MODULES MULTI-LIFT PLANNING ON INDUSTRIAL SITE: DEVELOPMENTS OF DECISION SUPPORT TOOLS

by

Mahmoud Farajmandi

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

Master of Science
in
Construction Engineering and Management

Department of Civil and Environmental Engineering
University of Alberta

Abstract

Properly planning module installation on industrial sites is a critical factor in ensuring that projects are delivered safely, on time, and within budget. In industrial modular construction, pre-fabricated modules are installed on site, in specific patterns, based on design documents. Depending on module size and weight, as well as crane availability, location, and configuration, different sizes of heavy-duty mobile cranes are used to safely pick, swing, and place modules for installation. High crane operating costs, number of options available to install the modules on site, and multiple crane-module technological constraints necessitate schedulers to spend weeks in a trial-and-error basis to prepare and improve module installation plans. A formalized framework for providing feasible, optimum installation plans will considerably minimize crane operation costs.

This thesis focuses on developing a novel framework (i.e., method, algorithm) for automating and improving module installation planning processes for modular construction on industrial construction sites with respect to number of crane relocations, number of crane reconfigurations, and total number of crane locations. Given the proposed framework, a decision tool is developed to facilitate the planning by ensuring that project constraints are satisfied while minimizing the crane operation cost. Two novel methodologies are presented in this thesis.

First, a heuristic-based methodology is proposed to build a module installation schedule iteratively by formalizing subject matter expert knowledge using heuristic rules. This methodology is implemented in a software prototype. A sample case study is provided to illustrate the calculation procedures and a practical case study is used to demonstrate the effectiveness of the developed tool.

Then, an artificial intelligence based Monte-Carlo Tree Search (MCTS) methodology is proposed where the optimum plan is searched for using biased sampling of the solution space. Based on this methodology, a decision support tool is developed for generating optimum plan. The same sample case study is used to demonstrate the procedure and calculation steps and features of the developed tool. It is found that the methodology efficiently discovers the optimum solution for the smaller scale problem. However, the MCTS-based method requires further development to be applied to large practical projects.

As a result of this research, the frameworks, along with their corresponding decision support tools, have been developed to automate on site module installation planning processes. The case studies investigated demonstrates that the heuristic-based rules efficiently minimize crane operation costs for large, complex projects. Although the current MCTS-based methodology is limited in its ability to formulate of module installation plans for large industrial construction projects, these limitations have the potential to be overcome.

Acknowledgment

The work I am presenting here would not be possible if it was not for the great people that have helped me during last two years. I would like to express my deepest gratitude to my supervisor, Dr. Simaan AbouRizk, for his continuous guidance and support. He has been a constant source of inspiration. I would also like to thank my thesis committee members, Dr. Ming Lu and Dr. Wei Victor Liu, for their interest in my research.

I would like to acknowledge PCL Industrial for providing me with data and facilitating my research. Specifically, I would like to thank Ulrich (Rick) Hermann for sharing his expertise, and for his patience and his desire to educate me along the way. Also, I am thankful to Dr. Hosein Taghaddos and Dr. Zhen Lei, who helped me by sharing their knowledge and vision for this research work.

Postdoctoral fellows Dr. Seungjun Ahn and Dr. Ming-Fung Francis Siu helped shape my research and constantly challenged me to higher standards by reviewing my work and providing me with feedback. I am very thankful for their time and effort.

Finally, I would like to thank my family for their support. I am grateful for my parents' encouragement and my brothers' and sisters' motivation during the last two years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACPO	Advance Crane Planning and Optimization program
ALPS	Automated Lift Planning System
CPM	Critical Path Method
E-GVRP	Equality – Generalized Vehicle Routing Problem
GVRP	Generalized Vehicle Routing Problem
MCS	Monte-Carlo Scheduling
MCTS	Monte-Carlo Tree Search
NP-Hard	Non-deterministic Polynomial Hard
OMC	Objective Monte-Carlo
PBBM	Probability to be Better than Best Move
PERT	Program Evaluation and Review Technique
RCPSP	Resource-Constrained Project Scheduling Problem
SME	Subject Matter Expert
TSP	Travel Salesman Problem
UCT	Upper Confidence bounds applied to Trees
VBA	Visual Basic for Application
VRP	Vehicle Routing Problem

Chapter 1: Introduction

1.1 Background

Scheduling construction projects on site can be a daunting task, requiring professional schedulers to spend weeks or months planning the project while consulting with different project stakeholders. Depending on project complexity and type, subject matter experts (SME) from multiple disciplines are often involved in establishing project requirements, activity precedence relationships, required resource types, and in determining the time required to safely deliver specified tasks. Projects are broken down into a series of tasks, and the precedence relationships of these tasks are established, the project planner can establish a project schedule that (a) meet project delivery deadlines, (b) satisfy project constraints and task precedence requirements, and (c) manage and level critical resource use throughout project duration. In addition, project schedulers have to investigate numerous execution details and sequencing options to minimize overall project makespan and cost.

Project scheduling techniques, such as the critical path method (CPM) and the Program Evaluation and Review Technique (PERT), are typically used to demonstrate step-by-step project schedules, illustrate resource usages, and to provide project duration forecasts. In CPM scheduling, activities' precedence and successor interdependencies are shown, and one can easily observe and analyze the effects of changes in individual schedule activities will impact interdependent activities. The activities for which a duration change will directly affect the overall project duration are said to be on the critical path. In PERT scheduling, the best- and worst-case scenarios for a given task are used to calculate the best estimate of task duration. While these techniques are very effective in showcasing the project plans and requirements, they are limited by their inability to integrate project constraint requirements into project schedule optimization. Project planners can use "what if" analyses to examine various scenarios using schedules built by CPM and PERT. However, performing a holistic project execution scenario check is not feasible for medium to large projects. Depending on the project type, specification and classification, different techniques and approaches are suggested to create improved schedule.

In industrial modular construction, prefabricated modules are sequentially installed on site according to module-specified patterns indicated in design documents. Different types of modules, such as pipe rack and equipment modules, are assembled in module yards. Module size and weight are dependent on both module type and level of modularization. Using mobile cranes, modules are picked up from trailers and are lifted into place for installation on site. Depending on module size and weight, as well as crane availability, location, and configuration, different sizes of heavy-duty mobile cranes are used to safely lift, swing, and place modules in designated locations for installation. This is done while factoring in crane-module technological constraints discussed in this research study (i.e., bottom-top module precedence relationships, neighbor module precedence relationships and module blocking precedence relationships). The high cost of operating these cranes demands that the schedulers to spend weeks improving crane operation schedules while still satisfying safety requirements and project constraints.

Mobile crane operation costs include mobilization, demobilization, foundation preparation, reconfiguration, relocation, and rental costs (Taghaddos et al. 2010). In planning module installation, larger cranes are used to maintain larger capacity and reachability, while smaller cranes are used to reduce leasing and supporting task costs. Planners balance the use of larger cranes, which minimizes foundation preparation, movement, and setup, with the use of smaller cranes, which minimize hourly crane rental and supporting tasks costs (Lin and Haas 1996). Typically, mobilization and demobilization costs are determined upon crane selection, while reconfiguration and relocation costs depend on the module installation plans. By minimizing frequency of crane reconfiguration, relocation, and the number of rigging changes during project execution, crane operation costs can be substantially reduced. On the other hand, any attempt to reduce crane operation costs should include evaluation of the feasibility of formulated module installation plans and satisfaction of technological constraints associated with module installation during project execution. Currently, schedulers manually sequence module installations based on a trial-and-error approach to ensure that the constraints are satisfied and a feasible module installation plan is developed.

The installation planning problem for industrial modules is a resource-constrained project scheduling problem (RCPSP). Alternative schedules are available for sequencing module installation using limited crane resources. The RCPSP is a non-deterministic polynomial hard (NP-hard) problem, i.e. there is no computationally feasible algorithm that guarantees that the problem can be solved to global optimum for a project with practical size and complexity. For example, formulation of a module installation plan for installation of 60 modules with 1500 feasible crane locations for lifting each module amounts to over 10^{210} solutions that require searching and examination.

The research presented in this thesis develops a novel framework for automating and improving module installation planning on industrial construction sites with respect to multiple criteria (i.e. number of crane relocations, number of crane reconfigurations and total number of ground locations). Two different methodologies to facilitate on-site module installation planning, based on heuristic rules or on an artificial intelligence Monte-Carlo Tree Search (MCTS), are proposed. Finally, to provide practitioners with decision support tools required for future module installation planning, these two methodologies are implemented into software.

1.2 Research Objective and Expected Contribution

The aim of this thesis is to develop a framework and algorithms that automate installation plan generation for module construction in industrial projects while minimizing on site crane operation costs. By examining the problem structure and comparing it to similar classes of problems in different research areas, appropriate algorithms and methods to solve this type of problem have been developed and are presented. The algorithm ensures that a feasible and (near) optimized installation sequence, including preferred crane location, type, and configuration, is chosen for a given project. The following objectives have been identified as means to achieve this aim:

1. Understanding the problem structure and dimensions as well as factors and decisions that can influence the final solution quality,
2. Understanding constraining factors in module installation scheduling and methods to avoid constraint violations;
3. Understanding and comparing similar problems that exist in literature and the approaches currently recommended to solve these types of problems; and
4. Developing a framework that formalizes an approach to determine feasible installation sequences for module installation to minimize crane operation cost, in which either a heuristic methodology or the Monte-Carlo Tree Search (MCTS) technique is used.

The completion of the objectives outlined above would result in the following contributions:

1. Determination of different problem aspects and factors that affect the cost of crane operation and module installation,
2. Formalization of an approach that ensures that only feasible module installation plans meeting all technological constraints are presented,
3. Development of a heuristic methodology that can be implemented to provide a good quality solution to the problem is developed, and

4. Investigation and presentation of the results of a MCTS method applied to the search for an optimal module installation plan.

These advancements and developments will result in the creation of an application that would provide a user-friendly tool for a planner to utilize for scheduling and sequencing module installation on industrial projects. The tool will shorten the time a planner needs to spend in scheduling module installation as well as provide a better quality solution compared to schedules currently selected by industry practitioners.

1.3 Research Methodology

In order to achieve the stated goals of this thesis, research was conducted in different phases. First, multiple subject matter experts were contacted, consulted and interviewed to understand the dimensions of the problem. The subject matter experts were asked to provide both the factors and constraints that are currently considered in planning module installation, as well as the decision-making processes that are currently followed by experts to devise installation plans. Furthermore, current and previous sample projects were reviewed to better understand the typical project size, layout, inputs and type of outputs that would be required for site wide installation planning.

Next, different aspects of mobile crane planning and scheduling that were previously covered in literature were reviewed. The past advancement and achievement in solving similar problems were studied, and their progress as well as shortcomings were considered. Furthermore, the literature related to similar problems in other research areas (e.g. operation research and job shop scheduling) were reviewed, and the similarities and differences of established problem types compared to our problem were recognized. Further, suggested approaches in solving these types of problems were reviewed and noted, and their limitations and restrictions were examined.

Finally, recognizing the gap between past advancements and the approaches required to solve this problem, two different frameworks and methodologies were developed and suggested. First, a heuristic rule-based method was established by formalizing the technological constraints in module installation scheduling and developing option ranking rules based on current practices and the desired solution quality. Next, a state-of-the-art artificial intelligence Monte Carlo Tree Search (MCTS) method was proposed and utilized to search the solution space for an optimum module installation plan.

1.4 Thesis Organization

The following chapters in this thesis are organized as follows:

Chapter 2 provides insight into different aspects of module installation planning on construction sites, as well as previous developments and advances in the field. It provides some background information on the types of constraints and limitations a planner would typically encounter when working on this type of problem. Then, an overview of current industry practices is provided. Next, similar type problem in literature is reviewed and proposed approaches to solve this type of problem are investigated. Finally, a gap analysis is provided, comparing the current state of the art solutions for these types of problem and the required approaches to prepare a feasible optimum module installation plan.

Chapter 3 focuses on developing a heuristic framework based on current industry practices. Once the application and importance of heuristic rules and knowledge-based automation tools in solving similar problems are explored, a heuristic framework is proposed to solve the problem at hand. The framework would consider the multiple factors that were recognized earlier in chapter 2 as major cost items, and heuristic rules are defined to minimize these major cost items. Further, a case study is presented to demonstrate the effectiveness and advantage of utilizing this new framework in planning module installation.

Chapter 4 provides an introduction to MCTS, and its merits for use in combinatorial type problems are laid out. The implementation details of MCTS to the problem are explained. The merits and the effectiveness of the proposed MCTS are shown through a case study. Finally, in this chapter, an application toolkit that was developed based on MCTS algorithm implementation is presented.

Chapter 5 contains the conclusion of this thesis as well as the academic and industrial contribution of the developed algorithms. Recommendations for future works are also stated in this chapter.

Chapter 2: Problem Statement and Literature Review

2.1 Introduction

In this chapter, the problem dimensions and constraints are defined in detail to begin. Different factors and how they impact solution quality are reviewed. The inputs available for solving the problem are listed and explained, and the desired outputs are described. Next, previous work and advances in different areas of crane selection and planning for industrial construction site are provided as a background, and specific attempts and shortcoming of previous studies to devise a tool to plan site-wide module installation are reviewed. Further, current industry practices in site-wide crane planning are detailed. Giving consideration to the problem description and dimensions, similar types of problems studied in the literature are reviewed, and their similarities and differences in relation to the module installation planning problem are explained. Finally, in this chapter, the gap between the current state of the art solutions in literature and the desired approach to solve the problem is explained and the paths followed in this thesis to fill this gap are suggested.

2.2 Problem Overview

2.2.1 Problem Description and Constraints

The module installation planning problem consists of scheduling and allocating limited crane resource(s) to installation of a set of modules by determining (i) the optimum crane configuration, (ii) crane location, and (iii) installation sequence for the duration of the project while meeting precedence relationship requirements. These different decision factors and constraints are explained next.

Each crane assigned to the project can have multiple configurations that dictate the crane capacity and reachability from any location on site. Furthermore, each mobile crane configuration is the combination of boom length, superlift type, superlift radius from crane center and superlift weight associated with the configuration. Figure 2.1 shows a crawler crane with three different configurations: (i) long boom length and no superlift, (ii) long boom length with wagon type superlift, and (iii) short boom length with wagon type superlift.

