CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Automated Look-Ahead Schedule Generation and Optimization for the Finishing Phase of Complex Construction Projects

By

Ning Dong

CIFE Technical Report #TR211
June 2012

STANFORD UNIVERSITY

If you would like to contact the authors, please write to:

c/o CIFE, Civil and Environmental Engineering Dept.,
Stanford University
The Jerry Yang & Akiko Yamazaki Environment & Energy Building
473 Via Ortega, Room 292, Mail Code: 4020
Stanford, CA 94305-4020

AUTOMATED LOOK-AHEAD SCHEDULE GENERATION AND OPTIMIZATION FOR THE FINISHING PHASE OF COMPLEX CONSTRUCTION PROJECTS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF

CIVIL AND ENVIRONMENTAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Ning Dong

June 2012

Abstract

Look-ahead schedules (LASs) are the last opportunity for general contractors to allocate resources for maximum effectiveness. Unfortunately, in the finishing phase of complex construction projects, project planners, site engineers, and construction engineers struggle to use LASs to effectively organize and allocate limited project resources such as crews and rooms on a daily basis because (1) the LAS generation process is time-consuming, even with the help of the existing commercial tools; (2) the LASs created are error-prone when the site engineers and project planners need to consider constraints including precedence constraints, room and crew availabilities, and engineering constraints, such as zone and blocking constraints; (3) there is no way to tell whether the LASs created are the best means by which to achieve specific project goals, such as shortest project duration and minimum project cost, even if accurate LASs can be quickly generated.

This dissertation describes an integrated approach I have developed to automating LAS generation and quickly discovering optimized LASs in the finishing phase of a complex building project. The approach builds on three theoretical foundations: automated construction schedule generation, computer simulation, and artificial intelligence for schedule optimization. The approach consists of an automated LAS generation (ALASG) method that ensures the rapid creation of error-free LAS. Coupled with computer simulation and an optimization method based on a genetic algorithm (GA), the ALASG method also finds near-optimal LAS quickly. The ALASG method is composed of an information model that integrates the project databases at the appropriate levels of detail to facilitate the sound formation of operations and the consideration of constraints and a LAS generation process model that simulates the daily LAS generation process on site. The GA-based optimization method interacts with the information model and the process model to create LASs optimized towards specific project goals. I have also implemented a software prototype based on the ALASG method and the GA-based optimization methods. The results from the use of this prototype in student and engineer design charrettes and two comparison studies provide evidence for the power of this approach to construct more high-quality LASs faster.

The dissertation includes three interrelated papers. The first paper, chapter 2, describes the method for automating LAS generation for the finishing phase of complex projects based on information modeling, process modeling, and simulation methods. This chapter identifies “room” as a core component for LAS generation and depicts different perspectives from which to view the room. That is, from the product perspective, a room is a part of the final building product to be used by end users; from the process perspective it belongs to a certain fragnet, and from the resource perspective, it is a type of resource. The chapter also describes the implementation of the prototype. Based on this prototype, the second paper, chapter 3, discusses the practical value of the prototype and its possible applications in the construction industry. Specifically, I define measurements for resource utilization and then evaluate its relation to project goals in the finishing phase of complex projects. The third paper, chapter 4, presents a GA-based method of finding optimized LASs. In addition to addressing the traditional constraints, such as operation precedence constraints and resource availability, it considers three key practical aspects that project planners and construction managers encounter frequently on site: the engineering priorities of each individual room, the zone constraint, and the blocking constraint. To encompass these aspects, the GA-based method interacts with the information model and the process model described in the first paper.

Collectively, these three papers illustrate an automated and integrated method that liberates site engineers and project planners from the tedious and time-consuming LAS generation process and provides them with accurate work assignments to guide field work so that they can channel their time and energy towards other project tasks. This dissertation is one of the few studies in the field of construction schedule automation and optimization to date that (1) addresses automatied LAS generation for the finishing phase of complex projects and (2) explores LAS optimization in light of engineering constraints.

Acknowledgments

A decade ago, when I had just gotten my Bachelor's in civil engineering and was ready to start my graduate study in computer science in Beijing, I knew I would not settle for just staying in the capital of China and getting a white-collar job. I dreamed of going to other countries and bringing technological innovation back to China's construction industry, which has been indolent in adopting new technologies, especially information technology, to improve productivity. But I never thought that someday I would carry out research in construction automation across four continents while learning from world experts in that field at Stanford University.

This wildly escalated version of my dream would not have come true had I not been accepted into Professor Martin Fischer's doctoral group. As my principal advisor, Professor Fischer has instilled in me a vision of and passion for construction innovations that have been waiting for the chance to be realized through collaborations between industry and academia. Professor Fischer encouraged me to start by investigating practical problems in construction management, introduced me to the Consolidated Contractor's Company (CCC) with whom I worked on such investigations, gave me wise advice at virtually every step in my research, and encouraged me all along the way. I am also greatly indebted to his understanding and support at the last stage of my research when I had to take leave for almost a year to care for my family after my dad was diagnosed with stage IV gastric cancer. Professor Fischer probably has no idea how inspirational his care for his students can become during such difficult times.

I have also been extremely fortunate to have Professor Raymond Levitt as a committee member. I am always amazed by his sharp eye for discerning the missing elements in my work and am continually intrigued by his gift for conveying his ideas with graceful precision, warm encouragement, and concrete suggestions. I am still trying to figure out how he rapidly processes complex and abstract concepts, and then asks extremely relevant questions on the basis of his deep theoretical knowledge. He sets such a high standard for scholarly work that it motivates me to keep improving and learning.

In addition, without Professor Dongdong Ge I wouldn't have been able to untangle the schedule optimization problem that is intertwined with complex constraints, which are critical to my research. Professor Ge had crystal clear insight about where my work was heading and generously offered invaluable suggestions. His advice in operations research and artificial intelligence helped me tremendously in developing my optimization approach. I especially appreciate the time he spent with me in iterations to figure out the mathematical formulation for my proposed optimization problem.

I would further like to thank Dr. John Kunz, who provided constant guidance, support, and encouraging feedback throughout my doctoral work. I also appreciate Professor Michael Lepech for serving on my defense committee and Professor Kenneth Waldron for serving as Chair; both provided very helpful comments during my oral exam. My gratitude also goes to Professor Yingyu Ye and Professor Warren Hausman, both from the Management Science and Engineering (MS&E) Department, who have opened the door of operations research for me.

I am very grateful as well for the opportunity to work with a group of extraordinary research colleagues (past and present) in the civil engineering program: Ragip Akbas, Calvin Cam, Henry Chan, Jack Cheng, Cheryl Chi, Sangwoo Cho, Forest Flager, Victor Gane, Matthew Garr, Ju Gao, Timo Hartmann, Stephan Jooste, Atul Khanzode, Jung In Kim, Tae Wan Kim, Tetsu Kondo, Wendy Li, Rene Morkos, Claudio Mourgues, Michael Murray, Forest Peterson, Reid Senescu, Mauricio Toledo, Thomas Wingate, and Peggy Yee. Special thanks also to Jinping Gou and Ramprasath Palanisamy, who participated in a part of my research validation along with some of my research colleagues.

Additionally, I wish to express my gratitude to many talented people from other departments at Stanford. Barry Chai from the Computer Science (CS) Department, who passed away in a tragic car accident in 2010, helped me get started with the graphic user interface of my prototype. Philip Guo, also from CS, helped me figure out how to use Python to develop the first version of that prototype. Jaosn Su, from the Electrical Engineering (EE) Department, introduced to me the powerful functions of MATLAB in data analysis and graphic outputs. Wei Qian, a Stanford alumnus and now a doctoral candidate in statistics at the University of Minnesota, Twin Cities, participated in the statistical analysis of the results created by my prototype. I would also like to thank Ioannis Antonellis, Chris Archibald, Adam Beberg, and Joel Brandt from CS; and John Carlsson, Thomas Chung, and Benjamin Yolken from MS&E for patiently answering my questions. When it comes to scientific writing, I must thank Professor Sandra Drake and in particular Dr. Lauren Rusk who meticulously read and helped me with my writing.

Beyond the ivory tower, field research and collaboration with industry are important parts of my research. This project would not have been such a wonderful and eye-opening experience had I not worked closely with Mr. Zuhair Haddad, my industry advisor and mentor, Vice President of Corporate Affairs and Chief Information Officer at CCC. An expert and visionary in construction automation, Mr. Haddad has enlightened me concerning what innovation means in the context of construction. He also provided amazing opportunities for me to conduct field research at CCC's headquarters in Athens, Greece, as well as on job sites in Cairo, Egypt, and Doha, Qatar. The experience I obtained in these countries not only enabled me to find an appropriate topic for my doctoral project, but also allowed me to communicate with industry experts and uncover the problems my dissertation addresses. Personally, these experiences have given me invaluable chances to learn about diverse cultures, histories, and traditions, and to make life-long friends in several countries.

In particular, I would like to thank Rayan Jreije, Salpie Kechichian, Benjamin Justin, Firas Hijazi, Yasser Hijazi, and Gerard Moubarak from CCC Athens, who were extremely helpful and supportive throughout my exploration in model-based integrated construction management and planning; Watheq Haddad, Andaz Ali, Ziad Baydoun, and Roy Sayegh from the Carnegie Mellon University Campus Project at CCC Qatar, who taught me how to do project planning and control in the finishing phase of complex projects and generously helped validate my research; Ashraf Roshdy, Ahmed El-Hadidy, Amr Manaa, Tamer Maher, Issam El-Absi, and Ziad Bishouty from CCC Cairo, who provided me valuable training in engineering and generous hospitality that made my stay in Cairo extremely rewarding; Mira Al Arab, Rami Santina, and Jean Pierre Wehbe from Beirut, who offered their expertise in programming and helped me partially validate my research. My gratitude also goes to many others at CCC, including Dr. Amr El-Sersy, Ula Papayannopoulou, Suhair Haddad, and George Venetsanopoulos. They helped me in many ways that made my traveling and experience at CCC smooth and pleasant.

I am grateful, moreover, for the generous financial support of both CCC and the Center for Integrated Facility Engineering (CIFE). Special thanks, as well, to Teddie Guenzer from CIFE, who has been constantly supportive and dedicated to making a student's life easy.

Back at Stanford, besides my colleagues, whose friendship provided novel and diverse cultural lessons and inspiring pep talks, I would like to thank many other friends for offering the joy, support, and comfort that added enormous fun and warmth to my journey. For this, I particularly want to thank Jiang Bian, Barry Chai, Xin Du, Zeng Fan, Ruo Ge, Jinping Gou, Po Hu, Ruoting Jiang, Zhuang Liu, Xiang Ma, Weiwei Shen, Bo Wang, Qiang Wang, Shenming Wang, Xing Xie, Bangpeng Yao, Chenhao Zhu, Yuan Zhuang, and Hong Zi.

Most importantly, I would like to honor my family; first my parents, who planted in me the spirit of wonderment about the universe and taught me to take action to pursue my quest. I am also grateful to my mother- and father-in-law, who provided tremendous support during the difficult time when my father was hospitalized. And especially, I thank my wife Zheng Zhuang; I could not have finished my dissertation without her constant support, encouragement, and understanding.

Finally, I would like to dedicate my dissertation to my father, Kechang Dong, and my two grandfathers, Zhenzhi Dong and Qingchun Zhao. I inherited from these great men the qualities of perseverance, patience, and diligence. A mechanical engineer himself, my father used to tell me that it is engineers who change the world. With the knowledge I have obtained from my Ph.D. program, I am ready to embark on a journey to prove that he was right.

Table of Contents

List of Tables

List of Illustrations

Chapter 1: Introduction

This dissertation concerns generating and optimizing look-ahead schedules (LASs) for the finishing phase of complex construction projects for general contractors. Because of the vast amount of data and the complex work sequence and constraints involved in such projects, existing methods are insufficient to help construction managers and site engineers generate LASs quickly, and in enough detail, to serve as a daily guide to organizing the limited resources on site while achieving challenging schedule, budget, quality, and safety goals. As a result, work conflicts and rework occur, which lead to increases in project cost and/or duration and a decrease in quality and safety.

Look-ahead planning is a method of medium-term planning, which is proposed to shape workflow (Ballard 1997) so as to identify and remove constraints that may interfere with the continuous progression of work (Tommelein 1998). The term medium-term planning derives from the work of Laufer and Tucker (1987), who suggest that the construction planning process should be divided into different levels – long, medium, and short term planning. A master schedule represents the long-term plan managing all the activities scheduled for a project (Ballard 1997, Soares et al. 2002). It is typically created by project planners and site engineers at the beginning of the construction phase to show the project stakeholders important project goals and milestones. However, such master schedules cannot be accurately detailed too far into the future because of lack of information about actual activity/operation duration and resource deliveries. Instead, LASs are used to guide the site engineers for the daily resource allocation and work coordination processes when the availability of resources, construction methods and activity duration become known (Ballard, 2000). A further advantage of LASs is that site engineers receive feedback on how good the

LAS worked at the end of each day enabling, in theory, adjustments of the upcoming LASs and related planning information. However, manual adjustments of the LASs on a daily basis for complex projects are very time-consuming, if not undoable. Ballard and Howell (1994) consider commitment planning to be short-term planning, which can improve work performance by selecting appropriate amounts of work with correct sequences. They differentiate look-ahead planning from commitment planning, which often takes the form of weekly work plans produced for each crew or sub-crew of each trade. The WorfFace planning method detailed by the Construction Owners Association of Alberta (2012) has similar goals to commitment planning except that the WorkFace planning method is developed to facilitate the short-term scheduling of large oil and gas projects. In my research, I use LAS to represent both medium-term and short-term planning. As such, the LASs generated are already detailed enough to guide crews' daily work while the resource allocation is optimized to achieve project goals (such as shortest construction duration or lowest construction cost).

LASs can be manually generated for small or repetitive projects such as housing, high-rise, tunnel and highway construction projects. Generating LASs for such projects normally does not require a lot of time from the site engineers and project planners, because the allocation of available resources and the coordination of different trades/subcontractors are relatively simple. On the other hand, manual generation of LASs for complex projects, particularly in the finishing phase, is very difficult. A single process pattern cannot be used as the template for the coordination of multiple trades (civil, mechanical, electrical, plumbing, and so on), the allocation of various types of available resources, the consideration of complicated work sequences in many rooms, and the accommodation of engineering constraints within and among certain rooms.

Therefore, it is crucial to provide an automated way of LAS generation to save the time and energy of the site engineers and project planners in the schedule generation process, to eliminate work conflicts and rework resulting from low-quality, insufficiently detailed schedules used on job sites, and to improve the efficiency of the site engineers in supervising the crews' daily work. The optimization of LASs can further help the construction managers to adjust their resource allocation strategies according to dynamic, constantly-changing site conditions while adhering to specific project goals.

1 Motivating Case

In 2008, I spent six months at the construction site of the Carnegie Mellon University Campus in Qatar, investigating possibilities of automating the generation of LASs in the finishing phase to improve the efficiency of the site engineers' work assignment process .

The project consists of two main buildings, each three stories high. The finishing phase was staggered into three stages, with each story of both buildings forming a stage. For each phase, more than 200 rooms needed to be completed. These rooms can be categorized into different types such as computer class rooms, small conference rooms, lecture halls, auditoriums, prayer rooms, plant rooms, electrical rooms, IDF rooms and data centers, security rooms, faculty and staff offices, storage rooms, toilets, and so on. Each type of room has its unique architectural design, material requirements, and functionality and thus has a unique construction work sequence in the finishing phase. On average more than 20 operations are needed to finish a room. Therefore more than 4,000 operations are involved at each stage. Each operation needs to install at least one type of material. At each stage, more than 100 types of materials are installed. In this thesis, the term "operation" is used to represent a higher level of detail than "activity". Most operations can be carried out in a given room. In addition, the site engineers need to organize more than 10 types of crews and subcontractors that include plastering, screed, painting, carpentry, carpeting, electrical, HVAC, fire protection, plumbing, and so on.

This vast amount of information needs to be considered in the LAS generation process. On top of it, the project planners and site engineers also need to consider certain crucial engineering constraints. One constraint is referred to as *zone constraint*, which requires certain operations in multiple rooms to be synchronized with the same crew(s) assigned to all such operations at the same time. Another constraint is generally related with corridors. When a *blocking operation* starts in a corridor, the access to all its adjacent spaces is suspended and no other crews are allowed to step on the floor until the operation is finished. A typical blocking operation is “Terrazzo flooring” in a corridor. The constraint related with such an operation is called *blocking constraint*. *Engineering priority* is another constraint that also needs to be taken into account. When the owner requires certain rooms to be finished first, these rooms will have higher priority than others; thus resources should be allocated to these rooms first.

Three problems were discovered from the field research project. First, LAS generation is time-consuming when the schedules are generated manually or with the help of existing scheduling tools. In a scheduling workshop conducted at the field research job site, when given a 20-room case for LAS generation, i.e., a test case about 1/10th of the complexity of the actual project, one of the project planners stated that “I would not waste my time in creating a LAS for 20 rooms when I need to organize and synthesize so much information when scheduling, knowing the inevitability of making a mistake.” This statement confirms the second problem: if LASs are created for the finishing phase of complex projects, they are likely to contain significant errors. Thirdly, the current scheduling methods do not enable site engineers to check whether a particular LAS, even if it happened to be error-free, is the best LAS in support of specific project goals such as lowest construction cost or shortest construction duration because of the existence of many alternatives (Dong et al. 2011a).

2 Summary of the Requirements for This Research

The big idea of this research is to quickly generate accurate and close-to-optimum LASs for the finishing phase of complex construction projects. To realize this big idea, there is a theoretical and practical need for modeling, computer simulation, and optimization.

Modeling is the foundation of automated LAS generation. The requirement in modeling is composed of two parts. First, since a LAS needs to show who (the crews) will work on what (which operation) when (from which date to which date) and where (at which location), an information model is needed to integrate various project data sources in order to permit effective retrieval of the relevant data in the LAS generation process. Second, a process model is required to automate the LAS generation process using the information model as an agent to allocate available resources without violating the engineering constraints.

The computer simulation should be conducted on the basis of this modeling and should provide LAS alternatives sorted according to the project goals specified by the users. It should allow the users to analyze the relation between resource utilization and project goals and to conduct time-cost studies.

Finally, the optimization method should discover better LAS alternatives faster than the simulation-based method.

3 Research Questions

Automation and optimization of LAS generation are the two main components of this study. In the first component I developed a modeling-based approach to automating the LAS generation in the finishing phase of complex projects. To enable automatic LAS generation, I investigated the types of data required and how they should be connected to schedule an operation: who will carry out the operation, when it should start/continue, when it should end, where it should take place, and how it will affect the other operations. Once all the data for one operation could be automatically retrieved, I needed to find out how to schedule all the operations automatically, without violating critical constraints including resource availability, progress updates, work sequences, and engineering constraints. The second component builds upon the first one to explore how to optimize LAS generation when two types of resources (room and crew) and the engineering constraints are considered. I developed two major research questions, one for each component, as follows.

1. How can a LAS be generated quickly and accurately in the finishing phase of complex construction projects, considering the work content and work sequence in each room, resource availability, crew productivity, and engineering constraints?

- *To enable such generation I needed to construct an information model that integrates different data sources to facilitate LAS generation.*
- *Then I needed to build a LAS generation process model that can utilize the information model to generate a LAS.*

2. How can close-to-optimum LASs be created quickly?

4 Research Methods

4.1 Information modeling to integrate data to get ready for LAS generation

A basic function of a LAS is to predict how each operation should proceed without violating any constraints. Naturally the modeling of operations and constraints becomes the core of information modeling for look-ahead scheduling. Therefore, when automating LAS generation for the finishing phase of complex projects, I need an information model which can establish the correct data connection to achieve two purposes: (1) facilitate the creation of operations with the form of [crew, material, start and end dates, room] and (2) correctly link operations to the relevant constraints including precedence, crew and room availability, and OSS constraints.

4.2 Process modeling to automate LAS generation

The schedule generation method in ACP (Waugh 1990) categorizes activities into types including TODO, CANDO, DOING, ENDED, and DONE. For each incremental time period, ACP selects eligible activities from the TODO bucket and puts them in the CANDO bucket. Then it assign crews to the activities in the CANDO bucket until no available crews are left or the CANDO bucket is empty. The activities with crew assigned are put into the DOING bucket. After one time period, ACP takes the completed activities out of the DOING bucket, puts them into the ENDED bucket and updates the project status, then moves these activities into the DONE bucket.

My dissertation expands the ACP process model into one that manages rooms as well as engineering constraints. Since each room includes a set of operations, the schedule generation process should start with room. Rooms with the highest engineering priorities should be selected first. Then in each room operations can be processed using the aforementioned buckets. In other words, the bucket needs to have two levels of detail: room □ operation. The work sequence (i.e., the precedence constraints of operations) inside a type of room can be defined by a fragnet (Dong et al. 2011a). Most operations can be scheduled using the operation-level bucket. But since both the zone constraint and the blocking constraint involve a number of rooms, operations related to such constraints need to be processed at both the room and the operational level.

4.3 Simulation and analysis of results

Once the information model and the process model for LAS generation are created, computer simulation is used for the generation of multiple LAS alternatives. After the users provide the necessary inputs required for LAS generation, they can specify the number of simulations they want to run and then start the simulations. Each simulation generates one LAS based on the information model and the process model. Up to this point, the research has reached the goal of automated LAS generation. Chapter 2 details the modelling and simulation for such a goal.

Once the LAS generation is automated, a large number of simulations can be carried out to conduct time-cost trade-off study or investigate the relations between resource utilization and project goals. Statistic methods need to be used to analyze the large set of data generated from simulations. Chapter 3 provides more details in this regard.

4.4 LAS optimization based on a genetic algorithm (GA)

LAS generation considering limited resources can be categorized as a type of resource-constrained project scheduling problem (RCPSP) and can be addressed by a genetic algorithm approach (Feng et al. 2010). However, as mentioned by Payne (1990), up to 90% of all projects occur in a multi-project context. In this context, a project is composed of multiple sub-projects and each sub-project consists of an operation network (with embedded operation precedent constraints) that draws from shared pools of multiple types of resources which are normally not large enough for all the sub-projects, and thereby operations, to work concurrently. LAS generation for the finishing phase of complex construction projects conforms to this context (Dong et al. 2011b). In the finishing phase, each room is relatively independent so that most operations in one room do not interfere with those in another. In addition, each type of room entails a specific work sequence for crews to work on. If all rooms are entirely independent, each room can be considered as a sub-project, thus making the LAS generation problem become a resource-constrained multi-project scheduling problem (RCMPSP). GA is widely used in RCPSP/RCMPSP to find good solutions with few computational requirements within a reasonable time period. A GA applies the principles of biological evolution to solve optimization problems by combining and altering existing solutions in order to form new ones (Holland 1975; Goldberg 1989). It is used for this LAS optimization problem because it is more flexible in accommodating additional constraints compared to the other types of local search methods. Chapter 4 describes a GA-based algorithm to address RCMPSPs in the finishing phase of complex construction projects.

5 Prototype Implementation

I implemented a software prototype based on the modeling, simulation, and optimization methods discussed in Section 4. The prototype is developed in Microsoft visual C# and contains around 18,000 lines of code. After providing necessary inputs, users can either run the schedule simulation or the GA-based algorithm to get the desired LASs. The graphic user interface (GUI) of the prototype allows the user to provide any changes on the job site so that the LASs generated reflect actual project conditions. The GUI is also an interface that allows users to access certain project databases related to LAS generation. It screens the sophisticated data connections among various project databases, only providing users what they need to know, by retrieving relevant data using the information model. The user inputs include crew availability, room availability, crew productivity, operation duration when it cannot be calculated, progress updates, direct and indirect costs, and engineering constraints. The outputs of the prototype include the best LASs found using the html format as well as the corresponding crew’s daily work plans. Users can also view a list of assorted project duration/cost of all the alternatives generated in the simulation/optimization process from the GUI. Chapter 2 explains the prototype implementation in detail.

6 Research Validation

The validation of my research includes two parts: internal validation and external validation. The internal validation tests the correctness of the LASs as the outputs of the prototype. To achieve this goal, I developed a schedule-checking program to scrutinize the following aspects on the schedules: work sequence in each space, crew allocation conflicts on each day, conformance to engineering constraints, duration of each operation in the schedule, and the calculation of project duration and cost. The program was applied to test all the schedules generated and found no errors.

The charrette test (Clayton et al. 1998) is the main method for the external validation of my research. I conducted the test with two controlled groups of participants, practitioners and Stanford graduate students from the Department of Civil and Environmental Engineering. The goal of this test was to prove that the LAS generation process using my prototype is superior to using the existing methods/tools. The metrics used for such validations include the number of errors in the LASs generated, the total construction duration/cost of these LASs, and the time spent in creating them. Chapter 5 describes the setup and the results of the charrette validations in detail.

7 Research History

My research at Stanford includes three stages, through which I have explored three main areas that provide sources of knowledge and inspiration: construction management (particularly model-based integrated project monitoring and control), operations research, and artificial intelligence. As a CIFE research fellow, I have gained fundamental knowledge of construction management and IT applications in construction through CIFE’s collaboration with the Consolidated Contractors Company (CCC).

My field research experience at the headquarters and Cairo office of CCC in 2007 represents the first stage, during which I studied model-based integrated project monitoring and control. A two-month training experience at CCC's headquarters in Athens, Greece opened my eyes regarding how building information modeling (BIM) is actually used in project monitoring and control on site. It was particularly refreshing to learn what databases are essential and how to link them in order to use BIM for project monitoring and control, how to integrate progress updates into BIM, and how to generate project progress reports and LASs based on such integration. Then my five-month field research experience of implementing C3D, CCC's in-house application for integrated project monitoring and control, the core function of which is the BIM-based look-ahead scheduling, stimulated my intuition that forging the appropriate connection between various project databases is essential for LAS generation, and can significantly improve project planners' efficiency in scheduling and reduce errors in the LASs created. The Saudi Arabia Embassy Project in Cairo was in the structural phase while I was there. The project consisted of two high-rise buildings, each floor structurally similar to another in the same building so that the work sequence of a floor could be reused. Therefore, LAS generation for this project at the structural phase was not very complicated, and the site engineers and project planners could use C3D to generate a LAS within an hour, once they were familiar with its functionalities.

The second stage started with my field research experience at Carnegie Mellon University Campus in Qatar, where the project planners, site engineers, and construction managers faced many more issues on a daily basis because of the complexity of the project described in Section 1. This experience further strengthened my intuition that I needed to create an information model to integrate different project databases necessary for LAS generation at the appropriate levels. More importantly, the issues the site engineers had with respect to project monitoring and control motivated me to establish an automated method to generate LASs to help them better organize the limited resources on site. The case studies discussed in Chapter 2, 3, 4, and 5 are all derived from this project experience.

Compared to the first two stages of my research, which involved intensive field research including site observations, interviews with site engineers and construction managers to discover the problems they faced, and workshops with project planners and other construction professionals to discuss the existing construction planning and scheduling techniques, my work in the third stage was carried out mainly at Stanford. At this stage, I built the information model for LAS generation and created a process model to automate LAS generation, which are detailed in Chapter 2. I also implemented a prototype to allow users to find accurate schedules via simulations (Dong et al. 2011a). After running the simulation multiple times, I was able to analyze and evaluate the relations between resource utilization and project goals as well as conduct time-cost trade-off studies, which are discussed in Chapters 2 and 3. Because simulation-based method might not be fast enough to find optimized LASs for complex large-scale projects, I dove into the operations research domain and first tried to use integer programming methods to address the LAS optimization problem. After discovering that this problem is an NP-hard problem, I switched to the artificial intelligence domain and explored approximation algorithms and efficient heuristics to tackle the problem (Dong et al. 2011b). Thus I developed a GA-based method that can find better solutions faster than the simulation-based method (Chapter 4).

8 Explanation of Dissertation

In this chapter, I presented an overview of my dissertation, including the practical and theoretical motivations, research questions and methods, prototype implementations, research validation, and roadmap. The chapters that follow proceed in order of the ALASG method and related prototype implementation; application of the prototype, particularly in the area of resource utilization in LASs; LAS optimization; research validation and contributions.

Chapter 2, which will be submitted to Automation in Construction, describes the method to automate LAS generation for the finishing phase of construction projects based on information modeling, process modeling, and simulation methods. It also describes the implementation of the prototype. This chapter addresses the first research question.

Chapter 3, which will be submitted to Journal of Construction Engineering and Management, evaluates the relationship between resource utilization and project goals using the prototype. This chapter discusses the practical value of the prototype and its possible applications in the construction industry.

Chapter 4, which has been accepted by Advanced Engineering Informatics, presents my GA-based method for LAS generation in the finishing phase of complex construction projects. This chapter addresses the second research question.

Chapter 5 explains the internal and external validation for this research, its contributions and practical impacts.

9 References

Ballard, G. (1997). "Lookahead planning: the missing link in production control." *Proceedings of 5th Annual Conference of the International Group for Lean Construction (IGLC-5),* Gold Coast, Australia, 13-25.

Ballard, G. (2000). "The last planner system of production control." Ph.D. Dissertation, Faculty of Engineering, School of Civil Engineering, The University of Birmingham, UK.

Chapter 1

Ballard, G., and Howell, G. (1994). "Implementing lean construction: stabilizing work flow." *Proceedings of the 2nd Annual meeting of the international Group for Lean Construction,* Santiago, Chile.

Clayton, M. J., Kunz, J., and Fischer, M. (1998). "The charrette test method." Stanford University, Center for Integrated Facility Engineering, Technical Report 120.

Construction Owners Association of Alberta, WorkFace Planning, <http://www.coaa.ab.ca/Productivity/WorkFacePlanning/Summary.aspx> (June 4, 2012).

Dong, N., Fischer, M. and Haddad, Z. (2011a). "Automatic look-ahead schedule generation system for the finishing phase of complex projects for general contractors." *Proceedings of the 2011 ASCE International Workshop on Computing in Civil Engineering,* Miami, FL.

Dong, N., Ge, D., Haddad, Z., and Fischer, M. (2011b). "Applying genetic algorithm for look-ahead schedule generation for the finishing phase of construction projects for general contractors." *Proceedings of the 2011 eg-ice Workshop,* University of Twente, The Netherlands.

Feng, C., Chen, Y., and Huang, J. (2010). "Using the MD CAD model to develop the time–cost integrated schedule for construction projects." *Automation in Construction,* 19(3), 347-356.

Goldberg, D.E. (1989). Genetic Algorithm in search, optimization, and machine learning, Addison-Wesley, Reading, MA.

Chapter 1

Holland, H.J. (1975). Adaptation in Natural and artificial systems, University of Michigan Press, MI.

Laufer, A., and Tucker, R. L. (1987). "Is construction project planning really doing its job? A critical examination of focus, role and process." *Construction Management and Economics,* 5, 243-266.

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

Soares, A. C., Bernardes, M. M. S., and Formoso, C. T. (2002). "Improving the production planning and control system in a building company: contributions after stabilization." *Proceedings of 10th Annual Conference of the International Group for Lean Construction (IGLC-10),* Gramado, RS, Brazil, 477-487.

Tommelein, I. (1998). "Pull-driven scheduling for pipe-spool installation: simulation of lean construction technique." *Journal of Construction Engineering and Management,* 124(4), 279-288.

Waugh, L. (1990). "A construction planner," Ph.D. Dissertation, Stanford University, Stanford, CA.

Chapter 2: A Method to Automate Look-ahead Schedule (LAS) Generation for the Finishing Phase of Construction Projects

Ning Dong, Martin Fischer, Zuhair Haddad, and Raymond Levitt

1 Abstract

Look-ahead schedules (LASs) in the finishing phase of complex projects produced with sufficient detail to guide crews' work can help prevent conflict, ensure the correct work sequence for work units, and facilitate a fluent workflow for a crew, preventing rework. Unfortunately, such LASs are rarely used in these projects because of the time-consuming schedule generation process, the high probability of introducing errors into the LASs and the inability to find the close-to-optimum LAS from the vast number of feasible alternatives. This paper proposes an automated LAS generation (ALASG) method addressing these challenges. This method is composed of an information model and a LAS generation process model. The information model organizes various sources of project data, integrates them at the appropriate levels of detail, and gets the project data ready for LAS generation. The LAS generation process model automates the site engineers' daily work assignment process and delivers error-free LASs. A software prototype developed based on the ALASG method is presented and a simulation-based approach is introduced to find close-to-optimum LASs quickly in terms of construction duration or cost for a case study. The proposed method contributes to the field of automated project scheduling, particularly automated look-ahead scheduling, through the

quick generation of error-free, close-to-optimum LASs for the finishing phase of complex projects.

2 Introduction

Look-ahead scheduling is one of the most popular and effective ways to guide the crews' work on job sites to complete projects on time and on budget. A good look-ahead schedule (LAS) helps prevent work conflict, ensure the correct work sequence, and facilitate a fluent workflow for a crew, preventing rework.

However, it is challenging for project planners and site engineers to create executable LASs with detailed work assignments for the finishing phase. We discovered three scheduling-related challenges from field research we conducted on a university building project in Qatar, in collaboration with Consolidated Contractors Company (CCC). In this project, the site engineers needed to manage, on a daily basis, the finishing work for more than 50 rooms (out of a total of about 600 rooms) with an average of 20 operations in each room, more than 10 types of their own crews, and more than 10 subcontractors. These rooms can be categorized into 18 different types, each type relating to a specific work sequence to complete a room. Each type can be represented by a fragnet, a predefined operation network (Aalami et al. 1998). We call projects of such scale with a non-repetitive nature complex projects. The term "operation" is used in this paper to represent a higher level of detail compared to "activity". The challenges discovered in this project are related to the schedule generation process and the results, i.e., the LASs.

First, the LAS generation process is time-consuming when the schedules are generated manually or with the help of existing scheduling tools. To generate a LAS, site engineers need to first collect progress updates from the job site and then generate the LAS starting from the next work calendar date D. Based on the actual and assumed progress until the start of date D, they determine all the rooms available for construction on date D. If a room is available, they check which operation(s) can start (i.e., all its predecessors are finished) in this room. If an operation can start, they check whether the crew resource is ready. If so, they calculate the duration of this operation and schedule it in the LAS and then continue to check the next room until all the rooms are processed. By the end of D, they release the rooms and crews for the operations which are scheduled to be completed. Then they roll to date D+1 and repeat the above steps. This process stops when all the finishing operations in all the rooms are completed. This process would have been very time-consuming in our field research project. Assuming the construction manager wanted to keep 50 rooms active for crews' daily work assignments, the project planners would have had to calculate the duration of operations 1,000 times (20 operations per room on average), assuming each duration is calculated only once. Each time, they would have needed to find the materials and their quantities relevant to an operation within a room from the bill of quantities (BOQ) database and find the corresponding crew and their productivity from the cost/budget database. In addition, they would have had to check the operation precedence relations and the crew availability more than 1,000 times in this LAS generation process. Since it was not realistic with today's tools to follow such a formal LAS process on site, the site engineers assigned crews to rooms as best as they could based on perceived priorities. As a result, the authors observed work conflicts and rework on a daily basis.

