998). "Resource leveling of linear schedules using integer linear programming." *J. Constr. Eng. Manage.*, 124(3), 232-244.

Menesi, W., Golzarpoor, B., and Hegazy, T. (2013). "Fast and near-optimum schedule optimization for large-scale projects." *Journal of Construction Engineering and Management*, 139(9), 1117-1124.

Menesi, W., Golzarpoor, B., and Hegazy, T. (2013). "Fast and near-optimum schedule optimization for large-scale projects." *J. Constr. Eng. Manage.*, 139(9), 1117-1124.

Menesi, W., and Hegazy, T. (2014). "Multimode resource-constrained scheduling and leveling for practical-size projects." *Journal of management in engineering*, 31(6), 04014092.

Nee, K. E., Chin, J. F., Loh, W. P., and Tan, M. C.-L. (2014). "A Constraint programming-based genetic algorithm for capacity output optimization." *Journal of Industrial Engineering and Management*, 7(5), 1222.

Neumann, K., and Schwindt, C. (1998). "A capacitated hierarchical approach to make-to-order production." *Journal européen des systèmes automatisés*, 32(4), 397-413.

Nuijten, W. P., and Aarts, E. H. (1996). "A computational study of constraint satisfaction for multiple capacitated job shop scheduling." *European Journal of Operational Research*, 90(2), 269-284.

O'Brien, W. J., Formoso, C. T., Ruben, V., and London, K. (2008). *Construction supply chain management handbook*, CRC press.

O’Connor, J. T., O’Brien, W. J., and Choi, J. O. (2014). "Critical success factors and enablers for optimum and maximum industrial modularization." *Journal of Construction Engineering and Management*, 140(6), 04014012.

OptaPlanner (2018). "OptaPlanner - Constraint satisfaction solver (Java^{TM}, Open Source) [online]." <http://www.optaplanner.org/>. (2018-02-15).

Patil, D., Shrotri, A., and Dandekar, A. (2012). "Management of uncertainty in supply chain." *International Journal of Emerging Technology and Advanced Engineering*, 2(5), 303-308.

Payne, J. H. (1995). "Management of multiple simultaneous projects: a state-of-the-art review." *International journal of project management*, 13(3), 163-168.

Platje, A., Seidel, H., and Wadman, S. (1994). "Project and portfolio planning cycle: project-based management for the multiproject challenge." *International Journal of Project Management*, 12(2), 100-106.

PMI "A guide to the project management body of knowledge (PMBOK guide)." Project Management Institute.

Polat, G., Arditi, D., and Mungen, U. (2007). "Simulation-based decision support system for economical supply chain management of rebar." *Journal of Construction Engineering and Management*, 133(1), 29-39.

Rossi, F., van Beek, P., and Walsh, T. (2006). *Handbook of Constraint Programming*, Elsevier Science Inc., New York, USA.

Rossi, F., van Beek, P., and Walsh, T. (2006). *Handbook of Constraint Programming*, Elsevier Science.

Rostami, M., Bagherpour, M., Mazdeh, M. M., and Makui, A. (2017). "Resource Pool Location for Periodic Services in Decentralized Multi-Project Scheduling Problems." *Journal of Computing in Civil Engineering*, 31(5), 04017022.

Rushton, A., Croucher, P., Baker, P., Logistics, C. I. o., and UK., T. i. t. (2006). *The Handbook of Logistics and Distribution Management*, Kogan Page.

Safa, M., Shahi, A., Haas, C. T., and Hipel, K. W. (2014). "Supplier selection process in an integrated construction materials management model." *Automation in Construction*, 48, 64-73.

Said, H., and El-Rayes, K. (2010). "Optimizing material procurement and storage on construction sites." *Journal of Construction Engineering and Management*, 137(6), 421-431.

Said, H., and El-Rayes, K. (2013). "Automated System for Multi-Objective Optimization of Construction Supply and Site Logistics." *Annual Conference of the Canadian Society for Civil Engineering 2013: Know-How - Savoir-Faire, (CSCE 2013)* Canadian Society for Civil Engineering, 204-213.

Shaw, P. "Using constraint programming and local search methods to solve vehicle routing problems." *Proc., International Conference on Principles and Practice of Constraint Programming*, Springer, 417-431.

Shokri, S., Ahn, S., Lee, S., Haas, C., and Haas, R. (2015). "Current status of interface management in construction: Drivers and effects of systematic interface management." *J. Constr. Eng. Manage.*, 142(2), 04015070.

Simonis, H. "A problem classification scheme for finite domain constraint solving." *Proc., Proceeding of workshop on constraint applications, CP96, Boston*, Citeseer.

Siu, M.-F. F., Lu, M., and AbouRizk, S. (2015). "Resource supply-demand matching scheduling approach for construction workface planning." *J. Constr. Eng. Manage.*, 142(1), 04015048.

Siu, M.-F. F., Lu, M., and AbouRizk, S. (2015). "Zero-one programming approach to determine optimum resource supply under time-dependent resource constraints." *J. Comput. Civ. Eng.*, 30(2), 04015028.

Son, J., and Skibniewski, M. J. (1999). "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." *J. Constr. Eng. Manage.*, 125(1), 23-31.

Song, L., and AbouRizk, S. M. (2008). "Measuring and modeling labor productivity using historical data." *J. Constr. Eng. Manage.*, 134(10), 786-794.

Sonmez, R., and Uysal, F. (2014). "Backward-forward hybrid genetic algorithm for resource-constrained multiproject scheduling problem." *Journal of Computing in Civil Engineering*, 29(5), 04014072.

Sonmez, R., and Uysal, F. (2014). "Backward-forward hybrid genetic algorithm for resource-constrained multiproject scheduling problem." *J. Comput. Civ. Eng.*, 29(5), 04014072.

Speranza, M. G., and Vercellis, C. (1993). "Hierarchical models for multi-project planning and scheduling." *European Journal of Operational Research*, 64(2), 312-325.

Sprecher, A., and Drexl, A. (1998). "Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm." *European Journal of Operational Research*, 107(2), 431-450.

Spuhler, R. W., and Biagini, R. G. (1990). "The role and weaknesses of top management in internal projects." *Handbook of management by projects*, Manzsche Verlag, Austria.

Stork, F. (2001). "Stochastic resource-constrained project scheduling." Ph.D. Thesis, Technische Universitat, Berlin.

Taghaddos, H., AbouRizk, S., Mohamed, Y., and Hermann, U. (2011). "Simulation-based auction protocol for resource scheduling problems." *J. Constr. Eng. Manage.*, 138(1), 31-42.

Talbot, F. B. (1982). "Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case." *Management science*, 28(10), 1197-1210.

Tang, Y., Liu, R., and Sun, Q. (2014). "Schedule control model for linear projects based on linear scheduling method and constraint programming." *Automation in construction*, 37, 22-37.

Tatum, C. B. "Integrated construction engineering activities to satisfy challenging project objectives." *Proc., Construction Research Congress 2012: Construction Challenges in a Flat World*, ASCE, 139-148.

Thomas, H. R., Riley, D. R., and Messner, J. I. (2005). "Fundamental principles of site material management." *Journal of Construction Engineering and Management*, 131(7), 808-815.

Thomas, H. R., and Sanvido, V. E. (2000). "Role of the fabricator in labor productivity." *Journal of construction engineering and management*, 126(5), 358-365.

Tserng, H. P., Yin, S. Y., and Li, S. (2006). "Developing a resource supply chain planning system for construction projects." *Journal of Construction Engineering and Management*, 132(4), 393-407.

Van de Vonder, S., Demeulemeester, E., Herroelen, W., and Leus, R. (2005). "The use of buffers in project management: The trade-off between stability and makespan." *International Journal of production economics*, 97(2), 227-240.

Van de Vonder, S., Demeulemeester, E., Herroelen*, W., and Leus, R. (2006). "The trade-off between stability and makespan in resource-constrained project scheduling." *International Journal of Production Research*, 44(2), 215-236.