Given the details and configuration of the crane, feasible crane location coordinates are provided for each to-be-sequenced module as the feasible installation options. The selected crane and its associated configuration should be capable of lifting, swinging and placing the desired module from the specified crane location without exceeding the maximum allowable crane utility value. Note that these options are established based on previously developed software, such as the Advance Crane Planning and Optimization (ACPO) introduced by Hermann et al. (2010). ACPO or similar program provide potential installation options for each module and illustrate how individual option feasibility changes as on site module installation progresses. During sequencing module installation, the following precedence relationships should be respected:

The *bottom-top module precedence relationships* exist between the bottom module installation and top module installation. The finish-to-start precedence relationships are imposed between installing the bottom module and its top module. Note, when there are multiple levels of modules stacked on top of each other, a finish-to-start precedence relationship exists between installing the module at the bottom and the module directly stacked on top. This relationship between modules is established by reviewing site drawings and recording this information into databases.

The *neighbor module precedence relationships* dictate that any module belonging to a defined module group is not allowed to be installed between any two previously installed modules from the same module group. A group is defined as a straight run of modules either from East to West or from South to North. The constraint is shown in Figure 2.2. The middle module cannot be installed between the two previously installed or sequenced modules.

Module-blocking precedence relationships represent the fact that previously installed or earlier-sequenced module installation would eliminate some installation options for to-be-sequenced modules. Figure 2.3 demonstrates this precedence relationship. The left hand side of Figure 2.3 shows that the lifted module can be installed when the crane is positioned at a specific location. However, the right hand side of the same figure shows that the path for installing the lifted module from the same crane location would not be feasible if the upper module was installed in the preceding installation iteration (as circled in Figure 2.3). Lack of sufficient clearance between any part of a lifting crane and the installed/sequenced module could prevent and block installation of a specific to-be-sequenced module from specific crane locations using specific crane configurations. As such, the finish-to-start relationships for specific modules are imposed with respect to particular crane locations and crane configurations.

Multiple cranes on site cannot be located at the same location. Further, based on current practices when there are two or more cranes being utilized on the job site at the same time, a minimum distance (d_{min}) equal to the sum of the Tail Swing (T) dimension of the cranes plus five (5) feet should be allowed between the two cranes (Ulrich Hermann, Manager of Engineering at PCL Industrial, personal communication).

The goal of the proposed framework is to formalize the approach in determining the installation plan (including sequence, crane location and crane configuration) for module installation on industrial sites. This research study proposes two planning methodologies to produce feasible installation plans while crane operation costs are minimized and all the technological constraints discussed above are satisfied. First, a novel heuristic rule-based methodology is suggested, with the module installation plan built iteratively. Then, a MCTS methodology is proposed to search the solution space for an optimum module installation plan. The goal of the both methodologies is to generate feasible installation plans while minimizing the crane operation costs. In the next section, different cost factors that affect crane operation costs are briefly stated and explained.

Figure 2.1: Multiple crane configurations

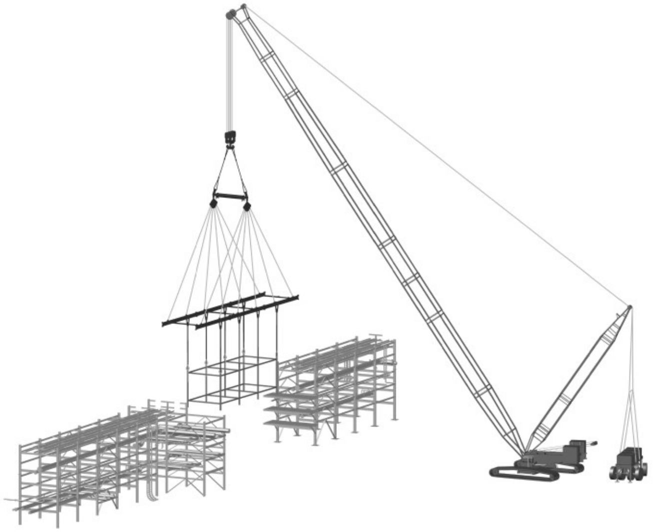


Figure 2.2: Neighbor module precedence relationships, the lifted module cannot be installed between two previously installed modules

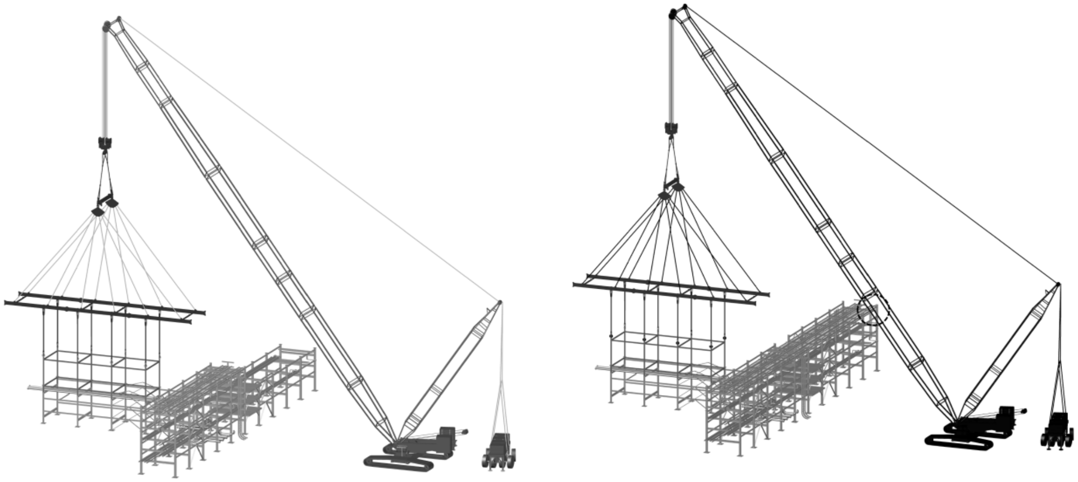


Figure 2.3: Module blocking precedence relationships

2.2.2 Cost Factors

There are multiple elements that can affect the overall mobile crane operation cost. In order to minimize the total cost of crane operation on site, multiple cost factors need to be considered. In this thesis, based on previous research work (Hornaday et al. 1993, Lin and Haas 1996, Taghaddos et al. 2011, Westover et al. 2014) and one–on-one interviews with subject matter experts, the following cost factors were recognized and considered in determining the heuristic rules and in evaluating the solution quality in MCTS approach. These are:

- Ground preparation cost,
- Mat installation/removal cost,
- Crane relocation cost,
- Crane reconfiguration cost,
- Rigging change cost, and
- Crane rental cost.

These cost factors are briefly explained in this section.

Ground Preparation Cost: Crane location preparation includes ground leveling and compaction tasks. The total cost of ground preparation at each location depends on the size of the area to be prepared. The size of the ground preparation at each location is determined by the crane types and configuration planned for use at each location for the duration of the project. In this thesis, the location of the crane is assumed to be square and the dimension of the crane location is calculated as 29 feet plus the largest superlift radius of the crane that planned to be at the specified location for the duration of the project based on the current practice on site (Ulrich Hermann, Manager of Engineering at PCL Industrial, personal communication,).

Mat Installation/Removal Cost: Depending on crane size, ground condition and the weight of the modules being lifted, one or two layers of mat are required underneath the crane for stability. Similar to ground preparation cost calculations, the cost of the mat installation and removal depends on the area of ground the mat is placed on. In this thesis, the size of the mat placement area is taken to be identical to the size of the ground preparation area explained in previous paragraph.

Crane Relocation Cost: The crane relocation cost entails crane operator cost as well as the cost of the crew needed to help moving move the crane from one location to another, if the required crane location needs to be changed for subsequent module installation. The crane relocation cost is only considered for crane movement distances that exceed a specified limit for each crane.

Crane Reconfiguration Cost: If the required crane configuration is different from the preceding crane configuration, the crane has to be reconfigured. The cost of crane reconfiguration depends on both the specific initial and final crane configuration. The cost should account for the assistant crane and crew as well as the period of time required to make the changes.

Rigging Change Cost: Each module requires a specific rigging arrangement dependant on the size, weight and configuration of the module. If rigging requirements for two subsequent modules are different, rigging change costs will be associated with the alteration. Generally, it costs more to change from one traditional rigging specification, with specific spreader bar and sling arranged in a specific way, to another traditional rigging specification compared to the changes required when using lift frame rigging (Westover et al. 2014). For traditional rigging, the cost would include the cost of the main crane, the assistant crane, and the crew required to make the changes. For lift frame rigging, only the cost of lift frame transportation from one site location to another is considered in this thesis.

Crane Rental Cost: The crane rental cost depends on the type of crane and the specific crane configurations that are used for the duration of the project. The crane rental period includes the duration the crane is required to complete all module installation and support tasks. In this thesis, the following assumptions are made based on interviewing subject matter experts: the crane takes one day for each module lift, one day is required for each major crane reconfiguration, one day is required for crane movement, and one day is needed for changing traditional rigging.

Figure 2.4 summarizes the inputs and the desired output for the installation planning algorithm. Most of the available input information is either stored in a database system or provided by user at the start. However, it is preferable to convert this data into a table in a database in order to effectively manage and utilize the large amount of information required for each project. In the remaining sections of this chapter, past developments and advancements in different areas of mobile crane planning are reviewed. Then, similar problems in other domains, notably in operation research, are noted. Compared with past research work, the research gap between the current advances and the desired outcomes are recognized. A path forward is then suggested to advance the current state of the art models for planning module installation on industrial sites.

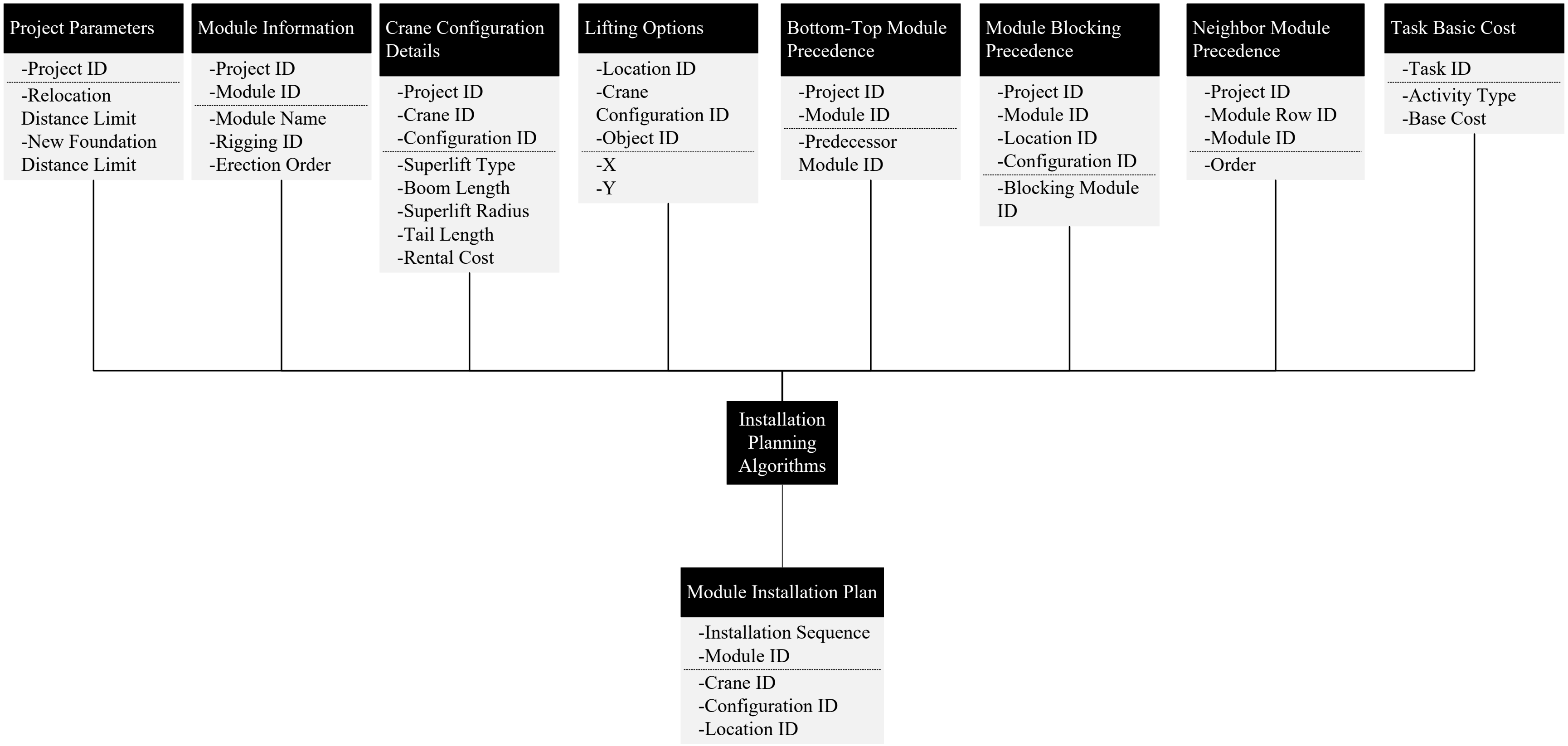


Figure 2.4: Overview of module installation planning input and output information flow

2.3 Literature Review

In this section, a general overview of past work and advances in the optimization and automation of different components of mobile crane planning on industrial construction site are provided. Further in this section, the current practices of industry experts are reviewed and explained. In Section 2.3.2, similar problems that appear in operation research literature are presented, and the similarities and differences to site wide crane planning are explained. The approaches suggested in the literature to solve these types of problems are detailed.

2.3.1 Current State of the Art in Heavy Lift Planning

To ensure lift safety and feasibility on industrial projects, multiple design and planning activities are needed. In the past, researchers have developed computer-aided planning tools to facilitate many of these decision-making processes. The design and planning activities include the selection of the rigging design and crane model, ground bearing pressure calculations, crane location and configuration assignments for each module or vessel, path planning, access planning, and formulating an overall lifting plan (Haas and Lin 1995, Lei et al. 2015). Much of the research focused on the automation of these mobile crane design and planning activities on industrial sites. For instance, Hornaday et al. (1993) and Al-Hussein et al. (2001, 2005) developed computer-aided systems to automatically identify potential crane locations based on crane capacity, lifting range, and the crane utilization percentage. Haas and Lin (1995), Reddy and Varghese (2002), Olearczyk (2014), Lei et al. (2013) and Lei et al. (2015) analyzed the lifting, swinging and placement for a single object, and automated clash detection based on site constraints and crane configuration. Lei et al. (2014) and Han et al. (2014) analyzed crane walking paths for instances where a crane picks up and travels with an object (e.g. modules, equipment, or vessels) before placing the object in its final position. Hermann et al. (2010) and Olearczyk et al. (2015) propose integrating the above analysis in an integrated software platform for preparation of engineered lift drawings and detection of potential conflicts on the job site while considering the crane capacity, object weight, rigging requirements and site constraints. In addition, industry has developed in-house planning tools. For example, the Automated Lift Planning System (ALPS) developed by Bechtel can be used to provide a visualization environment for each lift (William and Bennett 1996), or the 3D Lift Plan program developed by A1A Software (2009) can be used to select crane size and configuration.