What makes the LAS generation process even more complicated is that certain critical engineering constraints need to be considered. In the finishing phase, individual rooms are used as the basic units of work for most operations. A room is also commonly used as the basic unit for quality inspection. However, certain constraints related to the features of some operations prevent the rooms from being treated independently when scheduling. One is the *zone constraint* which requires the same type of operation to be synchronized across multiple rooms. For example, an HVAC work zone requires that the "HVAC duct installation" operations be carried out in multiple adjacent rooms simultaneously. If a subcontractor is responsible for these operations, the GC can hand over this HVAC zone to the subcontractor for a certain number of days. The zone constraint entails a hierarchy of operations (Waugh 1990; Winstanley et al. 1993) and requires aggregation of project data to accommodate different levels of detail, which account for discrete organizational and management responsibility levels (Darwiche et al. 1988). Operations that are not subject to a *zone constraint* can be carried out in a single room. Such operations are related to plastering, screed, painting, carpeting, gypsum, woodwork, waterproofing, lighting and the installation of certain electrical/mechanical devices, etc. These operations are normally independent from those carried out in another room, except for the operations related to a *blocking constraint*, which makes access to multiple rooms difficult or impossible. For instance, if the "suspended ceiling installation" operation in a corridor requires scaffolding which takes up the whole corridor floor area, this operation blocks all other operations in the adjacent rooms (Figure 2-1). Since the *zone* and *blocking constraints* are not only related to rooms, but also to certain operations and their construction methods, these constraints are named *operation-specific spatial (OSS) constraints* in this paper. If the OSS constraints are ignored in scheduling, work conflicts could occur in a LAS. However, it is formidable for project planners and site engineers to generate a LAS at the aforementioned scale with the speed required on dynamic construction projects with the consideration of these constraints, even with the help of the existing tools (e.g., Primavera, MS Excel, and MS Project). In a scheduling workshop conducted at the field research job site, when given a 20-room case for LAS generation, i.e., a test case about 1/10th of the complexity of the actual project, one of the project planners stated that "I would not waste my time in creating a LAS for 20 rooms when I need to organize and synthesize so much information when scheduling, knowing the inevitability of making a mistake."

This statement confirms the second challenge: if LASs are created, for the finishing phase of complex projects, they are likely to contain significant errors. In the schedule generation process, it is almost inevitable to make mistakes when manually calculating operation durations and taking into account all the scheduling constraints including operation precedence constraints, room and crew availability, and OSS constraints. As a result, errors occur in the LASs, which would lead to conflicts and rework if used on site.

The third challenge is that current methods do not enable site engineers to check whether a particular LAS, even if it happened to be error-free, is the best LAS in support of specific project goals such as minimum construction cost or shortest construction duration. This challenge concerns the quality of a LAS when the first two challenges are fully addressed. When multiple rooms are competing for the same crew, the allocation of this crew to each room leads to one schedule alternative. Site engineers and construction managers surely prefer the best alternative towards the intended project goals. However, since generating one error-free LAS is already hard enough, it is almost impossible to iterate through all the feasible alternatives to find the best LAS for a complex project.

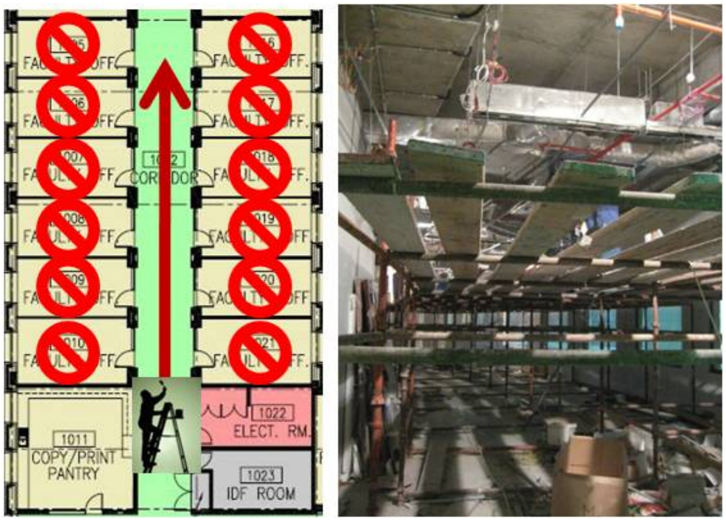


Figure 2-1. An instance of the blocking constraint – the scaffolding used when “suspended ceiling installation” is in progress in the corridor blocks the access to the corridor’s adjacent rooms completely.

The LAS method described in this paper fully addresses the first two challenges and paves the way to address the third challenge by proposing an automated LAS generation (ALASG) method and implementing a prototype based on this method. ALASG consists of an information model and a LAS generation process model. The information model organizes and integrates various information sources for the schedule generation. It ensures the accuracy of these data and feeds them to the LAS generation process model. The LAS generation process model automates the LAS generation process, arranging each crew’s work on a daily basis from the point of the most recent update towards the end of the project. It entails quick generation of a LAS without violating critical constraints including precedence, crew and room availability, and OSS constraints. The prototype developed on the basis of the ALASG method automates LAS generation and delivers error-free LASs. A computer simulation method is applied to the prototype to partially address the third challenge.

The next section provides a summary review of the relevant concepts and the limitations of the existing methods related to automated schedule generation. Based on this review, Section 4 describes the proposed ALASG method. Section 5 describes the implementation of the prototype and demonstrates an application of the prototype to look-ahead scheduling for the finishing phase of a six-room construction project. Section 6 explains the validation of the ALASG method and illustrates the validation results. Finally, the conclusions and future steps are discussed in Section 7.

3 Key Concepts

3.1 Information modeling for look-ahead scheduling

A basic function of a LAS is to establish when and how each operation should proceed without violating any constraints. Naturally, the modeling of operations and constraints becomes the core information modeling effort for look-ahead scheduling. An operation in a LAS created for the finishing phase of complex projects should accurately show which crew(s) will be working on what material(s) in which room, as well as its start and end dates. In other words, the operation should have the form of [*crew, material, start and end dates, room*]. The inclusion of the first two elements–crew and material–are discussed by Darwiche et al. (1988), who propose to use an object, action, and resource (OAR) model to represent an activity/operation before generating a construction schedule. In the finishing phase, materials are the objects to be installed in an operation, and crew is a type of resource to execute an operation. The OAR model takes into account the type of crew and type of material involved in an operation and thus is suitable for representing operations in fragnets, in which operations are not yet connected to specific crews and rooms. Such operations are called *generic operations* with the form of [type of crew, type of material] in this paper.

Once a generic operation is connected to a room, it takes the form of [type of crew and number of crews required, material (type and quantity), duration, room]. We call operations with such a form *operation instances*. In the finishing phase of complex projects, a room is a type of workspace that affects the interactions of operations. Researchers including Thabet and Beliveau (1997) and Akinci et al. (2002a) have proposed models of workspaces measured by length, width, and height. In this paper, rooms are considered the basic units of workspaces because (1) they provide a natural and convenient representation for spatial reasoning (Cherneff et al. 1991), (2) they are the basic production units for quality inspection for most of the finishing operations, and (3) they account for the spaces where the crews and equipment must be allocated regardless of their shapes (which cannot always be measured merely using length, width, and height).

Given a specific construction method, the size of a room directly determines how many crews can work simultaneously on an operation, which, in turn, establishes the duration of this operation. Winstanley et al. (1993) introduce five steps to estimate operation duration manually: (1) reference the operation by material; (2) find the recorded productivity rate for the crew; (3) determine the quantity of material; (4) divide the quantity into the productivity; and (5) use the resulting figure to represent the number of crew hours required to accomplish the operation. All these steps can be automated, but the first three require proper data integration to find the relevant crew production rate, material type, and quantity. Quite a few studies in the area of automated schedule generation imply that such integrations are applied (Zozaya-Gorostiza et al. 1989; Waugh 1990; Yau et al. 1991; Chevallier and Russell 1998), but to our knowledge none has explicitly explained how to do so for the finishing phase of complex projects.

Once the start and end dates are specified, an operation instance can be transformed into an actual operation in a LAS with the form of [crew, material, start and end dates, room]. Henceforth, the term *operation* refers to an actual operation. The start and end dates implicitly dictate an operation’s conformance to the critical constraints on site including precedence, crew and room availability, and OSS constraints. The determination of the start and end dates, which is related to the schedule generation process, is discussed in Section 3.2.

In addition to the modeling of operations, we should also model the critical constraints in look-ahead scheduling and establish the correct connection between an operation and its relevant constraints. An example is the connections between an operation and the resource constraints, i.e., the crew and room availability. For instance, to start the “carpet installation” operation in room 110, both the required tiling crew(s) and the workspace (room 110) must be available. Therefore, proper integration between an operation and its related resources is required for automated generation of LAS. The SCaRC method (Thabet and Beliveau 1997) presents a way to connect both crew and spatial resources to an activity when scheduling repetitive floors in multi-story projects. However, when modelling spatial resources, this method focuses on dividing a floor area into work blocks and linking these blocks, instead of individual rooms, to activities/operations. In addition, SCaRC does not create connections between OSS constraints and activities/operations nor between a fragnet and a room (so that operations in a room can be correctly linked to precedence constraints). Therefore, SCaRC is not suitable for LAS generation in the finishing phase of complex projects. The WorkFace Planning method extensively discussed by the Construction Owners Association of Alberta (2012) is another method that has been gaining popularity for look-ahead scheduling for large oil and gas projects. In the design phase, the design engineers need to design discipline-specific engineering work packages (EWPs). Then the contractors create discipline-specific construction work packages (CWPs) accordingly. The project planners and site engineers divide these CWPs into discipline-specific field installation work packages (FIWPs), which are used for look-ahead scheduling. The WorkFace Planning method can accurately describe crew, material, and start and finish dates for an operation. However, it focuses on disciplines instead of rooms and, thus, cannot provide adequate sequencing information in a room, and cannot accommodate location-specific constraints such as OSS constraints.

To sum up, when automating LAS generation for the finishing phase of complex projects, we need an information model which can establish the correct data integration to achieve two purposes: (1) facilitate the creation of operations with the form of [*crew, material, start and end dates, room*] and (2) correctly link operations to the relevant constraints including precedence, crew and room availability, and OSS constraints.

3.2 Schedule generation method

Once operation instances are appropriately formed and connected with relevant constraints, they need to be scheduled without violating these constraints. Critical path method (CPM) is the commonly used network technique for scheduling construction projects (Clough 1991; O'Brien and Plotnick 2005). However, the CPM method performs poorly in considering constraints critical for LAS, e.g., resource availability (Feng et al. 2010) and OSS constraints. Manually considering these constraints when scheduling for complex projects using CPM is time-consuming and error-prone. Therefore, for work assignments at the day-to-day level, CPM is ineffective (Koskela and Howell 2002) and is generally used for master scheduling, not weekly or daily production control by general contractors (Brodetskaiaa et al. 2011). Another drawback of CPM is the technical difficulty of representing a CPM schedule as a Gantt chart in a comprehensible manner when hundreds of activities/operations are scheduled (Brodetskaiaa et al. 2011). Further issues, such as difficulties in relating an activity/operation to physical components of a building and tracking the paths and flows of crews, have been identified (Jaafari 1984; Sriprasert and Dawood 2002; Brodetskaiaa et al. 2011).

The location-based scheduling method (Kenley and Seppänen 2009) resolves some of these flaws. It allows the consideration of room availability and presentation of the work content in a room and the flow of crews. However, it is still cumbersome and time-consuming to manually allocate a specific crew to a specific room when more than 20 types of crews and subcontractors are needed to be managed simultaneously on a daily basis, under the circumstance that rooms of different types have different work sequences and rooms of the same type might allow different numbers of crews to work on an operation (subject to the size of these rooms), as was the case for our field research project introduced in the Section 2. More importantly, the location-based scheduling method lacks an efficient mechanism to handle OSS constraints that affect multiple rooms/locations because the basic unit of modeling is a single crew performing a single task in a single location (Brodetskaiaa et al. 2011).

To accommodate such constraints, many researchers have developed knowledge-based expert systems (KBESs) to automate schedule generation (Darwiche et al. 1988; Zozaya-Gorostiza et al. 1989; Waugh 1990; Winstanley et al. 1993; Chevallier and Russell 1998; Aalami 1998; Kataoka 2008; Kanit et al. 2009; Feng et al. 2010). Most of the studies do not consider workspace as a type of resource in their scheduling procedures except SCaRC (Thabet and Beliveau 1997). However, as mentioned above, the spatial availability considered in SCaRC is at the floor level, not individual room level. In the domain of space-scheduling, quite a few researchers have proposed methods to identify and resolve space conflicts in existing schedules (Riley 1994; Riley and Sanvido 1997; Guo 2002; Akinci et al. 2002a; Akinci et al. 2002b; Dawood and Mallasi 2006), yet to our knowledge none have developed methods to directly create schedules that ensure zero conflict. As to handling precedence constraints, one popular trend in the KBES-related domain is to create activity sequences based on the reasoning of the relations among physical components as well as the construction methods (Aalami 1998; Kataoka 2008; Feng et al. 2010). However, these studies elaborate on the structural phase of construction projects. In the finishing phase of complex projects, the reasoning of the component relations for the purpose of activity/operation sequence generation is much more complicated and requires a highly detailed product model. Therefore, we found that fragnets are most appropriate to represent the work sequence within a room (Chevallier and Russell 1998). On the other hand, reasoning about the physical relations among rooms, subject to certain construction methods, can be helpful to accommodate OSS constraints in scheduling. Specific methods for such purpose have not yet emerged.

In sum, the existing scheduling methods do not account for at least one critical constraint in the finishing phase of complex projects. Therefore, a constraint-based schedule generation process model is needed to automate LAS generation so that accurate LASs can be created quickly.

4 The ALASG Method

To quickly generate accurate LASs, we propose the ALASG method which consists of an information model and a LAS generation process model. The function of the information model is to generate operation instances with the form of [type of crew and number of crews required, material (type and quantity), duration, room] and correctly connect these operation instances to relevant constraints. The function of the LAS generation process model is to automatically convert operation instances to operations with the form of [crew (actual crews assigned), material, start and end dates, room]. In this conversion process, the process model specifies the start and end dates for an operation without violating relevant constraints. The process model retrieves scheduling-related data through the information model. The information model organizes and integrates these project data. Upon receiving a query from the LAS generation process model, the information model answers the query with processed data. The details of the inputs needed for the LAS generation in the finishing phase of complex projects are listed in Table 2-1. These inputs are considered essential for LAS generation in the finishing phase of complex building projects according to interviews with two project planners, five site engineers (two electrical, two mechanical, and one civil), and one construction manager from CCC (six of these eight practitioners were from the field research project). In their discussion with respect to work assignments, prior scholars have already considered some of these inputs such as fragnet (Chevallier and Russel 1998), material and crew (Darwiche et al. 1988; Winstanley et al. 1993; Chevallier and Russel 1998; Mourgues 2009; Feng et al. 2010), and construction costs (Feng et al. 2010). However we are not aware any discussion regarding rooms and OSS constraints, not to mention all the inputs in one place. The structure of the ALASG method and its inputs and output are illustrated in Figure 2-2. Although the execution of the LAS generation process model only results in one accurate LAS, through computer simulation, a prototype developed based on the ALASG method can generate thousands, even millions of alternatives, subject to users' requests and the scale of a project, and present the best one(s) in terms of minimum construction cost or shortest construction duration. The implementation of the prototype also includes setting up the project-specific databases for the information model to retrieve, organize, and integrate inputs. Three databases, i.e., scheduling, estimation/budget, and design and engineering, are shown in Figure 2-2Figure 2-2. Structure of the ALASG method and its inputs and output. for illustration purposes. Based on our field research, site engineers, construction managers, and project planners consider the inputs listed in Table 2-1 as essential for LAS generation. Once the finishing phase starts, site engineers can provide changes and updates regarding certain inputs so that the databases reflect the current conditions of the job site.

Table 2-1. Inputs for the ALASG method.

Type of input	Details
Fragnet	- Generic operations o Required crew type o BOQ code - Precedence constraint
Blocking constraint	- The operation related to a blocking constraint - The rooms this operation blocks
Zone constraint	- The operations related to a zone constraint - The rooms forming this zone
Room	- Room ID - Room available date for the finishing phase - Room type (fragnet)
Material	- BOQ code - Quantity - Unit
Crew	- Crew type - Total number of crews for a certain type - Crew ID - Productivity related with a BOQ code - Availability
Construction costs	- Construction indirect cost - Crew direct cost (daily cost and mobilization/demobilization cost)
Other inputs	- Duration estimation method of an operation (calculation or direct input from site engineers) - Work progress by the time a new LAS needs to be created

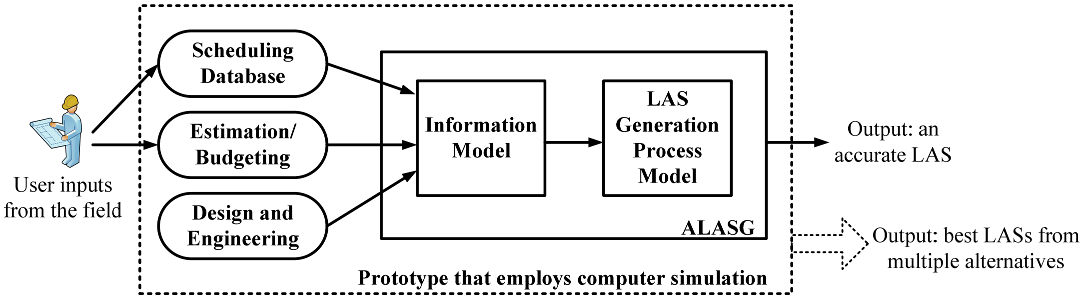


Figure 2-2. Structure of the ALASG method and its inputs and output.

4.1 Design of the information model

The extended construction-method-model (CMM) (Akinci et al. 2002) is an information model which represents work spaces with respect to different construction methods. It has considered the following inputs listed in Table 2-1: material, crew, and space. These inputs are fundamental for LAS generation in the finishing phase. When designing our information model we have inherited from CMM such features as the definition of classes and their relations and the definition of subclasses and their relations to superclasses, using UML (Fowler and Scott 1999). However, as mentioned in Section 3.1, CMM is not sufficient to be used for look-ahead scheduling in the finishing phase because it does not specify the relations between activities (i.e., precedence constraints), the OSS constraints, crew productivity against the type of material to be installed, the calculation/retrieval of operation duration, and construction costs. SCaRC (Thabet and Beliveau 1997) has considered connecting activities/operations to the precedence constraints, yet it sill misses the rest. When creating our information model, we have considered these missing inputs responding to the requirements of the site engineers, construction managers, and project planners. Since the scheduling-related inputs are scattered in different data sources with different levels of detail (Table 2-1), we developed the structure of the information model using the product, organization, and process (POP) model (Kam and Fischer 2004; Fischer and Kunz 2004; Kunz and Fischer 2005) so that these inputs are managed consistently. Figure 2-3 illustrates the information model designed and how the inputs are arranged from different POP perspectives. In this model we use the term "organization" loosely because the entities categorized in the organization model are actually resources (i.e., crew and room resources) that are essential for the formation of operations.

Multiple levels of abstraction, representing the class-instance (or superclass-subclass) relations between two entities (Marshall et al. 1987; Fowler and Scott 1999), exist in the product, organization, and process model. In the process model, the generic operation entity contains attributes including an operation's required type(s) of crew and the type(s) of material to be installed. Once a room is assigned a type/fragnet (Table 2-1), a generic operation in this room turns into an operation instance, which inherits the attributes from the generic operation.

Once the process model assigns start and end dates to an operation instance, it becomes an actual operation, which is an instance of the operation instance. The status attribute of an operation entity describes its progress towards the current calendar date (i.e., whether it has been completed; If not, which crew(s) are assigned to it, and when it is supposed to be completed). The three levels of abstraction (i.e., generic operation→operation instance→operation in a LAS) help the project planners establish necessary connections among different types of project data without creating connection redundancies. It is worth mentioning that a generic operation can entail multiple operation instances because multiple rooms can belong to the same room type (i.e., fragnet) whereas an operation instance can only have one operation as its instance. As in the relation between a generic operation and an operation instance, in the product model, a material is delineated by its type, represented by a BOQ code, and an instance, loaded with a quantity in a given room. Similarly, in the organization model, a crew is delineated by its type, for the execution of a type of finishing work, and an instance, loaded with its availability for the next task and specific productivity relevant to the type of material to be installed.

As demonstrated in Figure 2-3, a room has representations in multiple models. From the product perspective, a room is deemed a production unit, which contains a number of materials to be installed. From the organization/resource perspective, it can be treated as a spatial resource without which an operation cannot be scheduled. From the process perspective, it is categorized by a certain fragnet and consists of a number of operation instances. The relation between a fragnet and a room from the process perspective is similar to that between a generic operation and an operation instance. As an illustration of the roles of entities and their relations to others, a room entity appears in both the process model and the product model in Figure 2-3. In the actual implementation of the information model, only one data entity is needed for a room.

As discussed in Section 3.1, rooms are the basic units of workspaces. However, when it comes to a zone constraint, certain types of materials in a group of rooms (i.e., a zone) form a work package. Therefore, two levels of detail for operations in a LAS should be allowed: operations executed in a zone and operations executed in a room. In accordance with this requirement, we define a zone entity in the product model so that multiple rooms can be grouped for operations subject to zone constraints.

In this paper the construction-related cost is composed of daily indirect costs and crews' direct costs. The daily indirect costs include expenses in site supervision, management of field offices (e.g., IT facilities, site QA program, and payroll administration), site cleanup, and general housekeeping. These costs are incorporated into the attributes of the work calendar entity in the process model. The crews' direct costs include the crews' daily costs as well as their mobilization/demobilization costs. These costs are represented as attributes of the crew type entity.

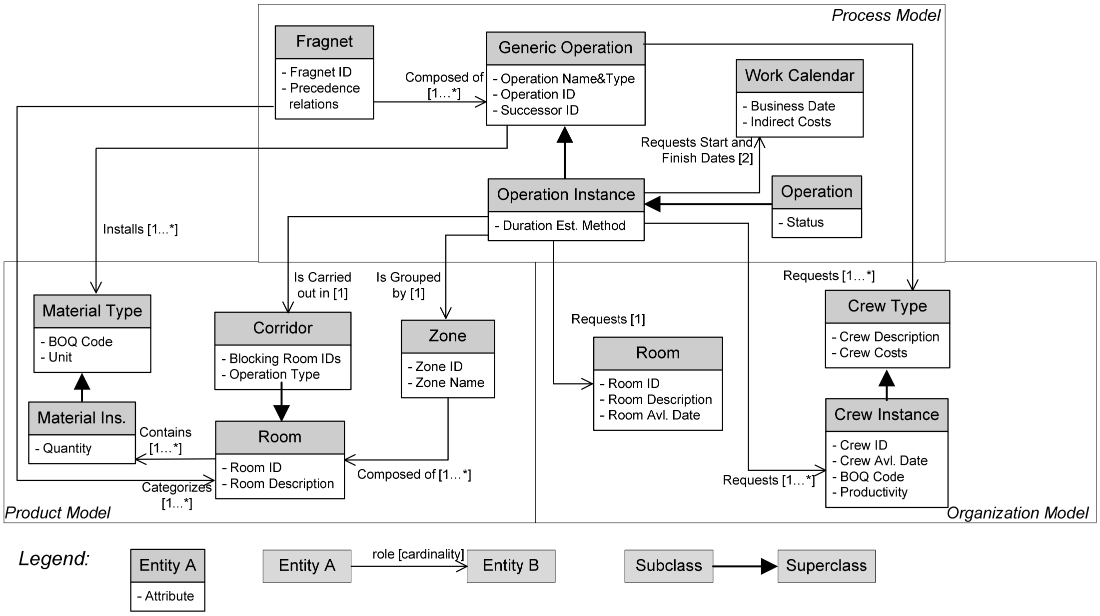


Figure 2-3. Information model for LAS generation from product, organization and process perspectives.

4.1.1 Formation of operation instances

Operation instances that take the form of [type of crew and number of crews required, material (type and quantity), duration, room] are the final results that the information model produces for the LAS generation process model to schedule. Through the connections between entities and the attributes within the entities, operations instances can be formed. The connections linking the generic operation entity to the material type entity and the crew type entity (Figure 2-3) entail the formation of a generic operation [type of crew, type of material]. An operation instance inherits such connections from the generic operation and thus includes the type of crew and type of material. Using an operation instance's connection to the room entity coupled with the instance's related type of material, the information model can find the quantity of material in the product model. The operation instance entity also connects to the crew instance entity (in the organization model) with the arrows specifying the number of crews required. A crew instance's productivity can be determined on the basis of the type of material to be installed. At this point, all the data entries to form an operation instance should be ready except the duration of an operation. The "duration estimation method" attribute in the operation instance entity determines how the duration is estimated, either by using the site engineers' estimate or by calculating the duration based on quantity of material, crew productivity, and the number of crews required in a room. When the information model cannot retrieve data entries related to the quantity of material or crew productivity from the relevant databases, it ignores the value of the duration estimation method attribute and directly uses the site engineer's estimate as the duration of the operation instance. The work calendar does not affect the creation of an operation instance, but affects the transformation of an operation instance to an actual operation in a LAS because it rules out all the holidays so that crews won't be scheduled to work when they are not supposed to. In this transformation process, the process model retrieves crew IDs via the crew instance entity and assigns them to the operations in a LAS.

4.1.2 Connecting an operation instance to relevant constraints

The connections between different types of data established in the information model not only serve to generate operation instances with the correct form but also facilitate linking the operation instances to relevant constraints. If an operation instance is not properly connected to all its related constraints (including precedence, crew and room availability, and OSS constraints), the LAS generated will not be accurate.

In the finishing phase of complex projects, two features are related to precedence constraints: (1) rooms are relatively independent – they only have interactions when they are involved with OSS constraints, and (2) a fragnet represents a type of room and describes the precedence relations of operations within a room. Therefore, the links of fragnet → generic operation → operation instance in the process model (Figure 2-3) are sufficient to connect an operation instance to the relevant precedence constraints (i.e., its immediate predecessors and successors). In this research project, we define fragnets on the basis of the seminal work of Echeverry et al. (1991) as well as the best practice of CCC. The design and definition of the fragnets are not within the scope of this paper.

The attributes in the room entity and crew instance entity specify the room and crew availability. The connections between the operation instance entity and these two entities are sufficient to link an operation instance to the relevant resource constraints.

As for the zone constraint, the information model needs to aggregate certain operation instances into one so that (1) the operation instances share the crew resource during their execution and (2) the quantities of material are summed to facilitate duration calculation and help site engineers prepare enough materials to carry out the tasks in a zone. The zone entity in the product model (Figure 2-3) is critical to this aggregation. Its connection to the operation instance entity indicates the crew requirement for a zone; i.e., site engineers should specify the number of crews required for a zone. The connection of the zone entity to the room entity determines which quantities should be retrieved and added up by the process model.

Finally, for the blocking constraint, a key message the information model needs to send to the LAS generation process model is which rooms should be blocked once an operation instance related to a blocking constraint starts. For such a purpose, we introduce, in the product model, a corridor entity, which is a subclass of the room entity and thus inherits its attributes and connections to other entities. The "Blocking Room IDs" attribute of the corridor entity specifies the rooms, which will be blocked by a type of operation (e.g., suspended ceiling or terrazzo flooring installation). With the corridor entity connected to the operation instance entity, the connection between a blocking constraint and the relevant operation instance is established in the information model.

4.2 Design of the LAS generation process model

The function of the LAS generation process model is to automatically sequence the operation instances formed by the information model without violating any critical constraints, including precedence, crew and room availability, and OSS constraints. Therefore, a constraint-based scheduling approach (Darwiche et al. 1988; Waugh 1990) is appropriate for the automation of LAS generation. Waugh (1990) proposes ACP, an integrated scheduling approach that considers constraints sequentially. In ACP, activities are categorized into buckets including TODO, CANDO, DOING, ENDED, and DONE. For each incremental time period, ACP selects eligible activities (that meet the precedence constraints) from the TODO bucket and puts them into the CANDO bucket. Then it assigns resources to the activities in the CANDO bucket until no available resources are left or the CANDO bucket is empty. The activities with resources assigned are put into the DOING bucket. After one time period, ACP takes out the completed activities, puts them into the ENDED bucket, updates the project status, and moves these activities into the DONE bucket. Such a scheduling algorithm is suitable for projects in which all types of constraints are treated at the same level of detail. However, in the finishing phase of complex projects, we need to address the constraints at two levels, the room level and the operation level.

To schedule an operation instance that is not related to OSS constraints, we first need to check, at the room level, whether the related room is available (e.g., whether it is already occupied or blocked). If the room is available, we can follow the steps of ACP to address the precedence constraint and crew availability sequentially (on the operation level). As for the operation instances that are related to a zone constraint, we need to ensure, at the room level, that all rooms grouped by the zone are available, before checking each room individually for the satisfaction of the precedence constraints (on the operation level). To start an operation instance that is related to a blocking constraint, we also need to make sure that all the adjacent rooms are not occupied by on-going operations (room level).

In the LAS generation process model, we propose three key steps to schedule zone-constraint-related, blocking-constraint-related, and non-OSS-constraint-related operation instances. In each of these steps, we use two types of buckets – the room bucket and the operation bucket – to handle the constraints at different levels. Figure 2-4.a presents an overview of the proposed LAS generation process model. By processing the non-OSS-constraint-related operation instances last, we give the OSS-constraint-related operation instances a chance to start at a relatively early stage in a LAS. Otherwise, such operation instances would always be pushed to the end of the schedule, because the processing of non-OSS-constraint-related operation instances constantly makes rooms less available. Zone constraints are processed before blocking constraints because, according to our field research, blocking-related operations are generally arranged after zone-related ones.

Each key processing step of the LAS generation process model contains a loop, which is intended to process multiple rooms, as illustrated in Figs. 4b, 4c, and 4d. The room bucket reflects the room requirement of an operation and contains two categories, ROOM TODO and ROOM CANDO. The operations are categorized into buckets including OP TODO, OP CANDO, OP DOING, and OP DONE, as in the activity categorization of ACP.

In each of the three processing steps, the room availability is checked before the other constraints. Checking the rooms first accommodates the room priority constraint by sorting the rooms/zones according to their priority and selecting them accordingly. Here we use the processing of zone-constraint-related operation instances (Figure 2-4.b) as an example to explain the detailed procedure in each step. First, a zone is selected from the zone pool of the project, and all the rooms involved in this zone are put to ROOM TODO. When all the rooms are available, they are put into the ROOM CANDO bucket. Next the LAS generation process model puts the zone-constraint-related operation instances into the OP TODO bucket, and checks them to determine whether they are ready (i.e., the relevant precedence constraints are satisfied). If so, these operation instances are moved from the OP TODO bucket to the OP CANDO bucket, waiting for crew assignments. If enough crew(s) are available, these operation instances are moved to OP DOING bucket. The movement of the rooms and operations across the buckets are dependent on the feedback from the information model in response to the query of the LAS generation process model. The OP DONE bucket is not shown in Figs. 4b, 4c, and 4d because these figures only demonstrate the constraint-handling steps before the increment of time. The procedure of processing the non-OSS-constraint-related operation instances (Figure 2-4.d) is similar to that of zone-constraint-related operation instances except that only one room needs to be checked at a time and only one operation can be moved from OP TODO to OP CANDO at a time, if the corresponding precedence constraints are satisfied. Compared to these two processing steps, when processing blocking-constraint-related operation instances (Figure 2-4.c), the LAS generation process model checks the room availability of the adjacent rooms after an operation instance is moved from OP TODO to OP CANDO. Such an order ensures that the blocking-constraint-related operation instances are ready before checking the readiness of the rooms blocked by them.

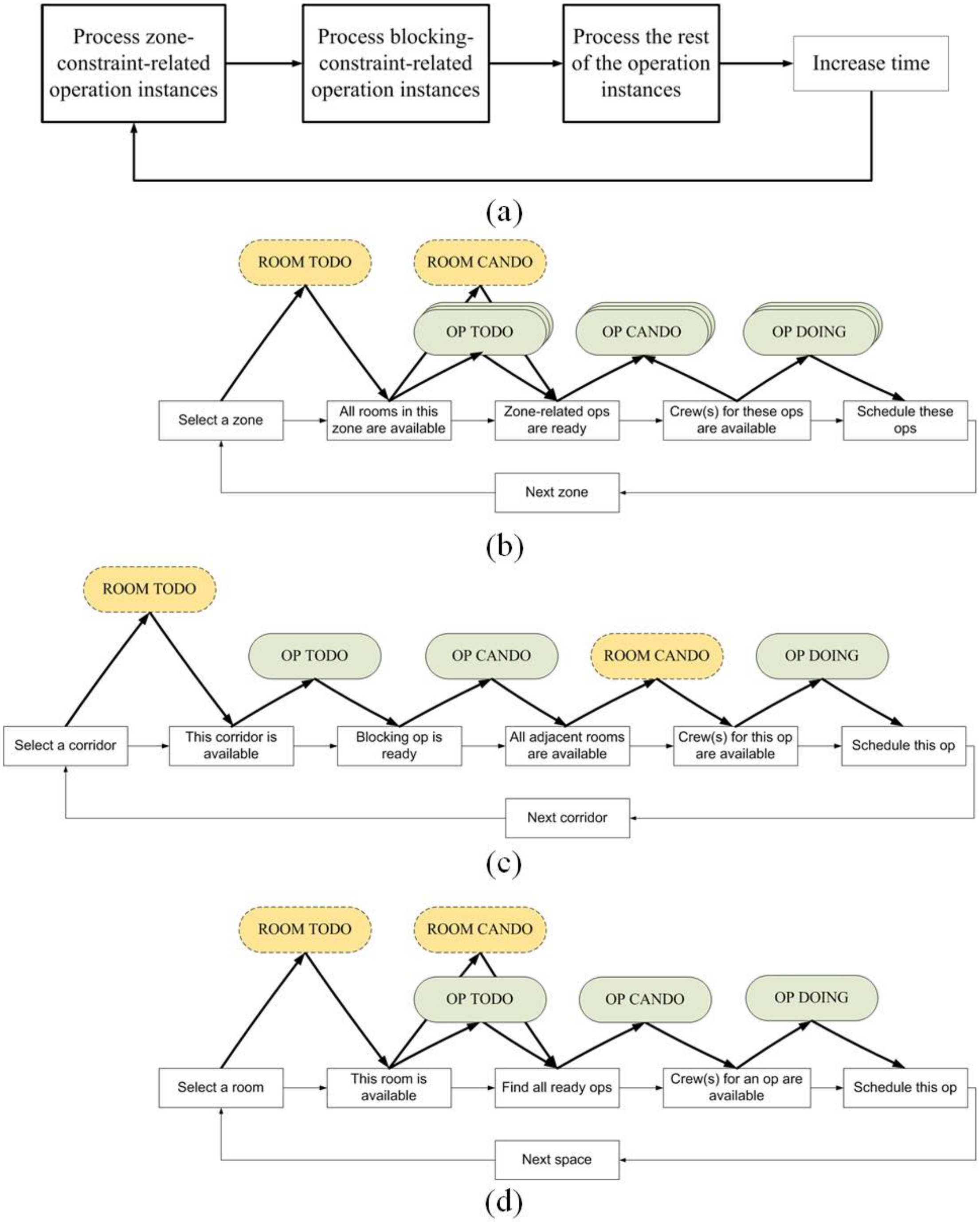


Figure 2-4. The key processing steps of the proposed LAS generation process model (a); the algorithm for processing the zone-constraint-related operation instances (b); the algorithm for processing the blocking-constraint-related operation instances (c); the algorithm for processing other operation instances (d).