Van de Vonder, S., Demeulemeester, E., Leus, R., and Herroelen, W. (2006). "Proactive-reactive project scheduling-trade-offs and procedures." *International Series in Operations Research and Management Science*, 92, 25).

Walsh, K. D., Hershauer, J. C., Tommelein, I. D., and Walsh, T. A. (2004). "Strategic positioning of inventory to match demand in a capital projects supply chain." *Journal of Construction Engineering and Management*, 130(6), 818-826.

Wideman, R. M. (2004). *A Management Framework: For Project, Program and Portfolio Integration*, Trafford Publishing.

Xu, X., Zhao, Y., and Chen, C.-Y. (2016). "Project-driven supply chains: integrating safety-stock and crashing decisions for recurrent projects." *Annals of Operations Research*, 241(1-2), 225-247.

Xue, X., Wang, Y., Shen, Q., and Yu, X. (2007). "Coordination mechanisms for construction supply chain management in the Internet environment." *International Journal of project management*, 25(2), 150-157.

Yaghootkar, K., and Gil, N. (2012). "The effects of schedule-driven project management in multi-project environments." *International Journal of Project Management*, 30(1), 127-140.

Yang, K.-K., and Sum, C.-C. (1993). "A comparison of resource allocation and activity scheduling rules in a dynamic multi-project environment." *Journal of Operations Management*, 11(2), 207-218.

Yi, W., and Chan, A. P. (2013). "Critical review of labor productivity research in construction journals." *Journal of Management in Engineering*, 30(2), 214-225.

Zapata, J. C., Hodge, B. M., and Reklaitis, G. V. (2008). "The multimode resource constrained multiproject scheduling problem: Alternative formulations." *AIChE Journal*, 54(8), 2101-2119.

Zhang, H., Li, H., and Tam, C. (2006). "Particle swarm optimization for resource-constrained project scheduling." *International Journal of Project Management*, 24(1), 83-92.

Zheng, Z., Shumin, L., Ze, G., and Yueni, Z. (2013). "Resource-constraint multi-project scheduling with priorities and uncertain activity durations." *International Journal of Computational Intelligence Systems*, 6(3), 530-547.

Zhou, J., Love, P. E., Wang, X., Teo, K. L., and Irani, Z. (2013). "A review of methods and algorithms for optimizing construction scheduling." *Journal of the Operational Research Society*, 64(8), 1091-1105.

APPENDIX A: OPTIMIZATION PROGRAMMING LANGUAGE (OPL) CODE SNIPPETS FOR THE PROPOSED FRAMEWORK

This appendix presents the detailed optimization programming language (OPL) code to implement the proposed framework in IBM ILOG CPLEX Optimization Studio by taking the case study in chapter 4 as an example. In IBM ILOG CPLEX Optimization Studio, the model and data are defined in separated files. In a model file, it includes the declaration of data and decision variables and the definition of objective functions and constraints. A data file is used to initialize the data defined in the model file. A "*run configuration*" is used to link model, data, and specific settings (if any) for solving the model. For one model, it can be linked to different data files for various instances.

The code snippets are presented by following the steps introduced in Section 3.2. The first step is to determine the duration range for each project. The programming code of the model is listed as follows.

```
/*************************************************************
 * OPL 12.6.3.0 Model
 * Purpose: determine the duration range for each project
 *************************************************************/
  using CP;
  // Data declarations
  int NTasks = ...;
  int NRscrs = ...;
  range RsrcIds = 0..NRscrs-1;
  int Rsrc[RsrcIds] = ...;

  tuple Task {
    key string id;
    int nmethods;
  }
  {Task} Tasks = ...;

  tuple Ordering {
    string pred;
    string succ;
 }
 {Ordering} Orderings = ...;
```

```
  tuple Method {
    key string taskId;
    key int id;
    int dur;
    int cost;
  }
  {Method} Methods = ...;
  tuple RequiredRsrc {
    key string taskId;
    key int methodId;
    key int RsrcId;
    int dm_R;
  }
  {RequiredRsrc} RequiredRsrcs = ...;

  // Decision variables
  dvar interval task[t in Tasks];
  dvar interval method[m in Methods] optional size m.dur;

  cumulFunction Q_UsedRsrc[r in RsrcIds] = sum (m in Methods, rsrc in
RequiredRsrcs: m.taskId == rsrc.taskId && m.id == rsrc.methodId && r ==
rsrc.RsrcId) pulse (method[m], rsrc.dm_R); // Time-dependent resource demand

  execute {
      cp.param.FailLimit = 10000000;
      cp.param.TimeLimit = 300;
  }
  // Objective function: minimize project duration
  minimize (max(t in Tasks) endOf(task[t]));

  subject to { // Constraints
      // Alternative mode
      forall (t in Tasks)
        alternative(task[t], all(m in Methods: m.taskId == t.id) method[m]);
      // Resource limit constraint
      forall (r in RsrcIds)
        Q_UsedRsrc[r] <= Rsrc[r];
      // Precedence relationship constraint
      forall (o in Orderings: o.succ != "Z")
        endBeforeStart(task[<o.pred>],task[<o.succ>]);
  }
```

To be able to solve the model, a data instance should be provided. The following gives the code snippet of the data file of Project 1 in the case study of Chapter 4. The data can be hard-coded in the data file or read from an external spreadsheet. By linking with the data file, the model would give the maximum and minimum duration of Project 1. In a similar way, the duration range of Project 2 can be determined by linking with the data file of Project 2.

```
/*********************************************
 * OPL 12.6.3.0 Data
 * Purpose: data file for Project 1
*********************************************/

  // Connect the input spreadsheet
  SheetConnection sheet("Project 1.xlsx");

  //read information from the spreadsheet to initialize data defined in the
model file
  NTasks from SheetRead(sheet, "'Basic Info.'!B1");
  NRscrs from SheetRead(sheet, "'Basic Info.'!B2");
  Tasks from SheetRead(sheet, "Act.!A2:B26");
  Orderings from SheetRead(sheet, "Order!A2:B37");
  Methods from SheetRead(sheet, "Method!A2:D50");
  RequiredRsrcs from SheetRead(sheet, "Rsrc!A2:D50");

  Rsrc = [18]; // hard-coded data
```