In addition, some researchers have attempted to automate the planning and scheduling of multiple lift sequences. Lin and Haas (1996) proposed an interactive platform that allows the selection of an optimum schedule for a single crane using linear programming. Lin and Haas (1996) have also proposed a semi-automated approach for the formulation of a lifting schedule for one crane. However, they did not consider crane reconfiguration or site constraints (e.g. top-bottom module relationships) that can have an impact on project cost and duration. Reddy et al. (2007) presented a multiple lift planning tool that visualizes the simulation of an installation schedule of heavy vessels in accordance with particular crane types and site locations. Taghaddos et al. (2011) optimized crane lift schedules by using an ascending auction protocol. When computer-aided planning tools are not utilized, planning multiple heavy lifts in a congested industrial site is complicated, error-prone and time consuming (Taghaddos et al. 2010, Olearczyk et al. 2015). In current practice, the practitioners plan the lifting sequence in a semi-automatic manner by a heuristic rule approach (e.g. minimizing the number of crane relocations). The solutions are manually determined using a trial and error method based on the experience and expertise of the planners (Hermann et al. 2010). In an iterative process, the subject matter expert (SME) chooses the most critical modules (in terms of weight and size) to be processed and determines a crane location for the selected modules which could be used in future module installation. If any previously established location could be utilized to lift the current module, it would be selected over a new crane location. This process is repeated until a feasible crane schedule for the project is determined.

2.3.2 Similar Types of Problems in Operation Research

The traveling salesman problem (TSP) and vehicle routing problem (VRP) are other examples of similar combinatorial type problems in the literature with NP-Hardness. The TSP can be defined as a special case of the VRP, and the VRP matches the module installation planning problem better. For this reason, the VRP is reviewed, the past advances for solving the VRP are discussed, and similarities and differences to the problem described in this thesis are noted.

The VRP can be defined as the problem of servicing a set of customers with known locations and requirements using a single depot (starting/finishing point) and vehicles of known capacities. Variations of this basic problem, such as the VRP with time windows and the generalized VRP, will be reviewed later. Similar to

RCPSP, smaller cases of VRP have been solved using exact algorithms (Laporte and Nobert 1986). Some researchers have used heuristic methods to solve larger-sized VRP (Gendreau et al. 1994, Laporte et al. 2000). Furthermore, metaheuristic algorithms have been suggested as a means of improving heuristic solutions and escaping local optimum solutions. Examples of metaheuristic algorithms that are used to solve the VRP include genetic algorithms, simulated annealing, and ant colony algorithms (Baker and Ayechew 2003, Osman 1993, Bell and McMullen 2004).

The Equality-Generalized Vehicle Routing Problem (E-GVRP), introduced by Ghiani and Improta (2000), is the closest match to the module installation planning problem found in operation research literature. The E-GVRP is an extension of VRP. The objective of E-GVRP is to design optimal service routes, which are subject to vehicle capacity restrictions, from a given depot to a number of predefined, mutually exclusive and exhaustive node-sets (clusters) (Pop et al. 2010). The vehicle should only visit one node within each node cluster before returning to the depot. Since the problem was introduced in 2000, there has been very limited research on the subject. Pop et al. (2009) and Pop et al. (2010) proposed using an ant colony system and a genetic algorithm to solve the GVRP, respectively. Bektas (2011) proposed a branch and cut algorithm to solve the GVRP. Ha et al. (2014) presented a metaheuristic algorithm to solve the GVRP to obtain a near-optimal solution.

In comparison to the problem statement presented in Section 2.2.1, there are some differences between the structure of the module planning problem and the E-GVRP.

In the E-GVRP definition, all clusters are available for a visit at all times, while in the module planning problem the module availability for installation changes dynamically as predecessors are scheduled (i.e., according to the bottom-top module precedence relationship and neighbor module precedence relationship). In addition, the node-set (i.e., crane location/configuration) that can be used to install each module changes dynamically depending on the previously installed module (i.e., the module blocking precedence relationship). This is in contrast to the classic definition of E-GVRP, where the availability of nodes within each cluster does not change. Finally, while in E-GVRP the node-sets are mutually exclusive clusters, in module installation planning, the crane locations and configurations are not necessarily mutually exclusive to one module, and there are different crane locations and configurations that can be used to lift multiple modules.

2.4 Research Gap and Path Forward

The module planning problem defined above is a combinatorial type problem with multiple technological constraints where the objective is to devise a feasible installation plan for which the cost of crane operation is minimized. The size of the project, however, could obstruct the possibility of approaching this problem by carrying out an exhaustive search. For a medium size project with 60 modules each having approximately 1,000 potential crane locations, there can be as many as 8.3×10^{261} ($1000^{60} \times (60-1)!$) options to search through. Reviewing the previous work and advances, both in the crane planning literature and in similar problems (e.g. RCPSP and VRP) in operation research, have shed light on some general approaches that can be used in solving this problem. However, these approaches do not provide or suggest a comprehensive method and framework that can efficiently be used to prepare a good-quality plan that takes into account all the project constraints. In the existing body of knowledge, there is no formalized approach to determine the sequence for module installation.

Given the problem scope and difficulty, the goal of this research was to find, discover or develop an algorithm and framework that automates the search and building of a near optimum industrial site overall module installation plan for the medium- to large-size problem, with all project-associated technological constraints embedded in the suggested solution. This research builds on previous work of Al-Hussein et al. (2001, 2005) and Hermann et al. (2010), which identify potential lifting options for individual objects, that of Lei et al. (2013) and Lei et al. (2015), which analyze clash detection automation of lifting single objects, and proposes a framework for site-wide installation planning capable of considering multiple options and constraints. Further, it was concluded from past work on similar problems that depending on the solution accuracy required and the time and resources the end user is willing to spend to find a solution, heuristic (e.g. priority-based) or metaheuristic (e.g. genetic algorithm) methods can be used to find an appropriate near-optimal solution to the problem. Moreover, recent developments in machine learning techniques and artificial intelligence methods have provided new opportunities to approach these combinatorial type problems.

In Chapter 3, the applicability and benefits of using heuristic methods are reviewed. Then, a novel planning methodology is proposed to produce feasible installation plans for modular construction. The proposed approach uses the heuristic rules to formulate a module installation plan while minimizing cost and ensuring that technological constraints are satisfied. Due to the complicated nature of iterating the module installation sequence, the development of a decision support tool for automatically generating the module installation plan is essential to reduce the planning time and effort. A software system prototype is also presented in this thesis to automate the iterative process. The ease of use and high solution quality provided by the developed framework and semi-automated tool, particularly when compared to current industry practices, is illustrated by the use of a case study.

Chapter 4, noting the deficiencies of the heuristic method proposed in Chapter 3, the use of an artificial intelligence optimization and solution searching method, namely the Monte Carlo Tree Search (MCTS), is proposed. This MCTS-based method is then implemented in an automation tool to find a good-quality solution to the complex module installation problem. The advantages and disadvantages of the MCTS methodology for module installation planning are discussed.

Chapter 3: Heuristic Based Decision Support Tool for Module Installation Planning

3.1 Introduction

In this chapter, the importance and advantages of automating the planning process using heuristic-based rules are discussed. Then, a novel multi-stage heuristic methodology is proposed to determine the module installation sequence for industrial projects based on (i) a list of modules, (ii) the available crane, and (iii) the configurations of the available crane (i.e., boom length, superlift type, and superlift weight). This last variable (iii) takes into consideration the crane-module technological constraints. A sample case study is used to demonstrate the proposed concepts and calculation steps. A practical case study is used to illustrate the prototyped decision support tool developed to provide the plan automatically in practical settings. It is shown that the software system can significantly reduce the planning time and effort required to devise a feasible module installation plan.

The remaining of this chapter will be as follow:

Section 3.2 provides an overview of similar work in using heuristics to plan and schedule difficult combinatorial type problems and developing automation tools for construction task planning.

Section 3.3 describes the proposed heuristic methodology in detail. In this section, the required input information is first listed. Then, detailed step-by-step processes are described. Finally, the expected output information from this process is stated.

Section 3.4 provides a small case study where the step-by-step processes outlined in Section 3.3 were used to solve the problem. Detailed calculations are shown for two iterations.

Section 3.5 shows the result of the application of this methodology to a practical case study. It is shown that a large-sized project can be planned in a reasonable time using the prototype application developed based on the proposed methodology.

Section 3.6 summarizes the chapter and provides the conclusion and direction for future advancement.

3.2 Heuristic Optimization and Automation Tools in Planning

Heuristic algorithms have been commonly proposed and used in literature to solve complex large NP-hard combinatorial type problems such as RCPSP and VRP. Kolisch and Hartmann (1999) have acknowledged that heuristic procedures are essential in solving large, practical, NP-hard problems. The heuristic procedure has been proposed to solve RCPSP (Boctor 1990, Kolisch 1996, Kolisch and Hartmann 2006) and VRP (Laport et al. 2000, Pisinger and Ropke 2007) in situations where the exact method fails to provide an optimal solution in a reasonable timeframe.

The uses of heuristic knowledge-based approaches to plan construction tasks are also common. Knowledge-based schedule generation tools are developed and recommended in construction to facilitate installation planning and eliminate errors by incorporating automated mechanisms to check for different project constraints. Examples of knowledge-based tools are reported in work by Navinchandra et al. (1988), Shaked and Warszawski (1995), and Mikulakova et al. (2010). Applications of these tools can be found in building construction (Koo et al. 2007; Chen et al. 2013; Dong et al. 2013), offshore platform installation (Hendrickson et al. 1987), and bridge construction (Wu et al. 2010).

While there are many benefits to fully automating the generation of the installation plan (Morad and Beliveau 1991, Dong et al. 2013, Chua et al. 2013), practical, knowledge-based tools, where the user maintains some control over the final decision, would be more appropriate in the short term (Chevallier and Russell 1998). Waly and Thabet (2003) have concluded that the construction industry lacks the planning tools that allow users to apply their creativity and experience so as to make a knowledgeable decision. Shah (2014) proposed a decision-support tool that allows the users to select and change the equipment and site conditions in order to observe the impact on the project schedule.

As discussed in Chapter 2, past developments in the planning automation of site-wide modular construction using mobile cranes is limited, and does not provide industry practitioners a solid framework or tool to plan module installation for industrial projects. The current practice of planning modular installation projects manually based on a trial-and-error approach is a time-consuming and error-prone process. Knowledge-based tools involving user inputs are valuable for reducing the error and effort involved in formulating the module installation plan. As such, this chapter presents an automatic tool that assists planners in sequencing module installation while also allowing selection of the preferred installation sequence. By using the proposed methodology and the automation tool, a module installation plan with the minimum crane operation cost can be generated in a reasonable time frame. The proposed new methodology is presented in the next section.

3.3 Proposed Methodology

In this section, a novel multi-stage heuristic method for planning the module installation sequence on industrial projects is proposed. The proposed approach produces a module installation sequence for modular construction based on (i) a list of modules, (ii) the rigging requirements for module installation, (iii) the available crane, and (iv) available crane configurations (i.e., boom length, superlift type, and superlift weight), given the crane/module technological constraints that were discussed in Chapter 2. To generate the solution, heuristic rules are proposed to determine possible installation sequences factoring in feasible installation options. Based on the formalized approach, the module installation plan minimizes the number of crane locations, crane relocations, crane reconfigurations, and the crane moving distance. The following subsections discuss the input, process, and output of the proposed approach.

3.3.1 Input: Module Layout, Crane Details and Crane/Module Technological Constraints

Various project details and constraints are necessary to properly generate a module installation plan. The following inputs are minimum requirements: (i) the feasible crane configurations, (ii) the feasible crane location coordinates associated with the individual modules, and (iii) the module installation precedence relationships. As discussed in Section 2.2.1, the feasible crane configurations and crane locations for each module are determined using some of the tools previously developed. The precedence relationships (also discussed in Section 2.2.1) are: (i) bottom-top module precedence relationships, (ii) neighbor module precedence relationships, and (iii) module blocking precedence relationships. This input information is assumed to be available in a database, or in a format that can easily be converted to a database.

3.3.2 Process for Formulating a Feasible Module Installation Plan using Proposed Heuristic Rules

Given the inputs, a feasible plan for module installation is formulated based on the proposed iterative procedure. The process consists of (i) feasible solution generation, (ii) solution ranking, and (iii) solution selection using the proposed heuristic rules. Figure 3.1 shows the methodology flowchart for the overall process.