4.3 Assumptions made in developing the ALASG method

- When calculating the duration of an operation, a day is the basic unit. If an operation in a room requires less than one day, a whole day is allocated to this operation, assuming the quality inspection for this operation will take up the rest of the day.
- Operations cannot be interrupted. Once assigned to a room, a crew is not allowed to move out of a room until the corresponding operation is finished.
- Average-sized crews are used for all types of rooms. The work assignments of individual skilled workers (Al-Bazi and Dawood 2010; Dong et al. 2011) are not within the scope of this paper.
- At any time, only one operation is allowed to be executed in a room. Thus, the finish-start relation between operations is sufficient to define fragnets.
- The duration of an operation is fixed for the following two reasons. (1) LASs are used for the purpose of mid-term or short-term scheduling. Thus, the construction method for an operation should have been decided before scheduling, and, therefore the productivity rate is fixed. (2) The one-operation-at-a-time assumption eliminates spatial conflict among operations in a room. Thus, there is no variation of productivity due to space sharing (Brodetskaiaa et al. 2011).
- The material, equipment, and engineering documents are assumed to be always available. Thus, only the availability of crews and rooms is considered.

5 System Implementation Illustrated Using a Six-room Case

We developed a prototype, implemented with Microsoft Visual C#, based on the described ALASG method using a six-room case. Computer simulation is applied to the prototype to find near-to-optimum schedules. Since the information model needs to access project databases to retrieve relevant data, we also designed the databases (with xml format) to ensure the availability of minimum sets of data for LAS generation. In accordance with previous studies (Winstanley et al. 1993; Chevallier and Russel 1998; Waugh 1990), we regard the users’ interaction with the scheduling tool as important. Therefore, we also implemented a graphic user interface (GUI) to allow users to input data to reflect the changes on site. These inputs are saved in the project databases ready for use by the information model.

5.1 The six-room case

The proposed six-room case represents a small portion of a complex project in the finishing phase. The data sets were extracted and tailored based on our field research project. Despite its simplicity, it represents the scheduling problem described in this paper. Four types of rooms are included – the IDF room, the electrical (ELE) room, the plant (PLT) room, and the corridor (COR), each of which possesses a unique work sequence.

5.1.1 Fragnets and rooms

Table 2-2 summarizes the fragnets and related rooms of this case study. The same operation name might show up in different fragnets, but their work contents (i.e., materials) and crew productivity rates could be different. For example, the “electrical final fix” in the ELE fragnet concentrates on finalizing the power boxes and panels while in the IDF fragnet it focuses on the installation of data servers. Appropriate crew productivity rates are determined by the information model, as discussed in Section 4.1.1.

Table 2-2. Four types of fragnets involved in the case study.

	ELE Fragnet	**IDF Fragnet**	**PLT Fragnet**	**COR Fragnet**
Rooms	ELE1, ELE2	IDF1, IDF2	PLT1	COR1
Generic operations	E-1. Plastering E-2. Screed E-3. Painting (First Two Coats) E-4. Electrical First Fix E-5. Painting (Last Coat) E-6. Epoxy Floor E-7. Doors & Wood Panels E-8. Electrical Final Fix E-9. Electrical Second Fix E-10. HVAC & Fire Protection	I-1. Plastering I-2. Painting (First Two Coats) I-3. Electrical First Fix I-4. Painting (Last Coat) I-5. Raised Floor I-6. Doors & Wood Panels I-7. Electrical Final Fix I-8. Electrical Second Fix I-9. HVAC & Fire Protection	P-1. Plastering P-2. Screed P-3. Painting (First Three Coats) P-4. Electrical First Fix P-5. Louvers P-6. Painting (Last Coat) P-7. Epoxy Floor P-8. Doors P-9. Electrical Final Fix P-10. Electrical Second Fix P-11. HVAC & Plumbing & Fire Protection	C-1. Conduit & Box C-2 Terrazzo Flooring C-3 Electrical First Fix (Ceiling) C-4 Suspended Ceiling C-5 Dry Wall C-6 Painting (First Two Coats) C-7 Painting (Last Coat) C-8 Electrical Final Fix (wall) C-9 Skirting C-10 HVAC & Plumbing & Fire Protection MEP C-11 Electrical Second Fix (Ceiling) C-12 Electrical Second Fix (Wall) C-13 MEP Final Fix (Ceiling)
Successors	E-1 – E-2 E-2 – E-3 E-3 – E-4, E10 E-4 – E-9 E-9 – E-5 E-10 – E-5 E-5 – E-6 E-6 – E-7 E-7 – E-8	I-1 – I-2 I-2 – I-3, I-9 I-3 – I-5 I-9 – I-5 I-5 – I-8 I-8 – I-4 I-4 – I-6 I-6 – I-7	P-1 – P-2 P-2 – P-3 P-3 – P-4, P-11 P-4 – P-10 P-10 – P-5 P-11 – P-5 P-5 – P-6 P-6 – P-7, P-8 P-7 – P-9 P-8 – P-9	C-1 – C-3, C-5 C-3 – C-10 C-10 – C-11 C-11 – C-4 C-4 – C-2, C-13 C-2 – C-9 C-5 – C-4, C-6, C-12 C-6 – C-8 C-12 – C-8 C-8 – C-7 C-7 – C-9

5.1.2 Types of crew

Each generic operation in Table 2-2 requires a specific type of crew/subcontractor. Table 2-3 lists the relation between a crew/subcontractor type and a generic operation. The relations described in Table 2-3 do not indicate the number of crews/subcontractors required because a generic operation is not linked to a specific room. Site engineers can decide the number of crews/subcontractors necessary in each room for each operation instance.

Table 2-3. Relation between generic operations and types of crew in the case study.

Crew Type	ELE Fragnet	IDF Fragnet	PLT Fragnet	COR Fragnet
Plastering	E-1	I-1	P-1	-
Screed	E-2	-	P-2	-
Carpenter	E-7	I-5, I-6	P-5, P-8	C-4, C-5,
Painting	E-3, E-5, E-6	I-2, I-4	P-3, P-6, P-7	C-6, C-7,
Electrical	E-4, E-8	I-3, I-7	P-4, P-9	C-1, C-3, C-8
Tiling	-	-	-	C-2, C-9
Cable Pulling Sub	E-9	I-8	P-10	C-11, C-12
HVAC Sub	E-10	I-9	P-11	C-10, C-13

5.1.3 OSS constraints

Zone constraint

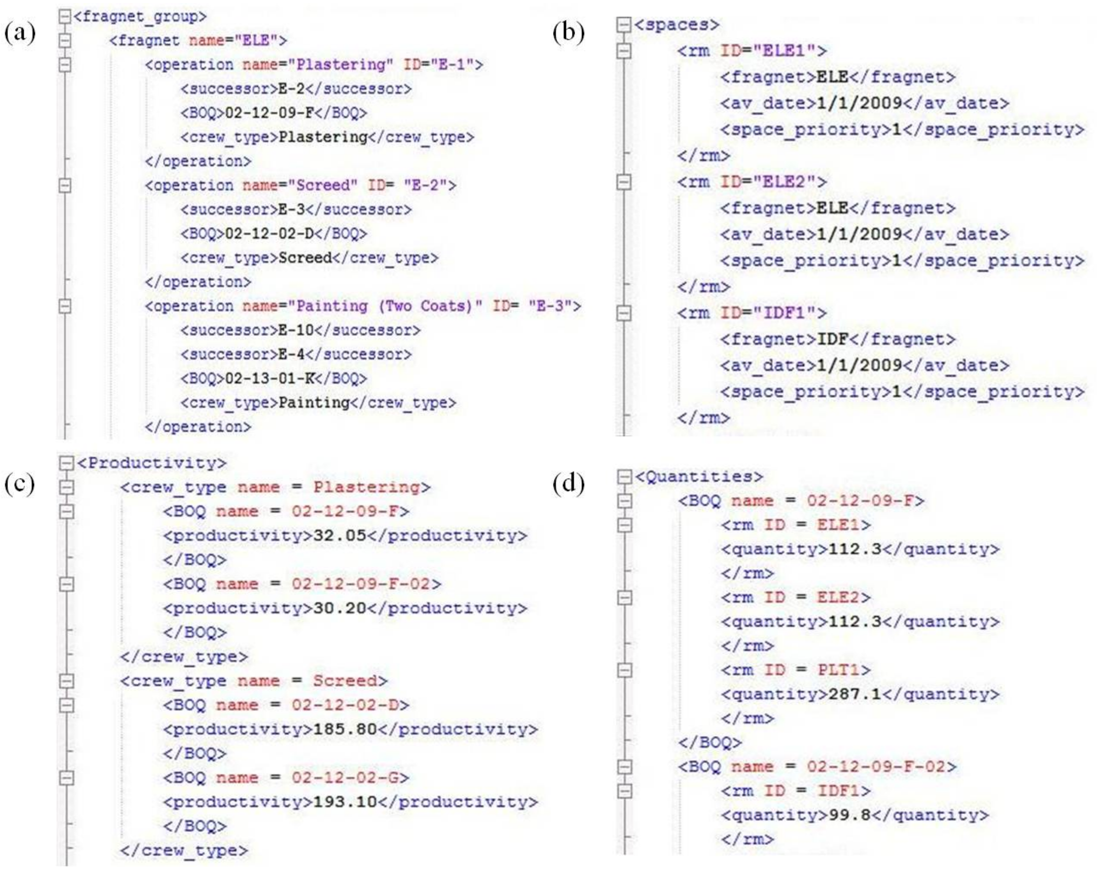
In this case study, it is assumed that there is one zone constraint related with the operation instances “HVAC & Fire Protection” in ELE2, IDF1, and IDF2.

Blocking constraint

The finishing work in corridor COR1 contains two blocking-constraint-related operation instances – “suspended ceiling” and “terrazzo flooring”. The installation of suspended ceiling requires condensed scaffolding which blocks the access to COR1’s adjacent rooms. As to the “terrazzo flooring”, the sanding and terrazzo-applying processes prevent other crews from walking on the floor. COR1 blocks all the other five rooms when these two operation instances are in progress.

5.2 The design of the databases

We created three types of databases (scheduling, design and engineering, and estimation) to meet the data requirements of LAS generation according to Table 2-1. Each database contains a number of xml files, some of which are shown in Figure 2-5. Most of the xml files only contain data relevant to a specific database. For example, the fragnet-related inputs are stored in the scheduling database with one xml file called “fragnet” (Figure 2-5.a). For the convenience of implementation, a few xml files might contain data from two databases such as the “room” file (Figure 2-5.b) – the room ID comes from the design and engineering database and the available date from the scheduling database. The productivity of a type of crew with respect to a material type is recorded in the “productivity” file (Figure 2-5.c). As for the material quantities (Figure 2-5.d), they are first grouped by the BOQ code. Within each group, the material quantity related to a certain room can be found. The inputs shown in Figure 2-5 do not require frequent updates on site when scheduling. However, users should still be allowed to change/update some of them when necessary. Users also need to provide inputs to reflect the dynamic conditions on site. The next section discusses the design of the GUI of the prototype for these purposes.



(a)

```
<fragnet_group>
    <fragnet name="ELE">
        <operation name="Plastering" ID="E-1">
            <successor>E-2</successor>
            <BOQ>02-12-09-F</BOQ>
            <crew_type>Plastering</crew_type>
        </operation>
        <operation name="Screed" ID= "E-2">
            <successor>E-3</successor>
            <BOQ>02-12-02-D</BOQ>
            <crew_type>Screed</crew_type>
        </operation>
        <operation name="Painting (Two Coats)" ID= "E-3">
            <successor>E-10</successor>
            <successor>E-4</successor>
            <BOQ>02-13-01-K</BOQ>
            <crew_type>Painting</crew_type>
        </operation>
```

(b)

```
<spaces>
    <rm ID="ELE1">
        <fragnet>ELE</fragnet>
        <av_date>1/1/2009</av_date>
        <space_priority>1</space_priority>
    </rm>
    <rm ID="ELE2">
        <fragnet>ELE</fragnet>
        <av_date>1/1/2009</av_date>
        <space_priority>1</space_priority>
    </rm>
    <rm ID="IDF1">
        <fragnet>IDF</fragnet>
        <av_date>1/1/2009</av_date>
        <space_priority>1</space_priority>
    </rm>
```

(c)

```
<Productivity>
    <crew_type name = Plastering>
        <BOQ name = 02-12-09-F>
        <productivity>32.05</productivity>
        </BOQ>
        <BOQ name = 02-12-09-F-02>
        <productivity>30.20</productivity>
        </BOQ>
    </crew_type>
    <crew_type name = Screed>
        <BOQ name = 02-12-02-D>
        <productivity>185.80</productivity>
        </BOQ>
        <BOQ name = 02-12-02-G>
        <productivity>193.10</productivity>
        </BOQ>
    </crew_type>
```

(d)

```
<Quantities>
    <BOQ name = 02-12-09-F>
        <rm ID = ELE1>
        <quantity>112.3</quantity>
        </rm>
        <rm ID = ELE2>
        <quantity>112.3</quantity>
        </rm>
        <rm ID = PLT1>
        <quantity>287.1</quantity>
        </rm>
    </BOQ>
    <BOQ name = 02-12-09-F-02>
        <rm ID = IDF1>
        <quantity>99.8</quantity>
        </rm>
```

Figure 2-5. Structured xml files for system inputs: fragnet (a); room-related inputs (b); crew productivity (c), and material quantities (d).

5.3 User inputs and the design of the prototype

To ensure the LASs generated reflect the most up-to-date conditions on site, users need to provide inputs including crew availability, room availability, crew productivity, operation duration, zone constraint, corridor constraint, progress update, and cost-related data.

5.3.1 Crew availability

The number of available crews/subcontractors of a certain trade could vary as the project goes on. Therefore users might need to provide the latest crew availability. As shown in Figure 2-6.a, each type of crew (Table 2-3) entails an entry in the crew tab of the prototype's GUI.

5.3.2 Room availability

If the structural work of a room is delayed, users need to update the room availability (Figure 2-6.b) for the finishing phase so that no finishing-related operations can be scheduled earlier than the actual available date. Users can also specify how urgently they want a room to be completed by providing the information regarding room priority. The available date and priority are saved into the “room” xml file (Figure 2-6.b).

5.3.3 Crew productivity

Users can update the productivity of a crew related to a specific type of material using the GUI shown in Figure 2-6.c to reflect the learning effect of a type of crew as the project goes on. How a crew’s productivity is shown is different from how it is stored in the “productivity” xml file (Figure 2-6.c). Instead of showing the productivity relevant to the obscure BOQ code, the GUI shows it in a more intuitive and understandable way for users. After selecting a fragnet and a generic operation, the crew type, number of crews required, BOQ type, and crew productivity show up in the productivity tab (Figure 2-6.c). Users can change the crew productivity according to the actual site conditions. Showing the crew productivity in such a structure requires the GUI to communicate with the information model so that the correct data sets can be presented to users and then updated according to their inputs.

5.3.4 Duration

There are times when certain data entries, such as BOQ code, quantity of material, and crew productivity, are not available. When this happens, the duration of an operation cannot be calculated. In this case, users can provide the duration directly as indicated in Figure 2-6.d. Users can also change the number of crews required for this operation.

5.3.5 Zone constraint

Figure 2-6.e shows how users can define a zone according to the design specification and field requirements. Users can add desired rooms to a zone and then specify its name, duration of the related operations, the crew type, and number of crews required. Once specified, the crews will be working in the zone without differentiating in which rooms they need to work on a certain day. If users do not provide the duration, it will be calculated based on the integration and aggregation functions of the information model, as discussed in Section 4.1.

5.3.6 Blocking constraint

Users can define blocking constraints in a similar way to the zone constraint. As indicated in Figure 2-6.f, first the specific corridor needs to be selected. Then, users need to specify which operations in the corridor are related with blocking constraints. Finally, they need to indicate which rooms the operation in this corridor blocks.

5.3.7 Progress update

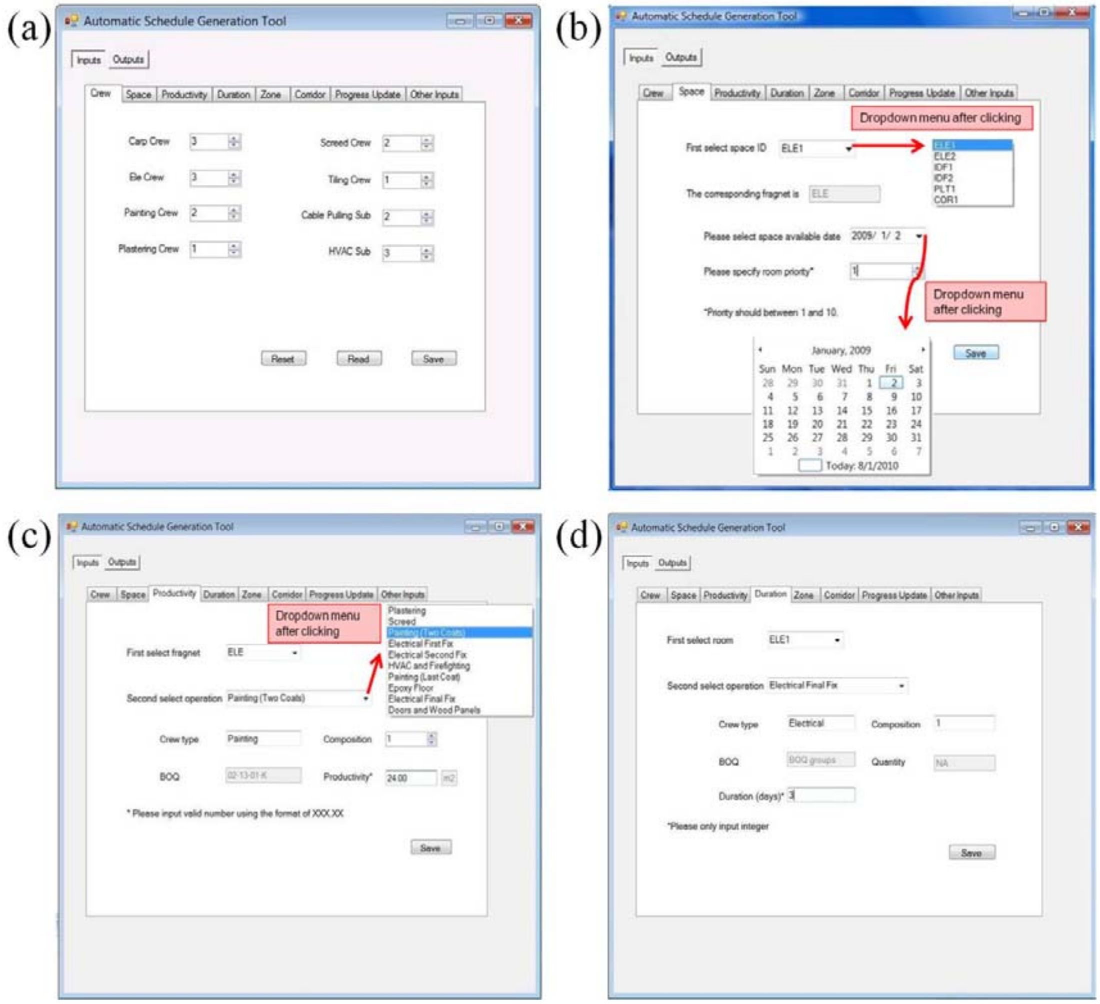
Progress update of operations is another crucial aspect the prototype takes into account so that the LASs generated can give crews the most updated and accurate work assignments. Figure 2-6.g shows an example of operation progress update. Users need to specify the date for which they received the most recent update on site. Then, they select the room and operation they need to update. To track the performance of certain crews/subcontractors, they need to provide their IDs. Also, if the operation did not start as planned in the last LAS, users need to input the actual start date. The prototype then automatically calculates the remaining days of this operation based on the provided inputs. If users think the calculated remaining days do not conform to the actual condition on site, they can make modifications. If an operation is already completed, users need to enter the date on which it was finished. All the inputs fed to the GUI of the progress update tab are stored back to the scheduling database.

5.3.8 Cost

To generate a cost-loaded LAS, construction cost is calculated as the total of direct costs and indirect cost. For each discipline/trade, the direct costs include a crew's daily cost as well as the mobilization and demobilization costs. Using the example in Figure 2-6.h, if a crew does not have work for more than 10 days, a GC should demobilize this crew (and relocate it to another project site) to cut the daily direct cost. Otherwise, they should not be demobilized even if they have to stay idle (and get paid) for a few days. As for the indirect cost, the construction manager should provide these data according to the company's historical data, site conditions, and actual project features. The construction cost can be calculated using Eq. (1).

$$cost = I * P + \sum_{i=1}^{n} (M_i + L_i) * D_i \tag{1}$$

The indirect cost is calculated by multiplying the daily indirect cost I by the construction duration P. The direct cost is represented by a given crew's cost. For a crew i, M_i is the sum of the mobilization and demobilization costs measured by days. L_i is the total number of days during which crew i stays on site (regardless whether they are working or not). D_i is the daily cost of crew i.



(a)

Carp Crew 3

Screed Crew 2

Ele Crew 3

Tiling Crew 1

Painting Crew 2

Cable Pulling Sub 2

Plastering Crew 1

HVAC Sub 3

(b)

Dropdown menu after clicking

First select space ID ELE1

The corresponding fragnet is ELE

Please select space available date 2009/ 1/ 2

Please specify room priority* 1

*Priority should between 1 and 10.

Dropdown menu after clicking

(c)

Dropdown menu after clicking

First select fragnet ELE

Second select operation Painting (Two Coats)

Crew type Painting

Composition 1

BOQ 02-13-01-K

Productivity* 24.00 m2

* Please input valid number using the format of XXX.XX

(d)

First select room ELE1

Second select operation Electrical Final Fix

Crew type Electrical

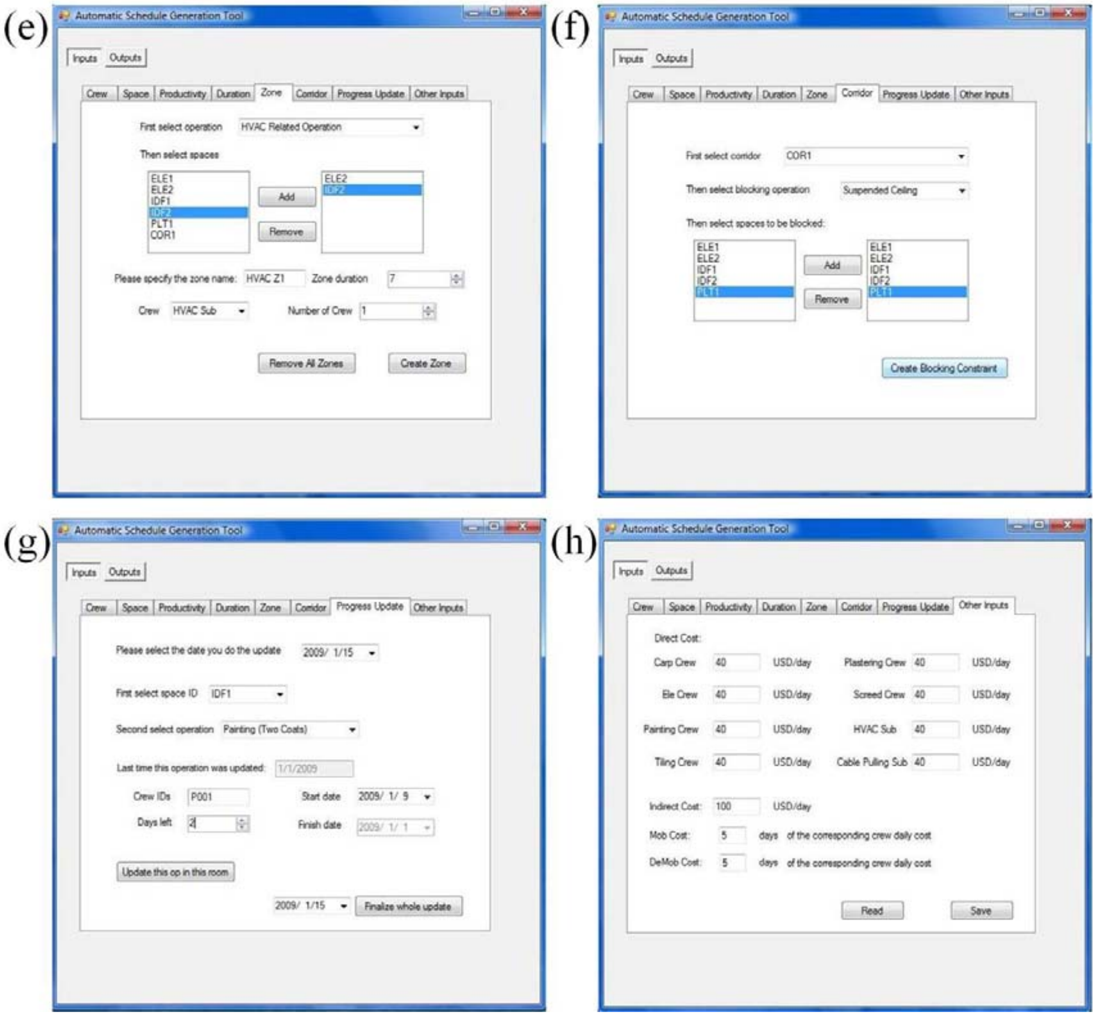
Composition 1

BOQ BOQ groups

Quantity NA

Duration (days)* 3

*Please only input integer



(e)
First select operation: HVAC Related Operation
Then select spaces: ELE1, ELE2, IDF1, IDF2, PLT1, COR1 → ELE2, IDF2
Please specify the zone name: HVAC Z1
Zone duration: 7
Crew: HVAC Sub
Number of Crew: 1

(f)
First select corridor: COR1
Then select blocking operation: Suspended Ceiling
Then select spaces to be blocked: ELE1, ELE2, IDF1, IDF2, PLT1 → ELE1, ELE2, IDF1, IDF2, PLT1

(g)
Please select the date you do the update: 2009/ 1/15
First select space ID: IDF1
Second select operation: Painting (Two Coats)
Last time this operation was updated: 1/1/2009
Crew IDs: P001
Start date: 2009/ 1/ 9
Days left: 2
Finish date: 2009/ 1/ 1
2009/ 1/15

(h)
Direct Cost:

Carp Crew	40	USD/day	Plastering Crew	40	USD/day
Ele Crew	40	USD/day	Screed Crew	40	USD/day
Painting Crew	40	USD/day	HVAC Sub	40	USD/day
Tiling Crew	40	USD/day	Cable Pulling Sub	40	USD/day

Indirect Cost: 100 USD/day
Mob Cost: 5 days of the corresponding crew daily cost
DeMob Cost: 5 days of the corresponding crew daily cost

Figure 2-6. The GUI for user inputs: crew availability (a); room availability (b); crew productivity (c); duration of an operation (d); zone constraint (e); blocking constraint (f); progress update (g); and cost (h).

5.4 Application of computer simulation to the prototype

Computer simulation can be used to generate multiple schedule alternatives to test and evaluate the effectiveness of various resource allocation policies towards specific project goals (Ahuja and Nandakumar 1985; Pidd 1998; Alberto et al. 2002). In this research project, we adopt computer simulation in our prototype to present users the best LAS alternative(s) out of a certain number of runs of simulation (specified by the users) towards a certain project goal (i.e., shortest construction duration or lowest cost). We developed our simulation-based scheduling method in the prototype on the basis of the activity-based construction (ABC) modelling and simulation method, which is composed of an ABC modelling (ABC-Mod) component and an ABC simulation (ABC-Sim) component (Shi 1999). ABC-Mod models construction processes using an activity-centered approach; ABC-Sim executes the ABC model using three steps in loops: schedule activities, advance simulation, and release resources. We chose the ABC method as the foundation of our simulation-based scheduling method because (1) the activity-centered approach in ABC-Mod resonates with our information modelling approach, in particular with the core function to create operations with the appropriate form, and (2) the three steps used in ABC-Sim can be smoothly extended to simulate the processing steps of the proposed LAS generation process model.

ABC-Mod models construction processes assuming all the activities belong to one single project. In our simulation-based approach, we expanded ABC-Mod by directly using the information model of ALASG to model operations and their related constraints with a three-tiered hierarchy: project, sub-project (i.e., room), and operation. ABC-Sim checks all the constraints (i.e., resource and precedence constraints) in the first step of its three-step loop. We expanded this step by accommodating the additional OSS constraints using the three key processing steps and the two types of buckets of the LAS generation process model introduced in Section 4.2. LAS alternatives are created after a resource is randomly assigned to one operation out of multiple operations competing for the same limited resource at the same time.

5.5 Outputs and possible applications

5.5.1 LAS outputs with duration distribution

Using the six-room example with only one crew available for each discipline, the project schedule distribution based on a 10,000-run simulation is shown in Figure 2-7. The given project condition is that (1) all six rooms possess the same priority and availability; and (2) no finishing operations have started yet. On a PC with an Intel Core(TM)2 Duo CPU (2.53GHz), the simulation takes 8.7 minutes. The schedule duration ranges from 74 days to 99 days. As indicated in Figure 2-7, the possibility of getting a schedule with the duration of 81 days is much higher than the other schedule options under the given project settings.

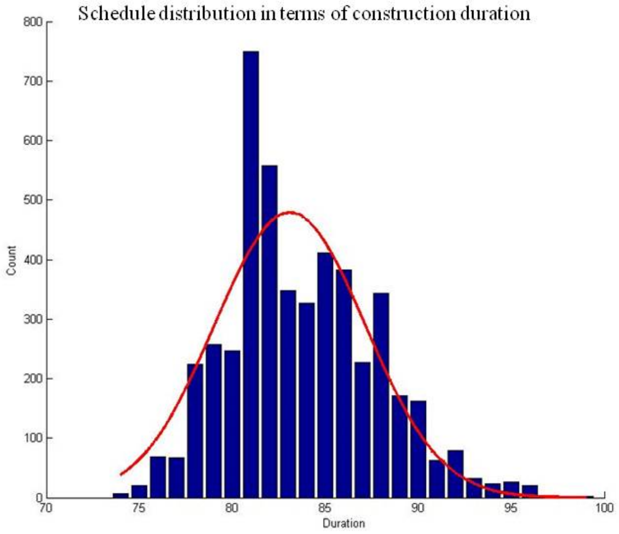
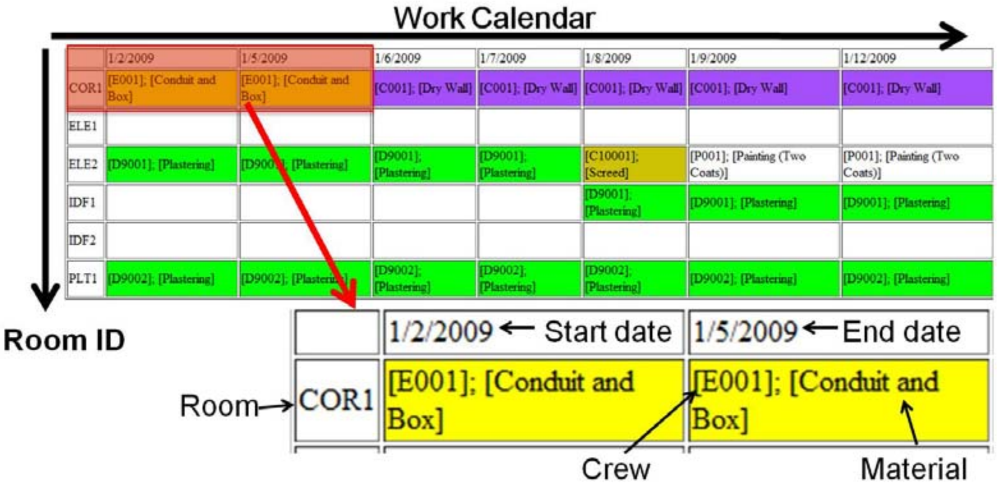


Figure 2-7. Schedule distribution of 10,000 runs of the simulation.

Figure 2-8 illustrates the beginning part of a sample LAS generated. Horizontally, it shows the operations on a daily basis. Vertically, it shows the work content inside each room. The “conduit & box installation” operation in room COR1 is enlarged to illustrate the operation with the form of [crew, material, start and end dates, room] or, in this case, [E001, conduit and box, 1/2/2009 and 1/5/2009, COR1].



	1/2/2009	1/5/2009	1/6/2009	1/7/2009	1/8/2009	1/9/2009	1/12/2009
COR1	[E001]; [Conduit and Box]	[E001]; [Conduit and Box]	[C001]; [Dry Wall]	[C001]; [Dry Wall]	[C001]; [Dry Wall]	[C001]; [Dry Wall]	[C001]; [Dry Wall]
ELE1							
ELE2	[D9001]; [Plastering]	[D9001]; [Plastering]	[D9001]; [Plastering]	[D9001]; [Plastering]	[C10001]; [Screed]	[P001]; [Painting (Two Coats)]	[P001]; [Painting (Two Coats)]
IDF1					[D9001]; [Plastering]	[D9001]; [Plastering]	[D9001]; [Plastering]
IDF2							
PLT1	[D9002]; [Plastering]	[D9002]; [Plastering]	[D9002]; [Plastering]	[D9002]; [Plastering]	[D9002]; [Plastering]	[D9002]; [Plastering]	[D9002]; [Plastering]

	1/2/2009	1/5/2009
COR1	[E001]; [Conduit and Box]	[E001]; [Conduit and Box]

Figure 2-8. A sample LAS with an enlarged operation with the form of [crew, material, start and end dates, room].

5.5.2 Construction time/cost trade-off study

With simulation runs, the prototype can help construction managers evaluate the relation between construction duration and cost. Figure 2-9 shows the time/cost trade-off of the 10,000-run simulation. The construction cost of the shortest schedules (74 days) ranges from $23,440 to $23,800. However, the minimum cost found is $22,900, which corresponds to the duration of 77 days. Therefore, the shortest construction duration does not necessarily entail minimum cost.

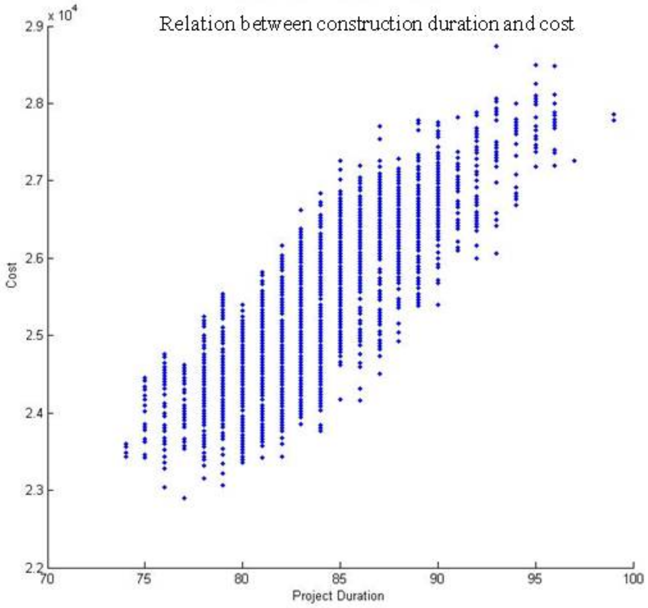


Figure 2-9. Construction time/cost trade-off study for the six-room case based on the 10,000-run simulation.

5.5.3 Resource utilization study

Prior studies point out that efficient resource utilization leads to better schedule in terms of construction duration or cost. However, such studies either target repetitive projects (Harris and Ioannou 1998; Vanhoucke 2006) or discuss cases from manufacturing (Al-Bazi and Dawood 2010). For non-repetitive construction projects in the finishing phase, little research has been conducted to study the relation between the utilization of two types of resource and project goals. The ALASG-based prototype supports such a study. Figure 2-10 depicts the relation between resource utilization and construction cost based on the 10,000-run simulation. Resource utilization is represented by the total idle days of the room or crew resource. Figure 2-10 indicates that eliminating crew idle days helps to reach minimum construction cost while increasing room occupancy does not necessarily help achieve the same purpose.