The next step is to determine the resource use profile over the duration. The code snippet of the model is presented as follows.

```
/*********************************************
 * OPL 12.6.3.0 Model
 * Purpose: determine the resource use profile over the duration
*********************************************/
  using CP;

  // Data declarations
  int NTasks = ...;
  int NRscrs = ...;
  int deadline = ...;
  range RsrcIds = 0..NRscrs-1;
  int Rsrc[RsrcIds] = ...;

  tuple Task {
    key string id;
    int nmethods;
  }
  {Task} Tasks = ...;

  tuple Ordering {
    string pred;
    string succ;
  }
 {Ordering} Orderings = ...;

  tuple Method {
    key string taskId;
    key int id;
    int dur;
    int cost;
  }
  {Method} Methods = ...;
```

```
  tuple RequiredRsrc {
    key string taskId;
    key int methodId;
    key int RsrcId;
    int dm_R;
  }
  {RequiredRsrc} RequiredRsrcs = ...;
  // Decision variables
  dvar interval task[t in Tasks];
  dvar interval method[m in Methods] optional size m.dur;
  // Peak resource demand for resource leveling
  dvar int peakUsage[r in RsrcIds] in 0..Rsrc[r];

  cumulFunction Q_UsedRsrc[r in RsrcIds] = sum (m in Methods, rsrc in
RequiredRsrcs: m.taskId == rsrc.taskId && m.id == rsrc.methodId && r ==
rsrc.RsrcId) pulse (method[m], rsrc.dm_R); // Time-dependent resource demand

  dexpr int cost = sum(m in Methods)(presenceOf(method[m]) * m.cost * m.dur);
  dexpr int duration = max(t in Tasks) endOf(task[t]);

  execute {
      cp.param.FailLimit = 10000000;
      cp.param.TimeLimit = 300;
  }
  // multi-objective optimization, cost with higher priority
  minimize staticLex(cost, peakUsage[0]);
  subject to {
      // Alternative mode
      forall (t in Tasks)
        alternative(task[t], all(m in Methods: m.taskId == t.id) method[m]);
      // Resource limit
      forall (r in RsrcIds)
        Q_UsedRsrc[r] <= peakUsage[r];
      // Predecessor
      forall (o in Orderings: o.succ != "Z")
        endBeforeStart(task[<o.pred>],task[<o.succ>]);
      // Duration limit
      forall (t in Tasks)
        endOf(task[t]) <= deadline;
  }

  //Flow control for iterating the deadline from the min to max duration
  main{
      thisOplModel.generate();
      var schedule = thisOplModel;
      var def = schedule.modelDefinition; // Load model
      var data = schedule.dataElements; // Access members
      var deadline = thisOplModel.deadline;

   // max duration (i.e., limit) for project 1: 27 days, project 2: 29 days.
      var limit = 27;
      // files for saving scheduling results
      var ofile = new IloOplOutputFile("RawRsrc.txt");
      var ofile1 = new IloOplOutputFile("ActivitySch.txt");
      ofile.writeln("Duration\tCost");
      ofile1.writeln("Duration\tCost");
```

```
      while (deadline <= limit){ // Flow control block
            var flag;
            flag = cp.solve();
            if (flag){
                  writeln("duration: ", schedule.duration, "\t",
schedule.cost, "\t", schedule.peakUsage);
                  ofile.writeln("==============================");

            //Save the deadline, cost and resource usage over time to files
                  ofile.writeln(deadline, "\t", schedule.cost);
                  ofile.writeln(schedule.Q_UsedRsrc);
                  ofile1.writeln("==============================");
                  ofile1.writeln(deadline, "\t", schedule.cost);
                  ofile1.writeln("TaskID\tModeID\tStart\tFinish");
                  for (var m in schedule.Methods) {
                        if (schedule.method[m].present){
                              ofile1.writeln(m.taskId, "\t", m.id,
"\t",schedule.method[m].start, "\t", schedule.method[m].end);
                  }
            }
      }
            if (schedule != thisOplModel){
                  schedule.end();
            }
            // create new OPL model instance
            schedule = new IloOplModel(def,cp);
            deadline++; // change data
            data.deadline = deadline; // change data
            schedule.addDataSource(data); // add data to model instance
            schedule.generate(); // generate the new OPL model instance
      }
      ofile.close();
      ofile1.close();
  }
```

A data file is also needed to initialize data defined in the model file. The following code snippet is for Project 1.

```
/**********************************************
 * OPL 12.6.3.0 Data
 * Purpose: data file for Project 1
*********************************************/

  SheetConnection sheet("Project 1.xlsx");
  NTasks from SheetRead(sheet, "'Basic Info.'!B1");
  NRscrs from SheetRead(sheet, "'Basic Info.'!B2");
  Rsrc = [18];
  deadline = 20;
  Tasks from SheetRead(sheet, "Act.!A2:B26");
  Orderings from SheetRead(sheet, "Order!A2:B37");
  Methods from SheetRead(sheet, "Method!A2:D50");
  RequiredRsrcs from SheetRead(sheet, "Rsrc!A2:D50");
```

After macro modes of all projects are generated, they are combined into a multi-mode resource-constrained project scheduling problem (MRCPSP) model to minimize the total cost subject to total available resources. The detailed code for the MRCPSP model is shown as follows.

```
/*********************************************
 * OPL 12.6.3.0 Model
 * Purpose: Multi-project scheduling, MRCPSP model
*********************************************/
  using CP;
  // Data declaration
  int NbProjs = ...;
  int NbRsrcs = ...;
  range RsrcIds = 0..NbRsrcs-1;
  range ProjIds = 0..NbProjs-1;
  int Rsrc[RsrcIds] = ...;
  int indirectCost[ProjIds] = ...;
  int indirectShopCost = ...;

  tuple Project {
    key int ID;
    int nmodes;
  }
  {Project} Projects = ...;

  tuple MacroMode {
    key int projId;
    key int id;
    int dur;
    int cost;
    int nsubProj;
  }
  {MacroMode} MacroModes = ...;

  // MacroMode is divided into several sub-projects which are with constant
resource demand over their own duration.
  tuple subProj {
      key int projId;
      key int subProjId;
      key int modeId;
      int dur;
      int succ;
  };
  {subProj} subProjs = ...;

    // Define resource demand for each sub-project
  tuple RequiredRsrc {
    key int projId;
    key int subProjId;
    key int modeId;
    key int RsrcId;
    int dm_R;
  }
  {RequiredRsrc} RequiredRsrcs = ...;
```

```
  // decision variables
  dvar interval project[p in Projects];
  dvar interval macromode[m in MacroModes] optional size m.dur;
  dvar interval subproj[sub in subProjs] optional size sub.dur;

  cumulFunction Q_UsedRsrc[r in RsrcIds] = sum (rsrc in RequiredRsrcs, sub in
subProjs, m in MacroModes: m.projId == sub.projId && m.id == sub.modeId &&
rsrc.projId == m.projId && r == rsrc.RsrcId && sub.subProjId ==
rsrc.subProjId && rsrc.modeId == m.id)  pulse (subproj[sub], rsrc.dm_R);

  cumulFunction UsedRsrc_proj[r in RsrcIds, p in ProjIds] = sum (rsrc in
RequiredRsrcs, sub in subProjs, m in MacroModes: m.projId == sub.projId &&
m.id == sub.modeId && rsrc.projId == m.projId && r == rsrc.RsrcId &&
sub.subProjId == rsrc.subProjId && rsrc.modeId == m.id && m.projId == p+1)
pulse (subproj[sub], rsrc.dm_R);

  // Calculate the direct cost
  dexpr int cost = sum(m in MacroModes)(presenceOf(macromode[m]) * m.cost);
  // Calculate the duration and end time for each project
  dexpr int duration[projId in ProjIds] = sum(m in MacroModes: m.projId ==
projId+1)(presenceOf(macromode[m]) * m.dur);
  dexpr int End[projId in ProjIds] = max(p in Projects: p.ID == projId+1)
endOf(project[p]);

  execute {
      cp.param.FailLimit = 10000000;
      cp.param.TimeLimit = 300;
  }

  // minimize total cost (direct cost + project indirect cost + shop indirect
cost)
  minimize(cost + sum(projId in ProjIds)duration[projId]*indirectCost[projId]
+ indirectShopCost * max(m in MacroModes) endOf(macromode[m]));
  subject to {
      // Alternative mode
      forall (p in Projects)
        alternative(project[p], all(m in MacroModes: m.projId == p.ID)
macromode[m]);
      // Make sure when one macromode is selected, the corresponding sub-
projects are selected as well.
      forall (m in MacroModes, sub in subProjs: sub.projId == m.projId &&
sub.modeId == m.id)
        presenceOf(macromode[m]) == presenceOf(subproj[sub]);
      // Resource limit
      forall (r in RsrcIds)
        Q_UsedRsrc[r] <= Rsrc[r];
      // Link macromode with subprojects
      forall (sub in subProjs, m in MacroModes: sub.projId == m.projId &&
sub.modeId == m.id && sub.subProjId == 1)
        startAtStart(macromode[m], subproj[sub]);
      forall (sub in subProjs, m in MacroModes: sub.projId == m.projId &&
sub.modeId == m.id && sub.subProjId == m.nsubProj)
        endAtEnd(macromode[m], subproj[sub]);
      // Link different subprojects
      forall (sub in subProjs, m in MacroModes: sub.succ != 0)
        startAtEnd(subproj[<sub.projId,sub.succ,sub.modeId>], subproj[sub]);
  }
```

```
// write scheduling results to file

  execute {
      writeln("Total cost = ", cp.getObjValue());
      writeln("Total duration = ", duration);
      writeln("Rsrc = ", UsedRsrc_proj);
      writeln("End time = ", End);
      writeln("Total direct cost = ", cost);
      var ofile = new IloOplOutputFile("Rsrc_MultiProj.txt");

      ofile.writeln("ProjectID\tST\tDuration\tResource");
      for (var sub in subProjs) {
            if (subproj[sub].present){
                  for (var rsrc in RequiredRsrcs){
                        if (rsrc.subProjId == sub.subProjId && rsrc.projId ==
sub.projId && rsrc.modeId == sub.modeId)
                              ofile.writeln(sub.projId + " \t " +
subproj[sub].start + "\t" + sub.dur + "\t" + rsrc.dm_R);
                  }
            }
      }
      for (var projId in ProjIds){
            ofile.writeln(projId+1, "\t", End[projId], "\t0\t0")
      }
  }
```