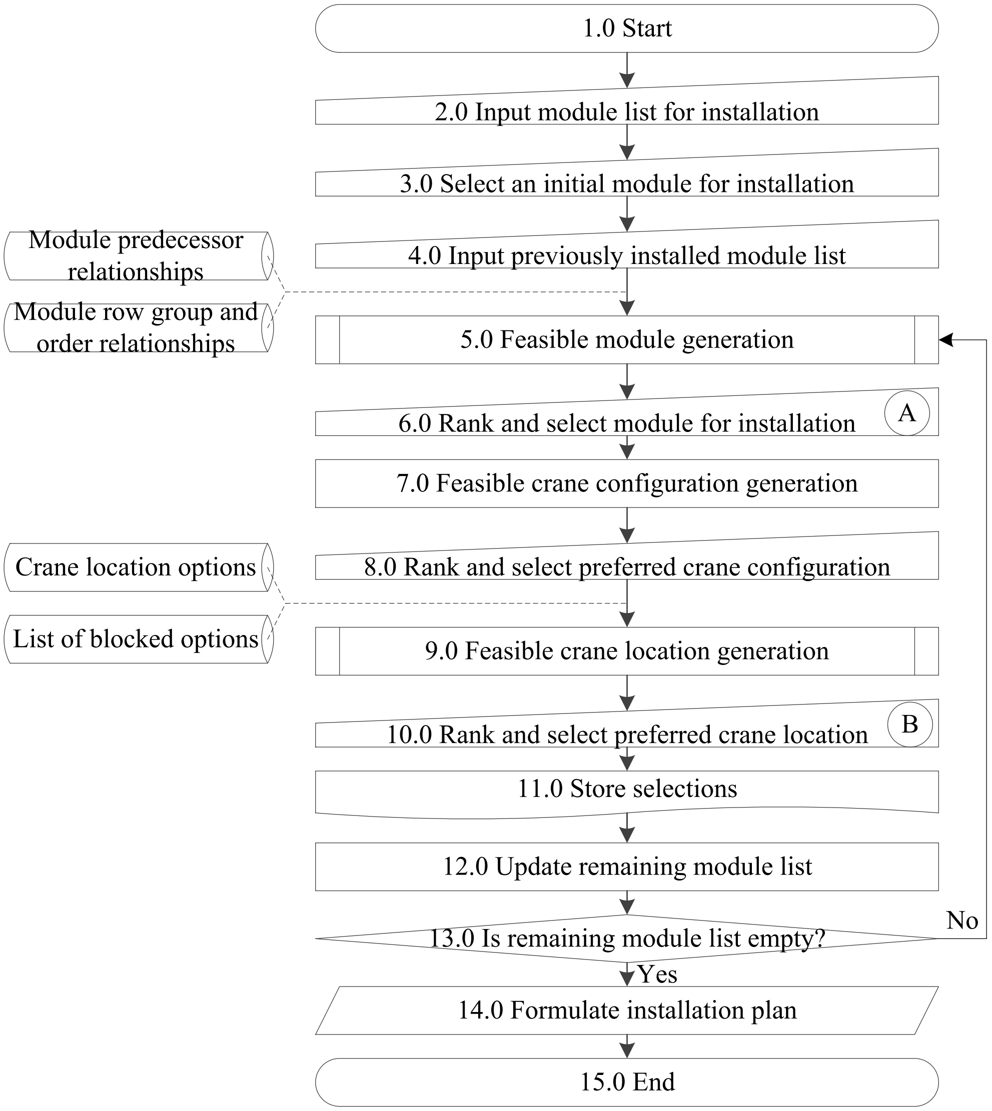


Figure 3.1: Overall flowchart for heuristic-based methodology for development of module installation plan

The process steps are detailed as follow:

1.0 Start: Start planning the module installation sequence for a modular construction project.

2.0 Input module list for installation: Select and list the modules for installation planning from project modules list.

3.0 Select an initial module for installation: Specify the first module for installation in Installation Iteration 1.

4.0 Input the previously installed module list: Select the installed modules from the module list for the project.

5.0 Feasible module generation: Generate a list of the to-be-installed modules, taking into account the module precedence relationships and the list of previously installed modules. The list of previously installed modules is initially identical to the installed module list specified by the planner. However, the module list is updated during later iterations by adding the sequenced modules to the list. The modules are selected based on three filtering processes: (1) previously installed modules are eliminated, (2) the modules with bottom-top module precedence relationship are excluded if the predecessor module (i.e., bottom module) is not installed, and (3) the modules violating the neighbor module precedence relationship are omitted.

6.0 Rank and select the module for installation: Rank the modules in the to-be-installed module list prepared in Step 5 based on multiple criteria, and select the module with the lowest ranking for installation. The flowchart in Figure 3.2 demonstrates the heuristic ranking rules (the numbers on the flowchart are chosen in a way to demonstrate the importance of each factor in relation to each other). The heuristic rules for ranking are as follow:

    6.1 Check if the previous crane location and configuration can be used to lift the module. If it can be reused, the project cost can then be reduced in terms of the crane foundation preparation cost, crane relocation cost, and crane reconfiguration cost,

6.2 Check if the same crane location can be reutilized by changing the crane configuration. If the previous crane location can be reused, the project cost can be reduced in terms of the crane foundation preparation cost and crane relocation cost,

6.3 Rank the modules with neighbor precedence relationships based on how close they are located to the edge of the group. This allows the crane to install the modules starting at one end of the construction site and moving toward the other end of the construction site,

6.4 Rank the modules by calculating the net number of available options for installing each module. The net number of options is determined by subtracting the number of blocked options (as the result of imposing module blocking precedence relationship) from the total number of available options. This gives higher installation priority to the to-be-installed module with the least number of options before exhausting all its available options.

Once the modules are ranked based on the criteria stated above, the module with the lowest ranking value is selected for installation.

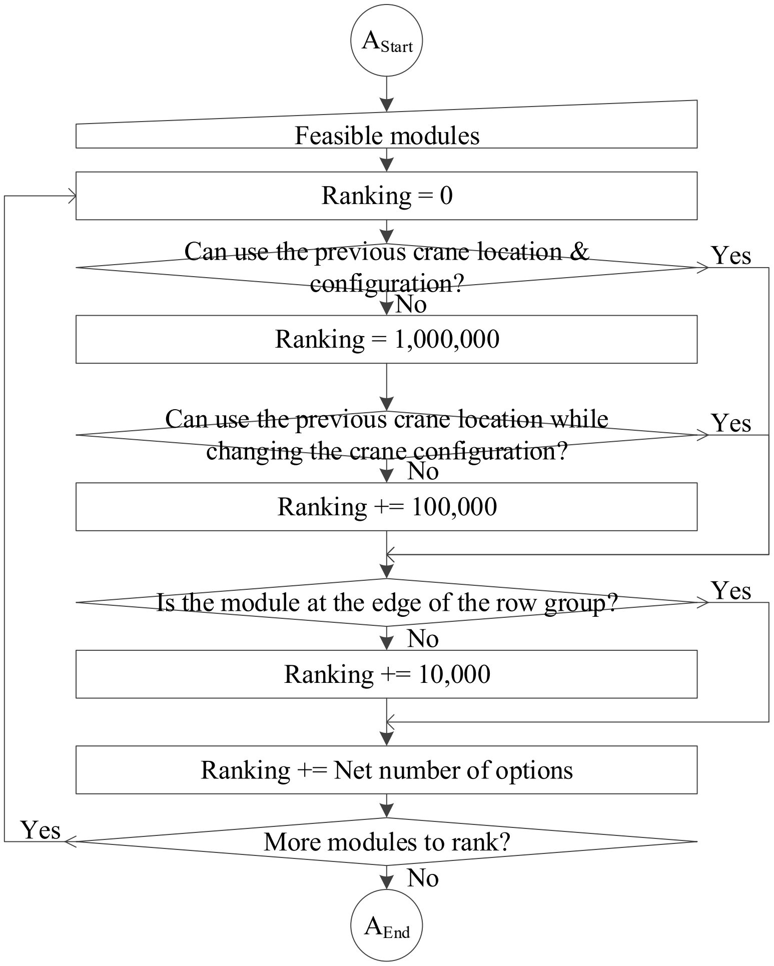


Figure 3.2: Ranking modules flowchart for heuristic-based methodology

7.0 Feasible crane configuration generation: List the feasible crane configurations for installing the selected module.

8.0 Rank and select crane configuration: Rank the feasible crane configurations based on two criteria. The ranking criteria are: (1) check if the previous crane configuration can be reused – a lower rank value is given if the crane configuration can be reused, and (2) check for how many modules the crane configuration can be utilized for installing the to-be-sequenced modules, if the previous crane configuration cannot be reused. The crane configuration that can be utilized most for future module installation is given a lower ranking. As a result, the crane configuration with the lowest rank is selected.

9.0 Feasible crane location generation: Prepare the location list for the selected module and crane configurations, taking into account the feasible crane lifting options and the module-blocking precedence relationships. The two filtering processes are: (i) eliminate options based on the selected modules in Step 6 and the selected crane configuration in Step 8, and (ii) remove the infeasible crane lifting options by considering the module-blocking precedence relationships. The result is a list of feasible crane locations that can be utilized to install the selected module.

10.0 Rank and select the crane locations: Rank the crane locations from Step 9 based on the ranking process flowchart, as shown in Figure 3.3 (the ranking numbers are chosen based on the importance of each factor and to avoid any interference of factors number with each other). The crane locations are ranked based on multiple criteria, as outlined below:

    10.1 Check if the current location is identical to the previously used crane location. Reutilization of an identical location will eliminate the crane movement, and it will allow the reuse of the existing ground preparation,

    10.2 Check if the crane location matches any previously utilized crane locations. This rule prioritizes a crane location that would not incur any additional cost of ground preparation, 10.3 Evaluate how many times a crane location could potentially be utilized for installing the remaining to-be-sequenced modules. This criterion would minimize the cost for preparing new crane locations,

10.4 Evaluate how many instances a crane location can be used for lifting both sequenced modules and to-be-sequenced modules throughout the project duration. This provides an alternative option for lifting previously installed modules if the original selected option could not be utilized (for any reasons),

10.5 Measure the crane moving distance from its previous crane location to the new crane location. This ensures that crane movements are minimized, and

10.6 Evaluate whether the new crane location has the same easting or northing as previous crane locations. The mat placements and crane movements are simplified if cranes are moved in one direction on site.

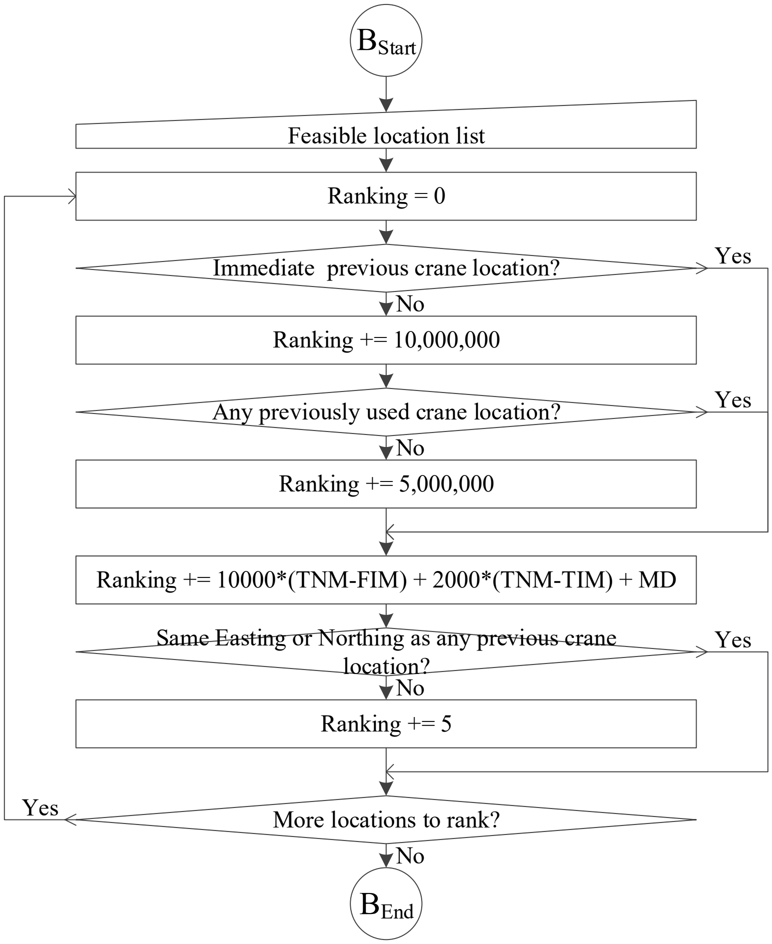
Once the crane locations are ranked based on above criteria, the location with the lowest ranking is selected.

11.0 Store selections: Store the selected module, crane configuration, and crane locations as part of the installation plan for this installation iteration.

12.0 Update remaining module list: Move the selected module from the to-be-sequenced module list to the sequenced module list.

13.0 Is the remaining module list empty: Check if all the modules have been sequenced. If not, repeat Steps 5 through 12.

14.0 Present the installation plan: Once all module installations are sequenced, the final installation plan is presented.

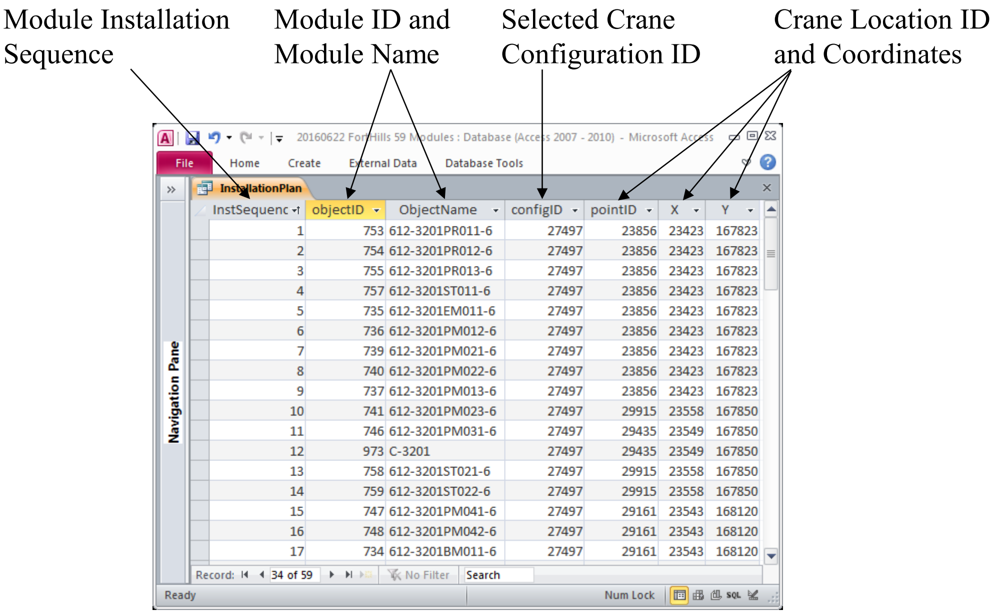


TNM = Total Number of Modules" in the project, FIM = "Future Installable Module" from the same crane location, TIM = "Total Installable Module" from the same crane location, and MD = "Moving Distance" of the crane from the last assigned location.

Figure 3.3: Ranking crane location flowchart for heuristic-based methodology

3.3.3 Output: Module Installation Plan

The output of the proposed methodology is a module installation plan. The installation plan specifies the installation sequence, as shown in Figure 3.4. In addition, the crane configuration and crane location have been determined for installing each module. As a result of the provided module installation plan, the precedence relationships for installing the modules are satisfied while the costs of crane foundation preparation, crane relocation, and crane reconfiguration are minimized for installing the modules.