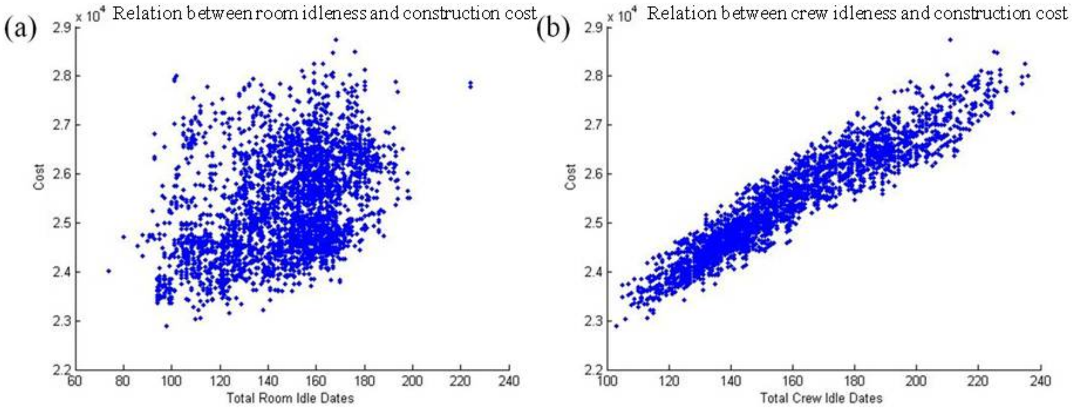


Figure 2-10. The relation between resource utilization and construction cost based on the 10,000-run simulation: room idleness and construction cost (a); and crew idleness and construction cost (b).

6 Validation

Two methods were used to validate the ALASG method. First, we developed a schedule-checking program to examine the accuracy of schedules produced by the prototype. The aspects we checked with respect to schedule accuracy, according to our interviews previously mentioned in Section 4, included: work sequence in each room, crew allocation conflicts on each day, conformance to the constraints, duration of each operation in the schedule, and the calculation of construction duration and cost. The program was applied to test all the schedules generated. The schedule-checking program did not discover any errors with regard to the aforementioned aspects.

Next, the ALASG method was validated using two charrette tests (Clayton et al. 1998), one with three practitioners (two project planners and one site engineer) from CCC and another with seven graduate students from the Construction Engineering and Management Program at Stanford University. The goal of the charrette tests was to compare the manual LAS generation method with the ALASG method (via the prototype) in terms of time spent in LAS generation and the quality of the LASs created, measured by the number of errors found in the LASs and their construction duration or cost. In these tests, the participants were first asked to manually generate one LAS for the six-room case (Section 5.1), towards a specific project goal (lowest construction cost for the practitioners and shortest construction duration for the graduate students), with permission to use any commercial scheduling tools such as Microsoft Project, Excel, Primavera, and Vico Control. All the necessary inputs for the LAS generation, detailed in Table 2-1, were provided in Excel spreadsheets. After the participants created the LASs, they were asked to use the prototype to run the simulation for 100 times to find the best LAS(s) toward the same goal. Table 2-4 shows part of the charrette test results. On average, the participants spent 96% less time in LAS generation using the prototype compared to using the manual LAS generation method. The participants spent more than 90% of their time operating the prototype and doing inputs before configuring and running the simulations. Only one error-free LAS was found out of the 10 LASs created by the 10 participants using the manual LAS generation method. This LAS was created by one project planner and its construction cost was $26,820. This same project planner found a LAS with the cost of $24,680 when using the prototype, 8% lower than that of the LAS manually created. The average construction duration of the LASs created using the prototype by the seven graduate students is 75.2 days, 6.7% shorter than using the manual method. However, since none of the LASs created manually were correct, such a comparison might eclipse the power of the ALASG method.

Table 2-4. Partial results of charrette tests of LAS generation.

Type of Participants	**Manual LAS Generation**		**LAS Generation Using the Prototype**	
	Time Spent (seconds)	**Average # of Errors found**	**Time Spent (seconds)**	**Average # of Errors found**
3 Practitioners	5982	1	240	0
7 Graduate students	5178	2.3	228	0

7 Conclusions

The proposed ALASG method can quickly generate error-free, close-to-optimum LASs for the finishing phase of complex projects. It extends prior research on information modelling for construction scheduling and automated construction schedule generation by introducing an information model and a LAS generation process model. The information model creates operation instances with the following form: [type of crew and number of crews required, material (type and quantity), duration, room] and accurately connects relevant constraints to these operation instances. The LAS generation process model transforms the operation instances to operations with the following form: [crew, material, start and end dates, room] using a constraint-based approach. To organize project inputs effectively, the information model employs a POP-based architecture to integrate and aggregate project data from various project databases at appropriate levels of detail. The LAS generation process model utilizes two types of buckets–room and operation–to enable automated scheduling without violating any constraints. The LASs created by ALASG address the commonly found constraints on site (i.e., precedence constraints, crew and room availability, and OSS constraints) and provide sufficient detailed to guide the crews' daily work.

The prototype developed on the basis of the ALASG method facilitates the user interactions with the scheduling system to ensure that the most up-to-date inputs are readily available before scheduling. According to the findings of the six-room case study, the ALASG method, when used with computer simulation, can save site engineers and project planners significant amounts of time in LAS generation and produce accurate LASs of higher performance in terms of construction duration or cost. Although the data sets for the case study were extracted from our field research on a university building project, the design of ALASG makes it equally applicable for the look-ahead scheduling in the finishing phase of construction projects such as hospitals and medical centers, offices, data centers, laboratories, and research facilities.

Coupled with the simulation-based approach, ALASG can help site engineers and project planners investigate the relation between resource utilization and project goals in look-ahead scheduling in the finishing phase of complex projects, as is further discussed Chapter 3. Although the simulation-based approach can help deliver accurate LASs in a timely fashion, compared to other optimization methods, it could consume a great amount of time in obtaining optimal or near-optimal solutions, depending on the scale of a project (Zhang and Li 2004). Therefore, it would be helpful to apply other optimization methods to the proposed scheduling problem in this paper and compare their performance to that of the simulation-based method. In addition, more complex cases could be used to evaluate the effectiveness of these optimization methods for larger projects. These steps are described in Chapter 4. One key assumption when creating LASs is that only one operation can be carried out in a room at a time. When a room becomes very large (such as a 200-seat lecture hall), multiple operations can be performed at the same time, if they are independent from each other. When scheduling for such rooms, sub-spaces should be defined using the workspace definitions proposed by prior studies (Thabet and Beliveau 1997; Akinci et al. 2002) to avoid work conflict and decrease of crew productivity. Operation relations in addition to finish-start should be used for these large rooms. Another limitation of ALASG is that the material, equipment, and engineering documents are assumed to be always available. ALASG needs to be extended to accommodate the actual availability of these items so that the LASs created can reflect the specific conditions on site. When calculating operation duration, ALASG assumes that (1) crew productivity is fixed and (2) the a day is the basic unit for duration. Learning curve effects and a higher granularity of time (such as hour or even minute) should be considered in the future so that the LASs generated can bear more practical value to the general contractors.

8 References

Aalami, F., Kunz, J., and Fischer, M. (1998). "Model-based mechanisms for automated activity sequencing." Stanford University, Center for Integrated Facility Engineering, Technical Report 50.

Aalami, F. (1998). "Using method models to generate 4D production models." Ph.D. Dissertation, Stanford University, Stanford, CA.

Akinci, B., Fischer, M., Levitt, R., and Carlson, R. (2002a). "Formalization and automation of time-space conflict analysis." *Journal of Computing in Civil Engineering,* 16(2), 124-134.

Akinci, B., Fischer, M., and Kunz, J. (2002b). "Representing work spaces generically in construction method models." *Journal of Construction Engineering and Management*, 128(4), 296-305.

Akinci, B., Fischer, M., and Kunz, J. (2002c). "Automated generation of work spaces required by construction activities." *Journal of Construction Engineering and Management,* 128 (4), 306-305.

Ahuja, N.T.H., and Nandakumar, V. (1985). "Simulation model to forecast project computation time." *Journal of Construction Engineering and Management*, 119(2), 355-373.

Al-Bazi, A., and Dawood, N. (2010). "Developing crew allocation system for the precast industry using genetic algorithms." *Computer-Aided Civil and Infrastructure Engineering,* 25(8), 581-595.

Chapter 2

Alberto, L., Azcarate, C., Mallor, F., and Mateo, P.M. (2002). "Optimizaton with simulation and multi-objective analysis in industrial decision-making: a case study." *European Journal of Operational Research,* 140(2), 373-383.

Ballard, G. (1997). "Lookahead planning: the missing link in production control." Proceedings of 5th Annual Conference of the International Group for Lean Construction (IGLC-5), 13-25.

Brodetskaiaa, I., Sacksa, R., and Shapiraa, A. (2011). "A workflow model for systems and interior finishing works in building construction." *Construction Management and Economics,* 29(12), 1209-1227.

Cherneff, J., Logcher, R., and Sriram, D. (1991). "Integrating CAD with construction-schedule generation." *Journal of Computing in Civil Engineering,* 5(1), 64-84.

Chevallier, N., and Russell, A. D. (1998). "Automated schedule generation." *Canadian Journal of Civil Engineering,* 25, 1059–1077.

Clayton, M., Kunz, J., and Fischer, M. (1998). "The Charrette test method." Stanford University, Center for Integrated Facility Engineering, Technical Report 120.

Clough, R. H. and Sears, G. A. (1991). *Construction project management,* Jon Wiley & Sons, New York.

Construction Owners Association of Alberta, WorkFace Planning, <http://www.coaa.ab.ca/Productivity/WorkFacePlanning/Summary.aspx> (June 4, 2012).

Darwiche, A., Levitt, R., and Hayes-Roth, B. (1988). "OARPLAN: generating project plans by reasoning about objects, actions and resources." *AI EDAM,* 2(3), 169-181.

Dawood, N., and Mallasi, Z. (2006). "Construction workspace planning: assignment and analysis utilizing 4D visualization technologies." *Computer-Aided Civil and Infrastructure Engineering,* 21(7), 498–513.

Dong, N., Fischer, M., and Haddad, Z. (2011). "Automatic look-ahead schedule generation system for the finishing phase of complex projects for general contractors." *Proceedings of the 2011 ASCE International Workshop on Computing in Civil Engineering,* 134-142.

Dong, N., Ge, D., Fischer, M., and Haddad, Z. (2012). "A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects." *Advanced Engineering and Informatics,* to appear.

Echeverry, D., Ibbs, W., and Simon, K. (1991). "Sequencing knowledge for construction scheduling." *Journal of Construction Engineering and Management,* 117(1), 118–130.

Feng, C., Chen, Y., and Huang, J. (2010). "Using the MD CAD model to develop the time–cost integrated schedule for construction projects." *Automation in Construction,* 19(3), 347–356.

Fischer, M., and Kunz, J. (2004). "The scope and role of information technology in construction." Stanford University, Center for Integrated Facility Engineering, Technical Report 156.

Fowler, M., and Scott, K. (1999). *UML distilled,* Addison Wesley Longman.

Fricke, S.E., and Shenhar, A.J. (2000). "Managing multiple engineering projects in a manufacturing support environment." *IEEE Transactions on Engineering Management*, 47(2), 258-268.

Guo, S. J. (2002). "Identification and resolution of work space conflicts in building construction." *Journal of Construction Engineering and Management,* 128(4), 287-295.

Harris, R. B., and Ioannou, P. G. (1998). "Scheduling projects with repeating activities." *Journal of Construction Engineering and Management,* 124(4), 269-278.

Hendriks, M.H.A., Voeten, B., and Kroep, L. (1999). "Human resource allocation in a multi-project R&D environment.", *International Journal of Production Management*, 17(2) 181-188.

Jaafari, A. (1984). "Criticism of CPM for project planning analysis." *Journal of Construction Engineering and Management,* 110(2), 222-233.

Kam, C., and Fischer, M. (2004). "Capitalizing on early project decision-making opportunities to improve facility design, construction, and life-cycle performance–POP, PM4D, and decision dashboard approaches." *Automation in Construction,* 13(1), 53-65.

Kanit, R., Gunduz, M., and Ozkan, O. (2009). "Investigating the effectiveness of certain priority rules on resource scheduling of housing estate projects." *Journal of Construction Engineering and Management,* 135(7), 609-613.

Kataoka, M. (2008). "Automated generation of construction plans from primitive geometries." *Journal of Construction Engineering and Management*, 134(8), 592-600.

Kenley, R., and Seppänen, O. (2009). Location-based management for construction: Planning, scheduling and control, Spon Press, Abingdon, Oxon, UK.

Koskela, L., and Howell, G. (2002). "The underlying theory of project management is obsolete." *Proceedings of PMI Research Conference,* 293-301.

Kunz, J. and Fischer, M. (2005). "Virtual design and construction: themes, case studies and implementation suggestions." Stanford University, Center for Integrated Facility Engineering, Working Paper 97.

Marshall, G., Barber, T.J., and Boardman, J.T. (1987). "Methodology for Modelling a Project Management Control Environment." *IEE Proceedings,* 134(4), 287-300.

Mourgues, C. (2009). "Method to produce field instructions from product and process models." Ph.D. Dissertation, Stanford University, Stanford, CA.

O'Brien, J. J., and Plotnick, F. L. (2005). *CPM in construction management,* 6th Edition, McGraw-Hill, Inc., New York.

Pidd, M. (1998). *Computer Simulation in Management Science,* Wiley, Chichester, New York.

Riley, D. (1994). "Modelling the space behaviour of construction activities," Ph.D. Dissertation, Pennsylvania State University, University Park, PA.

Riley, D., and Sanvido, V. (1997). "Space planning method for multi-story building construction." *Journal of Construction Engineering and Management,* 123(2), 171-180.

Shi, J.J. (1999). "Activity-based construction (ABC) modeling and simulation method." *Journal of Construction Engineering and Management,* 125(5), 354-360.

Sriprasert, E., and Dawood, N. (2002) "Requirements identification for 4D constraint-based construction planning and control systems." *Proceedings of CIB W78 Conference,* 1-8.

Thabet, W. Y., and Beliveau, Y. J. (1997). "SCaRC: space-constrained resource-constrained scheduling system." *Journal of Computing in Civil Engineering,* 11(1), 48–59.

Vanhoucke, M. (2006). "Work continuity constraints in project scheduling." *Journal of Construction Engineering and Management,* 132(1), 14-25.

Waugh, L. (1990). "A construction planner," Ph.D. Dissertation, Stanford University, Stanford, CA.

Winstanley, G. Chacon, M., and Levitt, R. (1993). "Model-based planning: scaled-up construction application." *Journal of Computing in Civil Engineering,* 7(2), 199-217.

Yau, N., Melin, J., Garrett, J., and King, S. (1991). "Integrating the processes of design, scheduling, and cost estimating within an object-oriented environment." *In Preparing for Construction in the 21st Century, Proceedings of Construction Congress,* 342-347.

Zhang, H., and Li, H. (2004). "Simulation-based optimization for dynamic resource allocation." *Automation in Construction,* 13(3), 409-420.

Zozaya-Gorostiza, C., Hendrickson, C., and Rehak, D. (1989). *Knowledge-based process planning for construction manufacturing,* Academic Press Inc., San Diego, CA.

Chapter 3: Evaluation of Resource Utilization in Look-ahead Scheduling for the Finishing Phase of Non-repetitive Building Projects

Ning Dong, Martin Fischer, Wei Qian, and Zuhair Haddad

1 Abstract

General contractors (GCs) consider resource utilization as an important factor in construction scheduling, especially look-ahead scheduling, regardless of the existence of repetitive features on projects. Crew idleness is the commonly used measurement of resource utilization for repetitive projects. However, for the finishing phase of non-repetitive building projects, such as university buildings and hospitals, it is unclear whether crew idleness should be used as the sole indicator to represent resource utilization, when the utilization of rooms becomes a key factor to be considered in look-ahead schedule (LAS) generation. This paper discusses the most relevant measurements to measure resource utilization when scheduling the finishing phase of non-repetitive building projects. This paper then uses these measurements to evaluate the relations between resource utilization and project goals. Four measurements are defined and applied in an automated schedule generation method to produce a sufficient number of LASs to facilitate the evaluation process. Statistical analysis is conducted to evaluate the relations between key measurements and the relationship between resource utilization and construction duration. This study contributes to the field of resource utilization in LAS generation for the finishing phase of non-repetitive building projects through the following key findings: (1) interruption should be used as another measure of resource utilization, (2) the

workforce resource (i.e., skilled-workers and crews) and room resource should be treated separately, and (3) the correlation between the utilization of a type of resource and total construction duration depends on the number of units available for this type of resource.

2 Introduction

Look-ahead scheduling in the finishing phase of non-repetitive building projects can be very challenging for project planners, site engineers, and construction managers. Compared to repetitive projects, in which activities are repeated in units that have the same work sequence (El-Rayes and Moselhi 1998; El-Rayes 2001b), non-repetitive building projects in the finishing phase contain multiple types of units/rooms, each type entailing an unique work sequence. An example of such a non-repetitive project is the university building project in which we conducted field research. The project includes around 200 rooms that can be categorized into 18 types (e.g., faculty offices, conference rooms, lecture halls, computer classrooms, toilets, electrical rooms, and plant rooms), each of which entails a unique work sequence for the finishing phase. One of the challenges of look-ahead scheduling in the finishing phase of such projects is the management of limited crew and space resources.

Most of the prior studies on crew availability and scheduling methods concentrate on resource utilization at the crew level, assuming that a skilled worker stays in the same crew throughout a project for a particular type of work. In the context of creating LASs for the finishing phase of non-repetitive building projects, this assumption does not always hold, because the installation of different types of materials in different types of rooms can require different numbers of skilled workers (as well as laborers) in a given crew. Therefore, when generating LASs we need to consider resource utilization at the skilled-worker (and laborer) level (Al-Bazi and Dawood 2010; Dong et al. 2011). We also need to reduce the number of times workers interrupt work on similar units/rooms so that (1) learning curve effects (Shtub et al. 1996) can be maximized and (2) the extra effort associated with work interruptions including setup time, mobilization/demobilization, and temporary storage of tools/materials can be minimized (Thabet and Beliveau 1994a). This aspect of resource utilization has been referred to as *work continuity constraint* by prior researchers in the domain of scheduling for repetitive construction projects, such as high-rise buildings, housing projects, highways, pipeline networks, bridges, tunnels, railways, airport runways, and water and sewer mains (El-Rayes 2001b).

Thabet and Beliveau (1994b) propose to include the availability of workspace as another constraint in scheduling for repetitive building projects, for three reasons. First, due to the storage limitation for construction materials and equipment, materials end up being stored in the work areas, which affects the overall availability of spaces for the activities/operations executed in these areas. Second, certain activities require that a large amount of work area be reserved until their completion. For example, the installation of raised floors in a 12' by 12' IDF room will take up the entire work area in the room, preventing any other activities from proceeding until the installation is finished. Third, although a space may be available to accommodate crews of different trades working concurrently, it is sometimes implied (by mutual understanding of the trades people) or specifically stated (by the main contractor or construction manager) that only one trade at a time can use the space to work or store their materials/tools. Although Thabet and Beliveau use these three reasons to justify their consideration of workspace availability in scheduling for repetitive building projects, these reasons are equally applicable to non-repetitive building projects. The difference is that instead of considering workspace availability at the floor level, we need to consider this availability at the room level.

In the finishing phase of non-repetitive building projects, the room is the basic production unit, with specific work content. It provides a natural and convenient representation for spatial reasoning (Cherneff et al. 1991) and accounts for spaces in which the crews and equipment must be allocated regardless of the shape of the space, unlike the work spaces defined and measured by length, width, and height (Thabet and Beliveau 1997; Akinci et al. 2002). The effective use of rooms in the finishing phase of non-repetitive building projects leads to improved safety, reduced work conflict among workers, increased productivity, and higher work quality.

The efficient utilization of crew and room resources when scheduling need to bear some relation to project goals such as minimizing construction duration or cost, because these goals are what general contractors ultimately care about. In the aforementioned university building project in the finishing phase, we observed a debate between two groups of site engineers. Some site engineers insisted that if they made all the workers as busy as possible, they could achieve the shortest schedule; others argued that if they made all the rooms as occupied/busy as possible, they could obtain the shortest schedule.

Such a debate leads to the following questions when generating LASs in the finishing phase of non-repetitive projects: (1) How can "busyness" of a type of resource be defined in order to measure resource utilization? (2) Is the utilization of the crew resource independent from that of the room resource? (3) What is the relationship between resource utilization and project duration?

3 Prior Studies on Resource Utilization

Our literature review regarding resource utilization is in line with the three questions brought forward in the previous section.

3.1 Indicators of resource utilization

A major study of seven Swedish construction projects (Josephson 2005) cited by several researchers (Jongeling and Olofsson 2007; Lu and Olofsson 2009) reveals that only 15-20% of a construction worker's time is spent on direct work and about 45% on indirect work. The remaining 35% is spent on interruption, waiting, and so on, i.e., wasted. This waste is tackled by studies in the domain of repetitive project scheduling, via ensuring or enhancing work continuity, a way to maximize the continuous utilization of the crew resource. For linear projects (such as pipelines, railroads, and highways) that are repetitive due to their linear geometrical layout, the goals are to achieve zero crew idle time (Selinger 1980; Reda 1990), improve the crew's learning curve (Ashley 1980; Birrell 1980), and eliminate the mobilization/demobilization cost of crews and equipment (Moselhi and Hassanein 2003). In order to eliminate crew idle time, zero interruptions are allowed in such studies. However, as pointed out by Russell and Caselton (1988) and Moselhi and El-Rayes (1993), when scheduling for most real-world repetitive projects in construction, maintaining absolute continuity is very difficult, if not impossible, for reasons including the following. (1) Crew size for the same type of activities at different work locations/units can change. (2) Duration of these activities, which are later defined by El-Rayes and Moselhi (1998) as atypical repetitive activities, can vary. (3) An activity may not be present at all work locations/units. (4) A given activity may be expected to proceed simultaneously at multiple locations/units using multiple crews. (5) Soft dependencies – dependencies other than technological constraints and construction methods (Tamimi and Diekmann 1988; Fan and Tserng 2006; Huang and Sun 2006) – among activities may not contribute to establishing absolute precedence relations at a given location/unit.

Henceforth, most of the studies in the repetitive scheduling domain concentrate on improving resource utilization (by means of reducing resource idle time, specifically, crew idle time) while aiming for achieving project goals such as the lowest construction cost or shortest construction duration. Resource idleness is referred to as unforced idleness or waste, which stems from resources (such as crew or equipment) waiting for preceding resources to complete their work (Vanhoucke 2006). Therefore, factors such as bad weather or equipment breakdowns are not considered causes for resource idleness. When calculating construction cost, crew idle time is considered in many studies as part of the direct cost (Moselhi and El-Rayes 1993; Eldin and Senouci 1994; El-Rayes 2001a; Hegazy and Wassef 2001; Kang et al. 2001; Ipsilandis 2007; Hyari and El-Rayes 2009). As for construction duration, crew idle time is used as one of the controlling factors in scheduling and is adjusted to reach a better/optimum construction duration (Harris and Ioannou 1998; El-Rayes and Moselhi 1998; El-Rayes 2001b; El-Rayes and Moselhi 2001; Nassar 2005; Hyari and El-Rayes 2006; Vanhoucke 2006; Ipsilandis 2007; Liu and Wang 2007, Hegazy and Kamarah 2008). These studies all use the resource (specifically, crew) idle time as the sole measurement of resource utilization for repetitive projects.

In the finishing phase of non-repetitive building projects, such as the project introduced in Section 2, the number of resource interruptions could be another critical factor with regard to resource utilization. Ashley (1980) and Birrell (1980) stress that the interruption (i.e., “come-back” delays, or off-on movement) of crews on a project should be minimized once work has begun. The impact of the number of resource interruptions can be disguised when resource idleness is used as the only indicator of resource utilization. For example, a crew needs to work on the same activity in eight different rooms and has two options: (1) to work continuously on the first four rooms, rest for 10 days, and then complete the remaining four rooms without interruption; or (2) to work sequentially on all the rooms but rest for one day each time a room is finished. In terms of crew idleness, we have 10 days for option one and 7 days for option two. The construction manager might nevertheless pick option one to avoid the high frequency of interruptions and the potential rearrangement of materials and equipment of option two even though it entails less idleness. Another example of interruption is related to crew mobilization and demobilization (Huang and Sun 2006). If a crew has a total idleness of 50 days on a project but could either be mobilized twice (with each interruption leading to a mobilization/demobilization) or once, the construction manager would probably choose the single-mobilization option to reduce the negative efforts involved in the mobilization/demobilization (Tommelein et al.1999).

Although the number of resource interruptions could be another important indicator of resource utilization, we are unaware of prior literature that has discussed the relationship between idle time and the number of interruptions, particularly in the domain of non-repetitive project scheduling. If minimizing resource idleness means the same as minimizing the number of interruptions, only one of these two measurements is necessary to represent resource utilization. Otherwise, both the idleness and the number of interruptions should be considered in evaluating resource utilization.

A third possible indicator of resource utilization is the direction of workflow. In the domain of repetitive project scheduling, three types of workflow direction have been classified. When scheduling projects such as the construction of pipelines and highways, the workflow direction is horizontal, because the construction process within a work zone should be horizontally joined to the next zone (Chrzanowski and Johnston 1986; Russell and Caselton 1988; Kang et al. 2001). In contrast, the workflow for the concrete work of high-rise building should be vertical because the construction process is repeated vertically, floor by floor (Thabet and Beliveau 1994a; Kang et al. 2001). A third type of repetitive project involves construction processes that repeat both horizontally and vertically, in which case both work directions must be taken into account (Riley 1994; Thabet and Beliveau 1994a; Harris and Ioannou 1998; Kang et al. 2001; Jongeling and Olofsson 2007). Workflow directions in the domain of non-repetitive project scheduling can be much more complicated (Akbas 2004). In the finishing phase of non-repetitive building projects, dual or multi-skilled workers may be employed to form different crews (with different sizes) within the same trade to work on different operations (Kang et al. 2001). Using our field research project (i.e., the university building construction project) as an example, two carpenters are necessary to install wood skirting in the faculty offices, but three are required to cut and install white boards in the lecture halls. In this context, it will be difficult to maintain a consistent work direction for all the construction workers. Therefore workflow direction is not discussed further in this paper.

3.2 Space as a type of resource

Compared to the utilization of crew as a type of resource, space utilization has gained less attention in construction scheduling. In his discussion of the work continuity constraint, Vanhoucke (2006) mentions that although it is generally linked with the minimization of crew idle time, this constraint may not be restricted to crews only. He gives an example of the work of De Boer (1998), and points out that the continuous utilization of the spatial resource (i.e., dry docks in this case) is crucial in scheduling a shipyard. In building projects, most researchers in the area of space scheduling employ the methods of identifying and resolving space conflicts by defining workspaces. They then apply these methods to existing schedules (Riley 1994; Riley and Sanvido 1997; Akinci et al. 2002; Guo 2002; Dawood and Mallasi 2006). On the other hand, Thabet and Beliveau (1994b) propose to consider workspace as a type of resource. They also (1997) create a space-constraint and resource-constraint (SCaRC) method to incorporate the availability of space in the schedule generation process. As discussed in the introduction, in the finishing phase of non-repetitive building projects, the room should be used as the basic unit for spatial resource consideration. To our knowledge, no prior literature has discussed the relationship between crew utilization and room utilization in the domain of scheduling for non-repetitive building projects. In the repetitive project scheduling domain, some researchers including Russell and Caselton (1988), Eldin and Senouci (1994), and El-Rayes and Moselhi (2001) use unit/room to determine the crew interruption vectors when scheduling under the work continuity constraint, but they do not discuss how the crew utilization (measured by crew idle time) is affected by the room/unit utilization (which can be measured by the room/unit idle time).

3.3 Resource utilization and project goals

Russell and Wong (1993) state that work continuity can be considered, but not necessarily strictly enforced, in order to achieve certain project goals. This statement is supported by case studies of proposed scheduling methods for repetitive projects. For example, when scheduling a highway construction project, El-Rayes and Moselhi (1998) found that allowing the interruption of certain activities can help reduce construction duration. In another highway project, El-Rayes (2001a) discovered that when crew idle time was eliminated, the construction duration increased significantly. On the basis of several case studies of repetitive projects, Vanhoucke (2006) claims that the schedule with the shortest construction duration corresponds to a very large value of crew idle time compared to the schedules with longer durations. Nassar (2005) attempted to find the shortest schedule while keeping the crew idle time as low as possible. Uncertain of the relationship between the construction duration and the crew idle time, Liu and Wang (2007) developed separate objectives for their scheduling algorithm in order to find schedules with the shortest construction duration or lowest crew idle time respectively.

As to the relationship between resource utilization and construction cost, Selinger (1980) notes that, in scheduling repetitive units, work interruptions result in an increased direct cost because of the idle crew time, the violation of the work continuity constraint by allowing work interruptions may result in an overall duration reduction, reducing the corresponding indirect costs. Similar results have been found by other researchers (Hegazy and Wassef 2001; Hegazy and Kamarah 2008). However, since construction direct and indirect costs are affected by many factors and the way of calculating these costs may vary from project to project, we focus on the relationship between resource utilization and construction duration instead of cost.

4 Research Objectives

Based on the literature review, we aim to address the following research questions about look-ahead scheduling in the finishing phase of non-repetitive building projects:

1. What measurement(s) should be used to evaluate resource utilization?
2. Is the utilization of the construction workforce resource (i.e., skilled workers and crews) independent from that of the room resource?
3. Would optimizing the utilization of one or both types of resources lead to a reduced construction duration?

To address these research questions, we need to develop our research method based on the following site observations and assumptions regarding look-ahead scheduling for non-repetitive building projects:

1. The term “operation” is used in this paper to represent a higher level of detail than “activity”. An operation can only be carried out in a given room (e.g., office rooms, conference rooms, corridors, lobbies, toilets, etc.). Site engineers need to use the operation level when creating LASs in order to specify work assignments (Dong et al. 2011).

2. The scheduling method for the non-repetitive building projects is unit-based because each type of room corresponds to a unique type of work sequence, which can be illustrated by a fragnet (i.e., an operation-on-node-network). When scheduling, we need to consider multiple types of fragnets instead of one unique fragnet repeatedly used in discrete steps throughout a project as in scheduling of repetitive projects.
3. At any time, only one operation is allowed to be executed in a room; thus the finish-start relation between operations is sufficient in defining fragnets.
4. The quantity of materials and duration of the same type of operation in the same type of room can vary from room to room.
5. Unlike Maravas and Pantouvakis (2011), who assume crews' productivity rates are uncertain and imprecise over units, we consider crew productivity is fixed – by the time site engineers and project planners need to create LASs, they are supposed to know how fast a type of crew can finish their work in a given room.
6. The availability of multi-skilled workers is considered when they are managed by the general contractor – they can form different crews to perform different types of operations. For example, a carpenter, once he finishes the wood skirting installation in a faculty office in a crew of two carpenters, can start to install white boards in a lecture hall in a crew of three carpenters, if the other two are available. The availability of crews is considered when the corresponding operations are carried out by subcontractors or single-skilled workers within a fixed crew formation. To avoid confusion, we henceforth

treat a subcontractor’s crew or a crew with fixed size as one multi-skilled worker.

7. Only the skilled-workers and fixed-size crews are considered the most important workforce resources, i.e., they are associated with an operation assuming other workforce resources such as unskilled labourers are readily available.
8. Two types of resources need to be considered in LAS generation: the workforce resource (a group of multi-skilled workers) and the room resource.
9. The formation of a crew for a particular operation in a given room is fixed (and known) in lieu of flexible formation (Senouci and Eldin 1996) before look-ahead scheduling.
10. The work direction of an individual multi-skilled worker is not considered in scheduling as the worker may re-enter a room many times (Yang and Ioannou 2004; Brodetskaiaa et al. 2011).
11. The availability of rooms at the beginning of the finishing phase may vary. That is, the structural work in different rooms may be completed at different times.
12. Operations cannot be interrupted. Once started, the resources assigned to the operation cannot be released until the operation is finished.
13. Space-buffer, i.e., lead-distance, and time-buffer, i.e., lead-time, (Yang and Ioannou 2004) are not considered.
14. Time for the routing of various multi-skilled workers are not considered in scheduling.

5 Research Methods

To address the proposed research questions, we first defined the metrics to measure resource utilization for each resource type used in the finishing phase. With these metrics defined, we used a sufficient number of valid LASs at the level of detail necessary to quantify each of the measurements accurately so that we could investigate the correlation between certain measurements and the correlation between certain measurement(s) and the project duration. To produce the desired set of LASs, we used an iterative method, which automates LASs generation for the finishing phase of non-repetitive building projects. Finally, we applied the method to a case study and carried out statistical analysis using the LASs generated to determine which measurement(s) can represent the utilization of a type of resource, the interdependency of the two types of resources, and the relationship between resource utilization and project duration.

5.1 Resource utilization measurements

We introduce *idleness* and *interruption* for both types of resources as measurements for resource utilization (Table 3-1). We use one equation (Eq.1) to formalize resource idleness and another (Eq.2) to formalize resource interruption to eliminate the redundancy of defining separate equations for the same type of measurement. The term *unit* in this section (and this section only) refers to a room or a multi-skilled worker.

Table 3-1. Two by two metrics of resource utilization measurements.

Resource Type	Idleness	Interruption
Crew Resource	The total number of days of all the workforce not working according to a LAS	The sum of all the number of off-on actions for all the workforce in a schedule divided by the number of workforce
Room Resource	The total number of days rooms unoccupied according to a LAS	The sum of all the number of occasions when a room is unoccupied between the first operation and the last operation in all rooms in a schedule divided by the total number of rooms

Assuming U is the set of all resource units, for each resource unit $i \in U$, x_i represents the total number of days when the resource unit is idle (not working or not occupied), the idleness of this resource is:

$$D = \sum_{i \in U} x_i \qquad (1)$$

Let y_i represent the total number of interruptions in a resource unit and N represent the total number of units in U, the interruption of this resource is:

$$T = \frac{\sum_{i \in U} y_i}{N} \qquad (2)$$

With the idleness and interruption defined, we now have three ways to represent the resource utilization for each type of resource: idleness only, interruption only and a combination of idleness and interruption.

5.2 Automated schedule generation method

To obtain a set of feasible LASs to carry out the statistical analysis, we have developed a simulation-based automated LAS generation (ALASG) method, building on the activity-based construction (ABC) modeling and simulation method, which is composed of an ABC modeling (ABC-Mod) component and an ABC simulation (ABC-Sim) component (Shi 1999).

ABC-Mod models construction processes using an activity-centered approach; ABC-Sim executes the ABC model using three steps in loops: schedule activities, advance simulation, and release resources. Our simulation-based method only produces feasible LASs and expands the ABC method by modeling construction processes in the finishing phase using a three-tiered hierarchy: project, room, and operation. The design, implementation, and validation of this simulation-based method are detailed in Dong (2012).

5.3 Statistical analysis on the data obtained from the LASs generated

Based on the multiple schedule options generated from a case study, we conducted statistical analysis to address the proposed research questions via a bottom-up approach:

- Step 1: to determine whether idleness alone can represent the resource utilization for a type of resource we evaluated the dependency between idleness and interruption for a type of resource.
- Step 2: with the measurements for resource utilization confirmed, we determined whether resource utilization of one type can represent the other. In other words, we evaluated the relationship between worker idleness and room idleness, and the relationship between worker interruption and room interruption.
- Step 3: we determined the relationship between certain resource utilization measurement(s) and project duration. Through the evaluation of these relations, we aimed to discover key drivers of construction duration reduction.