The data file to initialize this model is shown as follows. By running the data file with the MRCPSP model, the start time, finish time, and resources allocated are determined for each project.

```
/*********************************************
 * OPL 12.6.3.0 Data
 * Purpose: data file for multi-project scheduling
 *********************************************/
  NbProjs = 3;
  NbRsrcs = 1;
  Rsrc = [18]; // daily resource availability
  indirectShopCost = 3000; // daily shop indirect cost

  // daily project indirect cost for project 1, 2, and 3 respectively
  indirectCost = [840, 1306, 1306];

  // number of macro modes for each project
  Projects = {
      <1, 8>,
      <2, 9>,
      <3, 9>,
  };
  // connect spreadsheet for reading information
  SheetConnection sheet("CaseStudyCh4.xlsx");
  MacroModes from SheetRead(sheet, "Mode!A2:E27");
  subProjs from SheetRead(sheet, "SubProj!A2:E332");
  RequiredRsrcs from SheetRead(sheet, "Rsrc!A2:E332");
```

At the lower level, the project manager is responsible for performing further activity scheduling subject to the allocated resources from the upper level. The programming code for the lower-level individual project scheduling is presented as follows. In this model, the resource availability is time-dependent.

```
/*********************************************
 * OPL 12.6.3.0 Model
 * Purpose: individual project scheduling subject to allocated resources
 *********************************************/
  using CP;
  // Data declaration
  int NTasks = ...;
  int NRscrs = ...;
  range RsrcIds = 0..NRscrs-1;
  int indirectCost = ...;

  tuple Task {
    key string id;
    int nmethods;
  }
  {Task} Tasks = ...;

  tuple Ordering {
    string pred;
    string succ;
 }
 {Ordering} Orderings = ...;

  tuple Method {
    key string taskId;
    key int id;
    int dur;
    int cost;
  }
  {Method} Methods = ...;

  tuple RequiredRsrc {
    key string taskId;
    key int methodId;
    key int RsrcId;
    int dm_R;
  }
  {RequiredRsrc} RequiredRsrcs = ...;

  // decision variables
  dvar interval task[t in Tasks];
  dvar interval method[m in Methods] optional size m.dur;

  cumulFunction Q_UsedRsrc[r in RsrcIds] = sum (m in Methods, rsrc in
RequiredRsrcs: m.taskId == rsrc.taskId && m.id == rsrc.methodId && r ==
rsrc.RsrcId)  pulse (method[m], rsrc.dm_R);
```

```
  dexpr int duration = max(t in Tasks) endOf(task[t]);
  dexpr int cost = sum(m in Methods)(presenceOf(method[m]) * m.cost * m.dur);

  execute {
      cp.param.FailLimit = 10000000;
      cp.param.TimeLimit = 30000;
  }

  //minimize project duration
  minimize (duration);

  subject to {
      // Alternative mode
      forall (t in Tasks)
        alternative(task[t], all(m in Methods: m.taskId == t.id) method[m]);

      // Resource limit, time-dependent resource availability
      forall (r in RsrcIds){
            alwaysIn(Q_UsedRsrc[r], 0, 2, 0, 11);
            alwaysIn(Q_UsedRsrc[r], 2, 9, 0, 15);
            alwaysIn(Q_UsedRsrc[r], 9, 15, 0, 18);
            alwaysIn(Q_UsedRsrc[r], 15, 16, 0, 17);
            alwaysIn(Q_UsedRsrc[r], 16, 17, 0, 15);
            alwaysIn(Q_UsedRsrc[r], 17, 19, 0, 8);
            alwaysIn(Q_UsedRsrc[r], 19, 21, 0, 7);
      }
      // Predecessor
      forall (o in Orderings: o.succ != "Z")
        endBeforeStart(task[<o.pred>],task[<o.succ>]);
  }

  // save scheduling results to a file
  execute {
      writeln("Total Duration = ", duration);
      writeln("Total Cost = ", cp.getObjValue());

      //var ofile = new IloOplOutputFile("ActSchwithMinDuration.txt");
      var ofile = new IloOplOutputFile("LowerLevel.txt");
      //var ofile = new IloOplOutputFile("ActSchwithMinCost.txt");
      ofile.writeln("Duration\tCost");
      ofile.writeln(duration, "\t", cost);
      ofile.writeln("TaskID\tModeID\tStart\tFinish");
      for (var m in Methods) {
            if (method[m].present)
              ofile.writeln(m.taskId + "\t" + m.id + "\t" + method[m].start +
"\t" + method[m].end);
      }
      ofile.close();

      var ofile1 = new IloOplOutputFile("Rsrc.txt");
      ofile1.writeln("RsrcId\tStep function")
      for (var j in RsrcIds){
            ofile1.writeln(j, "\t", Q_UsedRsrc[j]);
      }
      ofile1.close();
  }
```

The result shown in Figure 4.5 is derived by initializing the model using the following data file. When inputting data file for the other two projects, the project schedules can also be generated for the other two projects.

```
/*********************************************
 * OPL 12.6.3.0 Data
 * Purpose: data file for project 1 at the lower level
*********************************************/
  SheetConnection sheet("Project1.xlsx");
  NTasks from SheetRead(sheet, "'Basic Info.'!B1");
  NRscrs from SheetRead(sheet, "'Basic Info.'!B2");
  indirectCost = 840;

  Tasks from SheetRead(sheet, "Act.!A2:B26");
  Orderings from SheetRead(sheet, "Order!A2:B37");
  Methods from SheetRead(sheet, "Method!A2:D50");
  RequiredRsrcs from SheetRead(sheet, "Rsrc!A2:D50");
```

APPENDIX B: DOCUMENTATION OF THE DEVELOPED MS PROJECT ADD-ON

This document (1) describes the architecture and design for the developed scheduling add-on and (2) gives a general procedure to show how to use this prototype. The scheduling add-on is an MS project based addon that can derive production schedules for bridge girder fabrication. The add-on gets inputs from an MS project file, performs schedule optimization using the inbuilt optimizer (i.e., a constraint programming optimizer), and outputs the optimized schedule to the MS project file.

B.1 Architecture and Design

The purpose of this section is to describe the architecture and design of the scheduling add-on in a way that addresses the interests and concerns of developers only. The developers want an architecture that will minimize complexity and re-development effort.

In this section, the main functional components of the system are described. Relationships between components and their interactions are also presented. The modules of the system are first expressed in terms of high-level components (architecture) and progressively refined into more detailed components and eventually classes with specific attributes and operations.

The architecture consists of 5 major components, as shown in Figure B.1:

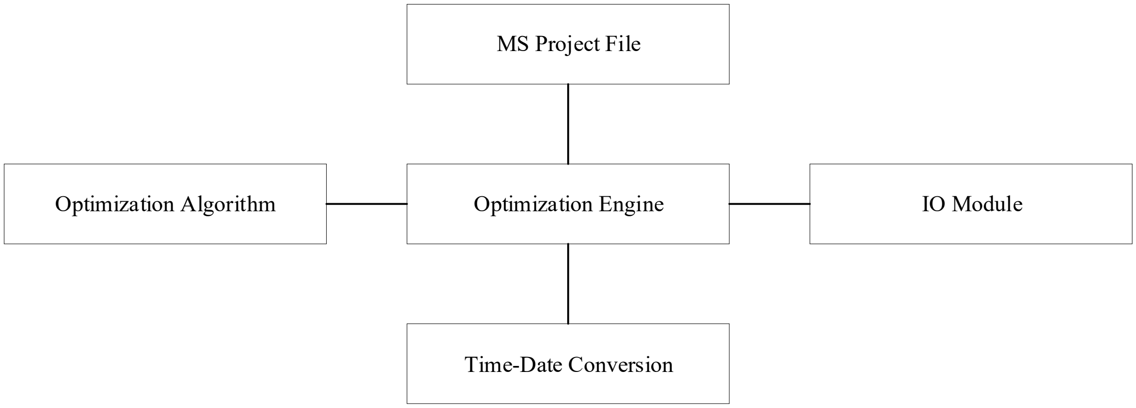


Figure B.1: System Architecture.