InstSequenc	objectID	ObjectName	configID	pointID	X	Y
1	753	612-3201PR011-6	27497	23856	23423	167823
2	754	612-3201PR012-6	27497	23856	23423	167823
3	755	612-3201PR013-6	27497	23856	23423	167823
4	757	612-3201ST011-6	27497	23856	23423	167823
5	735	612-3201EM011-6	27497	23856	23423	167823
6	736	612-3201PM012-6	27497	23856	23423	167823
7	739	612-3201PM021-6	27497	23856	23423	167823
8	740	612-3201PM022-6	27497	23856	23423	167823
9	737	612-3201PM013-6	27497	23856	23423	167823
10	741	612-3201PM023-6	27497	29915	23558	167850
11	746	612-3201PM031-6	27497	29435	23549	167850
12	973	C-3201	27497	29435	23549	167850
13	758	612-3201ST021-6	27497	29915	23558	167850
14	759	612-3201ST022-6	27497	29915	23558	167850
15	747	612-3201PM041-6	27497	29161	23543	168120
16	748	612-3201PM042-6	27497	29161	23543	168120
17	734	612-3201BM011-6	27497	29161	23543	168120

Figure 3.4: Sample module installation plan as the output of the proposed methodology

Note that the optimality of the sequence is dependent on the initial module which is selected in Step 3. In practice, the first module to be installed is determined by the project planners based upon module delivery schedules, criticality of the modules, or SME experience. It may be beneficial to generate the module installation plan by setting different starting points for the first to-be-sequenced module in order to achieve solutions with global optimality.

3.4 Sample Case Study

In this section, a case study example is used to explain the calculation procedures of the proposed methodology in order to formulate the module installation plan. Figure 3.5 shows the postulated site layout for the module installation. In this problem, there are 8 modules to be sequenced using one of 9 available crane locations. The one crane assigned to this project can take any of the two crane configurations described in the next section.

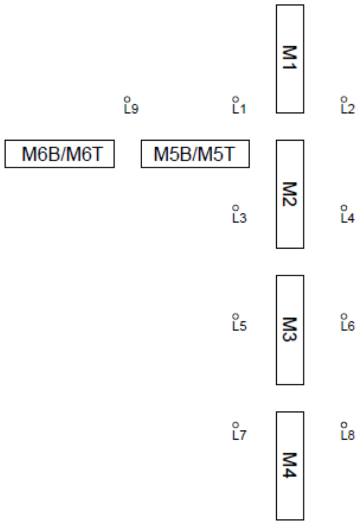


Figure 3.5: Site layout for sample case study with 8 modules and 9 potential crane locations

3.4.1 Input

Table 3.1 shows that bottom-top module precedence relationships exist between Modules M5B and M5T, and between M6B and M6T. Table 3.2 shows the neighbor module precedence relationships among the modules. M1, M2, M3 and M4 are classified as a group of modules with neighbor module precedence relationships (Group G1); and M2, M5B and M6B are classified as another group

(G2) with neighbor module precedence relationships. The Module Order in each group represents the module location with respect to other modules in the same group. Modules with subsequent order numbers are immediate neighbors. For example, M2, with Order 2, is the immediate neighbor of M1 with Order 1 as well as M3 with Order 3 in G1 (Group 1). Table 3.3 demonstrates three inputs: (i) all feasible crane configurations for lifting a particular module at each location, (ii) all feasible crane location options for lifting each single module, and (iii) the module blocking precedence relationships associated with each lifting option for each module. For example, Location L1 is disallowed for lifting M2 with crane configuration C1, because the previously installed M1 or M5 would block its lifting path. The option of using L1 to install M2 is therefore eliminated.

Table 3.1: Bottom-top precedence relationships for the sample case study

Module	Predecessor
M1	–
M2	–
M3	–
M4	–
M5B	–
M5T	M5B
M6B	–
M6T	M6B

Table 3.2: Neighbor module precedence relationships for the sample case study

Module	Group	Order	Group	Order
M1	1	1	–	–
M2	1	2	2	3
M3	1	3	–	–
M4	1	4	–	–
M5B	–	–	2	2
M6B	–	–	2	1

Table 3.3: Module blocking precedence relationships for the sample case study

Module	Crane Configuration	Crane locations	Blocking modules
M1	C1	L1	–
M1	C2	L1	–
M1	C1	L2	–
M1	C2	L2	–
M2	C1	L1	M1, M5T
M2	C1	L2	M1
M2	C1	L3	–
M2	C2	L3	–
M2	C1	L4	–
M2	C2	L4	–
M3	C1	L3	M2
M3	C1	L4	M2
M3	C1	L5	–
M3	C2	L5	–
M3	C1	L6	–
M3	C2	L6	–
M3	C1	L7	M4
M3	C1	L8	M4
M4	C1	L7	–
M4	C2	L7	–
M4	C1	L8	–
M4	C2	L8	–
M5B	C1	L1	–
M5B	C2	L1	–
M5B	C1	L2	M1, M2
M5B	C1	L3	–
M5B	C2	L3	–
M5B	C1	L9	–
M5T	C1	L1	–
M5T	C2	L1	–
M5T	C1	L2	M1, M2
M5T	C1	L3	–
M5T	C2	L3	–
M5T	C1	L9	–
M6B	C1	L1	M5T
M6B	C1	L9	–
M6T	C1	L1	M5T
M6T	C1	L9	–

3.4.2 Process Details

Given the above input data, the methodology process outlined in Section 3.3 is used to formulate the module installation plan. The detailed calculations are as follow:

1.0 Start.

2.0 Input module list for installation: {M1, M2, M3, M4, M5B, M5T, M6B, M6T}.

3.0 Select an initial module for installation: {M1}.

4.0 Input previously installed module list: {Null}.

Installation Iteration #1

5.0 Feasible module generation:

 5.1 Eliminate previously installed module: {Null}, since this is first iteration.

 5.2 Enforce the bottom-top module precedence relationships: {M5T, M6T} are eliminated from feasible module installation list.

 5.3 Enforce the neighbor module precedence relationships: {Null}, since no module associated with G1 or G2 is installed.

 5.4 The output feasible module list is {M1, M2, M3, M4, M5B, M6B}.

6.0 Rank and select the module for installation:

 6.1 Select the first module in the list: {M1}.

 6.2 Initiate the selected module ranking: $\text{Rank}_{M1} = 0$

 6.3 Check whether the previous crane location or configuration can be used: {Null}, since no previous crane location or configuration is available at the current stage.

6.4 Select “No”, and Rank_{M1} = 1,000,000

6.5 Check whether the previous crane location can be used by changing the crane configuration: {Null}, since no previous crane location or configuration is available at current stage.

6.6 Select “No” and Rank_{M1} = 1,000,000 + 100,000 = 1,100,000

6.7 Check whether the module is at the edge of a straight run group: {M1, G1}.

6.8 Add the number of options to the ranking number: Rank_{M1} = 1,100,000 + 4 = 1,100,004

6.9 Move to the next module.

Step 6 repeats for M2, M3, M4, M5B and M6B. The results are shown in Table 3.4. While M6B has the lowest ranking value, M1 is selected as the first module for installation in Step 3.

Table 3.4: Module ranking during the first iteration

Module	Ranking
M1	1,100,004
M2	1,100,006
M3	1,110,008
M4	1,100,004
M5B	1,110,006
M6B	1,100,002

7.0 Feasible crane configuration generation process: {C1, C2}.

8.0 Rank and select crane configuration:

8.1 Check if the configuration is identical to the previous crane configuration: “No”, since there is no previous crane configuration.

8.2 Check for how many modules the crane configuration can be utilized for installing the to-be-sequenced modules: Rank_{C1}= Total number of modules – Number of feasible modules using C1 = 8 - 8 = 0.

Repeating Step 8 for C2 would result in Rank_{C2} = 2. Since the ranking value of C1 is lower, C1 is selected as the crane configuration for installing M1.

9.0 Feasible crane location generation process:

9.1 Filter crane locations based on the selected module and crane configuration: {L1, L2}, using the list of options provided in Table 3.3.

9.2 Remove the blocked options based on the module blocking precedence relationships: {Null}, since there is no module previously installed.

10.0 Rank and select the crane locations:

10.1 Select the first location in the list: {L1}

10.2 Initiate ranking: Rank_{L1}= 0

10.3 Is this location the same as the immediate previous crane location: Select "No", since this is Installation Iteration 1.

10.4 Ranking update: Rank_{L1} = 10,000,000.

10.5 Is this location the same as any of the previously utilized crane locations: Select "No".

10.6 Ranking update: Rank_{L1} = 10,000,000 + 5,000,000 = 15,000,000.

10.7 Ranking update based on usability of the crane location:

10.7.1 How many times the location can be reused for installing future modules: Rank_{L1} = 15,000,000 + 10000 × (8-6) = 15,020,000

    - 10.7.2 How many times the location can be used for all modules: $\text{Rank}_{L1} = 15{,}020{,}000 + 2000 \times (8\text{-}6) = 15{,}024{,}000$
    - 10.7.3 Moving Distance (MD): MD = 0
  - 10.8 Does this location have an Easting or Northing coordinate similar to the previous crane location: Select "No"
  - 10.9 Ranking update: $\text{Rank}_{L1} = 15{,}024{,}005$
  - 10.10Move to the next location.

Repeating Step 10 for L2 would result in $\text{Rank}_{L2} = 15{,}048{,}005$. Since the ranking value of L1 is lower, L1 is selected as the crane location for installing M1.

11.0 Store solution: The selected M1, C1 with crane location L1 is stored for Installation Iteration 1.

12.0 Update remaining module list: M1 is moved from the to-be-sequenced module list to the previously installed module list.

13.0 Is there a module in the to-be-sequenced module list for installation: Select "Yes", then repeat Steps 4 to 12 until the iterations for all other 7 modules are processed.

Installation Iteration #2

5.0 Feasible module generation:

  - 5.1 Eliminate the previously installed module: {M1} is removed, since it was sequenced in Installation Iteration 1.
  - 5.2 Enforce the bottom-top module precedence relationships: {M5T, M6T} are eliminated from the feasible module installation list.

5.3 Enforce the neighbor module precedence relationships: {M3, M4}, since M1 (which is associated with G1 in Table 3.2) is already sequenced, M3 and M4 cannot be sequenced until M2 is sequenced.

5.4 The output feasible module list: {M2, M5B, M6B}.

6.0 Rank and select the module for installation:

6.1 Select the first module in the list: {M2}.

6.2 Initiate the selected module ranking: $Rank_{M2} = 0$

6.3 Check whether the previous crane location and configuration can be used: {Null}, since the previous crane Location L1 and Configuration C1 are blocked for Module M2 as a result of M1 being sequenced during Iteration 1.

6.4 Select "No", and $Rank_{M2} = 1{,}000{,}000$

6.5 Check whether the previous crane location can be used by changing the configuration: {Null}, since the previous crane location/configuration is not available at the current stage.

6.6 Select "No" and $Rank_{M2} = 1{,}000{,}000 + 100{,}000 = 1{,}100{,}000$.

6.7 Check whether the module is at the edge of the straight run group: {M2, G1}, since M1 is installed from G1, M2 would be at the edge of the remaining to-be-sequenced modules.

6.8 Add the number of options to the ranking number: $Rank_{M2} = 1{,}100{,}000 + 4 = 1{,}100{,}004$

6.9 Move to the next module.

Step 6 repeats for M5B and M6B. The results are shown in Table 3.5. M6B has the lowest ranking value and is selected.

Table 3.5: Module ranking during the first iteration

Module	Ranking
M2	1,100,004
M5B	10,005
M6B	2

7.0 Feasible crane configuration generation process: {C1}.

8.0 Rank and select crane configuration process:

8.1 Check if the configuration is identical to the previous crane configuration: Select “Yes”, since C1 was used in Installation Iteration 1, $\text{Rank}_{C1} = 0$.

8.2 Check for how many modules the crane configuration can be utilized for installing the to-be-sequenced modules if the configuration is not identical to previous crane configuration: Not applicable, since the previous crane configuration was C1.

8.3 Since C1 is the only available configuration, it is selected as the crane configuration for installing M6B.

9.0 Feasible crane location generation process:

9.1 Filter crane locations based on selected module and crane configuration: {M6B, C1, L1, L9} using the list of options provided in Table 3.3.

9.2 Remove the blocked options based on module blocking precedence relationship: {Null}, since sequenced module M1 does not block any of the two locations.

10.0 Rank and select the crane locations:

10.1 Select the first location in the list: {L1}

10.2 Initiate ranking: Rank_{L1}= 0.

10.3 Is this location same as immediate previous crane location: Select "Yes", since L1 was used in Installation Iteration 1.

10.4 Ranking update based on usability of the location:

    10.4.1 How many times the location can be reused for installing future modules: $\text{Rank}_{L1} = 0 + 10000 \times (8-5) = 30{,}000$

    10.4.2 How many times the location can be used for all modules: $\text{Rank}_{L1} = 30000 + 2000 \times (8-6) = 34{,}000$

    10.4.3 Moving Distance (MD): 0.

10.5 Does this location have an Easting or Northing coordinate similar to the previous crane location: Select "Yes"

10.6 Move to the next location.

Repeating Step 10 for L9 would result in $\text{Rank}_{L9} = 15{,}048{,}024$. Since the ranking value of L1 is lower, L1 is selected as the crane location for installing M6B.

11.0 Store solution: The selected M6B, C1 with crane location L1 are stored for Installation Iteration 2.

12.0 Update remaining module list: M5B is moved from the to-be-sequenced module list to the previously sequenced module list.

13.0 Is there a module in the module list for installation: Select "Yes", repeat Steps 4 to 12 until the iterations for all other 6 modules are processed.

3.4.3 Output

Table 3.6 shows the formulated installation plan obtained after completing the above process for 8 installation iterations. The installation plan provides the installation sequence for the modules and the crane location and configuration for installing each module. The precedence relationships for installing the modules are satisfied. The plan minimizes the number of crane foundations, relocations, and configurations, as well as the crane travel distance on site.

Table 3.6: Final installation plan for the sample case study

Installation Iteration #	Module	Crane Configuration	Crane Location
1	M1	C1	L1
2	M6B	C1	L1
3	M6T	C1	L1
4	M5B	C1	L1
5	M5T	C1	L1
6	M2	C1	L3
7	M3	C1	L7
8	M4	C1	L7

3.4.4 Method Validation

To validate the output produced in Section 3.4.3, three of the validation techniques presented by Sargent (2005) are utilized. First, individual behavior, priority, and ranking of modules were traced to ensure method logic is correct. Then, an animation for installing the modules in accordance with the formulated installation plan is created, reviewed, and scrutinized to ensure crane-module technological constraints are satisfied. Finally, face validation, where two knowledgeable individuals are asked to validate both the method behavior and result, are completed. Since the size of the problem is small, the optimality of the solution is reviewed; given the project input, a more optimal solution could not be found.