To conduct these analyses, linear models were created and compared using readily available model comparison methods, such as correlation of determination (R^2), F-tests and information criteria (Weisberg 2005). R^2 takes values between 0 and 1, and measures how well the linear model can explain the variation of model response. A R^2 value close to 1 indicates a high correlation between the model response and predictor variables, while a R^2 close to 0 means the model predictor variables can only poorly explain the response. This method is used throughout the above three steps to evaluate the relationship between one variable (e.g., room idleness) and one response (e.g., construction duration). Although simple and powerful, this method can only be used for evaluating correlations within a model, not among models. In this case, we used F-tests, which are widely used in simple linear model settings to compare nested models (Lomax 2007). If we suppose that model 2 includes all the predictor variables in model 1, i.e., model 1 is nested within model 2, then

$$\mathrm{F} = \frac{(\mathrm{RSS}_1 - \mathrm{RSS}_2)/(p_2 - p_1)}{\mathrm{RSS}_2/(n - p_2)} \tag{3}$$

where RSS_1 and RSS_2 are residual sums of squares for model 1 and model 2, p_1 and p_2 are numbers of predictor variables (including the intercept term) in model 1 and model 2, n is the sample size and F follows the F-distribution with degrees of freedom p_2-p_1 and n-p_2 under the null hypothesis that model 1 is the true model. In all the F-tests, we chose the significance level at 5%. If the p-value was less than 0.05, we concluded that model 2 is preferred over model 1. The F-tests were used in steps 1 and 3. For example, to investigate whether adding worker interruption better predicts construction duration compared to merely using worker idleness, we build model 1 as

$$V1 = \beta_0 + \beta_1 V2 + e \tag{4}$$

where $V1$ represents construction duration and $V2$ represents the predictor variable of worker idleness, coefficients β_i's can be zero and the error terms e's are assumed to be independent and follow some normal distribution. For all models in the analysis, estimates of model coefficients were obtained by ordinary least squares estimation (OLS). Similarly we can build model 2 as

$$V1 = \beta_0 + \beta_1 V3 + e \tag{5}$$

where *V3* represents worker interruption. Both model 1 and model 2 are nested in model 3 which is

$$V1 = \beta_0 + \beta_1 V2 + \beta_2 V3 + e \tag{6}$$

If the calculated p-values for the F-tests between model 1 and model 3 and between model 2 and model 3 are both less than 0.05, we can claim that the linear combination of worker idleness and worker interruption better predicts construction duration compared to using either one of them. In the remaining parts of the paper, we will use *V1 ~ V2+V3* as the equivalent model expression of Eq.6.

The F-tests described above do not apply if two models for comparison are not nested. In this case, information criteria is a popular method to compare models. It takes a balance between the complexity of a model and its lack of fit. For instance, we built model 4 as *V1 ~ V4* + *V5* where *V4* represents room idleness and *V5* represents room interruption. The F-tests cannot be used for comparing models that are not nested. For instance, model 3 and model 4 were not nested. In order to know which of the two linear models better estimates project duration, we employed information criteria to compare model 3 with model 4. Most commonly used information criteria include AIC (Sakamoto et al. 1987), BIC (Schwarz 1978) and C_p (Mallows 1973). In general, the smaller information criteria value indicates a better model. Since these different types of criteria give the same results in our analysis, we only show analysis results based on BIC. This statistic is defined by

$$\mathrm{BIC} = n\log(\mathrm{RSS}/n) + p\log(n) \qquad (7)$$

where RSS, n and p are a residual sum of square, sample size and number of variables, respectively. The BIC method was used only in step 3. It is worth noting that when the two models have the same number of predictor variables, the comparison using information criteria is equivalent to comparing R^2.

5.4 Case study - six-room example

We used a six-room case for which LASs were automatically created using the ALASG method (Dong et al. 2012) and we collected the data sets regarding resource utilization and construction duration. The case consists of three types of rooms–an electrical room, an IDF room and a plant room–each room requiring a unique sequence of tasks that calls for different types of multi-skilled workers and crews. The case represents a small portion of a complex university project in the finishing phase discussed previously. Despite its simplicity, it is representative of the scheduling problem for the finishing phase of non-repetitive building projects.

5.4.1 Fragnets and operations

We designed three fragnets to represent the work sequences in the three types of rooms. For example, the ELE fragnet contains an operation E-1: installing conduit and electrical boxes (Conduit & box), which precedes operation E-2: applying plastering (Plastering). Two ELE room instances – ELE1 and ELE2 – are linked to the ELE fragnet. Table 3-2 summarizes the fragnets, the room instances, and the related operations of this case study. Although certain fragnets contain the same operation names, the work contents and the related productivity rates are often different. For example, the “electrical final fix” in the ELE fragnet concentrates on finalizing the power boxes and panels, while in the IDF fragnet it concentrates on data servers above the raised floor. Table 3-2 also shows the successor(s) of each operation (using

"→"). Since most of the rooms are too small to allow multiple operations to proceed at the same time, we ddido not allow two or more operations be carried out in parallel in a room.

Table 3-2. Three types of fragnets involved in the case study.

	ELE Fragnet	**IDF Fragnet**	**PLT Fragnet**
Room instance	ELE1, ELE2	IDF1, IDF2	PLT1, PLT2
Operations	E-1. Conduit & box→ E-2 E-2. Plastering → E-3 E-3. Screed → E-4 E-4. Painting (first two coats) → E-5 & E-10 E-5. Electrical first fix → E-11 E-6. Painting (last coat) → E-7 & E-8 E-7. Epoxy floor → E-9 E-8. Electrical final fix → E-9 E-9. Doors & wood panels E-10. Electrical second fix → E-6 E-11. HVAC & Fire protection → E-6	I-1. Conduit & box → I-2 I-2. Plastering → I-3 I-3. Painting (first two coats) → I-4 & I-11 I-4. Electrical first fix → I-6 I-5. Painting (last coat) → I-7 I-6. Raised floor → I-9 I-7. Electrical final fix → I-8 I-8. Doors & wood panels I-9. Electrical second fix → I-5 I-10. HVAC & Fire protection → I-6	P-1. Conduit & box → P-2 P-2. Plastering→ P-3 P-3. Screed → P-4 P-4. Painting (first three coats) → P-5 & P-12 P-5. Electrical first fix → P-11 P-6. Louvers → P-7 P-7. Painting (last coat) → P-8 & P-9 P-8. Epoxy floor → P-10 P-9. Electrical final fix → P-10 P-10. Doors P-11. Electrical second fix → P-6 P-12. HVAC & Plumbing & Fire protection → P-6

Although most operations listed in Table 3-2 are carried out by crews formed by multi-skilled workers, certain operations are not. For instance, the plastering and screed operations have fixed crew formations with single-skilled workers in any fragnet. The operations related with electrical second fix, HVAC, fire protection, and plumbing are subcontracted. We treat each crew/subcontractor carrying out these operations as one multi-skilled worker in the LAS generation process. Table 3-3 describes each operation according to its type, crew formation, and availability. It also shows that the crew formation for an operation in one fragnet can be different from the operation with the same name in another fragnet because of the different work contents.

Table 3-3. Summary of crew formation for operations in the six-room case study and two scenarios of construction workforce availability.

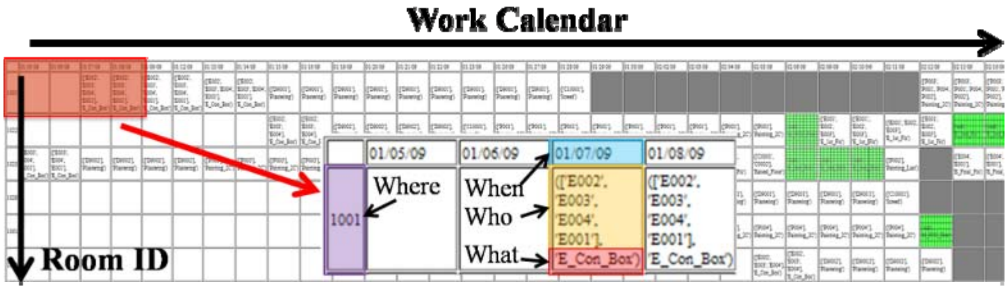
Operation Name	**Crew Formation**	**Available Resource – Scenario No.1**	**Available Resource – Scenario No.2**
Conduit & box (E-1 & I-1)	3 electricians	4 electricians – E001, E002, E003 and E004	6 electricians – E001, E002, E003, E004, E005 and E006
Conduit & box (P-1)	4 electricians		
Electrical first fix (E-5)	3 electricians		
Electrical first fix (I-4)	2 electricians		
Electrical first fix (P-5)	4 electricians		
Electrical final fix (E-8)	3 electricians		
Electrical final fix (I-7 & P-9)	2 electricians		
Louvers (P-6)	2 carpenters	2 carpenters – C001 and C002	3 carpenters – C001, C002 and C003
Doors & wood panels (E-9 & I-8)	2 carpenters		
Doors (P-10)	2 carpenters		
Raised floor (I-6)	2 carpenters		
Electrical second fix (E-10 & I-9 & P-11)	Subcontractor	1 crew – SubA1	2 crews – SubA1 and SubA2
HVAC & Fire protection (E-11 & I-10)	Subcontractor	1 crew – SubB1	2 crews– SubB1 and SubB2
HVAC & Plumbing & Fire protection (P-12)	Subcontractor		
Plastering (E-2+I-2+P-2)	Fixed formation 1 crew	2 crews – D9001and D9002	2 crews – D9001 and D9002
Screed (E-3+P-3)	Fixed formation 1 crew	1 crew – C1001	2 crews – C1001 and C1002
Painting (first two coats) (E-4+I-3)	1 painter	4 painters – P001, P002, P003 and P004	6 painters – P001, P002, P003, P004, P005 and P006
Painting (first three coats) (P-4)	4 painter		
Painting (last coat) (E-6+I-5)	1 painter		
Painting (last coat) (P-7)	4 painter		
Epoxy floor (E-7)	1 painter		
Epoxy floor (P-8)	2 painter		

5.4.2 Two scenarios of resource availability

In order to evaluate resource utilization under different resource availability conditions, we used two workforce resource availability scenarios (Table 3-3). In Scenario No.1, a very limited workforce situation is presented – for most skilled workers and crews there are only enough resources to finish one room at a time. For instance, among all the operations requiring electricians, two operations in the plant room need 4 electricians, which is the maximum requirement for electricians across all operations. Hence, we make only 4 electricians available throughout the entire project. In Scenario No.2, more crew members are available, increasing the chances of two or more rooms being occupied at the same time when they require the same type of skilled workers/crews. For both scenarios, all rooms are available for the finishing phase at the same time.

5.4.3 Sample LASs

The data sets for the statistical analysis regarding Scenarios No.1 and No.2 were obtained we run the schedule simulation one million times in the software prototype based on the ALASG method (Dong et al. 2012). Figure 3-1 presents two different views of a LAS created by the prototype after one simulation. Figure 3-1.a illustrates the room-centered view of a LAS with each row representing the work sequence in a room. Each cell specifies the multi-skilled workers or crews and the operation to take place in a room on a given day. For example, in room 1001 (where), on January 7, 2009 (when), four electricians (who) are required to work on “conduit & box” installation (what). This room-centered view clearly indicates the room idleness and interruptions. By rearranging the who-what-when-where elements, we get a worker-centered view of a schedule, as demonstrated in Figure 3-1.b, with each row representing the operation that a multi-skilled worker or crew will perform on a daily basis, and the location of that operation. A grey empty cell in either of these schedules indicates that a resource unit (room or skilled worker) is idle on a specific day. This crew-centered view clearly shows the crew idleness and interruptions.



a. Room-centered View of A Schedule

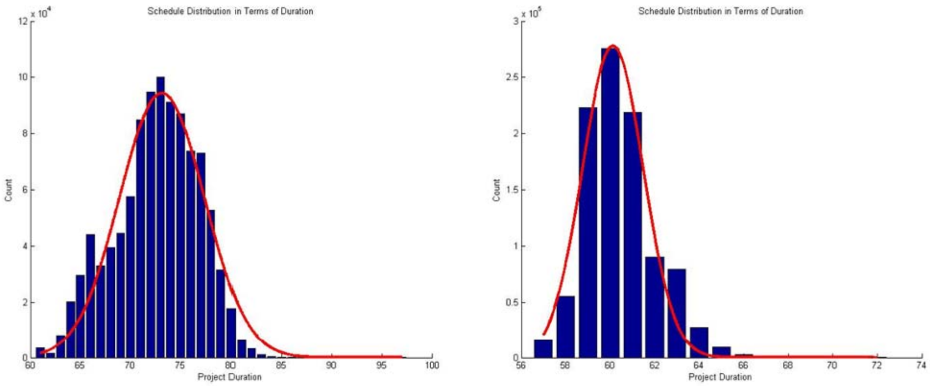
b. Worker-centered View of A Schedule

Figure 3-1. Two types of LASs created by the prototype (Dong et al. 2011).

6 Data Analysis

To evaluate resource utilization and its relationship with construction duration, we conducted data analysis based on data set No. 1 and data set No. 2, each data set corresponding to a scenario in Table 3-3. The five parameters we analyzed for each data set include **V1** - Project Duration, **V2** - Worker Idleness, **V3** - Worker Interruption, **V4** - Room Idleness, and **V5** - Room Interruption. Since each data set was obtained by running the schedule simulation one million times each data set contains one million records of these five parameters.

By randomly selecting available rooms and eligible operations for workforce assignments, the prototype generated a wide range of LASs for both scenarios in terms of construction duration. The distribution of these LASs in each data set is shown in Figure 3-2. Data set No. 1 has a wider duration distribution (61 to 95 days) compared to No. 2 (57-72 days). Since more workforce resource is available in data set No. 2, the shortest schedule found from data set No. 2 is 4 days shorter than the shortest schedule from data set No. 1; the longest schedule from data set No. 2 is 23 days shorter than the longest schedule from data set No. 1.

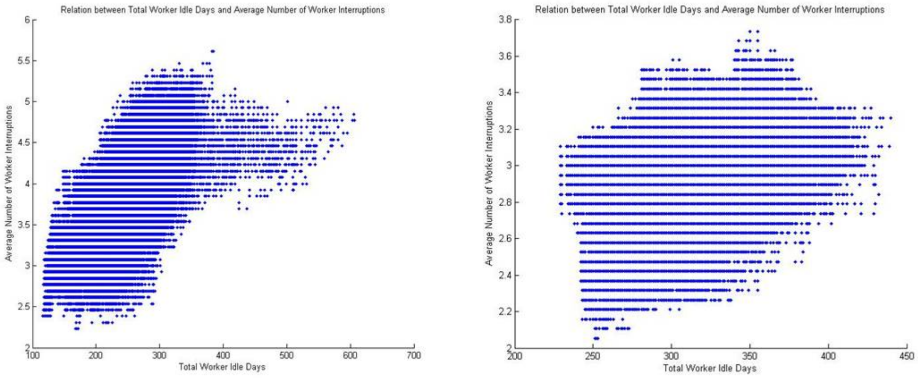


a. Schedule Distribution of Data Set No.1 **b. Schedule Distribution of Data Set No.2**

Figure 3-2. Schedule duration distribution of the two data sets.

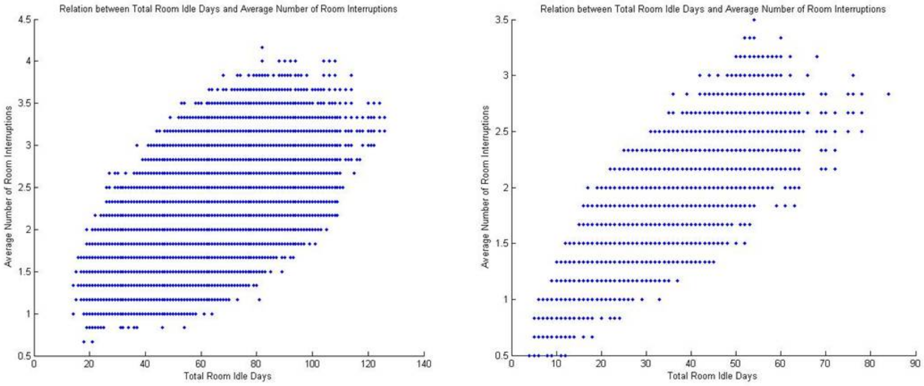
6.1 Relationship between idleness and interruption

To investigate whether idleness and interruption are relatively independent for the workforce resource and room resource, we used R^2 to evaluate the relationship between idleness and interruption within a resource for both resource scenarios (Figure 3-3). As indicated in Figure 3-3.d, room idleness (V4) and interruption (V5) show strong interdependence (R^2 equals 0.80) for the relatively abundant crew resource scenario (data set No. 2). Even so, it is impossible to use idleness to represent interruption entirely, or vice versa. For example, when room interruption reaches 2.0 in Figure 3-3.d, the room idleness spans from 16 days to 66 days. On the other hand, Figure 3-3.c indicates that room idleness and room interruption are relatively independent (R^2 equals 0.23). We can hardly establish a perfect linear relationship between worker idleness (V2) and interruption (V3) for both of the data sets, especially when with a relatively abundant workforce resource as indicated in Figure 3-3.b (R^2 equals 0.18). Therefore, in terms of resource utilization, it is worth bringing up interruption as an additional measurement.



a. Relation between worker idleness and interruption – data set No.1 $R^2 = 0.45$

b. Relation between worker idleness and interruption – data set No.2 $R^2 = 0.18$



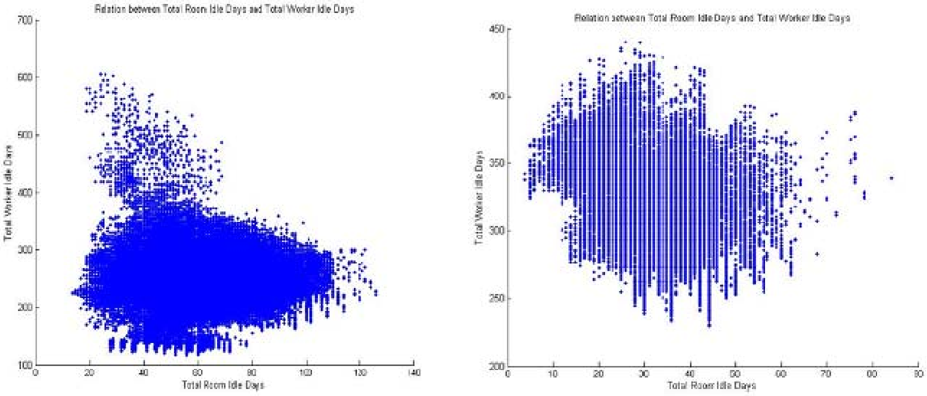
c. Relation between room idleness and interruption – data set No.1 $R^2 = 0.23$

d. Relation between room idleness and interruption – data set No.2 $R^2 = 0.80$

Figure 3-3. Idleness-interruption relations for the two scenarios.

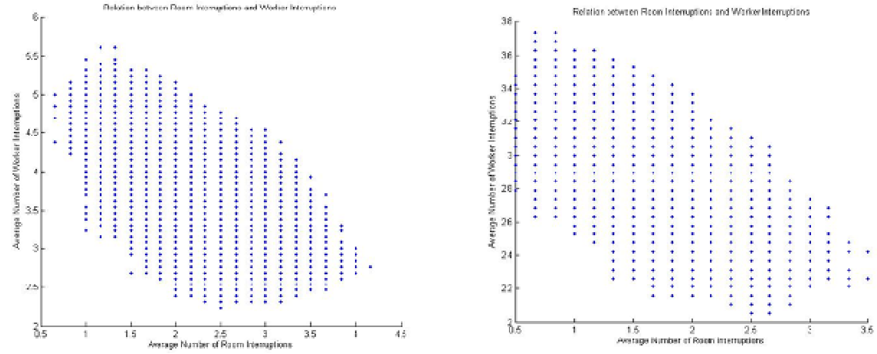
6.2 Relationship between workforce resource and room resource utilization

To evaluate the relationship between the utilizations of the two types of resource, we scrutinized their correlation with respect to both idleness and interruption using R^2 (Figure 3-4). In terms of idleness (V2 and V4), as indicated in Figure 3-4.a and Figure 3-4.b, the two resources are hardly related regardless of the availability of crews. As to interruption (V3 and V5), we observe a band-shaped relationship between the two resources for both data sets as shown in Figure 3-4.c and Figure 3-4.d. However, the R^2 values indicate that it is not possible to use the interruption of one resource to represent another. Therefore, these two resources cannot completely explain each other and should be considered separately when investigating resource utilization in the finishing phase of non-repetitive building projects.



a. Relation between worker idleness and room idleness   data set No.1 R^2 – 0.13

b. Relation between worker idleness and room idleness   data set No.2 R^2 – 0.48



c. Relation between worker interruption and room interruption – data set No.1 R^2 – 0.44

d. Relation between worker interruption and room interruption – data set No.2 R^2 – 0.32

Figure 3-4. Relationship between workforce resource utilization and room resource utilization.

6.3 Relationship between resource utilization and construction duration

To explain the relationship between resource utilization and project duration, we built linear models as shown in Table 3-4. Within group 1, either R^2 or BIC can be used to compare the four models and find which single resource utilization measurement (V2, V3, V4 or V5) best predicts construction duration (V1). Within group 2, the same method is used to compare the two models and to understand which resource can better predict V1. Between group 1 and group 2, since model 1-1 and model 1-2 are nested in model 2-1, we used F-tests to determine whether combining the worker idleness and interruption better predicts construction duration compared to using either idleness or interruption alone. We applied the same procedure to model 1-3, 1-4, and 2-2 to determine whether combining the room idleness and interruption better predicts construction duration compared to using either room idleness or interruption alone.

Table 3-4. Linear models for the analysis of the relationship between resource utilization and construction duration.

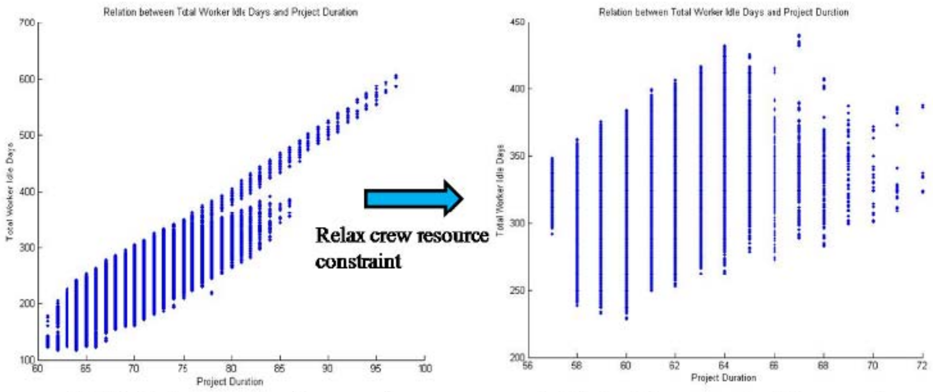
Group 1 – One Variable	**Group 2 – Two Variables**
Model 1-1: V1 ~ V2	Model 2-1: V1 ~ V2 + V3
Model 1-2: V1 ~ V3	Model 2-2: V1 ~ V4 + V5
Model 1-3: V1 ~ V4	
Model 1-4: V1 ~ V5	

6.3.1 Relationship between construction duration and one single resource utilization measurement

The BIC analysis results for both data sets No. 1 and No. 2 regarding the models described in Group 1 (Table 3-4) are listed in Table 3-5. From Table 3-5 we know that when there is a limited workforce, the worker idleness is the best choice for predicting construction duration. When more workforce resources are available, as in scenario No. 2, room idleness becomes most important. Figure 3-5 shows the plot of Model 1-1 and Model 1-3, comparing data set No. 1 with No. 2.

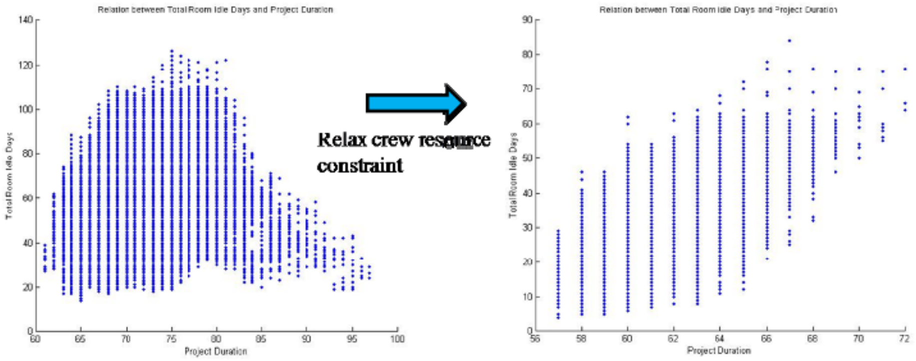
Table 3-5. BIC values of models of Group 1 in Table 3-4.

	Model 1-1	**Model 1-2**	**Model 1-3**	**Model 1-4**
Data set No. 1	1880235	2265628	2793029	2725680
Data set No. 2	940130	940498	720269	777412



a. Relation between worker idleness and project duration – data set No.1 $R^2 = 0.62$

b. Relation between worker idleness and project duration – data set No.2 $R^2 = 0.0007$



c. Relation between room idleness and project duration data set No.1 $R^2 = 0.07$

d. Relation between room idleness and project duration data set No.2 $R^2 = 0.20$

Figure 3-5. Relationship between construction duration and a particular resource utilization measurement.

Based on the data analysis shown in Table 3-5 and Figure 3-5, we can claim the following:

- When the workforce resource is extremely tight, trying to make the workers as busy as possible can lead to relatively short construction durations. However, since the relationship between worker idleness and construction duration is not perfectly linear (R2 = 0.62), we cannot claim that the minimum worker idleness leads to the shortest construction duration. From Figure 3-5.a we discover that when minimum worker idleness (117 days) is reached, we obtain schedules with durations of 62 days or 64 days, not 61 days (the shortest duration found from our schedule generation simulations). On the other hand, Figure 3-5.c indicates that room idleness barely has any relationship with construction duration.

- When more workforce resources become available, the fixed room resource becomes relatively more precious compared to the workforce resource. In this case, trying to make the rooms as occupied as possible can lead to a short construction duration. Figure 3-5.d shows that when room idleness reaches its minimum (4 days), the corresponding construction duration becomes shortest (57 days). On the other hand, in this scenario, minimizing crew idleness does not necessarily lead to schedules with the shortest construction duration, as indicated in Figure 3-5.b.
- We cannot guarantee that while the workforce resource keeps increasing, the room idleness becomes more important in predicting construction duration. An extreme case is that when there are enough crews available to satisfy any operations competing for the same type of multi-skilled workers or crews, the ALASG method generates only one schedule with all the rooms proceeding at the same time.

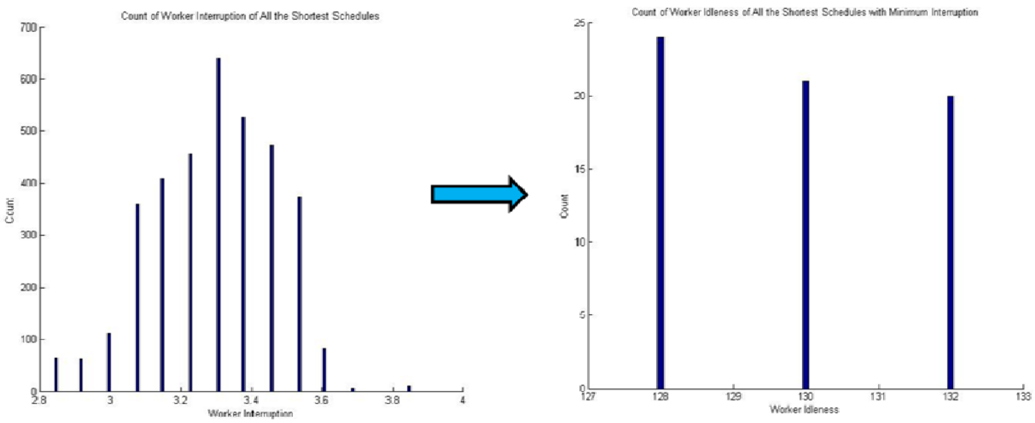
6.3.2 Expanding the measurement of resource utilization in the context of project duration

In the previous discussion regarding the relationship between the resource idleness and interruption, we notice that resource interruption does not bear a strong correlation with resource idleness. Thus, it should be used as an additional measurement for resource utilization. Here we discuss how well the linear models predict construction duration using this expanded resource utilization measurement based on F-tests results. Table 3-6 illustrates the p-values when comparing the models from the two groups described in Table 3-4. Table 3-6 shows that all the p-values are much less than 0.05, indicating that the expanded resource utilization measurement is more relevant in predicting construction duration than using either the resource idleness or interruption alone for the same type of resource.

Table 3-6. P-values representing the comparison of models between group 1 and group 2 in Table 3-4.

	Model Comparison	**p-value**
Data set No.1	Model 1-1 vs. Model 2-1	$< 10^{-15}$
	Model 1-2 vs. Model 2-1	$< 10^{-15}$
	Model 1-3 vs. Model 2-2	$< 10^{-15}$
	Model 1-4 vs. Model 2-2	$< 10^{-15}$
Data set No.2	Model 1-1 vs. Model 2-1	$< 10^{-15}$
	Model 1-2 vs. Model 2-1	$< 10^{-15}$
	Model 1-3 vs. Model 2-2	$< 10^{-15}$
	Model 1-4 vs. Model 2-2	$< 10^{-15}$

It is worth mentioning that the expanded resource utilization measurement can help single out the best schedule out of the vast number of LAS options. We use Figure 3-6, which is depicted based on the one million LASs alternatives generated for data set No.1, to illustrate how this measurement can help. First, we found 3,582 schedules with the duration of 61 days from the one million schedule alternatives (some of the alternatives are duplicates or identical to each other). Out of these 3,582 schedules we selected 65 schedules with a minimum worker interruption of 2.864 as indicated in Figure 3-6.a. Among these 65 schedules, the site engineers and construction managers might look for schedules with maximum worker idleness because more idleness can allow workers more rest and provide more flexibility to a LAS. Figure 3-6.b shows that there are 20 schedules with the greatest worker idleness – 132 days. After checking these 20 schedules, we confirmed that they are all the same. So we know that when we use the criteria of "shortest duration→minimum worker interruption→maximum worker idleness," only one LAS alternative is found and the repetition rate of this best schedule is 20 out of one million or 0.002%.



a. Worker interruption distribution of the 3582 schedules with duration of 61 days

b. Worker idleness distribution of the 65 schedules with duration of 61 days and interruption of 2.864

Figure 3-6. Process in finding the optimum schedule from one million iterations (data set No.1).

6.3.3 Importance of resource as a whole under the context of project duration

Model 2-1 and Model 2-2 (Table 3-4) represent the relationship between one type of resource utilization and construction duration. We use BIC to compare these models to determine which type correlates more closely to construction duration (Table 3-7). When comparing Model 2-1 with Model 2-2, as mentioned earlier in Section 5.3, a smaller BIC value indicates that the predictor variables in a model have stronger correlation with project duration. The BIC values in Table 3-7 show that when workforce resource is tight, its utilization is more relevant to construction duration than the room resource (1780975 vs. 2412026). However, when the workforce resource becomes relatively abundant, room resource utilization drives project duration (719659 vs. 940071).

Table 3-7. BIC values of models of group 2 in Table 3-4.

	Model 2-1 (Workforce Resource)	**Model 2-2 (Room Resource)**
Data set No.1	1780975	2412026
Data set No.2	940071	719659

7 Conclusion

The study described in this paper contributes to the field of resource utilization in LAS generation for the finishing phase of non-repetitive building projects by proposing key measurements to measure resource utilization, evaluating the relations of these measurements, and investigating the relationship between resource utilization and construction duration. Through statistical analysis of the many LAS options created for a case study with two resource availability scenarios, we found that, along with resource idleness, interruption should be considered as an additional measurement of resource utilization, thus extending the definition of the term. While most literature uses resource idleness to represent resource utilization, we analyzed the relationship between the workforce resource and room resource utilization using the proposed resource utilization measurements and found that these two resources can neither completely explain each other nor be considered completely independent. In other words, when a LAS alternative has the best workforce utilization, it does not necessarily have the best room utilization. Therefore merely considering workforce utilization in the finishing phase of non-repetitive projects is not sufficient to evaluate resource utilization.

Although only one case is used to make the aforementioned statements, this research bears generality because one single case is sufficient when it can negate the following statements when evaluating LASs in the finishing phase of non-repetitive building projects: (1) only idleness should be used to measure resource utilization and (2) the utilization pattern of workforce is the same as for rooms.

The case study also indicates that shortening construction duration by using resources more intensely, the significance of each resource varies according to its relative abundance compared to the other resource. For each type of resource, using the extended measurements of resource utilization is better than merely considering either the resource idleness or interruption. However, to simplify the derivation process, only one resource type could be considered according to the relative abundance of the two resources. These claims need to be further tested using more cases to demonstrate generality.

Resource utilization can be used to assist construction managers and site engineers in selecting the optimum LAS solution out of the large number of feasible alternatives. It is the site staff's decision whether to prioritize certain resource utilization measurements when many schedules possess the same shortest duration. For instance, site engineers could ignore how the room resource is utilized, focusing only on workforce interruption and idleness. In this way, they could find the best schedule (with the shortest duration), featuring minimum workforce interruption and maximum workforce idleness because more idleness allows higher flexibility when construction duration is the same.

For large-scale non-repetitive building projects in the finishing phase, running the ALASG-based prototype (Chapter 2) and selecting the best schedule is not the ideal way to find the close-to-optimum schedule(s) – the sheer number of feasible LAS options to analyze is formidable. The findings of this paper help uncover the relationship between resource utilization and the construction duration in the finishing phase. Such findings are useful in the search of close-to-optimum schedule alternatives using Artificial Intelligence for non-repetitive projects (Dong et al. 2012). We can appropriately design the objective function to consider both construction duration and resource utilization by assigning weights to these parameters according to resource availability. We can also use the measurements of resource utilization expanded in this paper to filter out the best LAS among multiple options.

8 References

Akbas, R. (2004). "Geometry-based modeling and simulation of construction processes," Ph.D. Dissertation, Stanford University, Stanford, CA.

Akinci,B., Fischer, M., Kunz, J., and Levitt R. (2002). "Representing work spaces generically in construction method models." *Journal of Construction Engineering and Management,* 128 (4), 296-305.

Akinci, B., Fischer, M., Levitt, R., and Carlson, R. (2002). "Formalization and automation of time-space conflict analysis," *Journal of Computing in Civil Engineering,* 16(2), 124-034.

Al-Bazi, A., and Dawood, N. (2010). "Developing Crew Allocation System for the Precast Industry Using Genetic Algorithms." *Computer-Aided Civil and Infrastructure Engineering,* 25 (8), 581-595.

Ashley, D. B. (1980). "Simulation of repetitive-unit construction." *Journal of the Construction Division,* 106(2), 185–194.

Birrell, G. E. (1980). "Construction planning - beyond the critical path." *Journal of the Construction Division,* 106(3), 389–407.

Brodetskaiaa, I., Sacksa, R., and Shapiraa, A. (2011). "A workflow model for systems and interior finishing works in building construction." *Construction Management and Economics,* 29(12), 1209-1227.

Cherneff, J., Logcher, R., and Sriram, D. (1991). "Integrating CAD with construction-schedule generation." *Journal of Computing in Civil Engineering,* 5(1), 64-84.