- The **MS Project File** provides (1) inputs on the detailed fabrication activities with durations, involved workstations, predecessors; and (2) an output platform to present the optimized schedule. An MS Project template is developed for standardizing the input format. The
- The **IO Module** (1) reads resource limits of each type and basic information of the MS Project file (e.g., project start time, working hours per day) and defined fabrication activities (including activity ID, duration, predecessors, and deadline) from the MS Project file; and attaches hard-coded, experience-based resource requirements for each fabrication activity; (2) writes optimized schedule back to the MS Project file.
- The **Optimization Algorithm** is the algorithm used to perform optimization. For now, the APIs of IBM CP Optimizer is used. It can be substituted by other optimization algorithm such as OR-Tools from Google (https://developers.google.com/optimization/).
- Given a relative time from the optimization engine, the **Time-Date Conversion** will convert the time point to the validate working date.

- The **Optimization Engine** is the main component of the prototype. It gets input from the MS project file, performs scheduling optimization using the optimization algorithm, and outputs the optimized schedule to the MS project file.

The UML sequence diagram presented in Figure B.2Figure B.2: UML Sequence Diagram.

shows the interactions between related objects.

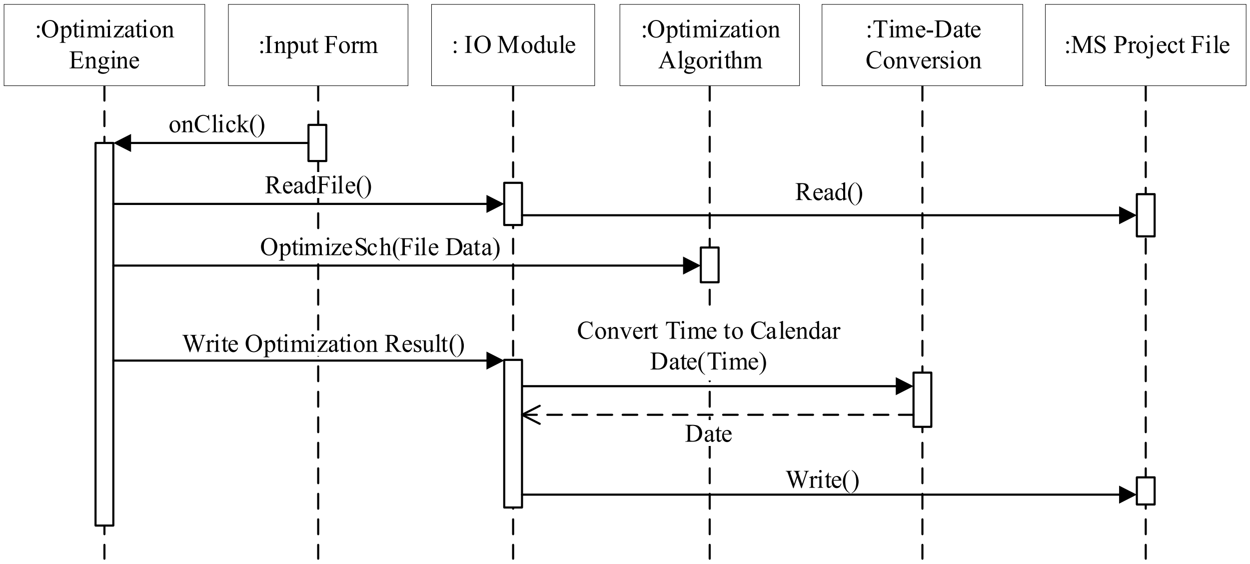


Figure B.2: UML Sequence Diagram.

Detailed Class Design

Input Form

The class "*OptiByStation*" in file "*OptiByStation.cs*" is used to initial the input form and save inputted information through the form.

IO Module

The IO model (including the file Data.cs and DataIO.cs) is defined under the “Data” Namespace. There are four classes in the *Data.cs* file. One “*ActivityList*” is to define the structure of activity list. Its class diagram is shown Figure B.3.

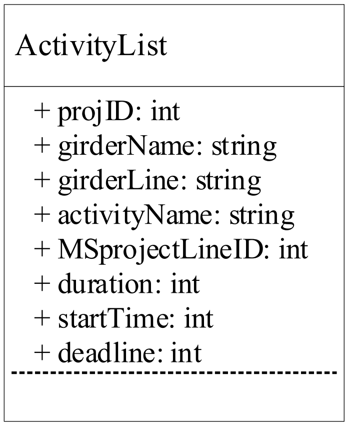


Figure B.3: Class diagram for class “*ActivityList*”.

Another one “*Ordering*” is to define the composition of precedence relationships. Its class diagram is shown in Figure B.4. The variable “*Type*” defines the relationships between the two activities (i.e., pre and succ). Four types relationships can be defined: FF (finish to finish), FS (finish to start), SS (start to start), and SF (start to finish). The variable “*lag*” specifies the lag time between the two activities.

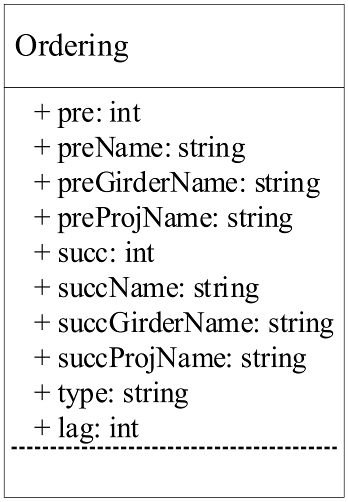


Figure B.4: Class diagram for class "*Ordering*".

Another class "*MSprojectLayout*" records the organization of the input MS project file. It records the start line ID and end line ID of each girder. The class diagram is shown in Figure B.5

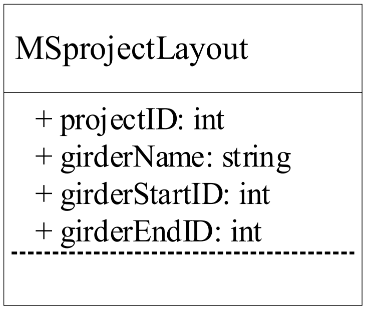


Figure B.5: Class diagram for class "*MSprojectLayout*".

Another class "*MSprojectInfo*" saves the basic information (as shown in Figure B.6) of the input MS project file.

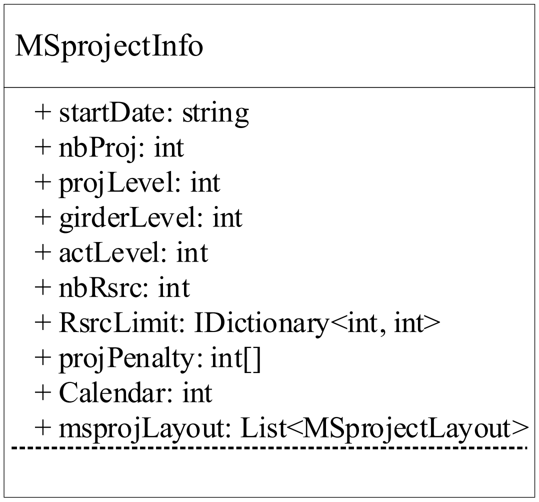


Figure B.6: Class diagram for class "*MSprojectInfo*".

Variables *"projLevel", "girderLevel", "actLevel"* specify the task level of the project, girder, and activity in the MS project file. By default, project level is 1, girder level is 2, activity level is 3; as shown in Figure B.7. The variable "*Calendar*" is the working hour per day.