Enforcement of the three constraints detailed in Section 2.2.1 is confirmed in the provided solution. Figure 3.6 demonstrates enforcement of bottom-top precedence relationships. As shown in screenshots from the animation, bottom module M6B is installed during Installation Iteration 2 prior to installation of top module M6T during Installation Iteration 3. Figure 3.7 demonstrates enforcement of neighbor module precedence relationships. After installing M6B during Installation Iteration 2, M5B is chosen for sequencing during Installation Iteration 4 (rather than M2). Finally, Figure 3.8 demonstrates the enforcement of the module blocking precedence relationship. Given that M2 could be installed from L1 (Table 3.3), the sequenced M5T blocks the path for installing M2 from L1. As such, the crane must be moved to a new location before installing M2 during Installation Iteration 6.

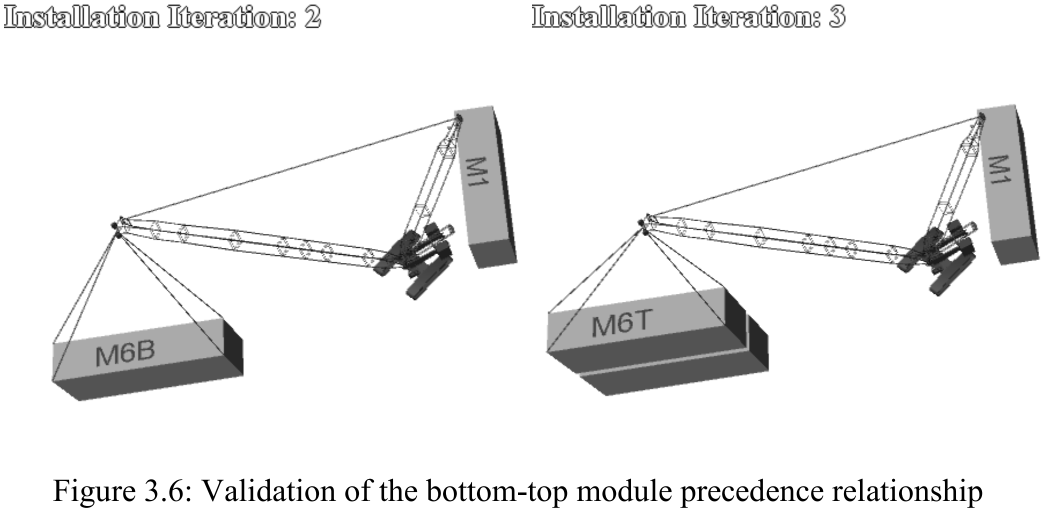


Figure 3.6: Validation of the bottom-top module precedence relationship

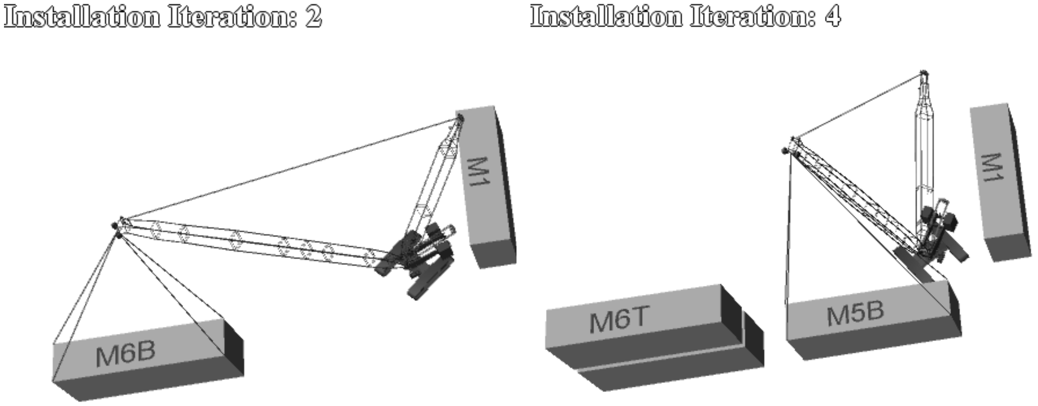


Figure 3.7: Validation of the neighbor module precedence relationship

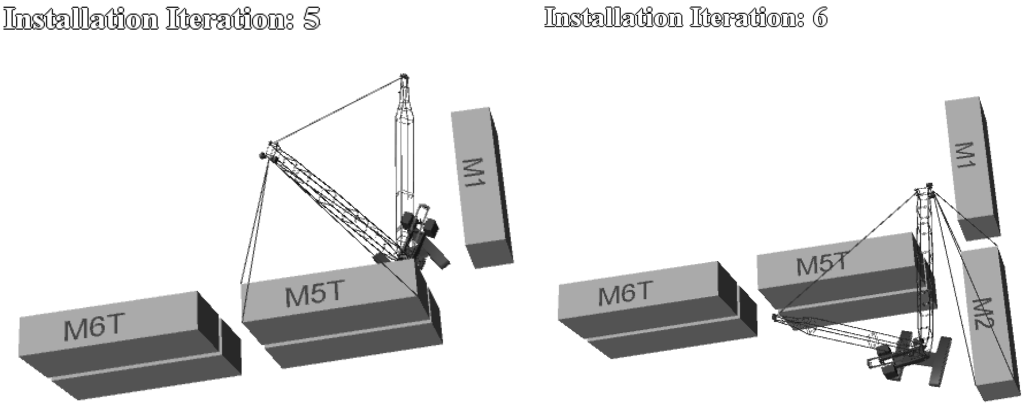


Figure 3.8: Validation of the module blocking precedence relationship

After verifying that the solution satisfies all the constraints, the optimality of the solution is investigated. The solution space comprises over 30 million possible combinations of module installation and crane location (note: some solutions violate the constraints). Even if the first module in the installation iteration is pre-determined, over 4 million solutions still exist. In order to check whether there is a better solution to this small-scale sample problem, three strategic questions were proposed and answered. First, is it possible to install all 8 modules using only 2 crane locations, in contrast to the current solution which requires 3 crane locations?

Second, is it possible to have fewer crane movements between different module installations? Third, is it possible to use crane locations with a shorter straight distance?

Answer to the first question: The combination of Location L1 and (L7 or L8) provides complete coverage of all module installations and accounts for all technological constraints; however, L1 cannot be used for installation of M2 as soon as M1 is installed during Installation Iteration 1. As a result, a third location is required to provide the coverage for installing M2 when M1 is selected as the first module.

Answer to the second question: No, considering the crane installs all these 8 modules from 3 distinct locations, the two crane movements (i.e., from L1 to L3; from L3 to L7) are the minimum requirements, as given in the current solution.

Answer to the third question: The distinction must be made between the straight path distance between crane locations and the actual path that the mobile crane travels on site. Currently, we are using the straight path distance as a measure of the traveling distance, and the locations L1, L3 and L7 are in straight line from North to South. Using these crane locations for crane movement thus provides the shortest possible crane travel distance.

As mentioned in Section 3.3.3, the selection of the initial module has a profound effect on the global optimality of the solution. While it has been shown that the above solution for this example is the best solution available given the input data (including the selection of the initial Module M1), different initial module selection would result in a different installation plan. For example, selecting M2 as the initial module for installation would result in a better installation plan, as shown in Table 3.7. If the initial module for installation is not determined by other project factors, it is recommended that various scenarios are checked by the user before finalizing the plan.

Table 3.7: An alternative installation plan for the sample case study

Installation Iteration #	Module	Crane Configuration	Crane Location
1	M2	C1	L1
2	M1	C1	L1
3	M5B	C1	L1
4	M6B	C1	L1
5	M6T	C1	L1
6	M5T	C1	L1
7	M3	C1	L7
8	M4	C1	L7

As mentioned in Chapter 1, the difficulty of solving this combinatorial problem is dependent on the size of the problem. The larger problem size, the more difficult it is to solve. The advantage of the proposed methodology is the ease of implementation of the proposed algorithms in a software system to automate the solving process for large projects. In the next section, a practical project with 68 modules is planned based on the program developed using the proposed methodology.

3.5 Practical Case Study

In this section, a practical case study is first presented to obtain a suggested plan based on the program developed using the proposed algorithm, in order to show the ease of obtaining an automated solution. The plan is then compared to the plan provided by industry practitioners based on the criteria of solution optimality.

Figure 3.9 shows the designated module layout. The project consists of 68 modules. The module types include pipe rack, electrical, building and equipment modules. The module weights range from 20,000 to 200,000 lbs, while the module lengths are anywhere from 18 to 36 meters. Three groups of straight run modules are identified where neighbor module precedence relationships exist. In addition, there are two or more modules stacked on top of each other in most areas, where bottom-top module precedence relationships exist.

On average, there are approximately 3,000 crane locations available for installing each module, with a total number of 200,000 options for installing all 68 modules. These various installation option for individual module as well as the module blocking precedence relationship between these installation options and the individual modules are generated using previously developed program ACPO (Herman et al. 2010).For example, Figure 3.10 shows the possible crane locations, represented as points 3 feet apart, for installing one module. These locations are shown regardless of the module blocking precedence relationship. It is assumed that one crane is used to install all these modules. The installation plan is evaluated based on the number of distinct crane locations, number of crane relocations, number of crane reconfigurations, and the total crane movement distance.

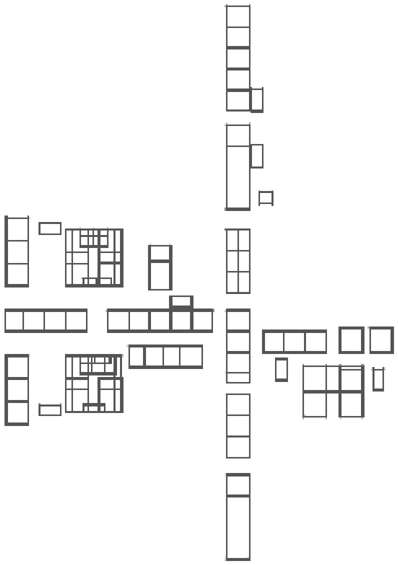


Figure 3.9: Site layout for installation of 68 modules for practical case study

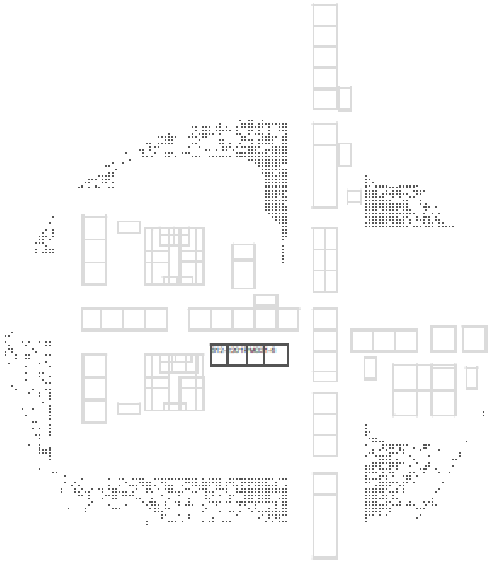


Figure 3.10: Possible crane locations for installing one module for practical case study

In order to plan module installation for a large project, the methodology and processes proposed in Section 3.3 were programmed and automated. Visual Basic for Application (VBA) in MS Access was used for implementing the coding of the algorithm. Figure 3.11 provides an overview of the different steps and processes that take place when preparing a solution using the automation tool. In Step 1, the project details are provided in the form of an information database. In Step 2, the project module list is displayed using the input information. The user can then select which modules are previously installed and which modules are to be sequenced for installation. Next, by clicking the “Start Planning” button, a list of feasible modules for Installation Iteration 1 are generated and ranked. In Step 3, the user should select a module as the first module for Installation Iteration 1, and click the “Generate Solution” button. In Step 4, the solutions are iterated using the proposed methodology. Alternatively, the user can select to navigate through the installation iteration on a step-by-step basis using “Next” and “Back” buttons. In Step 5, the solution is generated and stored in the database.

Figure 3.12 shows a larger screenshot of the module selection window. Once the user selects the project number, the list of module IDs in the project is populated. Next, the planner fixes the planning scope by choosing and moving the modules on the *Module ID* list to either the “To-Be-Sequenced Modules” list or to “Previously Installed Modules” list using the control button provided.