Chapter 3

Chrzanowski, E. N., and Johnston, D. W. (1986). "Application of linear scheduling." *Journal of Construction Engineering and Management*, 112 (4), 476-491.

Dawood, N., and Mallasi, Z. (2006). "Construction workspace planning: assignment and analysis utilizing 4D visualization technologies." *Computer-Aided Civil and Infrastructure Engineering,* 21 (7), 498–513.

Dong, N., Fischer, M., and Haddad, R. (2011). "Automatic Look-ahead Schedule Generation System for the Finishing Phase of Complex Projects for General Contractors." *Proceedings of the 2011 ASCE International Workshop on Computing in Civil Engineering,* Miami, Florida, 134-142.

Dong, N., Ge, D., Fischer, M., and Haddad, Z. (2012), "A genetic algorithm-based method for look-ahead scheduling in the finishing phase of construction projects," Advanced Engineering and Informatics, accepted.

De Boer, R. (1998). "Resource-constrained multi-project management-A hierarchical decision support system." Ph.D. Dissertation, Institute for Business Engineering and Technology Application, Enschede, The Netherlands.

Eldin, N. N., and Senouci, A. B. (1994). "Scheduling and Control of Linear Projects." *Canadian Journal of Civil Engineering,* 21(2), 219-230.

El-Rayes, K. (2001a). "Optimum planning of highway construction under the A+B bidding method." *Journal of Construction Engineering and Management,* 127(3), 199-205.

El-Rayes, K. (2001b). "Object-oriented model for repetitive construction scheduling." *Journal of Construction Engineering and Management,* 127(4), 261-269.

Chapter 3

El-Rayes, K., and Moselhi, O. (1998). "Resource-Driven Scheduling of Repetitive Projects." *Journal of Construction Management and Economics,* 16(4), 433-446.

El-Rayes, K., and Moselhi, O. (2001). "Optimizing resource utilization for repetitive construction projects." *Journal of Construction Engineering and Management,* 127(1), 18-27.

Fan, S., and Tserng, H. (2006), "Object-Oriented Scheduling for Repetitive Projects with Soft Logics." *Journal of Construction Engineering and Management,* 132(1), 35-48.

Guo, S. J. (2002). "Identification and resolution of work space conflicts in building construction." *Journal of Construction Engineering and Management,* 128(4), 287-295.

Harris, R., and Ioannou, P. (1998). "Scheduling Projects with Repeating Activities." *Journal of Construction Engineering and Management,* 124(4), 269-273.

Hegazy, T., and Wassef N. (2001). "Cost Optimization in Projects with Repetitive Nonserial Activities." *Journal of Construction Engineering and Management,* 127(3), 183-191.

Hagazy, T., Elhakeem, A., and Elbeltagi, E. (2004). "Distributed scheduling model for infrastructure networks." *Journal of Construction Engineering and Management,* 130(2), 160-167.

Hegazy, T., and Kamarah, E. (2008). "Efficient Repetitive Scheduling for High-Rise Construction." *Journal of Construction Engineering and Management,* 134(4), 253-264.

Huang, R., and Sun, K. (2006). "Non-unit-based planning and scheduling of repetitive construction projects." *Journal of Construction Engineering and Management,* 132(6), 585-597.

Hyari, K. H., and El-Rayes, K. (2006). "Optimal planning and scheduling for repetitive construction projects." *Journal of Management in Engineering,* 22(1), 11-19.

Hyari, K. H., and El-Rayes, K. (2009). "Automated trade-off between time and cost in planning repetitive construction projects." *Construction Management and Economics,* 27(8), 749-761.

Ipsilandis, P. (2007). "Multiobjctive linear programming model for scheduling linear repetitive projects." *Journal of Construction Engineering and Management,* 133(6), 417-424.

Jongeling, R., and Olofsson, T. (2007). "A method for planning of work-flow by combined use of location-based scheduling and 4D CAD." *Automation in Construction,* 16(2), 189-198.

Josephson, P. E. (2005). "Byggfel och andra icke värdeadderande aktiviteterk." Byggkostnadsforum seminar: från no-tech till hi-tech, Stockholm, Sweden.

Kang, L. S., Park, I. C., and Lee, B. H. (2001). "Optimal Schedule Planning for Multiple, Repetitive Construction Process." *Journal of Construction Engineering and Management,* 127(5), 382-390.

Liu, S., and Wang, C. (2007). "Optimization model for resource assignment problems of linear construction projects." *Automation in Construction,* 16(4), 460-473.

Chapter 3

Lomax, R. G. (2007). Statistical Concepts: A Second Course for education and the behavioral sciences, Longman, NY.

Lu, W., and Olofsson, T. (2009). "A continuous flow simulation model for probability repetitive projects." *Proceedings of 5th Nordic Conference on Construction Economics and Organization,* Reykjavík, Iceland, 242-254.

Mallows C. (1973). "Some comments on Cp." *Technometrics,* 15, 661-676.

Maravas, A., and Pantouvakis, J. (2011). "Fuzzy Repetitive Scheduling Method for Projects with Repeating Activities." *Journal of Construction Engineering and Management,* 137(7), 561-564.

Moselhi, O., and El-Rayes, K. (1993). "Scheduling of repetitive projects with cost optimization." *Journal of Construction Engineering and Management,* 119(4), 681-697.

Moselhi, O., and Hassanein, A. (2003). "Optimized scheduling of linear projects." *Journal of Construction Engineering and Management,* 129(6), 664-673.

Nassar, K. (2005). "Evolutionary optimization of resource allocation in repetitive construction schedules." *Journal of Information Technology in Construction,* 10, 265-273.

Reda, R. (1990). "RPM: Repetitive Project Modeling." *Journal of Construction Engineering and Management,* 116(2), 316-330.

Riley, D. (1994). "Modelling the space behaviour of construction activities." Ph.D. Dissertation, Pennsylvania State University, University Park, PA.

Riley, D., and Sanvido, V. (1997). "Space planning method for multi-story building construction." *Journal of Construction Engineering and Management,* 123 (2), 171-180.

Russell, A., and Caselton, W. (1988). "Extensions to Linear Scheduling Optimization." *Journal of Construction Engineering and Management,* 114(1), 36-52.

Russell, A., and Wong, W. (1993). "New Generation of Planning Structures." *Journal of Construction Engineering and Management,* 119(2), 196-214.

Sakamoto Y., Ishiguro M., and Kitagawa G. (1987). Akaike information criterion statistics, D. Reidel Publishing Co., Dordrecht.

Schwarz G. (1978). "Estimating the dimension of a model." *Annals of Statistics,* 6, 461-464.

Selinger, S. (1980). "Construction Planning for Linear Projects." *Journal of the Construction Division,* 106(2), 195-205.

Senouci, A. B., and Eldin, N. N. (1996). "Dynamic programming approach to scheduling of nonserial linear project." *Journal of Computing in Civil Engineering,* 10(2), 106-114.

Shi, J.J. (1999). "Activity-based construction (ABC) modeling and simulation method." *Journal of Construction Engineering and Management,* 125 (5), 354-360.

Shtub, A., LeBlanc, L. J., and Cai, Z. (1996). "Scheduling programs with repetitive projects: A comparison of a simulated annealing, a genetic and a pair-wise swap algorithm." *European Journal of Operations Research,* 88, 124-138.

Tamimi, S., and Diekmann, J. (1988). "Soft logic in network analysis." *Journal of Computing in Civil Engineering,* 2(3), 289-300.

Thabet, W. Y., and Beliveau, Y. J. (1994a). "HVLS: horizontal and vertical logic scheduling for multistorey projects." *Journal of Construction Engineering and Management,* 120(4), 875-892.

Thabet, W. Y., and Beliveau, Y. J. (1994b). "Modeling work space to schedule repetitive floors in multistory buildings." Journal of Construction Engineering and Management, 120(1), 96-116.

Thabet, W. Y., and Beliveau, Y. J. (1997). "SCaRC: space-constrained resource-constrained scheduling system." *Journal of Computing in Civil Engineering,* 11(1), 48–59.

Tommelein, I. D., Riley, D. R., and Howell, G. A. (1999). "Parade game: impact of work flow variability on trade performance." *Journal of Construction Engineering and Management,* 125(5), 304-310.

Vanhoucke, M. (2006). "Work continuity constraints in project scheduling." *Journal of Construction Engineering and Management,* 132(1), 14-25.

Weisberg S. (2005). Applied linear regression, 6th Edition, Jon Wiley & Sons, Inc., NJ.

Yang, T., and Ioannou, P. (2004). "Scheduling System with Focus on Practical Concerns in Repetitive Projects." *Journal of Construction Management and Economics,* 22, 619-630.

Chapter 4: A Genetic Algorithm-based Method for Look-ahead Scheduling in the Finishing Phase of Construction Projects

Ning Dong, Dongdong Ge, Martin Fischer, and Zuhair Haddad

1 Abstract

Genetic Algorithms (GAs) are widely used in finding solutions for resource constrained multi-project scheduling problems (RCMPSP) in construction projects. In the finishing phase of a complex construction project, each room forms a confined space for crews to conduct a series of activities and can thus be considered as an individual sub-project. Generating the look-ahead schedule (LAS) which takes into account the limited resources available at the job site falls in the domain of RCMPSP. Therefore GAs can be used to address this scheduling problem and help construction managers to guide the daily work on site. However, current GAs do not consider three key practical aspects that the project planers and construction managers deal with frequently at the job sites: the engineering priorities of each individual sub-project, the zone constraint and the blocking constraint. By addressing these aspects, this paper proposes a GA-based method that takes them into account in the search process for optimum project duration and/or cost. Two examples are used for the discussion of the effectiveness of this method and to showcase its capability in project scheduling when the scale of a project increases.

2 Introduction

As previously mentioned in Chapter 1, LAS is a useful tool for general contractors in planning their work at the level of detail such that they can actually consider the availability of crews and other resources on a daily basis. At this level, in the finishing phase of a construction project, the project planner needs to make it clear to all members of the project which crew (*who*) is working on what operation in which room (*where*) on which day (*when*), ensuring that everyone is on the same page. At a university project, to generate such a LAS, the site engineers needed to consider, on a daily basis, data sets including more than 15 types of rooms, more than 20 operations per room on average, their progress, more than 50 types of materials and their quantities in each room, more than 10 types of crews and their productivities, and the engineering constraints on the job site. Generating a reliable LAS manually for a project this complex became such a challenge that the site engineers and project planners eventually gave up. Therefore, they need an automated way for the LAS generation. In this paper, the term “operation” is used to represent a higher level of detail than “activity”. Most operations can be carried out in a given room (e.g., office rooms, conference rooms, corridors, lobbies, toilets, etc.), which is basically a functional space. Therefore, space is a key factor we need to consider when generating LASs. Prior research relevant to space resource management can be categorized, as pointed out by Guo (2002), into space-scheduling, site layout planning, and path planning. This research focuses on space-scheduling. In this area, prior researchers have mostly concentrated on identifying and resolving space conflicts by defining work spaces and then applying them to existing schedules (Riley 1994; Riley and Sanvido 1997; Akinci et al. 2002a, Akinci et al. 2002b, Dawood and Mallasi 2006). On the other hand, Thabet and Beliveau (1997) propose a space-constraint and resource-constraint method to incorporate the availability of space into scheduling. Such a method defines workspaces at the floor level, not the individual room level. In the finishing phase of complex projects, room is the basic production unit with specific work content. It provides a natural and convenient representation for spatial reasoning (Cherneff and Logcher 1991), and accounts for the spaces where the crews and equipment must be allocated regardless of their shapes, unlike the work spaces defined and measured by length, width, and height (Akinci et al. 2002a; Thabet and Beliveau 1997). Therefore, in this paper we consider room as the basic unit of workspace and take into account its availability in scheduling.

LAS generation considering limited resources can be categorized as a type of resource-constrained project scheduling problem (RCPSP) and can be addressed by genetic algorithms. For instance, Feng et al. (2010) use a genetic algorithm to find the best LAS for the structural phase of a construction project; Organización (2011) combines simulation with genetic algorithm to generate LAS for berth allocation for containerships. However, as mentioned by Payne (1995), up to 90% of all projects occur in a multi-project context. Under this context, a project is composed of multiple sub-projects and each sub-project consists of an operation network (with embedded operation precedent constraints) that draws from shared pools of multiple types of resources which are normally not large enough for all the sub-projects, and thereby operations, to work concurrently. LAS generation for the finishing phase of complex construction projects conforms to this context. In the finishing phase, each room is relatively independent so that most operations in one room do not interfere with those in another. In addition, each type of room entails a specific work sequence for crews to work on. If all rooms are entirely independent, each room can be considered as a sub-project, thus making the LAS generation problem become a resource-constrained multi-project scheduling problem (RCMPSP).

Such a problem has been first studied in the literature via the single-project approach, which adds dummy activities, i.e., “start” and “end”, and arcs to merge the sub-projects into a single mega-project. It reduces the RCMPSP to a RCPSP with a single critical path. *Activity priority rule* heuristics are the commonly used methods in this approach (Scheiberg and Stretton 1994; Wiley et al. 1998; Hendriks et al. 1999; Fricke and Shenhar 2000; Lova et al. 2000; Lova and Tormos 2001). As the complexity of a RCMPSP increases, researchers look into the multi-project approach, in which sub-projects are independent except that they draw from a common pool of resources of different types. Kurtulus and Davis (1982) point out that certain priority rules deliver better results when using such a approach for scheduling. The use of a multi-project approach is further justified by Browning and Yassine (2010). They deem that this approach is more realistic and that it presents a greater opportunity for improvement. More importantly, they point out that compared to the single-project approach, it has greater potential to provide alternative decision guidance for different management roles, e.g., individual project manager, general project manager, and portfolio manager.

Unfortunately, these approaches cannot be used directly to address the real-world LAS generation problem for the finishing phase of complex construction projects. One precondition of RCMPSP is that resources cannot be shared by multiple operations at the same time. But on a construction project, certain operations in multiple rooms need to be synchronized with the same resource(s) assigned to all such operations at the same time. This is referred to as zone constraint; the rooms involved are grouped as a zone. An example of a zone is a group of adjacent rooms sharing the HVAC system. The “HVAC duct installation” operation requires that a zone is formed so that all these rooms are reserved for the HVAC crews/subcontractors to work in before being handed over to other disciplines. While the zone constraint renders rooms less independent, another constraint – blocking constraint – adds further complexity to the spatial interactions. A typical blocking operation is “Terrazzo flooring” in the corridor. When such an operation starts in a corridor, the access to all its adjacent rooms is suspended and no other crews are allowed to step on the floor until the operation is finished. If interruption of an operation is not allowed, the blocking operation has to be squeezed into a schedule when no other operations are in progress in the adjacent rooms. In sum, a *zone constraint* requires that certain rooms/sub-projects be synchronized at a certain point, its corresponding zone operations sharing the same resource(s); while the *blocking constraint* requires that no operation in certain rooms/sub-projects be started until the blocking operation in a given room/sub-project is finished. Both constraints break the paradigm of looking at a group of rooms/sub-projects independently. The authors are not aware of prior research that has considered such practical constraints when using genetic algorithm to address RCMPSP.

Engineering priority is another factor existing literature has not yet taken into consideration when generating LAS. A LAS is normally derived from the project master schedule which requires certain project subareas to be completed first. Therefore, from time to time, engineering priorities are given to certain rooms/sub-projects, not individual operations. Without considering this factor, merely generating the best LAS using *priority rule* based genetic algorithms becomes less attractive to practitioners.

In this paper, a genetic algorithm based method is described to automate the LAS generation in the finishing phase of complex projects with the objective of minimizing project duration or project cost. Such a method takes into account the engineering constraints, i.e., blocking constraint and zone constraint, and engineering priorities in the schedule generation process so that all the schedules created satisfy practical needs. In the sections that follow, we consider the concepts of a room and a sub-project as interchangeable. An operation is related to only one sub-project in terms of materials to be installed/applied, although its execution (such as the execution of a blocking operation) might affect work in other sub-projects.

3 Problem Description and Mathematical Formulation

3.1 Problem description

A complex construction project consists of *N* types of rooms, each of which requires a number of operations with unique work sequence for the finishing work. A fragnet is used to model the finishing work of a type of room using activity-on-node network notation, where a set of *M* nodes represents *M* operations with arcs between the nodes to represent their precedence relations. In the research project this paper describes, only the “finish-to-start” relation is used when developing the fragnets. The remainder of this paper refers to a room as a sub-project. Figure 4-1 shows an example which is a miniature of a construction project in the finishing phase. The example consists of four sub-projects, each of which belongs to a particular type with a dummy start and a dummy finish operation. The duration of each operation is placed on top of each operation and the resource required is placed below. The constraint-related operations are also shown in Figure 4-1. Sub-project 4 contains a blocking operation and when it is started, all other sub-projects are blocked. Sub-projects 1, 2, and 3 each contain one zone operation (depicted as a grey node in Figure 4-1), which requires two H crews. These operations must start at the same time, sharing the same crews and lasting for three days.

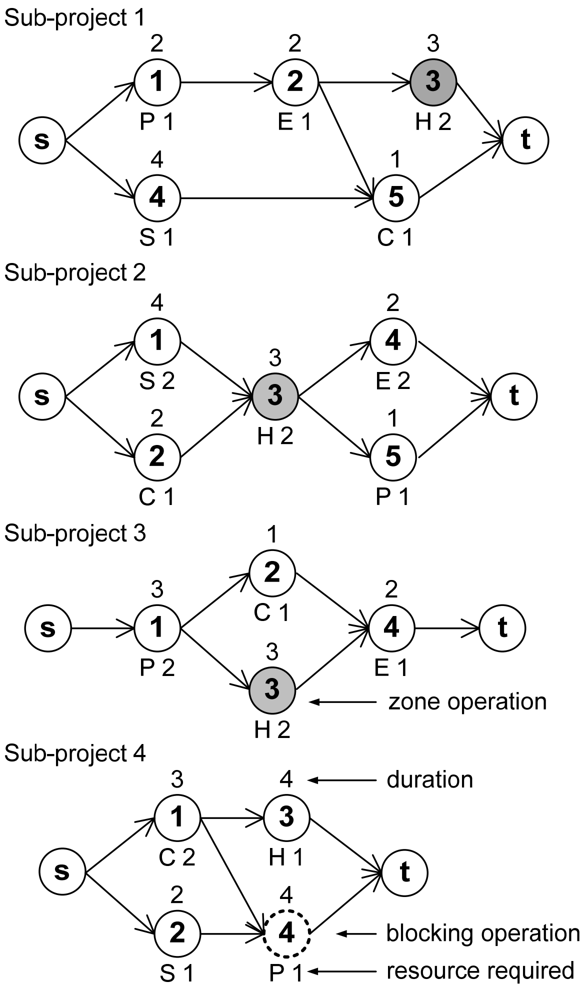


Figure 4-1. The activity-on-node fragnets of the four sub-projects constituting a sample construction project.

3.1.1 Precedence constraints

Unlike the multi-project genetic algorithm proposed by Kim et al. (2005), sub-projects in this paper do not have precedence constraints. In other words, all sub-projects can start at the same time as long as they are all ready (i.e., the structural related operations are all completed in a sub-project) at the same time with enough resources to get started. As to the precedence constraints of operations, the start time of each operation is dependent upon the completion of some other operations.

3.1.2 Resource consideration

To carry out an operation, multiple types of resources (e.g., crews, and equipment) should be employed together. However, only the most significant resource, i.e., crew, is associated with an operation assuming other types of resources are readily available. Different types of crews are available in limited quantities but renewable (i.e., can be reassigned) from period to period.

3.1.3 Crew productivity rate

By the time the site engineers start to create the LAS for the finishing phase of a construction project, the construction method for each operation and the productivity rate of the corresponding crews should already have been established. Therefore, the productivity rate for each crew for each type of operation is assumed to be fixed. Accordingly, the duration of each operation will not change over time.

3.1.4 Project cost

The project cost consists of two parts – direct cost and indirect cost. The indirect costs refer to costs related with site supervision, field office costs (e.g., IT facilities, site QA program, payroll administration, etc.), site cleanup and general housekeeping, etc. Here we calculate this cost by multiplying the daily indirect cost by the project duration. The direct costs are costs that are directly associated with the project, including subcontractors, hired labour, materials, supplies, equipment, bonds and permits. In this paper the direct cost is calculated in two parts. The first part is calculated by multiplying a crew’s daily expense, including their salary, by the number of days they are working on the job site. The second part is the mobilization and demobilization costs, which are calculated by multiplying a crew’s daily expense by the number of days the construction manager deems as the upper limit they can stay idle while getting paid. The cost calculation method is further explained in Chapter 2.

3.1.5 Single operation execution mode

Activities in the RCPSP/RCMPSP setting can have single execution mode or multi-mode (Kim et al. 2005). A mode is a processing alternative of an activity. When generating LAS for operations on a daily basis, only one mode is allowed. Operations are not allowed to be interrupted.

3.1.6 Zone constraint

Sub-projects form a zone when certain operations need to be synchronized. Synchronization means that all such operations have the same start and finish times and require the same crew(s). Such zone operations can be treated as one operation which has a lower level of detail compared to operations only being carried out in a given sub-project. A sub-project can be grouped to different zones when two or more zone operations are defined in its corresponding fragnet by the site engineers. Different zones do not necessarily consist of the same sub-projects.

3.1.7 Blocking constraint

A blocking operation is generally defined in a specific sub-project which serves as the hub for the transportation of resources and materials required by its adjacent sub-projects. These sub-projects form a blocked area. When a blocking operation occurs, the access to the blocked area is not available, thus no operation can start in the blocked area. Since operations are not allowed to be interrupted, the blocking operation requires the blocked area be cleared before it is started.

3.2 Mathematical formulation

The following notations are used for the formulation:

Indices:

i sub-project index, $i = 1, \ldots, M$. M is index set of sub-projects.

j operation index in each sub-project, $j = 1, \ldots, J_i$. J_i is the total number of operations in sub-project i.

t periods, $t = 1, \ldots, T$. In this paper, periods equal to days. T is the upper bound of the project duration.

k resource index, $k = 1, \ldots, K$. K is the total number of resource types needed for the construction project.

l the index of a zone operation in related sub-projects.

Variables:

y_t the binary variable with 1 indicating an operation is performed in a sub-project on day t.

$x_{i,j,t}$ the binary variable indicating whether operation j is performed in sub-project i on day t.

$x_{c,b,t}$ the binary variable indicating whether a blocking operation b is performed in sub-project c on day t.

$z_{l,t}$ the binary variable indicating whether the zone operation l is performed on day t.

Parameters:

R_k the total number of resource type k available.

$r_{i,j,k}$ the number of resource type k required by operation j in sub-project i. Generally only one type of crew is needed for an operation, but certain operations might require multiple types of crews to collaborate.

T_i The date by which sub-project i won't be available for the finishing work.

H_i	the set of operation pairs with precedence constraints in sub-project *i*.
$B_{c,b}$	the group of sub-projects blocked by the blocking operation *b* in sub-project *c*.
Z_l	the set of sub-projects forming a zone corresponding to the zone operation *l*.
L	the total number of sub-projects in zone Z_l.
$R_{k^l}^{Z_l}$	*the number of type-*k *resources required for zone operation* l *of zone* Z_l.

The mathematical formulation of this problem:

Minimize

$$f(time) \text{ or } f(cost) \tag{1}$$

Subject to

$$\textstyle\sum_{t=1}^{T} x_{i,j,t} = 1, \quad i = 1, \dots, M, j = 1, \dots, J_i \tag{2}$$

$$\textstyle\sum_{i=1}^{M} \sum_{j=1}^{J_i} r_{i,j,k} x_{i,j,t} \le R_k, \quad k = 1, \dots, K, t = 1, \dots, T \tag{3}$$

$$\textstyle\sum_{t=1}^{T_i} x_{i,j,t} = 0, t = 1, \dots, T_i \tag{4}$$

$$x_{i,j,t} \le y_t, \quad i = 1, \dots, M, j = 1, \dots, J_i, t = 1, \dots, T \tag{5}$$

$$\textstyle\sum_{j=1}^{m} \sum_{t=1}^{q} x_{i,j,t} \ \ge \ \sum_{j=1}^{n} \sum_{t=1}^{q} x_{i,j,t}, \quad 1 \le q \le T, (\mathrm{m}, \mathrm{n}) \in H_i, i = 1, \dots, M \tag{6}$$

$$x_{c,b,t} \textstyle\sum_{i \in B_{c,b}} x_{i,j,t} = 0, \quad j = 1, \dots, J_i, t = 1, \dots, T \tag{7}$$

$$\tfrac{1}{L} \textstyle\sum_{i \in Z_l} x_{i,l,t} = z_{l,t}, \quad t = 1, \dots, T \tag{8}$$

$$\frac{\sum_{i \in Z_l} r_{i,l,k^l} x_{i,l,t}}{\frac{1}{L} R_{k^l}^{Z_l}} = z_{l,t}, \quad t = 1, \dots, T \tag{9}$$

$$x_{i,j,t}, x_{c,b,t}, y_t, z_{l,t} \in \{0,1\}, \quad i = 1, \dots, M, j = 1, \dots, J_i, t = 1, \dots, T \tag{10}$$

The objective is to minimize the total project duration or project cost as described in Eq. (1). The project duration can be expressed as $\sum_{t=1}^{T} y_t$ where *T* is the last date of the entire project; the cost calculation method is explained in Section 3.1.4. Eq. (2) dictates that each operation can only be performed once in sub-project *i*. Two resource constraints exist for this problem. The general constraint is that for a certain resource type k, the total amount of such resource used by relevant operations in all projects cannot exceed the total available amount R_k on a daily basis, as formulated in Eq. (3). Another resource constraint is related with a zone and it is defined by Eq. (8) and (9). The number of resources assigned for a zone operation *l* is $R_{k^l}^{Z_l}$.

Eq. (8) denotes that once a zone operation l in a sub-project is started, all other zone operations in the same zone must start as well. Eq. (9) describes the special resource constraint in a zone – all the zone operations in a zone must share the same resource(s). Therefore we need to divide $R_{k^l}^{Z_l}$ by L, the number of zone operations in zone Z_l. Eq. (5) implies that $y_t = 1$ if any operation is done on day t. The constraint of room/sub-project availability is defined in Eq. (4), in which T_i *is the date by which sub-project i* won't be available for the finishing work (because of the incomplete structural work, such as concrete and block work). The precedence constraint is defined in Eq. (6), in which operation n is the successor of m. Eq. (7) defines the blocking constraint dictating that the blocking operation b in sub-project c and the operations in the blocked sub-projects (defined by B) are mutually exclusive.

After formulating this complex project management problem by integrating all its constraints, we observe that this mathematical program is a quadratically constrained integer program. It is well know that such a program is usually NP-hard: one cannot solve it to the optimality in polynomial time. Such an intrinsic obstacle lies in most project management problems, which has motivated researchers to develop a great variety of approximation algorithms and efficient heuristics to tackle the subject. Such algorithms include branch-and-bound (Alvarez-Valdes and Tamarit 1989), integer programming (Oguz and Bala 1994), sampling techniques (Kolisch and Drexl 1996) and local search, i.e., Tabu search (Baar et al. 1998), simulated annealing (Bouleimen and Lecocq 2003) and GA (Holland 1975).

4 The Genetic Algorithm-based Method for Look-ahead Scheduling

In this paper we focus on a detailed GA-based approach that integrates more constraints, such as engineering priorities, zone constraints and blocking constraints, than the models previously discussed (Kurtulus and Davis 1982; Scheiberg and Stretton 1994; Payne 1995; Wiley et al. 1998; Hendriks et al. 1999; Fricke and Shenhar 2000; Lova et al. 2000; Lova and Tormos 2001; Kim et al. 2005; Browning and Yassine 2010). We use this approach because GA is more flexible in accommodating additional constraints compared to the other types of local search methods. GA is widely used in RCPSP/RCMPSP to find good solutions with few computational requirements within a reasonable time period. A GA applies the principles of biological evolution to solve optimization problems by combining and altering existing solutions in order to form new ones (Holland 1975; Goldberg 1989). Scholars including Lee and Kim (1996), Hartmann (1998; 2002), Brucker et al. (1999) and Özdamar (1999) have conducted a great deal of work to use GA to address RCPSP problems. Based on their pioneering work, a number of studies (Kumanan et al. 2006; Gonçalves et al. 2008; Yassine et al. 2007; Chen and Shahandashti 2009) have been carried out to use GA to address the RCMPSP by transferring them into a RCPSP via the single-project approach. On the other hand, Kim et al. (2005) propose a two-stage encoding to apply the multi-project approach to their GA. Tasan and Gen (2008) and Xu and Zhang (2010) improve the performance of this GA by introducing enhanced fuzzy logic controllers to regulate the GA parameters (generation number, population size, crossover ratio and others). Regardless which approach is used, three main steps are used in GA in searching for the optimum solutions: initial population generation, reproduction, and selection. After a certain number (*POP*, which is assumed to be an even integer) of individuals are generated, the population is randomly partitioned into pairs of individuals. We then apply a crossover operator to such pairs to produce two new individuals (children). Subsequently, a mutation operator is applied to certain children to allow new features to appear in the offspring. The crossover and mutation operators are the most commonly used operators in the reproduction process. After the reproduction, a population size of 2 * *POP* is reached. The selection process reduces the population to its former size *POP* via certain criteria based on each individual's fitness. The reproduction and selection processes are repeated until a certain termination status (i.e., the maximum number of generations – *GEN* – is reached, the individuals in the new generation are identical, the upper limit of the CPU time is reached, etc.) is met. Through certain scheduling algorithms, individuals are created in both the initial population generation and the reproduction processes.

4.1 Scheduling algorithm

Since we need to take engineering priority into account, *priority rule* based heuristics become the starting point to create the scheduling algorithm for the proposed problem. Kolisch (1996) offers an extensive discussion regarding the two schedule generation schemes used by such heuristics – "serial" and "parallel", both of which create valid schedules based on certain priorities assigned to the activities. The serial schedule generation scheme (Kelly 1963) is an activity-oriented scheme and consists of *M* stages, where *M* is the number of activities to be scheduled. Each activity's priority is calculated (using a given priority rule) only once before the scheme starts. At each stage, an activity is selected and scheduled as soon as possible, according to its priority, taking into account the precedence relationships and availability of resources. The parallel schedule generation scheme (Kelly 1963; Kolisch 1996) is a time-oriented scheme and consists of *N* time stages. At each stage, a set of activities is scheduled and their priorities are determined at that stage. The parallel scheme is more appropriate for our proposed problem in which we need to address the two time-sensitive constraints – the zone constraint requires that a group of operations to be scheduled at the same time at a certain stage, whereas the blocking constraint restrains any operation in certain sub-projects to start while the blocking operation is being carried out. These two constraints also justify our choice of using the multi-project approach to address this problem as they introduce sub-project level interactions instead of sub-project precedence relationships. The data structure representing the multi-project example for the parallel scheme is shown in Table 4-1. All the information necessary for schedule generation can be retrieved from this structure except the operation precedence relationships shown in Figure 4-1. Table 4-1 indicates that sub-project 2 has the highest engineering priority and should be finished as early as possible. Sub-projects 1 and 3 have the same priorities which are the lowest among all the sub-projects. Operations O_{13}, O_{23} and O_{33} form a zone Z1, expressed by Z1 [O_{13}, O_{23}, O_{33}]. Operation O_{14} will block all the other sub-projects once it is started – we use B1 [O_{44}, (P_1, P_2 and P_3)] to represent such a blocking constraint.

Four operation sets are needed for the parallel scheme for the proposed problem: the *complete set C* for all the finished operations, the *in-progress set I* for all the scheduled but not yet finished operations, the *eligible set E* for the operations of which all the predecessors are already in *C*, and the *decision set D* for the operations to be scheduled at a period. At each period (a day in this case), the scheme first assigns resources to operations in *I*. If the remaining resources are sufficient for an operation to start, the scheme moves all such operations from *E* to *D* and ranks them according to their engineering priorities. If an operation is linked to a zone, it will not be put into *D* unless all other operations linked to the same zone are also in *E*. A blocking operation will only be selected into *D* if no other operations are ongoing in its corresponding blocked area. Next an operation from *D* is selected to be scheduled. Note that a zone operation, if any, will be selected before a regular operation if it has the same highest priority in *D*. To ensure the two types of constraints are satisfied, the selected operation will go through three steps: zone constraint processing, blocking constraint processing, and regular operation processing. If the operation is linked to a zone, all its related operations are scheduled as one zone operation regardless the priorities of these operations. If the selected operation is a blocking operation, it will be scheduled in the second step. If the selected operation does not belong to the above two categories, it will be scheduled in the third step. Chapter 2 gives an elaborate explanation regarding the details of these steps. The process of forming *D* and scheduling an operation will repeat as long as the remaining resources are sufficient for at least one operation in *D* to be scheduled. Such a daily schedule generation procedure is illustrated in Figure 4-2. The scheme repeats the daily schedule generation procedure day by day until all the operations are in the schedule (i.e., the whole schedule). The scheme and the daily schedule generation procedure (Figure 4-2) constitute our scheduling algorithm. In the rest of this paper, unless specifically stated otherwise, the term schedule refers to a whole schedule, not a daily schedule. The first two schedules in Figure 4-3 are two sample schedules generated using such an algorithm based on the example shown in Figure 4-1 and Table 4-1. It is notable that in Schedule 1, the blocking operation starts from day 10; no operations are allowed to start from this day on in sub-projects 1, 2 and 3 (marked by "*b*" in each cell) until it is completed. On the other hand, in Schedule 2, the blocking operation starts at the end of sub-project 4, when all the other sub-projects are already finished. Schedule 2 is more desirable than Schedule 1 in terms of the total project duration as well as the completion time of the sub-projects with higher engineering priorities.

Table 4-1. Data structure of the sample project for schedule generation.

	Sub-project1	Sub-project2	Sub-project3	Sub-project4
Engineering priority	3	1	1	2
Operation ID	1 2 3 4 5	1 2 3 4 5	1 2 3 4	1 2 3 4
Zone constraint	Z1[O_{13}, O_{23}, O_{33}]	Z1[O_{13}, O_{23}, O_{33}]	Z1[O_{13}, O_{23}, O_{33}]	Z1[O_{13}, O_{23}, O_{33}]
Blocking constraint	B1[O_{44}, (P_1, P_2, and P_3)]	B1[O_{44}, (P_1, P_2, and P_3)]	B1[O_{44}, (P_1, P_2, and P_3)]	B1[O_{44}, (P_1, P_2, and P_3)]

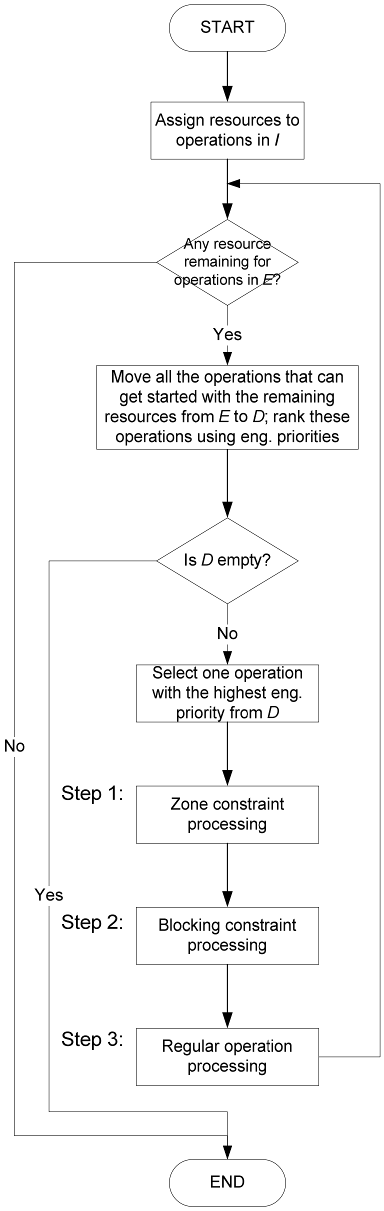


Figure 4-2. The proposed daily schedule generation procedure considering the engineering priority, zone, and blocking constraint.