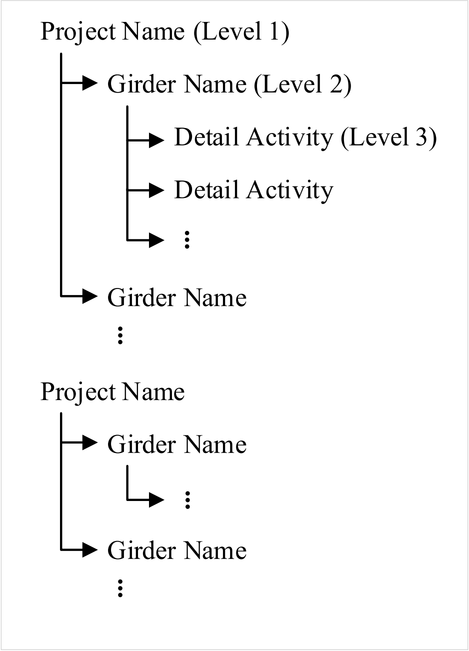


Figure B.7: The task arrangement in the input file.

There is one class (i.e., *DataFactory*) in the *DataIO.cs* file. The class diagram is presented in Figure B.8. It has two public functions and 5 private functions.

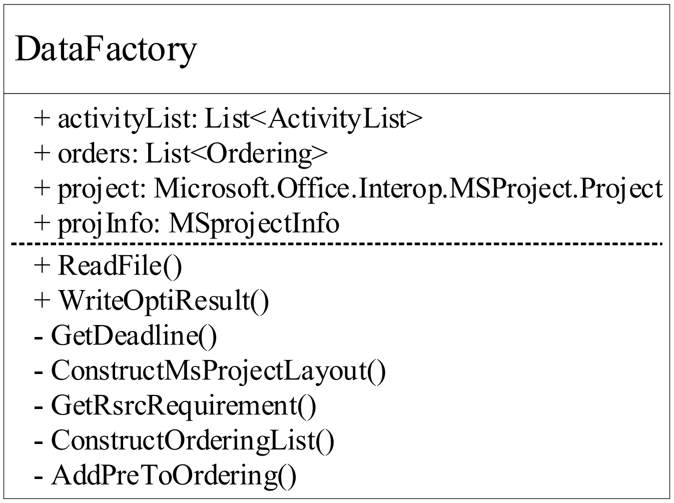


Figure B.8: Class diagram for class “*DataFactory*”.

“*ReadFile()*” function is to read required information (i.e., *MSprojectInfo, ActivityList, Ordering*) from the MS Project file. This function calls the five private functions for getting respective information: *GetDeadline()* attaches project deadline to each activity and also converts the deadline date to relative time duration; *ConstructMsProjectLayout()* gets the layout information about the MS Project file; *GetRsrcRequirement()* attaches the in-built resource requirements to each activity, the in-built resource requirements are based on the setting in the studied fabrication shop and shown in the Table B.1. Functions “*ConstructOrderingList()*” and “*addPreToOrdering()*” are to construct the precedence relationship lists and add required information (e.g., project name, girder name) to predecessors.

Table B.1: Inbuilt resource requirements.

Activity	Workstation	Labor
Cutting Flanges	Flange Cutting Station	2 journeymen
Straightening Top/Bottom Flanges	Flange Straightening Station	2 journeymen
Splicing Top/Bottom Flanges	Flange Splicing Station	1 journeyman
Splicing Web	Web Splicing Station	2 journeymen
Assembly Girder	Girder Assembly Station	2 journeymen
Welding Girder	Girder Welding Station	2 journeymen
Welding Stiffener	Stiffener Welding Station	2 journeymen
Splicing Girder	Girder Splicing Station	2 journeymen
Sandblast	Sandblast Station	2 sandblastmen

Optimization Engine

It corresponds to the file "*OptimizaitonEngine.cs*". Within this class, there are two functions, as shown in Figure B.9. Function "*OptimizationSch(fileData)*" is to control the program for reading data, optimizing schedules, and writing data back. The private function "*OptimizaitonAlgorithm(fileData)*" is to do the optimization work by using the IBM CP Optimizer APIs.

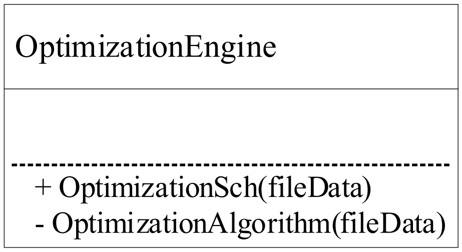


Figure B.9: Class diagram for class "*OptimizaitonEngine*".

TimeDateConversion class

It corresponds to the file "*TimeDateConversion.cs*". Within this class, there are two functions, as shown in Figure B.10. Function "*GetNumberofWorkingDays(start, end)*" is to convert a date (e.g., project deadline) to a relative time duration. The function "*GetEndDaysIncludingNonWorkingDays(start, time, calendar)*" is to convert a relative time ("*time*") to a calendar date.

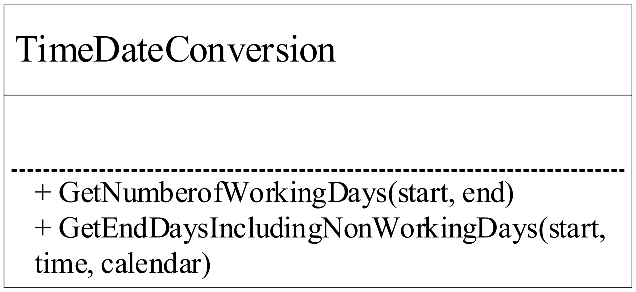


Figure B.10: Class diagram for class "*TimeDateConversion*".

B.2 User Manual

The add-on is customized for one shop (i.e., SSB) of the partner company. After installing the add-on, one tab named "*Bridge Sch. Generation*" will show on the ribbon of MS project, as shown in Figure B.11. It has two buttons, namely "*SSB Scheduler*" and "*Schedule in Shop View*".

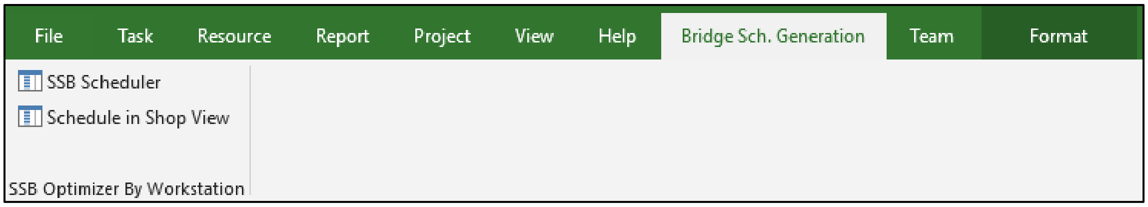
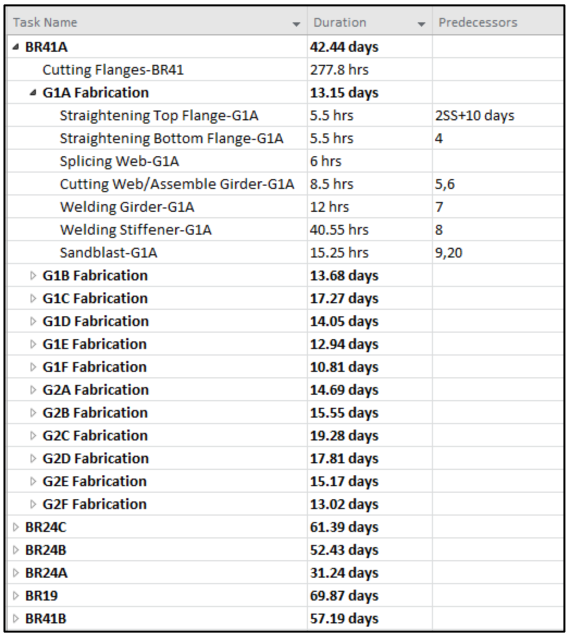


Figure B.11: Screenshot of the developed MS Project Add-on.