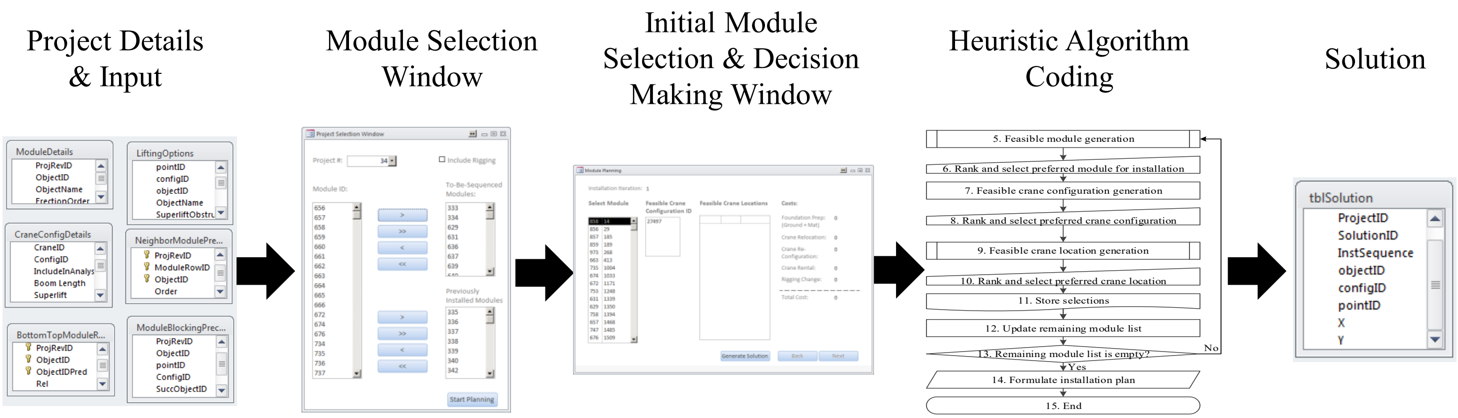
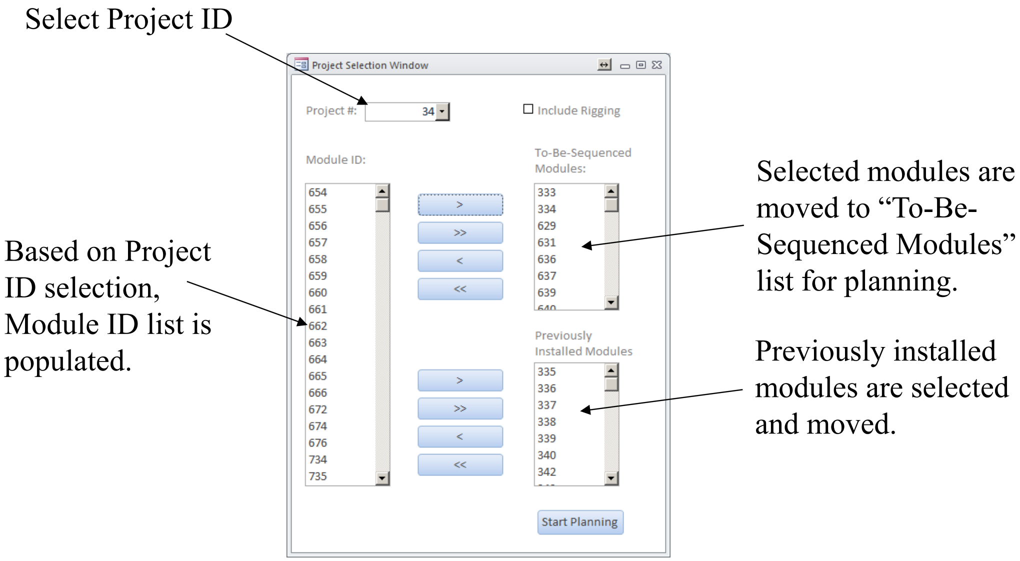


Figure 3.11: The planning process using the developed automation tool



Project #: 34

Select Project ID

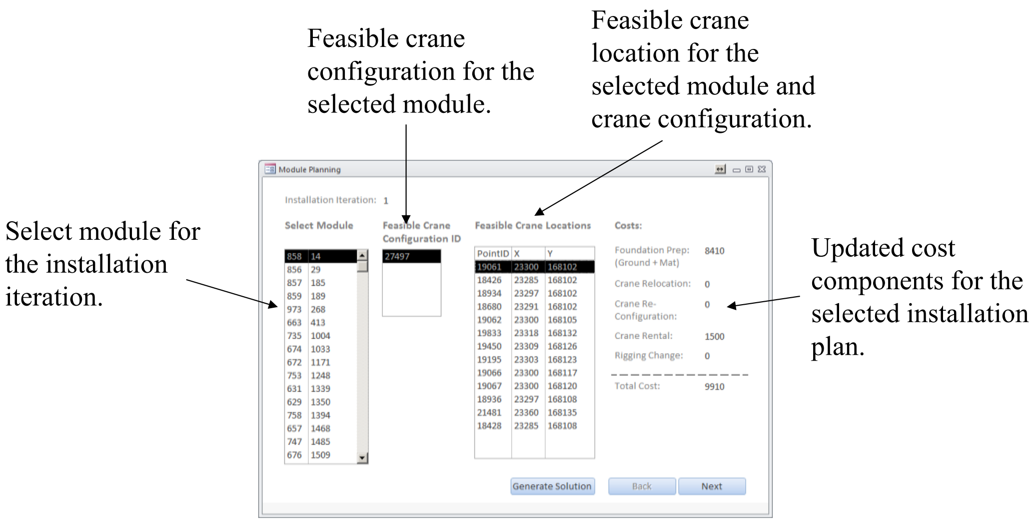
Based on Project ID selection, Module ID list is populated.

Selected modules are moved to “To-Be-Sequenced Modules” list for planning.

Previously installed modules are selected and moved.

Figure 3.12: Overview of initial project and module selection window

Figure 3.13 shows a screenshot of the module planning window. The user selects the starting module from the feasible list of modules generated by incorporating all the constraints detailed in Section 2.2.1. By clicking any module ID, the corresponding feasible crane configurations are displayed. Also, by clicking on any of the feasible crane configuration ID numbers, the list of feasible crane locations for the selected module and crane configuration are displayed. Once the starting module for Installation Iteration 1 is selected, the user can begin the automatic solution generation process by clicking the “Generate Solution” button. If preferred, the user can select the module ID, crane configuration ID and crane location in a step-by-step iteration. In this case, with each iteration, the module list, configurations and crane locations are ranked and displayed in the new order based on the heuristic rules presented in Section 3.3 and with the user choices as input. This can be extremely useful when there are other factors that dictate the installation of specific modules in a specific order. During this process, the cost of making any decision with regards to the selected module, crane configuration and crane location are updated for each iteration on the right side of the display for user reference. This provides the subject matter expert full control over making the final decision in planning the installation with only feasible options presented and ranked in each step.


Feasible crane configuration for the selected module.

Feasible crane location for the selected module and crane configuration.

Select module for the installation iteration.

Updated cost components for the selected installation plan.

Installation Iteration: 1

Select Module

858	14
856	29
857	185
859	189
973	268
663	413
735	1004
674	1033
672	1171
753	1248
631	1339
629	1350
758	1394
657	1468
747	1485
676	1509

Feasible Crane Configuration ID: 27497

Feasible Crane Locations

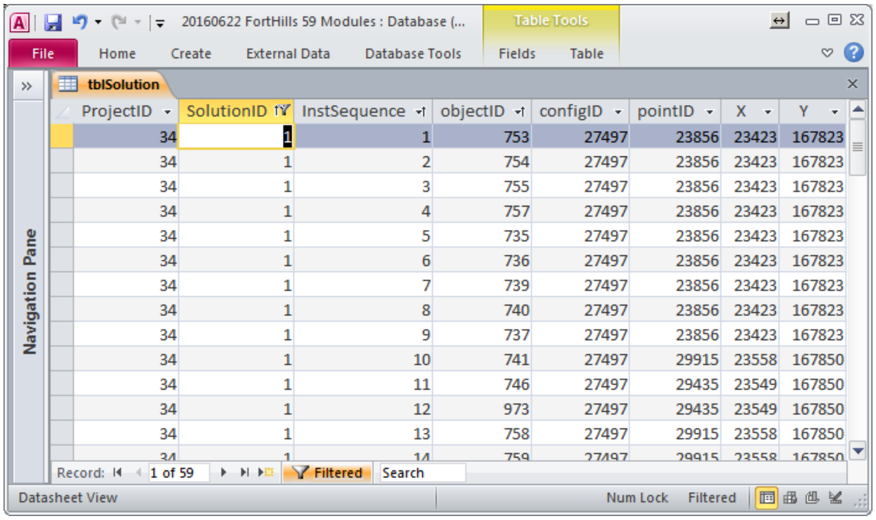
PointID	X	Y
19061	23300	168102
18426	23285	168102
18934	23297	168102
18680	23291	168102
19062	23300	168105
19833	23318	168132
19450	23309	168126
19195	23303	168123
19066	23300	168117
19067	23300	168120
18936	23297	168108
21481	23360	168135
18428	23285	168108

Costs:

Foundation Prep: (Ground + Mat): 8410
Crane Relocation: 0
Crane Re-Configuration: 0
Crane Rental: 1500
Rigging Change: 0
Total Cost: 9910

Figure 3.13: Module selection window overview

Using the methodology explained in Section 3.3, a feasible solution was found for the practical problem. Figure 3.14 shows part of the detailed solution stored in the database. In a process similar to the sample case study presented in Section 3.4, installation plan animations were created and reviewed to ensure that all constraints were enforced.



ProjectID	SolutionID	InstSequence	objectID	configID	pointID	X	Y
34	1	1	753	27497	23856	23423	167823
34	1	2	754	27497	23856	23423	167823
34	1	3	755	27497	23856	23423	167823
34	1	4	757	27497	23856	23423	167823
34	1	5	735	27497	23856	23423	167823
34	1	6	736	27497	23856	23423	167823
34	1	7	739	27497	23856	23423	167823
34	1	8	740	27497	23856	23423	167823
34	1	9	737	27497	23856	23423	167823
34	1	10	741	27497	29915	23558	167850
34	1	11	746	27497	29435	23549	167850
34	1	12	973	27497	29435	23549	167850
34	1	13	758	27497	29915	23558	167850
34	1	14	759	27497	29915	23558	167850

Figure 3.14: Details of the heuristics-based solution stored in the database

While the user or SME were heavily involved in method development, three validation methods stated in Section 3.4.4 were also used to ensure model correctness for this large-size practical problem. The validation methods included tracing the individual module behavior during the process, generating an animation displaying the module installation plan graphically, and face validation by knowledgeable individuals. Figure 3.15 shows a screenshot of the animation at Installation Iteration 61. To complete installation, 4 distinct crane foundation locations, along with 3 crane relocations, are used and a total of 898 feet of crane travel movement is required in this plan.

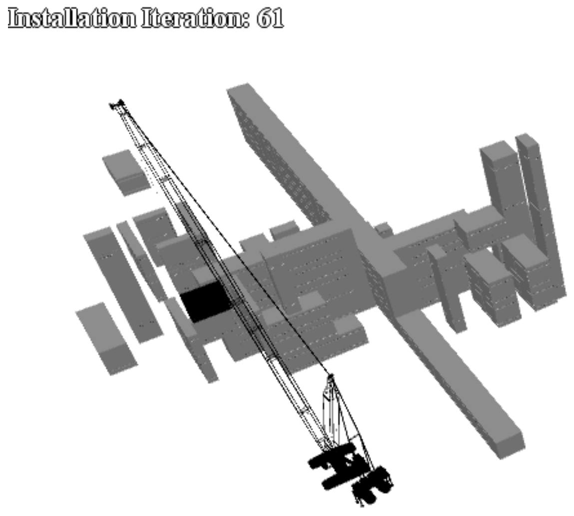


Figure 3.15: A screenshot of the animation of the installation plan generated for the practical case study

For the purpose of comparison, crane locations within a 45-feet radius are assumed to make use of the same crane foundation. Relocation is only considered when the crane moves over a distance of 45 feet to a new assigned crane location. Table 3.8 summarizes the comparison between two solutions provided by both the proposed methodology and practical experience. It shows that in the experience-based installation plan, 8 distinct crane foundations are used with 14 crane relocations and a total of 2270 feet of crane travel movement. The proposed installation plan based on the methodology developed in this work outperforms the experience-based installation plan, requiring a reduced number of crane foundations and relocations, and a substantial decrease in total crane travel movement.

Table 3.8: Comparison of solutions between the plan based on the heuristic-based decision support tool and the experience-based installation plan

Comparison Item	Proposed Installation Plan	Experience-based Installation Plan
Number of Crane Foundations	4	8
Number of Crane Relocations	3	14
Total Crane Travel Movement (ft)	898	2270

3.6 Conclusion

The past work and advancement in multiple-lift mobile crane planning is limited, and there is currently no formalized framework or methodology in place for preparing and automating multi-lift site plans for modular construction in order to determine an optimum module installation sequence. The current practice based on trial-and-error approaches is time-consuming and error-prone. The novel methodology presented in this work can be used to automate module installation planning in these cases. The methodology developed uses the project information (the list of modules, module rigging requirements, available crane, and available crane configurations) as inputs, enforces crane-module technological constraints, and ranks the sequencing options using heuristic-based rules. This facilitates the scheduling tasks involved in preparing an error-free plan for the installation of modules on site. The proposed methodology ensures that a feasible installation plan is generated while minimizing the crane operation cost by means of heuristic rules. The plan feasibility is ensured by enforcing: (i) bottom-top module precedence relationships, (ii) neighbor module precedence relationships, and (iii) module blocking precedence relationships. The crane operation cost is significantly reduced by the use of heuristic rules which minimize the number of distinct crane locations, number of crane relocations and number of crane reconfigurations.

A software system prototype has been developed by implementing the proposed methodology using VBA for Access, and used to automatically schedule a real-world modular construction project. The software system developed effectively prepares the module installation plan while satisfying all constraints. There are four significant advantages to using the software tool developed in this work. First, the software system allows the planner to choose preferred installation options in terms of module installation sequence and crane location. When the installation plan changes, the software allows the planner to investigate a potential path forward and update the project schedule. The software system also reduces the burden on the project team by ensuring all constraints are checked and satisfied. This is significant due to the large amount of information and the interdependence between module installations on the project. Finally, the creativity and expertise of the planner are utilized in planning and sequencing the module installation.

The work presented in this chapter is limited to the use of one crane on site. Also, it does not take into account the rigging requirements of the different modules. While the program is capable of tracking the number of rigging changes required to complete the module installation, it lacks the ability to consider the rigging requirements when planning the module installation sequence. In future research, proposed methodology can be expanded to include scenarios where multiple cranes are used simultaneously on site. Also, the possibility of taking into account module rigging requirements when sequencing module installation can be explored. Finally, the methodology can also be extended to allow the preparation of an installation schedule with specific dates for module installation, by incorporating the project start date and other project constraints such as module delivery dates into the input database.

Chapter 4: MCTS-Based Decision Support Tool for Planning Module Installation

4.1 Introduction

The novel heuristic-based methodology presented in Chapter 3 for planning module installation on industrial site has several limitations. It only covers cases where one crane is utilized on job site. Furthermore, in its current implementation, it does not take into account module rigging requirements when sequencing the installation. While the heuristic methodology could be advanced further to overcome these two limitations in future research, these types of limitations could restrict the application of the heuristic methodology to future changing conditions. As is the case for most heuristic optimization methodology, the proposed heuristic-based methodology is customized to a very specific problem description. Changes to the importance of different factors or requirements to add new criteria in evaluating the quality of the solutions can be an overwhelming task. Methodologies and optimization techniques that rely less on domain knowledge and are flexible to changes in evaluation criteria are desirable. Further, the recent success in utilizing MCTS in competitions in combinatorial type games suggests that this can be a viable, promising approach for the preparation of a module installation plan.

As it can be realized from problem description in Chapter 1 and the approach suggested in Chapter 3, module installation planning on industrial construction sites can be accurately represented as a tree-search problem where a sequence of decisions has to be made while project cost is minimized. Each potential partial solution (state), where a specified mobile crane lifts a specific module from a specific crane location using one of several crane configurations, can be represented as nodes of a tree, with edges between nodes representing the order of actions decided from one installation sequence to the next. A search tree algorithm takes the problem as input and returns a solution as a sequence of edges between tree nodes. In this chapter, the Monte Carlo Tree Search (MCTS) is demonstrated to provide a good basis for determining the installation plan for multiple modules.