4.2 Crossover

The one-point crossover operator is one of the most popular operators used in the literature for RCPSP (Hartmann, 1998; Elbeltagi et al. 2005). It is derived from the permutation based crossover operator and can be tailored according to specific features of a scheduling problem (Reeves 1995). To apply the one-point crossover to the proposed problem, we intend to find a cutting point that can act on all the sub-projects. To achieve this goal, we cut all the sub-projects at the same time so that changes in activity sequence can happen in each sub-project after the crossover. Such a way of cutting is different from the conventional one-point crossover as it picks the cutting point based on the position of a sorted activity list not time.

For any two given schedules *F* and *M*, we assume *F* costs *n1* days, *M* costs *n2* days. Then a number m from {1, 2, …, min{*n1*, *n2*}} is randomly chosen. Next we generate their children as follows:

Step 1: We choose the first m days in M as the first m days for the daughter *D*. Therefore *D* inherits the partial schedule from *M* up to day m.

Step 2: For each sub-project *P_i*, we sort its remaining work flow following the work flow of *F*, knowing that the remaining work sequence for *P_i* is still feasible (assuming that there is no iterative effect from other sub-projects).

Step 3: Starting from day *m+1*, we schedule all the work day by day. On day *m+1*, we check the resource requirement in each sub-project. If too many sub-projects ask for the same resource, we give the resource to the first k sub-projects (ranked by engineering priorities) it can satisfy and let the remaining sub-projects wait to try on the next day. The scheduling procedure is similar to that depicted in Figure 4-2, except that the work sequence in each sub-project should follow *F*.

When step 3 is finished, the operator will eventually generate a feasible LAS – a daughter *D* (child 1 shown in Figure 4-3), taking the partial schedule from S2 up to day *m*, then sorting the work flow for each sub-project from day *m+1* following *F*. Similarly we can generate a son *S* by switching the role of *M* and *F* in Steps 1 to 3.

This proposed crossover operator always generates legitimate children (although redundancy exists). We prove it with the following theorems.

The work sequence of a sub-project P_o in a schedule I is given by

$$W_{I_o} = (j_1^{I_o}, \dots, j_{J_o}^{I_o})$$

where J_o is the number of total operations in P_o. This work sequence is assumed to be a precedence feasible permutation of the set of operations. We consider two individuals selected for crossover, a mother *M* and a father *F*. The random cutting date m corresponds to a cutting position q_{M_o} in sub-project P_o in *M* and q_{F_o}in *F*, with $1 \leq q_{M_o}, q_{F_o} < J_o$. Since we use date instead of position as the cutting measurement, q_{M_o} and q_{F_o} are not necessarily the same. Through the proposed crossover operator, two children, a daughter *D* and a son *S*, are produced from the parents. In the work sequence of P_o in *D*, the positions $i = 1, \dots, q_{M_o}$ are taken from the mother, that is,

$$j_i^{D_o} := j_i^{M_o}$$

The work sequence of positions $i = q_{M_o} + 1, \dots, J_o$ of P_o in *D* is taken from the father. However the operations that have already been taken from the mother are not considered again.

Theorem 1 *If applied to precedence feasible parent individuals, the proposed crossover operator results in precedence feasible offspring.*

Proof. Let the work sequence of the parents *M* and *F* conform to the precedence assumption. We assume that child *D* produced by the crossover operator is *not* precedence feasible. That is, in a sub-project P_o, there are two operations $j_i^{D_o}$ and $j_k^{D_o}$ with $1 \leq i < k < J_o$; $j_i^{D_o}$ is before $j_k^{D_o}$, violating their precedence relations. Three cases can be distinguished (Hartmann 1998):

Case 1: We have $i, k \leq q_{M_o}$. Since all operations up to position q_{M_o} are taken from *M*, operation $j_i^{D_o}$ precedes $j_k^{D_o}$ in the work sequence of *M*, a contradiction to the precedence feasibility of *M*.

Case 2: We have $i, k > q_{M_o}$. As the relative positions are maintained by the crossover operator, operation $j_i^{D_o}$ precedes $j_k^{D_o}$ in the work sequence of *F*, contradicting the precedence feasibility of *F*.

Case 3: We have $i \leq q_{M_o}$ and $k > q_{M_o}$. Then operation $j_i^{D_o}$precedes $j_k^{D_o}$ in the work sequence of *M*, again a contradiction to the precedence feasibility of *M*.

Similar discussions can be made for any sub-project(s) using these three cases, thus the daughter *D* produced by the crossover operator is precedence feasible. As to son *S*, the positions $1, \ldots, \mathrm{q}_{\mathrm{F}_\mathrm{o}}$ of the son's work sequence are taken from the father and the remaining positions are determined by the mother. Therefore, Theorem 1 holds for both children.

Theorem 1 can be easily adapted to prove that the proposed crossover operator does not violate the engineering priorities. As to the zone constraint and the blocking constraint, we can use similar logic to testify the validity of the offspring. For instance we use Theorem 2 below to show how the zone constraint is fulfilled.

Theorem 2 *If applied to zone feasible parent individuals, the proposed crossover operator results in a zone feasible offspring.*

Proof. Let the parents *M* and *F* conform to the zone constraint. We assume that child *D* produced by the crossover operator is *not* zone feasible. That is, in sub-projects *P_m* and *P_n*, the zone operations $j_l^{D_m}$ and $j_l^{D_n}$ for Zone Z_l are not synchronized as they are supposed to be. Assuming $q_{M_m} \leq q_{M_n}$, three cases can be distinguished:

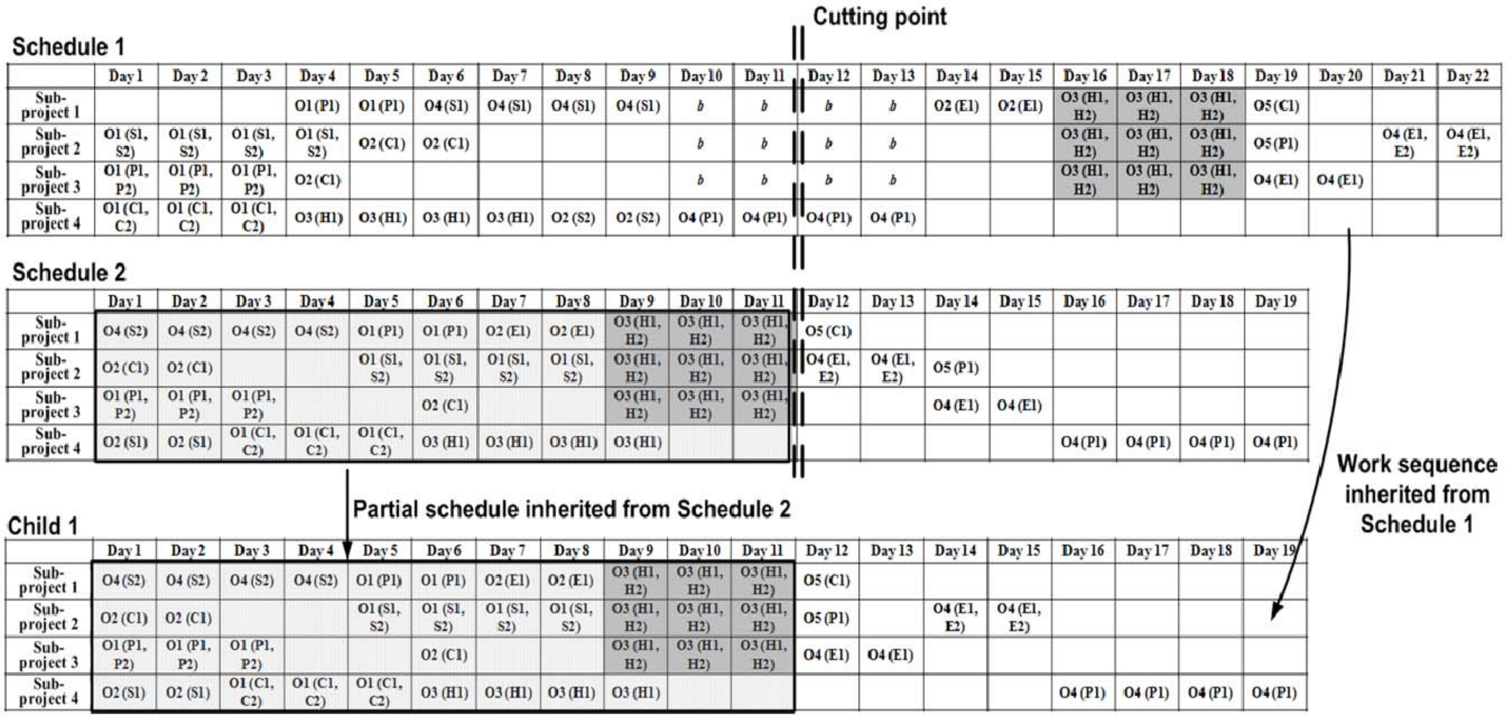
Case 1: We have $l \leq q_{M_m}$. Then operations $j_l^{D_m}$ and $j_l^{D_n}$ for Zone Z_l are not synchronized in *M*, a contradiction to the zone feasibility of *M*.

Case 2: We have $l \geq q_{M_n}$. Since the zone constraint is maintained by the crossover operator, the assumption that operations $j_l^{D_m}$ and $j_l^{D_n}$ for Zone Z_l are not synchronized in *D* contradicts the logic of the proposed daily schedule generation procedure in Figure 4-2. This procedure guarantees that all zone operations for a zone are synchronized in the schedule generation process.

Case 3: We have $q_{M_m} < l < q_{M_n}$. Then operations $j_l^{D_m}$ and $j_l^{D_n}$ for Zone Z_l are not synchronized in *M*, again a contradiction to the zone feasibility of *M*.

These cases also hold if $\mathrm{q}_{\mathrm{M_m}} > \mathrm{q}_{\mathrm{M_n}}$. Therefore the daughter *D* produced by the crossover operator conforms to the zone constraint. The same logic can be applied to son *S* and father *F*. So Theorem 2 holds for both children.

Figure 4-3 illustrates two parents (Schedule 1 and 2) and a child (Child 1) generated using the proposed crossover operator. From Figure 4-3, it can be observed that Child 1 has the same total project duration as Schedule 2. However, in terms of the degree of satisfaction to the engineering priorities, sub-project 2 (with the highest priority) completes on day 14 in schedule 2, whereas day 15 in Child 1. Therefore, the degree of satisfaction to the engineering priorities can be integrated into the time-based objective (for now we use only total project duration) if the project personnel deem it necessary.



Schedule 1

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19	Day 20	Day 21	Day 22
Sub-project 1				O1 (P1)	O1 (P1)	O4 (S1)	O4 (S1)	O4 (S1)	O4 (S1)	*b*	*b*	*b*	*b*	O2 (E1)	O2 (E1)	O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O5 (C1)			
Sub-project 2	O1 (S1, S2)	O1 (S1, S2)	O1 (S1, S2)	O1 (S1, S2)	O2 (C1)	O2 (C1)				*b*	*b*	*b*	*b*			O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O5 (P1)		O4 (E1, E2)	O4 (E1, E2)
Sub-project 3	O1 (P1, P2)	O1 (P1, P2)	O1 (P1, P2)	O2 (C1)						*b*	*b*	*b*	*b*			O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O4 (E1)	O4 (E1)		
Sub-project 4	O1 (C1, C2)	O1 (C1, C2)	O1 (C1, C2)	O3 (H1)	O3 (H1)	O3 (H1)	O3 (H1)	O2 (S2)	O2 (S2)	O4 (P1)	O4 (P1)	O4 (P1)	O4 (P1)									

Schedule 2

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19
Sub-project 1	O4 (S2)	O4 (S2)	O4 (S2)	O4 (S2)	O1 (P1)	O1 (P1)	O2 (E1)	O2 (E1)	O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O5 (C1)							
Sub-project 2	O2 (C1)	O2 (C1)			O1 (S1, S2)	O1 (S1, S2)	O1 (S1, S2)	O1 (S1, S2)	O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O4 (E1, E2)	O4 (E1, E2)	O5 (P1)					
Sub-project 3	O1 (P1, P2)	O1 (P1, P2)	O1 (P1, P2)			O2 (C1)			O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)			O4 (E1)	O4 (E1)				
Sub-project 4	O2 (S1)	O2 (S1)	O1 (C1, C2)	O1 (C1, C2)	O1 (C1, C2)	O3 (H1)	O3 (H1)	O3 (H1)	O3 (H1)							O4 (P1)	O4 (P1)	O4 (P1)	O4 (P1)

Child 1

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	Day 15	Day 16	Day 17	Day 18	Day 19
Sub-project 1	O4 (S2)	O4 (S2)	O4 (S2)	O4 (S2)	O1 (P1)	O1 (P1)	O2 (E1)	O2 (E1)	O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O5 (C1)							
Sub-project 2	O2 (C1)	O2 (C1)			O1 (S1, S2)	O1 (S1, S2)	O1 (S1, S2)	O1 (S1, S2)	O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O5 (P1)		O4 (E1, E2)	O4 (E1, E2)				
Sub-project 3	O1 (P1, P2)	O1 (P1, P2)	O1 (P1, P2)			O2 (C1)			O3 (H1, H2)	O3 (H1, H2)	O3 (H1, H2)	O4 (E1)	O4 (E1)						
Sub-project 4	O2 (S1)	O2 (S1)	O1 (C1, C2)	O1 (C1, C2)	O1 (C1, C2)	O3 (H1)	O3 (H1)	O3 (H1)	O3 (H1)							O4 (P1)	O4 (P1)	O4 (P1)	O4 (P1)

Figure 4-3. Two sample schedules (parents) and a child generated based on the proposed crossover operator.

4.3 Mutation

To avoid local optima, we develop a two-tier mutation operator. First, after scanning a schedule, if multiple sub-projects are competing for the same resource at the same time, it assigns the resource to a different sub-project of the same priority. At the operation level, it randomly picks a sub-project and randomly changes the order of two adjacent operations and test if the new schedule is still feasible. If so, the new schedule will be added to the selection pool as the candidate to go into the next generation. If not, the operator will choose a cutting date and start to use the proposed schedule generation algorithm to generate the rest of the schedule after the cutting date. Such a mutation operator guarantees the creation of a legitimate child every time the mutation occurs. But this child does not necessarily have to be a new one – if it is exactly the same as one of its parents, it will not be added to the selection pool.

4.4 Selection

We consider two alternative evolutionary mechanisms, namely elite selection and roulette wheel selection. Both mechanisms follow the procedures of initialization, selection, crossover, and mutation (Michalewicz 1995). In elite selection, after crossover and mutation, the optimal solution with the highest fitness value, i.e., minimum project duration, minimum project cost, etc., is retained and duplicated in the new population. While in the roulette wheel selection (Feng et al. 2010), individuals with better fitness values have a higher probability of being selected for the new population but there is a chance that they might not get selected. Though optional, the mutation operator is normally used to avoid the search being trapped in local optima. In the next section, we discuss its effect on search efficiency and whether it can help to find better solutions. In addition, the performance of the two selection methods is compared.

5 Illustrative Examples

We develop two examples with data extracted from an actual university building project to validate the proposed GA-based method for LAS generation. We do not use existing examples for this purpose because, to our knowledge, these examples do not consider engineering priorities, the blocking constraint, and the zone constraint. Therefore we compare the search results with the data obtained via a simulation-based method, which automates the generation of LAS taking into account the three types of constraints. The simulation-based method is developed based on the activity-based construction (ABC) modeling and simulation method, which is composed of an ABC modeling (ABC-Mod) component and an ABC simulation (ABC-Sim) component (Shi 1999). ABC-Mod models construction processes using an activity-centered approach; ABC-Sim executes the ABC model using three steps in loops: schedule activities, advance simulation, and release resources. Our simulation-based method expands the ABC method by (1) modeling construction processes in the finishing phase using a three-tiered hierarchy: project, sub-project, and operation; and (2) accommodating the three types of constraints in the first step of ABC-Sim. The design, implementation, and validation of this simulation-based method are detailed in Chapter 2. The objective functions of minimizing project duration and cost are both discussed in the two examples. The proposed GA-based method can be tailored in several ways. In terms of selection method, we discuss the search results when different combinations of selection method (Section 4.4) and mutation operator (Section 4.3) are used.

5.1 First example

5.1.1 Example setup

Chapter 2 introduced a six-room example to showcase the simulation-based LAS generation method. This example consists of one corridor (COR1), two electrical rooms (ELE1 and

ELE2), two IDF rooms (IDF1 and IDF2) and one plant room (PLT1). Therefore, four types of fragnets are introduced. In addition, one zone constraint and one corridor constraint are introduced. Sub-projects ELE2, IDF1 and IDF2 contain one zone operation “HVAC & Firefighting”. Two blocking operations – “Suspended Ceiling” and “Terrazzo Flooring” in COR1 – block all other sub-projects thus no operation can be carried out in them while the blocking operations are in progress. Only one crew is available for each discipline. For project cost items, we assign $100 to the daily indirect cost. As to the first part of direct cost, we assign $40 to the crew daily expense (assuming all disciplines cost the same). The second part of direct cost – a crew’s mobilization and demobilization cost – is considered to be eight days of a crew’s daily expense. If a crew do not have work for more than eight days, they will be demobilized with 4 days’ expenses for demobilization. Another four days’ expenses need to be paid for this crew to come back to the job site to work again.

5.1.2 Computational results

Table 4-2 compares the performance of the GA-based method with the simulation-based method. Four experiments (i.e., running the simulation for 1,000, 5,000 and 10,000 times and running the prototype based on the GA-based method with *POP* equals 100 and *GEN* equals 100) are conducted to compare their performance. Each is carried out ten times to get an average result recorded. The first row of Table 4-2 shows the average minimum project durations found by the four experiments and the corresponding project costs. The second row illustrates the average minimum project costs found and the corresponding project durations. Unlike the simulation-based method, the GA-based method searches the optimal schedule(s) according to the specific objective function. Therefore, the data sets related with the GA-based method in the first row and the second row are retrieved separately, while the data sets related with the simulation-based method are retrieved at the same time. In terms of minimum project duration, the GA-based method does not outperform the simulation-based method significantly, but it discovers the relative optimal schedule(s) faster (93 seconds vs. 390 seconds in the 5,000-time simulation). As to project cost, the GA-based method discovers a slightly lower project cost ($20,152 vs. $20,378 in the 10,000-time simulation) only using one fifth of the CPU time (161 seconds vs. 802 seconds). Although the simulation-based method can find schedules with lower cost when the number of simulations increases, it will spend much more time to reach similar results found by the GA-based method. Table 4-2 also indicates another two findings. First, the GA-based method reached the close-to-optimum duration faster than the cost (93 seconds vs. 161 seconds). This is because of the current setting of the termination criteria. When no better result is reached within 20 generations, the prototype stops. On average it does not discover a better schedule in terms of duration after the 40th generation. However, a lower cost keeps coming up until the final generation is reached. Second, both methods indicate that the relation between project duration and cost is not linear; the schedule with minimum project duration does not necessarily have the lowest project cost.

Table 4-2. Comparison between the GA-based method and the simulation-based method (1,000, 5,000, and 10,000 runs) in terms of project duration and cost for the six-room example.

Simulation -1,000 times			Simulation – 5,000 times			Simulation – 10,000 times			GA – 100*100		
Dur. (days)	Cost (USD)	CPU time (sec.)	Dur. (days)	Cost (USD)	CPU time (sec.)	Dur. (days)	Cost (USD)	CPU time (sec.)	Dur. (days)	Cost (USD)	CPU time (sec.)
74.2	20,818	75	74.0	20,724	390	74.0	20,520	802	74.0	20,470	93
77.2	20,464		77.0	20,460		76.4	20,378		76.0	20,152	161

The results using GA-based method shown in Table 4-2 are generated when the elite selection is chosen as the selection method and 10% of the population go through mutation. We also collected results based on different combinations of selection method and mutation rate as shown in Table 4-3. The setting of elite selection without mutation is the fastest in search of either minimum duration or cost. But it does not deliver the best result for cost. The roulette wheel selection takes more time than elite selection yet it does not give better results. The mutation operator appears to offer slightly better results when searching for minimum cost ($20,152 vs. $20,224 using elite selection; $20,158 vs. $20,166 using roulette wheel selection) yet it increases the search time. In sum, the setting of elite selection with 10% mutation gives the best results for the search of both duration and cost without incurring a major increase of CPU time compared to the other three settings.

Table 4-3. Computational results of four settings of the GA-based method – the six-room example.

	Elite with Mutation	Elite without Mutation	Roulette Wheel with Mutation	Roulette Wheel without Mutation
Av. Minimum Project Duration	74 days	74 days	74 days	74 days
Av. CPU Time Consumed	93 seconds	78 seconds	99 seconds	86 seconds
Av. Minimum Project Cost	20,152 dollars	20,224 dollars	20,158 dollars	20,166 dollars
Av. CPU Time Consumed	161 seconds	125 seconds	183 seconds	157 seconds

The results shown in both Table 4-2 and Table 4-3 are achieved assuming all operations have the same engineering priority. When we assign ELE1 the highest priority, IDF1 the next highest, keeping all the others the same, the lowest duration found by the GA-based method becomes 76 days and the cost $21,324. This result indicates that giving certain rooms priority for the limited resource might rule out the optimum resource allocation pattern, thus increasing project duration and/or cost.

5.1.3 Validation

Two methods are used to validate the results discussed in Section 5.1.2. First, we developed a schedule checking program to check the schedules in the following aspects: work sequence in each sub-project, crew allocation conflicts on each day, conformance to engineering priorities and constraints, duration of each operation in the schedule, and the calculation of project duration and cost. The program was applied to test all the schedules generated. No errors were found by the program. Second, the best schedules generated using the GA-based method and the 10,000-run simulation were reviewed and approved by two planning engineers and one site engineer from Consolidated Contractors International Company. Before letting them review the schedules, they were first asked to develop a LAS with all the relevant information (work sequence, crew-operation-productivity connection, quantities, crew availabilities, engineering priorities and constraints, etc.) provided in Microsoft Excel spreadsheets. Each spent an average of 5,980 seconds to generate one schedule, with an average duration of 79.3 days (a different mobilization cost number was given to the practitioners so project costs are not compared).

5.2 Second example

5.2.1 Example setup

Compared to the simulation-based method, the GA-based method discovers close-to-optimum schedules faster, but not significantly better, when applied to the six-room example. A key reason is that the scale of this example is not big enough so that close-to-optimum schedules can be found when the number of simulations becomes large enough without consuming too much CPU time (e.g., the shortest schedule can be found via the 5000-run simulation using about 390 seconds). We expect that the GA-based method outperforms the simulation method when it is applied to larger projects. Therefore, we built the second example introducing a new fragnet – faculty office with a total of 25 operations (Figure 4-4 and Table 4-4). This example consists of four electrical rooms, four IDF rooms, four plant rooms, one corridor, and seven faculty offices – a total of 20 rooms. Three zone operations are created related with “HVAC & Fire Protection”. Zone No.1 consists of rooms ELE1, ELE2, and IDF1; zone No.2 consists of rooms ELE3, IDF3, and IDF4, zone No.3 consists of rooms FAC1, FAC2, FAC3, and FAC4.

Similar to the first example, the two blocking operations – “Suspended Ceiling” and “Terrazzo Flooring” in the sub-project COR1 – block all other sub-projects once these operations start. All operations are assumed to have the same engineering priority. Two crews are available for the disciplines including carpenter, electrical, plastering, and painting. For the other disciplines, only one crew is available. Compared to the six-room example, the indirect cost increases to $240 to echo the increase of project scale while the direct cost remains the same for each crew. Since the project scales up, the *POP* increases to 400. The search does not stop even if no better schedules are found within a certain number of generations until the 200th generation (*GEN*) is reached.

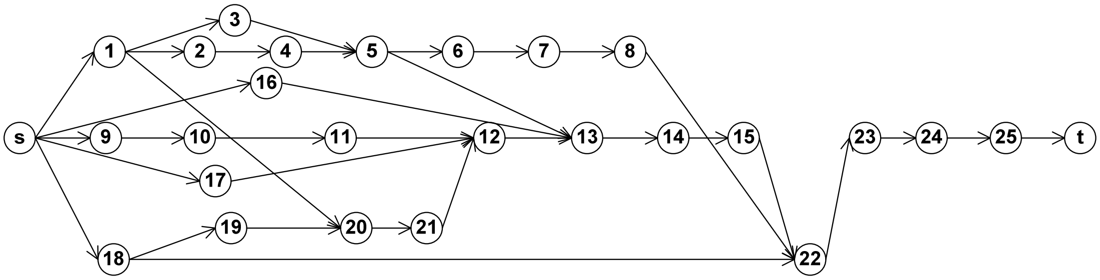


Figure 4-4. Activity-on-node network of the faculty office in the second example.

Table 4-4. Operations, precedence relations and required resources for the faculty office.

Operation #	Predecessors	Operation Name	Crew
1	-	Conduit & box above ceiling	1 Electrical
2	1	Electrical first fix above ceiling	1 Electrical
3	1	HVAC, plumbing, and fire protection above ceiling	1 HVAC
4	2	Electrical second fix - ceiling	1 Subcontractor
5	3, 4	Suspended ceiling	1 Carpenter
6	5	First coating (suspended ceiling)	1 Painting
7	6	Electrical final fix - ceiling	1 Electrical
8	7	Second coating (suspended ceiling)	1 Painting
9	-	Plumbing (precast wall)	1 HVAC
10	9	Dry wall	1 Carpenter
11	10	Conduit &box (dry wall)	1 Electrical
12	11, 17, 21	Electrical second fix (all walls)	1 Subcontractor
13	5, 12, 16	First coating (all walls)	1 Painting
14	13	Electrical final fix (all walls)	1 Electrical
15	14	Second coating (all walls)	1 Painting
16	-	Window (precast side)	1 Carpenter
17	-	Glass wall	1 Carpenter
18	-	Screed	1 Screed
19	18	Gypsum panel, first fix	1 Carpenter
20	1, 19	Conduit & box (gypsum panel)	1 Electrical
21	20	Gypsum panel, second fix	1 Carpenter
22	8, 15, 18	Carpeting	1 Tiling & Carpeting
23	22	Wood skirting & cladding	1 Carpenter
24	23	Furniture	1 Carpenter
25	24	Door	1 Carpenter

5.2.2 Computational results

Table 4-5 compares the search results using the simulation-based method and the GA-based method. All the searches were carried out ten times and the numbers in Table 4-5 are average numbers. As the number of simulations increases, better schedules are found for both project goals. This trend is illustrated in Figure 4-5, based on the data in Table 4-5. In terms of project duration, the GA-based method outperforms the simulation-based method by 1.5 days (112 days vs. 113.5 days in the 10,000-run simulation) while spending 27% less CPU time (7,235 seconds vs. 9,883 seconds) in the search. As to the project cost, it discovers schedules with an average of $66,014, more than one thousand less (1.58% less) than the average minimum cost found with the 10,000-run simulation, spending 26% less CPU time (7,361 seconds vs. 9,883 seconds). The results found using the GA-based method also indicate that the schedule with shortest project duration does not necessarily have the lowest project cost (Table 4-5). In terms of cost savings, the GA-based method offers better performance in the 20-room example compared to the 6-room example (1.58% vs. 1.11% or $1,058 vs. $226). We expect that as the complexity of a project increases, the GA-based method continues to give better results compared to the simulation-based method.

Table 4-5. Comparison between GA-based method and simulation-based method (1,000, 5,000, and 10,000 runs) in terms of project duration and cost for the 20-room example.

Simulation -1,000 times			Simulation – 5,000 times			Simulation – 10,000 times			GA – 400*200		
Dur. (days)	Cost (USD)	CPU time (sec.)	Dur. (days)	Cost (USD)	CPU time (sec.)	Dur. (days)	Cost (USD)	CPU time (sec.)	Dur. (days)	Cost (USD)	CPU time (sec.)
114.6	70,363	1,037	114.0	68,547	5,102	113.5	68,422	9,883	112.0	67,954	7,235
116.7	67,946		117.4	67,422		117.8	67,072		119.3	66,014	7,361

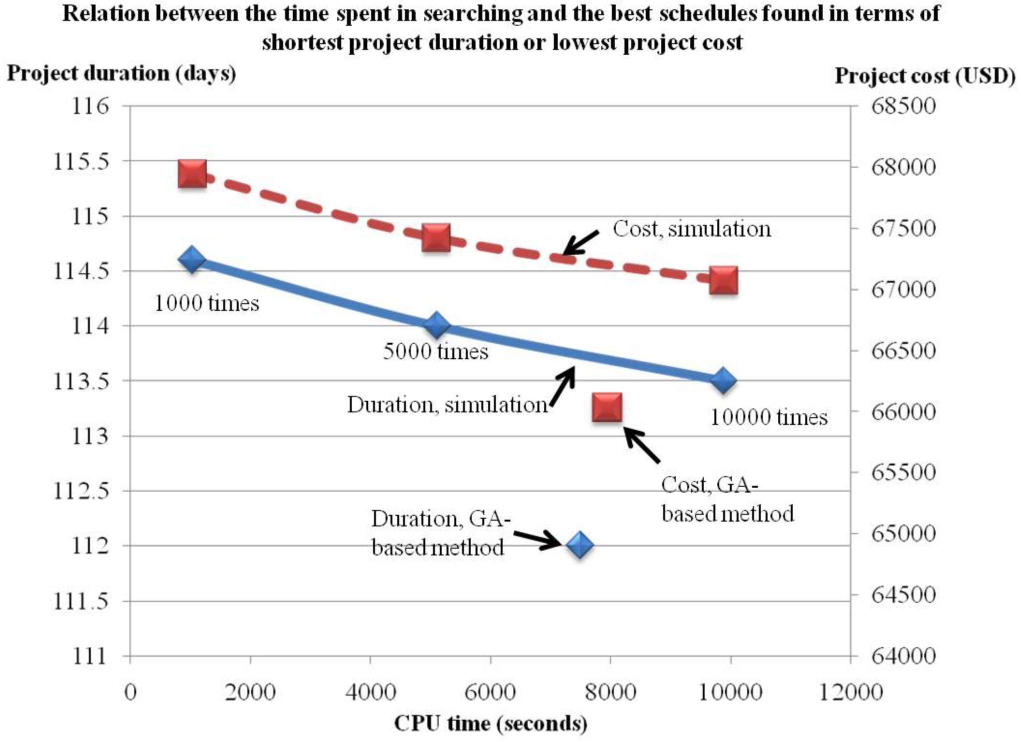


Figure 4-5. Trends in search of project duration and cost using the simulation-based method and the close-to-optimum results found by the GA-based method.

The results found using the GA-based method shown in Table 4-5 are generated when the elite selection is chosen as the selection method and 10% of the population go through mutation. Similar to Table 4-3, we collect and show in Table 4-6 the results using different combinations of selection method and mutation rate. Comparing Table 4-6 to Table 4-3, elite selection without mutation is still the forerunner in terms of CPU time spent but delivers poorer results compared to others. When mutation is applied better results are reached for both project duration and cost. The power of roulette wheel selection combined with mutation rises to surface as it delivers the best result in search of project cost, but spending 11% more CPU time compared to elite selection with mutation, which gives the second best result.

Table 4-6. Computational results of four settings of the GA-based method – the 20-room example.

	Elite with Mutation	Elite without Mutation	Roulette Wheel with Mutation	Roulette Wheel without Mutation
Av. Minimum Project Duration	112 days	112.7 days	112 days	112.2 days
Av. CPU Time Consumed	7,235 seconds	6,502 seconds	8,113 seconds	7,024 seconds
Av. Minimum Project Cost	66,014 dollars	66,236 dollars	66,004 dollars	66,012 dollars
Av. CPU Time Consumed	7,361 seconds	6,517 seconds	8,185 seconds	7,103 seconds

Since the results using the GA-based method in Tables 5 and 6 are obtained with the termination criteria of 200 generations (i.e., *GEN* equals 200), we expect that lower project cost could be found as the search goes on. Therefore, using the setting of roulette wheel selection with 10% mutation, we conducted a search to increase *GEN* to 600 and the results are shown in Figure 4-6. The lowest project cost found is $65,840, by the 500th generation.

5.2.3 Validation

All schedules generated for this 20-room example were examined by the schedule checking program described in Section 4.1.3 and no errors were discovered.

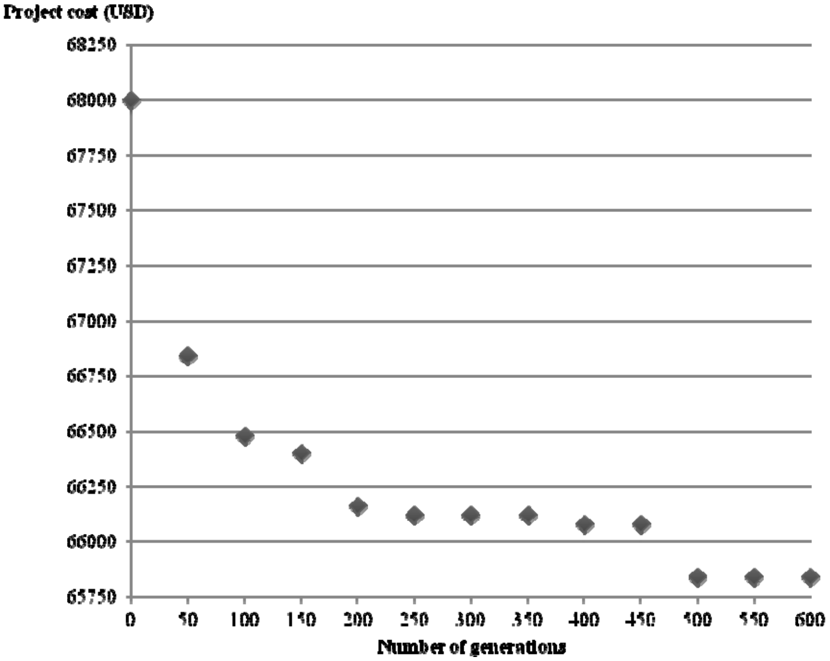


Figure 4-6. Minimum project cost found at a 50-generation interval within 600 generations using the GA-based method.

6 Conclusion

This paper presents a new GA-based method which automates LAS generation in the finishing phase of complex projects with the objective of minimizing project duration or cost. Besides the constraint of precedence relations between operations and the constraint of resource availability, the method also deals with three aspects that are common at the construction job sites of complex projects: engineering priorities, the blocking constraint and the zone constraint. The consideration of these aspects extends the previous related work in the area of GA, especially the multi-project GA approach, to address RCMPSP. The current multi-project GA literature either assumes each individual sub-project/room to have the same priority (then dynamically determine their priorities in the scheduling process) or directly assigns precedence relations between sub-projects according to particular project requirements. By considering the priority for each individual sub-project, the method allocates resources to the sub-projects in the order of their level of “hunger” for resources. The current GA-based RCMPSP literature only deals with the same level of detail in terms of activity/operation definition. By considering the zone constraint, the proposed method breaks this limitation and provides a new perspective to look at a group of sub-projects, which at certain time, need to be synchronized for a particular type of work sharing the same resources. It further extends the GA-based RCMPSP research by considering the blocking constraint, which presents another new perspective to look at a group of sub-projects – at a certain time, the work in one sub-project prevents all work in certain other sub-projects from being carried out. The proposed method can liberate the project planners from the tedious daily/weekly LAS generation process and focus on other issues on the project; it can benefit the construction managers by offering them alternative schedules to achieve varied goals at different project stages; it can also improve the efficiency of the site engineers in supervising the crews’ daily work.