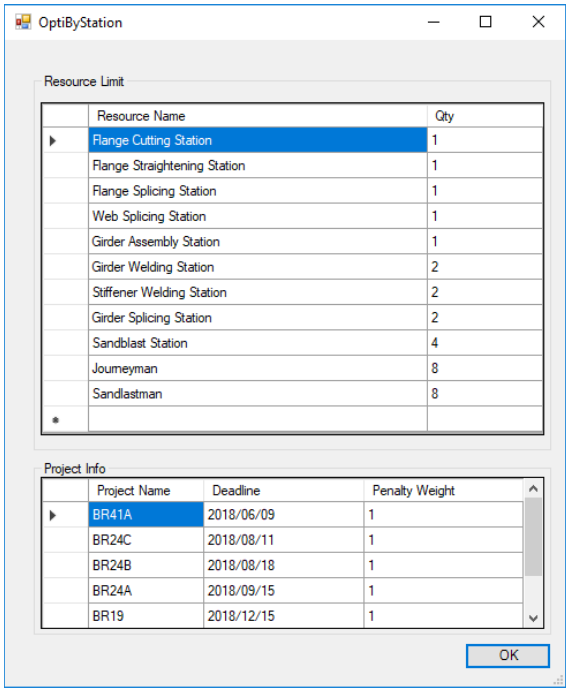
To use the add-on, users need to define detailed fabrication activities based on the provided MS Project template. Figure B.12 gives an example about the contents in the MS Project file. Users need to modify the activities and specify activity duration, predecessors, and deadlines for each project based on their fabrication practice. This file contains six bridges with different deadlines and different numbers of girders.



Task Name	Duration	Predecessors
BR41A	**42.44 days**	
Cutting Flanges-BR41	277.8 hrs	
G1A Fabrication	**13.15 days**	
Straightening Top Flange-G1A	5.5 hrs	2SS+10 days
Straightening Bottom Flange-G1A	5.5 hrs	4
Splicing Web-G1A	6 hrs	
Cutting Web/Assemble Girder-G1A	8.5 hrs	5,6
Welding Girder-G1A	12 hrs	7
Welding Stiffener-G1A	40.55 hrs	8
Sandblast-G1A	15.25 hrs	9,20
G1B Fabrication	**13.68 days**	
G1C Fabrication	**17.27 days**	
G1D Fabrication	**14.05 days**	
G1E Fabrication	**12.94 days**	
G1F Fabrication	**10.81 days**	
G2A Fabrication	**14.69 days**	
G2B Fabrication	**15.55 days**	
G2C Fabrication	**19.28 days**	
G2D Fabrication	**17.81 days**	
G2E Fabrication	**15.17 days**	
G2F Fabrication	**13.02 days**	
BR24C	**61.39 days**	
BR24B	**52.43 days**	
BR24A	**31.24 days**	
BR19	**69.87 days**	
BR41B	**57.19 days**	

Figure B.12: Example of the input MS Project file.

The button “SSB Scheduler” is designed to schedule multiple concurrent bridge girder fabrication projects by using the proposed multi-project scheduling framework in this study. By clicking the button “SSB Scheduler”, a form (as shown in Figure B.13) pops-up for users to define resource limits and project related information. Resources include workstations and labors, as illustrated in the case study of Chapter 5. Users can modify the available quantities of each resource based on the fabrication facility setting. For the project information, the columns “Project Name” and “Deadline” are initialized by using information specified in the MS Project file. Users can also modify the deadline from this form. The column “Penalty Weight” defines the relative importance of various projects, which serves as a priority index for resource allocation. By default, all projects are equally important.



Resource Limit

Resource Name	Qty
Flange Cutting Station	1
Flange Straightening Station	1
Flange Splicing Station	1
Web Splicing Station	1
Girder Assembly Station	1
Girder Welding Station	2
Stiffener Welding Station	2
Girder Splicing Station	2
Sandblast Station	4
Journeyman	8
Sandlastman	8

Project Info

Project Name	Deadline	Penalty Weight
BR41A	2018/06/09	1
BR24C	2018/08/11	1
BR24B	2018/08/18	1
BR24A	2018/09/15	1
BR19	2018/12/15	1

Figure B.13: Screenshot of the input form.

After finishing the input settings, click the “OK” button to start the resource allocation process. The in-built resource requirements shown in Table B.1 are first attached to fabrication activities defined in the MS Project file. Then the optimization engine schedules the start time of each activity subject to the limited resources and pre-defined deadlines. The optimized results are finally written back to the MS Project file for visualization and communication. The resulting resource usage can also be visualized through MS Project. Figure B.14 presents the resulting monthly resource usage throughout the total duration.

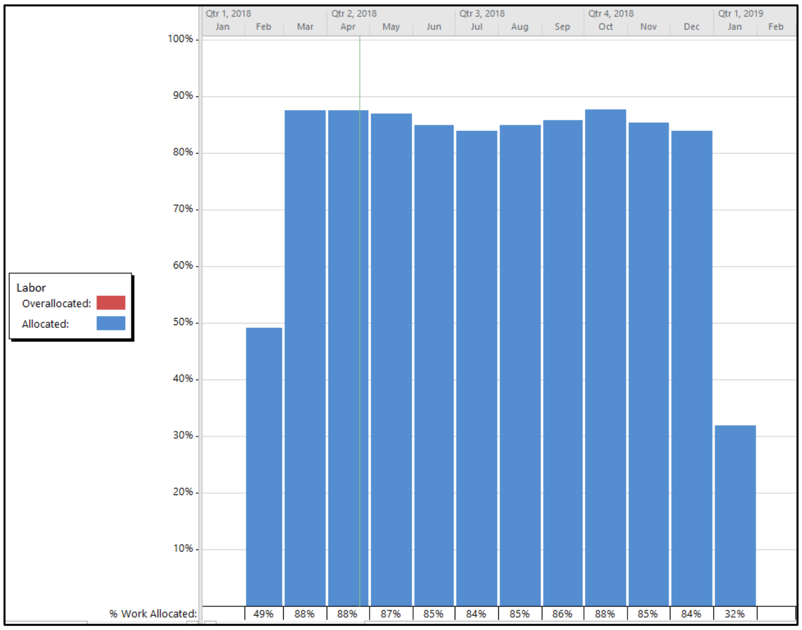
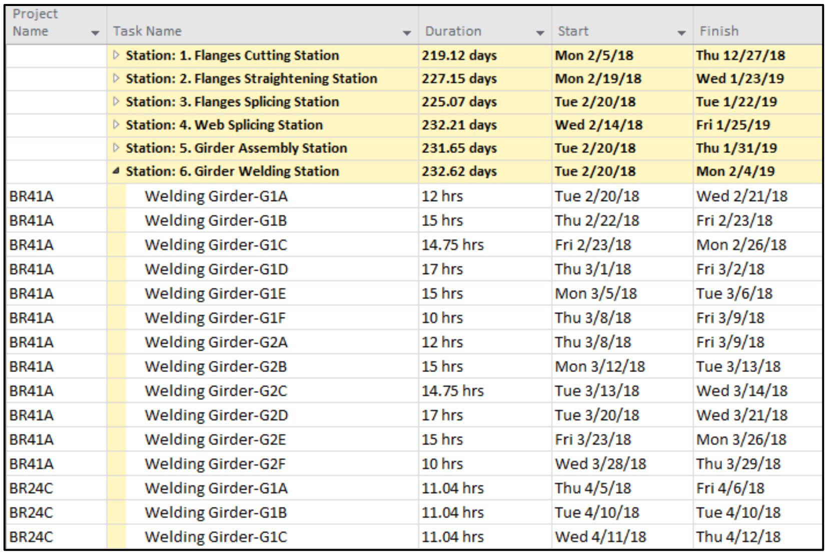


Figure B.14: Screenshot of the resulting monthly resource use graph.

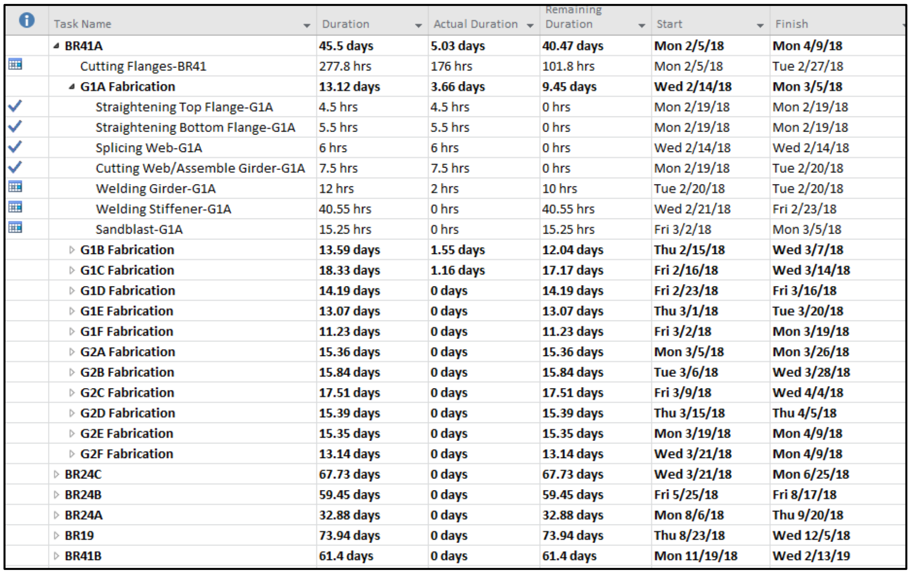
The resulting schedule can also be switched into shop view by clicking the button "Schedule in Shop View", as shown in Figure B.15. Schedule in the shop view shows the girder flow going through each workstation, which can provide guidance on daily work in the fabrication facility.



Project Name	Task Name	Duration	Start	Finish
	Station: 1. Flanges Cutting Station	**219.12 days**	**Mon 2/5/18**	**Thu 12/27/18**
	Station: 2. Flanges Straightening Station	**227.15 days**	**Mon 2/19/18**	**Wed 1/23/19**
	Station: 3. Flanges Splicing Station	**225.07 days**	**Tue 2/20/18**	**Tue 1/22/19**
	Station: 4. Web Splicing Station	**232.21 days**	**Wed 2/14/18**	**Fri 1/25/19**
	Station: 5. Girder Assembly Station	**231.65 days**	**Tue 2/20/18**	**Thu 1/31/19**
	Station: 6. Girder Welding Station	**232.62 days**	**Tue 2/20/18**	**Mon 2/4/19**
BR41A	Welding Girder-G1A	12 hrs	Tue 2/20/18	Wed 2/21/18
BR41A	Welding Girder-G1B	15 hrs	Thu 2/22/18	Fri 2/23/18
BR41A	Welding Girder-G1C	14.75 hrs	Fri 2/23/18	Mon 2/26/18
BR41A	Welding Girder-G1D	17 hrs	Thu 3/1/18	Fri 3/2/18
BR41A	Welding Girder-G1E	15 hrs	Mon 3/5/18	Tue 3/6/18
BR41A	Welding Girder-G1F	10 hrs	Thu 3/8/18	Fri 3/9/18
BR41A	Welding Girder-G2A	12 hrs	Thu 3/8/18	Fri 3/9/18
BR41A	Welding Girder-G2B	15 hrs	Mon 3/12/18	Tue 3/13/18
BR41A	Welding Girder-G2C	14.75 hrs	Tue 3/13/18	Wed 3/14/18
BR41A	Welding Girder-G2D	17 hrs	Tue 3/20/18	Wed 3/21/18
BR41A	Welding Girder-G2E	15 hrs	Fri 3/23/18	Mon 3/26/18
BR41A	Welding Girder-G2F	10 hrs	Wed 3/28/18	Thu 3/29/18
BR24C	Welding Girder-G1A	11.04 hrs	Thu 4/5/18	Fri 4/6/18
BR24C	Welding Girder-G1B	11.04 hrs	Tue 4/10/18	Tue 4/10/18
BR24C	Welding Girder-G1C	11.04 hrs	Wed 4/11/18	Thu 4/12/18

Figure B.15: Screenshot of schedule in shop view.

The add-on can also continuously update the schedule based on actual fabrication progress. It requires users enter actual duration happened and remaining duration still needed in the MS Project file (as shown in Figure B.16) and then re-run the add-on. The add-on will update the schedule from the current date.



ℹ	Task Name	Duration	Actual Duration	Remaining Duration	Start	Finish
	BR41A	**45.5 days**	**5.03 days**	**40.47 days**	**Mon 2/5/18**	**Mon 4/9/18**
▦	Cutting Flanges-BR41	277.8 hrs	176 hrs	101.8 hrs	Mon 2/5/18	Tue 2/27/18
	G1A Fabrication	**13.12 days**	**3.66 days**	**9.45 days**	**Wed 2/14/18**	**Mon 3/5/18**
✓	Straightening Top Flange-G1A	4.5 hrs	4.5 hrs	0 hrs	Mon 2/19/18	Mon 2/19/18
✓	Straightening Bottom Flange-G1A	5.5 hrs	5.5 hrs	0 hrs	Mon 2/19/18	Mon 2/19/18
✓	Splicing Web-G1A	6 hrs	6 hrs	0 hrs	Wed 2/14/18	Wed 2/14/18
✓	Cutting Web/Assemble Girder-G1A	7.5 hrs	7.5 hrs	0 hrs	Mon 2/19/18	Tue 2/20/18
▦	Welding Girder-G1A	12 hrs	2 hrs	10 hrs	Tue 2/20/18	Tue 2/20/18
▦	Welding Stiffener-G1A	40.55 hrs	0 hrs	40.55 hrs	Wed 2/21/18	Fri 2/23/18
▦	Sandblast-G1A	15.25 hrs	0 hrs	15.25 hrs	Fri 3/2/18	Mon 3/5/18
	G1B Fabrication	**13.59 days**	**1.55 days**	**12.04 days**	**Thu 2/15/18**	**Wed 3/7/18**
	G1C Fabrication	**18.33 days**	**1.16 days**	**17.17 days**	**Fri 2/16/18**	**Wed 3/14/18**
	G1D Fabrication	**14.19 days**	**0 days**	**14.19 days**	**Fri 2/23/18**	**Fri 3/16/18**
	G1E Fabrication	**13.07 days**	**0 days**	**13.07 days**	**Thu 3/1/18**	**Tue 3/20/18**
	G1F Fabrication	**11.23 days**	**0 days**	**11.23 days**	**Fri 3/2/18**	**Mon 3/19/18**
	G2A Fabrication	**15.36 days**	**0 days**	**15.36 days**	**Mon 3/5/18**	**Mon 3/26/18**
	G2B Fabrication	**15.84 days**	**0 days**	**15.84 days**	**Tue 3/6/18**	**Wed 3/28/18**
	G2C Fabrication	**17.51 days**	**0 days**	**17.51 days**	**Fri 3/9/18**	**Wed 4/4/18**
	G2D Fabrication	**15.39 days**	**0 days**	**15.39 days**	**Thu 3/15/18**	**Thu 4/5/18**
	G2E Fabrication	**15.35 days**	**0 days**	**15.35 days**	**Mon 3/19/18**	**Mon 4/9/18**
	G2F Fabrication	**13.14 days**	**0 days**	**13.14 days**	**Wed 3/21/18**	**Mon 4/9/18**
	BR24C	**67.73 days**	**0 days**	**67.73 days**	**Wed 3/21/18**	**Mon 6/25/18**
	BR24B	**59.45 days**	**0 days**	**59.45 days**	**Fri 5/25/18**	**Fri 8/17/18**
	BR24A	**32.88 days**	**0 days**	**32.88 days**	**Mon 8/6/18**	**Thu 9/20/18**
	BR19	**73.94 days**	**0 days**	**73.94 days**	**Thu 8/23/18**	**Wed 12/5/18**
	BR41B	**61.4 days**	**0 days**	**61.4 days**	**Mon 11/19/18**	**Wed 2/13/19**

Figure B.16: Screenshot of schedule update.