A six-room example is analyzed to validate the proposed method, and another 20-room example is presented to illustrate its power when a project scales up and its complexity increases. It is observed that compared to the simulation-based method, the proposed method can find good solutions in terms of either project duration or cost using less time. Comparing the two examples, the results denote that as the complexity of the project grows, the proposed method gives better results (i.e., more savings of schedule duration or cost) than the simulation-based method. Both examples also indicate that a trade-off between duration and cost exists for non-repetitive projects. The schedule results were generated when no finishing-related work had started yet in any sub-project. Similar results can be obtained when the finishing work is in progress in certain sub-projects and the LAS are generated based on the work progress.

Although we formulate the scheduling problem based on an university building project, the design of the proposed method makes it easily applicable to look-ahead scheduling in the finishing phase of various projects, such as hospitals and medical centers, offices, data centers, laboratories, and research facilities. More broadly speaking, since the proposed method utilizes a multi-project approach to address RCMPSP with more constraints, it can also be used to address multi-project planning problems, especially when (1) individual projects have specific priorities, (2) interdependencies exist among these projects, and (3) they compete for scarce resources. The first and third factors are automatically considered when addressing engineering priorities and resource constraints. For the second, we will need to decompose different types of interdependency into zone and blocking constraints and see whether we need to add other constraints (such as staggering of projects) to fully accommodate all the interdependencies. In addition, if the objective function is appropriately tailored, the proposed method can also be applied to project portfolio management (Reyck et al. 2005; Levine 2005) to help general contractors and subcontractors better select projects and properly prioritize them by going through multiple portfolio-mix scenarios.

In the current stage of development, the proposed method deals only with “finish-start” relations between operations because most of the operations are carried out in rooms with specific equipment and materials (and temporary structures) cramped inside, making it very difficult for more than one operation to proceed at the same time. The future research should take into consideration other types of relations when the rooms are big enough to allow multiple operations to go in parallel. As to the priorities of sub-projects, we discover that when resources are relatively rare (i.e., only one crew is available for each discipline), giving the resources first to the sub-projects with higher priorities does not necessarily guarantee that they finish before those with relatively lower priorities. Therefore, deadlines should be considered along with priorities to satisfy the construction managers’ requirements. Thirdly, priority-rule based heuristics are not considered in the proposed method because we focus on addressing the engineering priorities as well as the two constraints in the schedule generation process. When there is a tie between two operations with the same engineering priority competing for the same resources, we randomly assign the resources to one operation. In future research, when many sub-projects have the same priority, we can use priority-rules to break such ties and evaluate whether better schedules can be derived.

Finally, interruption of an operation is not allowed in the proposed method because suspending an operation to allow another operation to start in a sub-project would result in additional crew move-in, move-out, material/equipment in place and set-up time leading to extra time and cost. But sometimes such interruptions are unavoidable. Hence future research should take this factor into account as well.

7 References

Akinci, B., Fischer, M., Levitt, R., and Carlson, R. (2002a). "Formalization and automation of time-space conflict analysis." *Journal of Computing in Civil Engineering,* 16(2), 124-134.

Akinci, B., Fischer, M., and Kunz, J. (2002b). "Representing work spaces generically in construction method models." *Journal of Construction Engineering and Management*, 128(4), 296-305.

Alvarez-Valdes, R., and Tamarit, J.M. (1989). "Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis." *Advances in Project Scheduling,* Elsevier, Amsterdam,114-134.

Baar, T., Brucker, P., and Knust, S. (1998). "Tabu search algorithms and lower bounds for the resource-constrained project scheduling problem." *Meta-heuristics: Advances and trends in local search paradigms for optimization,* Kluwer Academic Publishers, 1-8.

Bouleimen, K., and Lecocq, H. (2003). "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version." *European Journal of Operational Research,* 149(2), 268-281.

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.

Brucker, P., Drexl, A., Möhring, R., Neumann, K., and Pesch, E. (1999). "Resource-constrained project scheduling: Notation, classification, models, and methods." *European Journal of Operational Research*, 112(1), 3-41.

Chen, P., 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.

Cherneff, J., Logcher, R., and Sriram, D. (1991). "Integrating CAD with construction-schedule generation." *Journal of Computing in Civil Engineering,* 5(1), 64-84.

Dawood, N., and Mallasi, Z. (2006). "Construction workspace planning: assignment and analysis utilizing 4D visualization technologies." *Computer-Aided Civil and Infrastructure Engineering,* 21(7), 498–513.

Elbeltagi, E., Hegazy, T., and Grierson, D. (2005). "Comparison among five evolutionary-based optimization algorithms." *Advanced Engineering Informatics,* 19 (1), 43-53.

Feng, C., Chen, Y., and Huang, J. (2010). "Using the MD CAD model to develop the time–cost integrated schedule for construction projects." *Automation in Construction,* 19(3), 347–356.

Chapter 4

Goldberg, D.E. (1989). Genetic Algorithm in search, optimization, and machine learning, Addison-Wesley, Reading, MA.

Gonçalves, J.F., Mendes, J.J.M., and Resende, M.G.C. (2008). "A genetic algorithm for the resource constrained multi-project scheduling problem." *European Journal of Operational Research*, 189(3), 1171-1190.

Guo, S. J. (2002). "Identification and resolution of work space conflicts in building construction." *Journal of Construction Engineering and Management,* 128(4), 287-295.

Hartmann, S. (1998). "A competitive genetic algorithm for resource-constrained project scheduling." *Naval Research Logistics*, 45(7), 733-745.

Hartmann, S. (2002). "A self-adapting genetic algorithm for project scheduling under resource constraints." *Naval Research Logistics*, 49(5), 433-448.

Holland, H.J. (1975). *Adaptation in Natural and artificial systems,* University of Michigan Press, MI.

Kelly, J.E. (1963). "The critical-path method: Resources planning and scheduling." *Industrial Scheduling,* Prentice-Hall, NJ, 347-365.

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). "Efficient priority rules for the resource-constrained project scheduling problem." *Journal of Operations Management*, 14(3), 179-192.

Kolisch, R., and Drexl, A. (1996). "Adaptive search for solving hard project scheduling problems." *Naval Research Logistics,* 43(1), 23-40.

Kumanan, S., Jegan Jose, G., and Raja, K. (2006). "Multi-project scheduling using an heuristic and a genetic algorithm." *International Journal of Advanced Manufacturing Technology*, 31, 360-366.

Kurtulus, I., and Davis, E.W. (1982). "Multi-project scheduling: categorization of heuristic rules performance." *Management Science*, 28(2), 161-172.

Lee, J.K., and Kim, Y.D. (1996). "Search heuristics for resource constrained project scheduling." *Journal of the Operational Research Society*, 47(5), 678–689.

Levine, H. (2005). *Project portfolio management: a practical guide to selecting projects, managing portfolios, and maximizing benefits,* Jossey-Bass, San Francisco, CA.

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 multiple project scheduling." *Annals of Operations Research*, 102, 263-286.

Michalewicz, Z. (1995). "Heuristic methods for evolutionary computation techniques." *Journal of Heuristics*, 1(2), 177-206.

Chapter 4

Oguz, O., and Bala, H. (1994). "A comparative study of computational procedures for the resource constrained project scheduling problem." *European Journal of Operational Research,* 72(2), 406-416.

Organización, I. D. (2011). "Berth allocation planning in Seville inland port by simulation and optimization." *Advanced Engineering Informatics*, 25(3), 452-461.

Özdamar, L. (1999). "A genetic algorithm approach to a general category project scheduling problem." *IEEE Transactions on Systems, Man, and Cybernetics,* Part C: Applications and Reviews, 29, 44-59.

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

Reeves, C.R. (1995). "Genetic algorithms and combinatorial optimization." *Applications of Modern Heuristic Methods,* Alfred Waller Ltd., Henley-on-Thames, 111-125.

Reyck, B.D., Grushka-Cockayne, Y., Lockett, M., Calderini, S.R., Moura, M., and Sloper, A. (2005). "The impact of project portfolio management on information technology projects." *International Journal of Project Management,* 23, 524–537.

Riley, D. (1994). "Modelling the space behaviour of construction activities," Ph.D. Dissertation, Pennsylvania State University, University Park, PA.

Riley, D., and Sanvido, V. (1997). "Space planning method for multi-story building construction." *Journal of Construction Engineering and Management,* 123(2), 171-180.

Scheiberg, M., and Stretton, J. (1994). "Multiproject planning: tuning portfolio indices." *International Journal of Project Management*, 12(2), 107-114.

Shi, J.J. (1999). "Activity-based construction (ABC) modeling and simulation method." *Journal of Construction Engineering and Management,* 125(5), 354-360.

Tasan, S.O., and Gen, M. "A priority-based genetic algorithm approach for solving multiple alternative project scheduling problems with resource constraints and variable activity times." *IEEE International Conference on SMC,* Singapore, 2537-2542.

Thabet, W. Y., and Beliveau, Y. J. (1997). "SCaRC: space-constrained resource-constrained scheduling system." *Journal of Computing in Civil Engineering,* 11(1), 48–59.

Wiley, V.D., Deckro, R.F., and Jackson, J.A. Jr. (1998). "Optimization analysis for design and planning of multiproject program." *European Journal of Operational Research*, 107(2), 492-506.

Xu, J., and Zhang, Z. (2010). "A fuzzy random resource-constrained scheduling model with multiple projects and its application to a working procedure in a large-scale water conservancy and hydropower construction project." *Journal of Scheduling*, online first.

Yassine, A.A., Meier, C., and Browning, T.R. (2007). "Multi-project scheduling using competent genetic algorithms." Pennsylvania State University, PDRL working paper PDL-2007-01, 2007.

Chapter 5: Research Validation and Conclusion

1 Research Validation

Validations are briefly discussed in Chapter 2 and 4. This section summarizes the research efforts made in order to (1) draw unambiguous conclusions from the results (Vaus 2001) and (2) provide empirical evidence to demonstrate the power of the developed methods (Ho et al. 2009). Since the big idea of this research is to generate accurate and close-to-optimum LASs quickly, I developed the validation methods around *accurate, quick,* and *close-to-optimum*.

1.1 The accuracy of the LASs created

Regardless how a LAS is created, it has to be accurate. This is the bottom line for a look-ahead scheduling method to offer both theoretical and practical value, and the first criterion I use to measure the quality of the LASs created using the proposed methods.

In this research project, I claim a LAS created is accurate if it (1) sequences operations without violating the precedence, resource availability, and critical engineering constraints including zone, blocking, and engineering priority constraints; (2) assigns the correct number of crews to the corresponding operations; (3) the operation duration is correctly calculated/retrieved. A basic method is to check the LASs created manually. However, errors can be overlooked due to human mistakes, especially when a great number of LASs must be examined or when the LASs are created for a large-scale simulated project (e.g., 100 rooms with a total of 2,000 operations).

To overcome this challenge, I developed an automated LAS checking tool (ALASCT) that can be used together with the prototype in two ways. First, it can be switched on in the

LAS simulation/optimization process. Once an error in the aforementioned three aspects is discovered, ALASCT terminates the ongoing process and returns to the user a report regarding the error. The report helps debug the prototype to determine whether an error resulted from a fault in the system design (in which case I need to change the corresponding process designed for LAS automation/optimization) or an oversight in coding. Second, ALASCT can be used to check the accuracy of the LASs created *after* a simulation/optimization process is completed. I can let ALASCT randomly select certain LASs from the results or allow it to select all LASs to check their accuracy, depending on the number of LASs created and the scale of the simulated project for which the LASs are created.

To check the accuracy of the LASs created based on ALASG coupled with the computer simulation method, I designed three experiments: 1,000,000 simulations for a 6-room case, 100,000 simulations for a 20-room case, and 10,000 simulations for a 100-room case. These three experiments were made after I had conducted more than three hundred smaller experiments using ALASCT and manual checking to ensure zero coding mistakes in the prototype. No errors were discovered after using ALASCT to check *all* LASs for the three experiments.

The accuracy of the LASs was also checked externally by a project planner and a site engineer from CCC using the 100,000-simulation 6-room case. Both of these civil engineering professionals worked in the field research project and participated in the charrette test (described in Chapter 2). Since it is impossible for them to check all the LASs created, I presented them one LAS with the shortest duration and another, which was randomly selected. They found no errors in these schedules. The graduate students who participated in the other charrette test (Chapter 2) were not asked to check the accuracy of these schedules because a method that is validated using students provides weaker evidence of power than a method that is validated by practitioners (Ho et al. 2009).

1.2 The closeness to optimum

The closeness to the optimum solution(s) is the second criterion I chose to measure the quality of the LASs created using the proposed methods. Construction duration and cost are the two measurements used to compare the methods in this regard. When comparing the ALASG method with the manual ALS generation method, I used charrettee tests with two groups of participants – graduate students (from the Construction Engineering and Management Program) and practitioners (two project planners and one site engineer from CCC) – who were required to generate LASs for a 6-room case (Chapter 2). First, the student group was asked to generate a LAS manually to achieve the shortest construction duration while the practitioners aimed for lowest construction cost. Then, both groups were asked to perform the same task using 100 simulations facilitated by the prototype. Only one accurate LAS was created in the first step out of the 10 results, with the construction cost 8.7% higher than the best LAS created by the same test subjects using the prototype. This validates our observation that staff on site can seldom generate an accurate LAS to assign work to crews and rooms, not to mention an optimized LAS. As mentioned in Chapter 2, when asked to generate a LAS manually for a 20-room case, a project planner said that “I would not waste my time in creating a LAS for 20 rooms when I need to organize and synthesize so much information when scheduling, knowing the inevitability of making a mistake.”

To test how close the LASs created based on the simulation method are to the optima, I applied the exhaustive search method (Slaney 1995) to the same 6-room case to find the optimum LAS with the shortest construction duration. The same shortest duration (74 days) was found in 5,370 seconds. I found this shortest duration 8 out of 10 times using the simulation method (1,000 simulations), which only took about 75 seconds each time (Section 5.1.2, Chapter 4). This demonstrates that, for small cases (in which the search space is small too), the ALASG-based simulation method can find the optimum very quickly. However, the search space grows exponentially as the scale of a project grows, and so it takes much longer (hours or even days) for the simulation method to find the actual optimum. Future efforts should be made to apply parallel computing (Kandil and El-Rayes 2006) and other advanced computing techniques to shorten the time spent on simulations.

It is also difficult to compare the optimization method with existing scheduling methods because, to my knowledge, the cases that have been used by prior researchers do not contain the set of constraints (i.e., precedence, room and crew availability, zone, blocking, and engineering priority constraints) this study addresses. For these reasons, to demonstrate the power of the GA-based LAS optimization method, I used computational experiments that compare the performance of the GA-based optimization method to the simulation-based ALASG method on the basis of the best LASs discovered in terms of construction duration or cost. As discussed in Chapter 4, for the 6-room case, although the GA-based method did not discover a shorter construction duration, it did find LASs with slightly lower construction costs. When the project was scaled up to 20-rooms, the GA-based method found LASs with a cost of 1.58% lower (on average) than the best LAS found by the simulation-based method. In terms of construction duration, the GA-based method found LASs that are 1.5 days shorter (on average) than the best LASs found by the simulation-based method. It can be expected that more savings in time and cost can be achieved by the GA-based method as the scale of a project increases.

The performance of the GA-based optimization method is dependent on the configuration of certain parameters against a given resource availability scenario and the desired objective (i.e., a project goal). Such parameters include the rate of crossover, mutation, replication, and so on. Therefore, I have conducted computer tests on tailoring these parameters to find an effective configuration of the optimization method. In future research, methods such as fuzzy logic can be used to automate the searching of the best configuration of these parameters (Xu and Zhang 2010).

1.3 The speed of LAS generation

The previous two sections have discussed the validation methods used to test the quality of the LASs generated by the proposed methods. This section focuses on the process of LAS generation instead of the LASs created. The measurement for such a purpose is time, i.e., how quickly an accurate LAS can be created.

To compare the speed of LAS generation using ALASG vs. the manual method, charrettee tests (Chapter 2) were used as the primary validation method. I found that the participants in the tests spent 96% less time in LAS generation using the prototype compared to using the manual LAS generation method. All LASs created are accurate, compared to the fact that only one LAS is discovered out of the 10 LASs manually created. Therefore, I can claim that even without the GA-based optimization method, the simulation-based method is already faster and more powerful than the manual method.

As discussed in Section 1.2, to validate the speed of LAS generation using the GA-based method, I can only compare it to the simulation-based method because no similar case studies with all the necessary constraints can be found. To compare these methods, computer tests were used (as discussed in Chapter 4), which showed that the GA-based method can find better result faster.

1.4 Other remarks regarding research validation

All the cases (i.e., the 6-room case, the 20-room case, and the 100-room case) mentioned in the previous sections were developed based on simulated data extracted from the field research project. I did not use real project data because the state-of-the-art research has not yet developed methods to handle the advanced complexities of real data (Maile et al. 2007). For example, in the field research project, operations were often executed in parallel in large rooms such as lecture halls and faculty lounge. In this study, I assume that only one operation can proceed in one room at a time. Therefore only simulated data were used and the charrettee as well as computer tests had to be carried out in a controlled environment so that all the constraints and assumptions could be properly considered (Ho et al. 2009).

With respect to the generality of this research, since all cases were developed based on an university building project in the finishing phase, it might appear that these conclusions are only valid when applied to look-ahead scheduling for this type of project. However, the ALASG and the GA-based optimization methods were developed with a room/unit as the core in the method design, making these methods easily applicable to look-ahead scheduling in the finishing phase of various projects, such as hospitals and medical centers, offices, data centers, laboratories, and research facilities. More test cases can be developed using the project data collected from these projects to provide further evidence of generality.

2 Summary of Theoretical Contributions

This research was motivated by the increasing challenges faced by project management staff on site when the amount of information from various project databases as well as the number of engineering constraints they need to process increase exponentially as the scale and complexity of construction projects escalate. Previous studies investigating automated construction scheduling and schedule optimization serve as a solid foundation for this study. Based on these prior studies as well as my field research, I have developed an integrated approach to automating LAS generation and optimization by constructing (1) an ALASG method that is composed of an information model that integrates various sources of data at the appropriate level of detail to prepare for LAS generation and a LAS generation process model that interacts with the information model to formalize the schedule generation process, and (2) a GA-based method that is built on top of ALASG to discover optimized LASs for the finishing phase of complex construction projects. In addition, through computer simulation developed on the basis of ALASG, I was able to conduct statistical analyses to evaluate the relationship between resource utilization and project goals.

2.1 Automated construction schedule generation

2.1.1 Project information modeling

This study first reveals that, to generate a LAS for the finishing phase of a complex project, the site engineers and project planners need to manage various sources of data, including crew and room resource availability, bills of quantity, crew productivity, progress updates, work sequences, and engineering constraints (such as zone constraints, blocking constraints, and engineering priorities). When entering an operation on a LAS, the site engineers and project planners need to connect these sources at the appropriate level of detail in order to avoid mistakes. Unfortunately, according to Cornick (1990), two-thirds of construction problems are caused by inadequate coordination and inefficient means of communication of project information and data. In terms of data integration for project planning and scheduling, construction method models (Aalami 1998; Akinci et al. 2002) provide state-of-the-art data integration for the formation of activities/operations and linking them to certain constraints. However, these models do not specify the relation between activities/operations (i.e., precedence constraints), the blocking and zone constraints (to allow different levels of detail in look-ahead scheduling), crew productivity against the type of material to be installed, the calculation/retrieval of operation durations, and construction costs. This study proposes an information model that links different project databases at the proper level of detail so that ALASG can quickly define operations for LAS generation.

In addition, chapter 2 identifies the room as one of the core components for LAS generation and depicts different perspectives from which to view the room and how it should be treated in the information model. From the product perspective, a room is deemed a production unit, which contains a number of materials to be installed; from the process perspective it is a fragnet; and from the resource perspective, a type of resource. Therefore, a room has multiple roles in the information model, each of which is linked with a particular type of project data.

Chapter 2 also defines zone and blocking constraints that frequently occur at construction job sites, particularly in the finishing phase. The consideration of such constraints in the information model extends the activity-precedence-relation. That is, the zone constraint defines a relation that combines start-start and finish-finish precedence constraints for multiple operations of the same nature in a group of rooms, and the blocking constraint specifies a negating relation between an operation in a particular room and an operation in the adjacent rooms.

2.1.2 Process modeling

This thesis extends the literature of automated construction scheduling by presenting an innovative process model that accommodates the critical constraints in LAS for the finishing phase of complex projects and details the processing of these constraints. Based on both the information model and the process model in particular, chapter 2 describes the implementation of a prototype that delivers LASs automatically through computer simulations. The prototype represents an attempt to (1) capture all necessary inputs so as to reflect dynamic conditions on site, (2) improve users' efficiency in the data entry process, and (3) facilitate users' review of the LASs as outputs.

2.1.3 Resource utilization in look-ahead scheduling in the finishing phase

Prior studies in the domain of scheduling for repetitive projects have indicated that how resources are utilized can affect project goals (i.e., construction duration and cost). For example, in a highway project, El-Rayes (2001) discovered that when crew idle time was eliminated, the construction duration increased significantly. In addition, Vanhoucke (2006) claimed, on the basis of several case studies of repetitive projects, that the schedule with the shortest construction duration corresponds to a very large value of crew idle time compared to schedules with longer durations. In the domain of scheduling for the finishing phase of complex non-repetitive projects, I am not aware of a study that discusses such relationships between resource utilization and project goals. If a particular relationship exists, it can be used for the quick discovery of optimized LASs. This research is one of the few efforts to date that explore relationships between resource utilization and project goals in the finishing phase of complex projects. By conducting statistical analyses of the data sets obtained from the prototype, I was able to define the metrics that help appraise the resource utilization in a LAS and to evaluate the relationship between resource utilization and project goals (Chapter 3). This evaluation paves the way for an objective definition of fitness functions in the GA-based optimization method described in Chapter 4 and lays the groundwork for multi-objective optimization investigations in future research. Chapter 3 also provides further evidence of the power of the ALASG method because skilled-workers and fixed-size crews are considered in the LAS generation process and the discussion of resource utilization. This demonstrates that the prototype developed on the basis of ALASG can handle multiple levels of detail regarding the workforce resource in addition to the multiple levels of detail regarding the room resource (i.e., rooms and zones).

2.2 Schedule optimization

If all rooms are entirely independent, each room can be considered a sub-project; thus the LAS generation problem becomes a resource-constrained multi-project scheduling problem (RCMPSP), which has been widely discussed. GA is one of the most widely used algorithms to address such problems. However, existing studies either assume that all sub-projects are completely independent or that they strictly conform to specified finish-start precedence relations (Kim et al. 2005). The GA-based method described in this thesis takes into account three engineering constraints, which dictate that certain sub-projects have to be considered concurrently for resource allocation. Firstly, by considering the priority for each individual sub-project, the method allocates resources to the sub-projects in order of their level of “hunger” for resources. Secondly, the method takes into account the zone constraint; thus this study breaks new ground by encompassing a finer grained level of detail of operations/activities on site. The zone constraint requires that operations in certain rooms be synchronized and share the same crew resource. Such operations in a zone can be considered as one operation at a lower level of detail in comparison to the regular operations, which are confined to their corresponding rooms. The method further extends the GA-based RCMPSP research by considering the blocking constraint, which presents another new perspective on a group of sub-projects, showing that, at certain times, the work in one sub-project can prevent all work in certain other sub-projects from being carried out. The two examples discussed in chapter 4 demonstrate the power of the GA-based method – it can find more effective LASs faster than the simulation-based method discussed in chapter 2.

3 Practical Implications

This research represents an effort to help site engineers, construction managers, and project planners improve their efficiency in project scheduling, monitoring, and control in the finishing phase of complex construction projects, such as the construction/renovation of campus and university buildings, hospitals and medical centers/offices, data centers, and laboratories and research facilities. The LASs generated as the result of this study can serve as a daily guide for work assignments because the LASs not only reflect the dynamic changes on site but also clearly indicate, on a daily basis, which crew will be working on which operation at which location considering these changes. Specifically, this study provides four key contributions to practice.

First, the LASs improve transparent communication among actors on site including site engineers, project planners, construction managers, foremen, subcontractors, quantity surveyors, design engineers, and QA/QC inspectors. These LASs put everyone on the same page by specifying a proper work sequence in each room and allocating resources to each operation scrupulously, thus preventing work conflicts and rework. They can also serve as a work calendar to illustrate the crews' tasks on a day-to-day basis. Furthermore, they demonstrate the level of work intensity at different project locations so that the relevant personnel can get prepared for the potential issues that might occur, especially when multiple parties are involved in certain locations.

Second, this study showcases an approach for integrated project management, exploiting the possibility of a comprehensive system for computer-aided project scheduling, monitoring, and control. It echoes the work of the technical leaders such as Consolidated Contractors Company (CCC), Skanska, and DPR, that have attempted to integrate progress updates and progress reporting into their BIM-based project control systems. To take one step further, the proposed information model and process model can be applied to such systems to expand their capacity for planning and scheduling significantly. The LASs can also serve as the foundation for material forward planning (Haddad 2011) to help implement just-in-time (Sugimori et al. 1977) material delivery and integrated supply chain management system.

Third, the simulation-based approach for LAS generation, detailed in chapter 2, can help site engineers and project planners improve their understanding of resource utilization and explore appropriate work patterns while allocating crew resources. For example, they can verify planners' intuitions regarding the relationship between the utilization of one type of resource and a specific project goal (see chapter 3). In addition, by switching room with crew on the LASs, the prototype presents the LASs from the crew-centered perspective, i.e., the crew daily work plans. Such plans help site engineers to improve the crew management and respond agilely to changes on site. For instance, if for some reason Crew A cannot carry out an operation on a certain day, the site engineer can use the crew daily work plan to find another crew of the same discipline that is idle on that day to perform the operation instead. The simulation-based approach can also help site engineers explore multiple options to evaluate how different project-related parameters can affect the achievement of certain project goals – e.g., they can observe the effect of the increase/decrease of a certain type of crew on project costs and check how the size of certain zones affects project duration.

Last but not least, the GA-based optimization method, detailed in chapter 4, helps construction managers and project managers to move the project forward towards desired goals. Combined with the simulation-based method, this study also allows these managers to consider time-cost tradeoff and provides options for them to choose what is best for the project.

4 Suggestions for Future Research

The LAS automation and optimization methods proposed in this research project are subject to the assumptions discussed in Chapters 2, 3, and 4, such as (1) only finish-start relations are used to define operation dependencies, (2) only one operation can be carried out in a room at any time, (3) the material, equipment, and engineering documents are readily available, (4) crew production rates are fixed throughout a project, (5) the unit of duration is a day (not hour or minute), (6) once started, an operation cannot be interrupted until it is finished. The elimination of each of these constraints constitutes a future study. For example, when a room becomes very large (e.g., a lecture hall large enough to seat 200 people), multiple operations can be performed at the same time, if they are independent from each other. To avoid work conflicts among such operations, sub-spaces within a room should be defined using workspace definition methods (Thabet and Beliveau 1997; Akinci et al. 2002).

A challenge in this area is how to automate the definition of these sub-spaces to facilitate automated LAS generation and optimization when a large number of operations (i.e., more than 40 operations in a lecture hall), which form a sophisticated operation-network structure, need to be carried out in a given room, which is a small part of a complex project. Another example is changing the granularity of operation duration from a day to an hour. Since construction operations on site are subject to many unpredictable factors, it is difficult to fully automate their execution and management (such as timely progress update and reallocation of crews from operations that need to be interrupted). Without this automation, checking the status of operations and moving crews on an hourly basis are difficult to implement. Therefore, the value of hour-based operation duration calculations in LAS still needs to be evaluated in future studies.

Another extension to this research is the construction cost calculation method. When calculating direct construction cost, only uniform daily crew cost is used. Future research can add costs such as rental of expensive equipment, setup cost of this equipment as well as crews, and the routing of material, crew, and equipment from the current location to the next location to form a comprehensive cost calculation mechanism.

A third area for the future research is the automated editing and updating of engineering constraints. In this study, the zone and blocking constraints are hard-coded in the prototype, which does not allow users to provide additional (engineering) constraints that can affect work sequences and resource allocation. A constraint, such as a safety regulation, could affect multiple groups of professionals on a job site. A distributed agent-based system (Kim and Paulson 2003) that allows different parties to negotiate and collaborate to finalize a constraint would be a useful add-on to the prototype, which would in turn deliver more practical value to professionals on site. One particular challenge in developing such an agent-based system is that the proposal of certain constraints by one party might affect the goals of another. For instance, a subcontractor might propose a constraint to reserve a certain floor area to work without disturbance. Such a constraint might be in conflict with the goals of the general contractor or other subcontractors. An arbitration module should be developed with respect to this conflict. The module would present LAS alternatives that show solutions of the "middle ground" to different parties.

Fourth, although a 100-room test case has been developed to test the accuracy of the proposed methods (Section 1.1), in practice, rooms are hardly ever processed in a batch of this scale, but rather in a group of 10 to 30 rooms, which form a construction sub-area (which is usually larger than a zone as defined in a zone constraint). Using such a 100-room test case (or other test cases), I can investigate the problem of centralization vs. decentralization (or optimization vs. sub-optimization) by evaluating the LASs created for a scenario without sub-areas and scenarios with sub-areas. A challenge would be how to define the roles of a site engineer in charge of a give sub-area and how to decide on a resource allocation/sharing strategy among the sub-areas to reflect the actual site conditions.

Last but not least, the proposed GA-based LAS optimization method is still a method in the domain of resource constrained multi-project scheduling problems (RCMPSP), which means that the availability of crews must be known (i.e., fixed) before LASs are created. In practice, general contractors typically care more about whether a system can give them a LAS that tells them the best crew configurations (i.e., the number of crews/subcontractors they should prepare for each trade) to achieve lowest construction cost or shortest duration. For example, for a 20-room case, when I change the crew availability from one crew per trade to five per trade, the construction cost found by the GA-based optimization method goes down first and then back up (Figure 5-1). However, these variations in crew availability were entered manually. Thus, efforts should be made to find the best crew configuration, which goes beyond the domain of RCMPSP in which crew availability is considered a (fixed) parameter, not a variable.

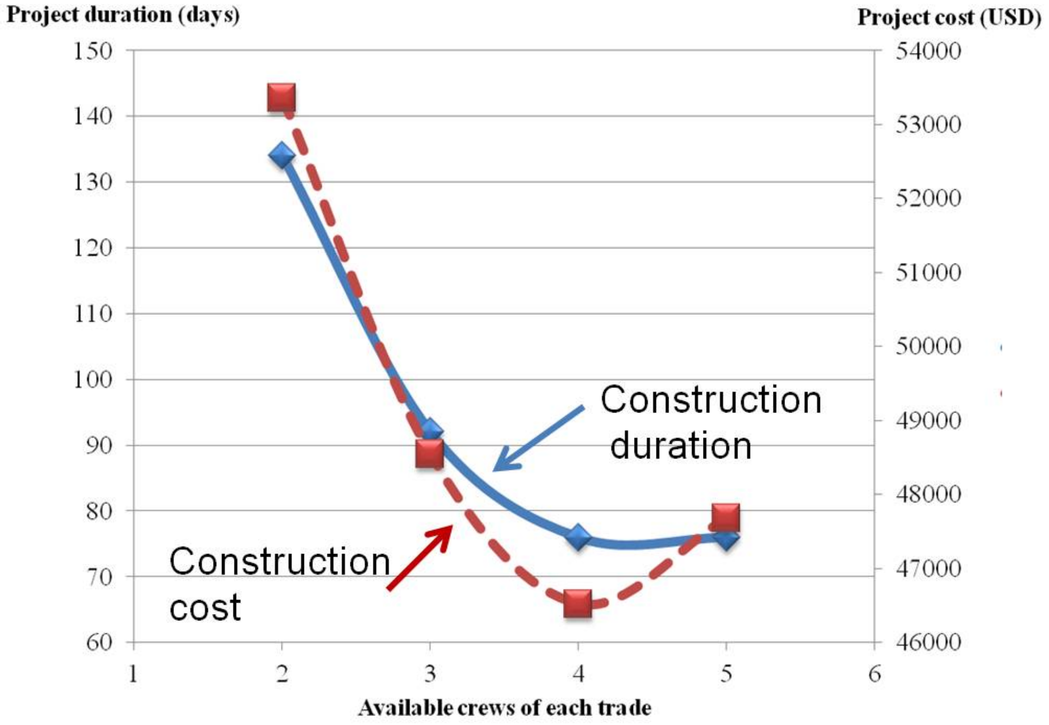


Figure 5-1. The construction duration and cost found by the GA-based method (Chapter 4) given different crew availability for a 20-room case.

5 References

Aalami, F. (1998). "Using method models to generate 4D production models." Ph.D. Dissertation, Stanford University, Stanford, CA.

Chapter 5

Akinci, B., Fischer, M., and Kunz, J. (2002b). "Representing work spaces generically in construction method models." *Journal of Construction Engineering and Management*, 128(4), 296-305.

Cornick, T. (1990). "Quality Management for Building Design." *Butterworth Architecture Management Guides*.

El-Rayes, K. (2001). "Optimum planning of highway construction under the A+B bidding method." *Journal of Construction Engineering and Management,* 127(3), 199-205.

Haddad, Z. (2011). "Information systems at CCC," Guest Lecture, Stanford University, Stanford, CA, December 14, 2011.

Ho, P., Fischer, M., and Cam, C. (2009). " Prospective validation of virtual design and construction methods: framework, application, and implementation guidelines." Stanford University, Center for Integrated Facility Engineering, Working Paper 123.

Kandil, R., and El-Rayes, K. (2006). "Parallel Genetic Algorithms for Optimizing Resource Utilization in Large-Scale Construction Projects." *Journal of Construction Engineering and Management,* 132(5), 491-498.

Kim, K., and Paulson Jr. B. C. (2003). " Agent-Based Compensatory Negotiation Methodology to Facilitate Distributed Coordination of Project Schedule Changes." *Journal of Computing in Civil Engineering,* 17(1), 10-18.

Maile, T., Fischer, M., and Bazjanac, V. (2007). "Building energy performance simulation tools - a life-cycle and interoperable perspective." Stanford University, Center for Integrated Facility Engineering, Working Paper 107.

Slaney, J. (1995). " Automated reasoning and exhaustive search: Quasigroup existence problems." *Computers & mathematics with applications,* 29 (2), 115-132.

Sugimori, Y., Kusunoki, K., Cho, F., and Uchikawa, S. (1977). " Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system." *International Journal of Production Research,* 15(6), 553-564.

Thabet, W. Y., and Beliveau, Y. J. (1997). "SCaRC: space-constrained resource-constrained scheduling system." *Journal of Computing in Civil Engineering,* 11(1), 48–59.

Vanhoucke, M. (2006). "Work continuity constraints in project scheduling." *Journal of Construction Engineering and Management,* 132(1), 14-25.

Vaus, D. d. (2001). "Tools for research design." *Research design in social research*, D. d. Vaus, ed., Sage Publications Ltd., London, UK, 17-33.

Xu, J., and Zhang, Z. (2010). "A fuzzy random resource-constrained scheduling model with multiple projects and its application to a working procedure in a large-scale water conservancy and hydropower construction project." *Journal of Scheduling*, online first.