The remaining of this chapter would be broken down as follow:

Section 4.2 provides a general overview of the MCTS technique. First, the origin of the MCTS is presented and past applications of the MCTS to similar problems are reviewed. While there are a few variations in the way that MCTS is implemented and used in literature (Coulom 2006, Kocsis and Szepesvari 2006, Chaslot 2006), a general framework of the MCTS algorithm and process is presented.

Section 4.3 discusses the details of MCTS in the context of the module installation planning problem. In this section, the changes that need to be made to the general framework presented in Section 4.2 are explained and discussed.

Section 4.4 presents the application of MCTS to a sample case study. First, the step by step processes of the MCTS are listed. Then, toolkit software that has been developed to implement the MCTS for module planning is presented and used to solve the sample case study.

Section 4.5 discusses the limitations of the proposed MCTS algorithm in solving real case examples.

Section 4.6 discusses potential improvements that should be examined in future research to enhance the results obtained from the MCTS algorithm. This section discusses the changes that can be made to selection policy as well as the simulation policy within MCTS to improve the optimality of the result.

Section 4.7 gives an outline of the chapter findings and results. It also summarizes the direction of future research to further advance the application of MCTS to the module planning problem.

4.2 MCTS Structure and Past Application

The MCTS approach provides an iterated solution by random exploration of the search space, where the results of previous exploration are used to guide and direct the future tree branching, as it successively becomes better at determining the most promising choices (i.e., actions) (Browne et al. 2012). In MCTS, each node represents a given state (i.e., a partial solution) of the problem, and two critical details are stored: (i) the current value of the state (there are different ways to determine this value, as will be discussed later), and (ii) the number of times the node has been visited. Figure 4.1 shows the four basic steps of MCTS. These four main steps take place iteratively – selection, expansion, simulation and back-propagation – and are described below.

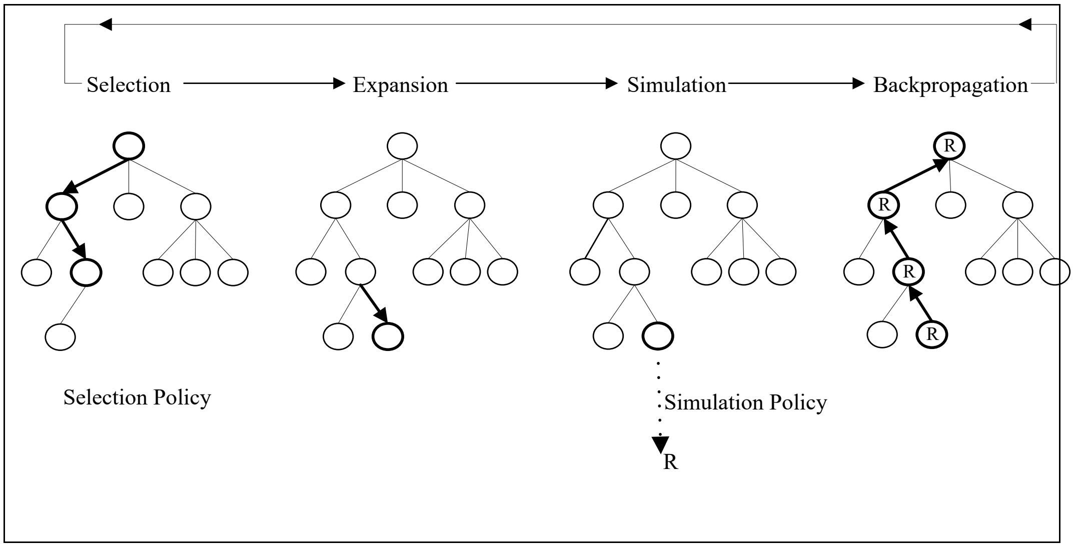


Figure 4.1: A general MCTS iteration

Selection: MCTS usually starts with a tree containing only the root node. The tree is expanded from the root node by selecting a potential child node that is not part of the tree yet. The selection policy should strike a balance between the exploitation of previously discovered good quality results and the exploration of less investigated tree branches. Several selection strategies have been suggested (Chaslot 2006, Coulom 2006, Gelly and Wang 2006). The Upper Confidence bounds applied to Trees (UCT) proposed by Kocsis and Szepesvary (2006) is the most common selection policy in literature and is as follows:

$$UCT = \overline{X_j} + 2C_p\sqrt{\frac{2\ln n}{n_j}} \qquad \text{(Equation 4.1)}$$

Where:

$\overline{X_j}$: the average outcome of the simulated solution that passes through node j,

n : the number of times the current (parent) node has been visited,

n_j : the number of times child j has been visited, and

C_p: a constant and can be adjusted to lower or increase the amount of exploration.

Expansion: In the expansion step, a selected child is added as a node to the tree. One node is added per simulation iteration.

Simulation: In the simulation step, actions are taken from the lowest expanded node to end of the problem for a complete solution. The actions can be either taken using random moves or pseudo-random moves according to a predefined simulation strategy. It is shown that an adequate simulation strategy can significantly improve MCTS results (Bouzy 2005, Gelly et al. 2006).

Backpropagation: The result of the simulation solution is propagated backward through all the previously expanded nodes that are part of the solution. There are different strategies on what information can be backpropagated. While the most popular and effective strategy is shown to be the average of all the simulated solution results throughout a specific node, other strategies have also been suggested in literature (Chaslot 2010, Coulom 2006, Chaslot 2006).

MCTS has been successfully implemented in playing and winning difficult games such as Go. Crazy Stone and MoGo programs defeated professional human Go players (Coulom 2006, Gelly et al. 2006) on small boards. The success of the MCTS framework in the computer Go provided motivation to utilize the same structure on other planning and combinatorial type optimization problem. Chaslot et al. (2006) applied MCTS to a production management problem where a fixed set of products need to be produced by going through a fixed set of production processes, and the goal is to maximize the amount of the final product within the constraint of limited time or money available. Matsumoto et al. (2010) used a Single-Player MCTS to schedule the printing process of automobile parts. Rimmel et al. (2011) used nested MCTS to reach a state of the art solution for the traveling salesman problem (TSP) with time window. Runarsson et al. (2012) proposed a MCTS-based scheduling approach called Monte Carlo Scheduling (MCS) for job shop scheduling.

The state of the art results reported in previous research on the application of MCTS to planning and solving the combinatorial optimization problem provided the motivation to implement MCTS for module installation planning. As concluded by Chaslot et al. (2006), Matsumoto et al. (2010), Rimmel et al. (2011) and Runarsson et al. (2012), MCTS needs to be customized to the problem in order to obtain good results. In the next section, the changes and modifications that were considered in the implementation of MCTS for module installation planning are detailed and explained.

4.3 MCTS Implementation Details for the Module Scheduling Problem

In MCTS a search tree is built iteratively and expanded by slowly adding leaves (nodes) to the tree according to the results of previous iterations. In module installation planning, each node in the tree represents a set of potential partial solutions at the specific installation iteration. Each node specifies the crane being used as well as the crane configuration and location. In addition, it specifies which modules are installed by each crane. Figure 4.2 presents the node information graphically. Any change in the details of the node, such as a change in crane configuration, crane location, or combination of cranes being used, would result in the creation of a new possible node in the tree at the specified installation iteration level.

As explained in the previous section, the MCTS algorithm consists of four main steps: (i) selection, (ii) expansion, (iii) simulation, and (iv) backpropagation. Figure 4.3 shows the flowchart of the overall steps that would take place in a typical MCTS procedure, as presented and explained in Section 4.2. In this section, the specific configuration of the MCTS components that were used to solve the problem is explained.

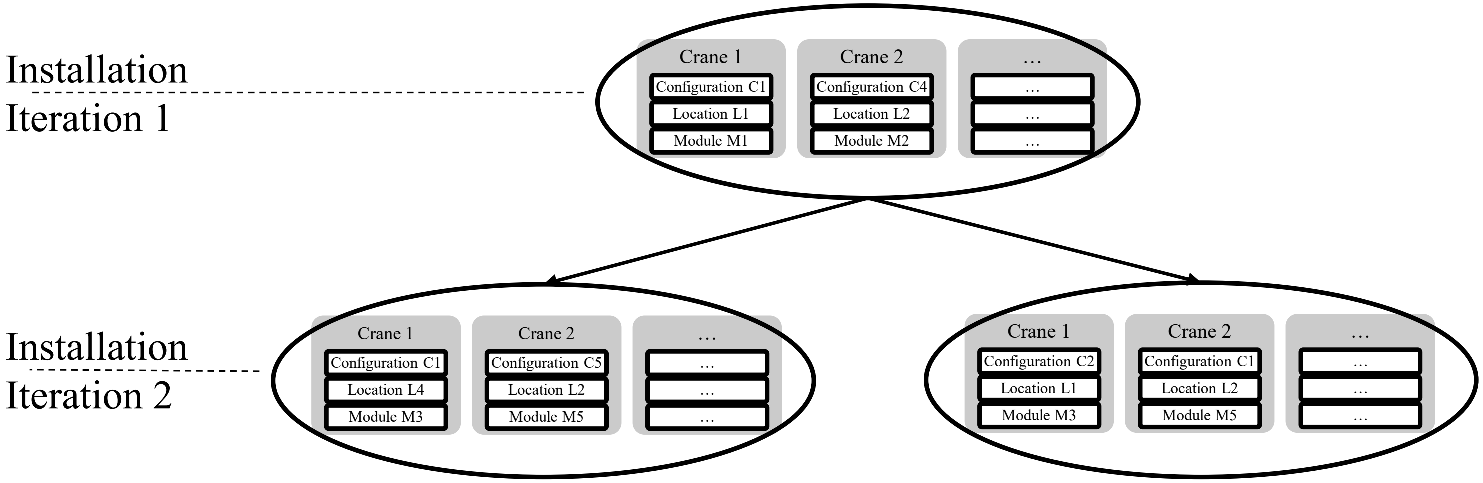


Figure 4.2: Example of tree node information

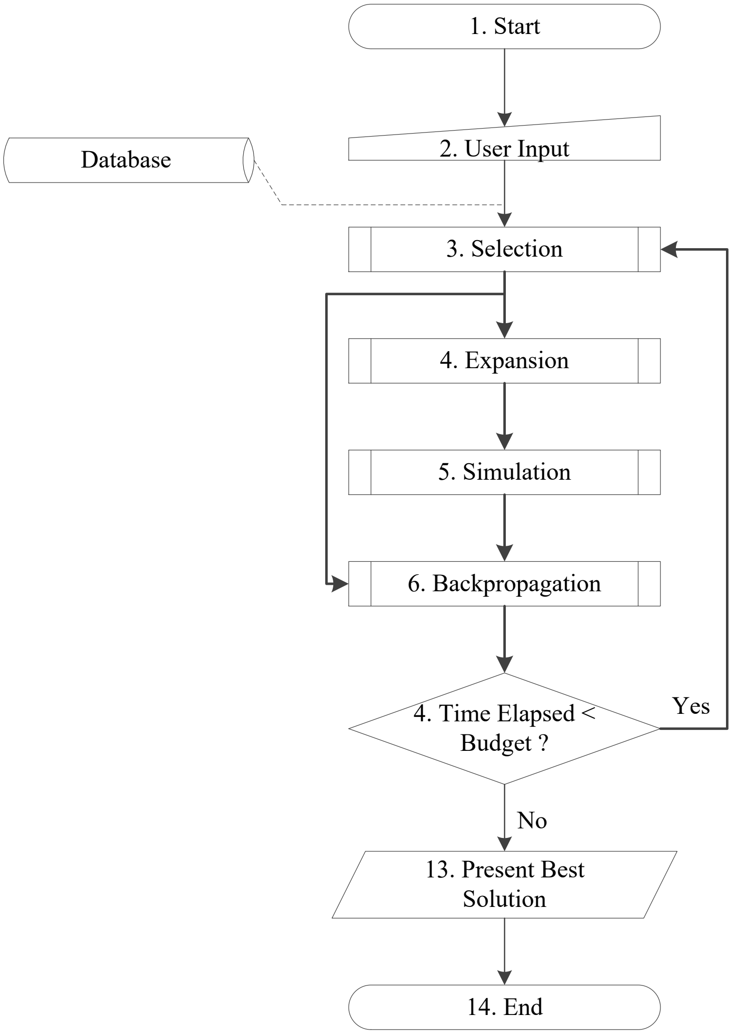


Figure 4.3: Overall MCTS procedure flowchart for module installation planning

4.3.1 User Input and Database for MCTS

Various project details and constraints are necessary to properly generate a module installation plan. The following minimum inputs are required: (i) the feasible crane configurations, (ii) the feasible crane location coordinates associated with the individual modules, and (iii) the module installation precedence relationships. As discussed in Section 2.2.1, the feasible crane configuration and location for each module are determined using previously developed tools. The precedence relationships need to be considered (also discussed in Section 2.2.1) are: (i) bottom-top module precedence relationships, (ii) neighbor module precedence relationships, and (iii) module blocking precedence relationships. The input information is assumed to be available in a database format, or in a format that can easily be converted to a database format.

In addition to the project information that are provided in the database, the user is required to provide specific information for the planning session. The user needs to specify which crane or cranes will be utilized on the specific job site. In addition, if there are any previously installed modules they need to be properly selected. Finally, the user needs to decide on which modules are planned for installation.

4.3.2 Selection

As mentioned earlier, an adequate selection policy and simulation strategy are the main factors in the successful application of the MCTS (Chaslot 2010). The multi-arm bandit selection policy is one of the most widely used selection policies in literature (Browne et al. 2012). In this case, the upper confidence bound (Kocsis and Szepesvari 2006) presented in Section 4.1 is used. The C_p value needs to be tuned experimentally based on the problem domain as well as the reward boundaries (Kocsis and Szepesvari 2006, Chaslot 2010). In this work, the average reward, X, for each solution, j, is calculated as:

$$\overline{X_j} = \frac{1}{SolutionCost} \qquad \text{(Equation 4.2)}$$

Figure 4.4 presents the flowchart of the processes that take place in the selection step. It travels down the tree branches iteratively based on the UCT calculation until it either hits a node that is not fully expanded, or it hits the end state of the problem. When it hits a node that is not fully expanded, it forwards the node to the expansion step. On the other hand, when the process hits a non-expandable node representing the end state of the problem, it moves to the backpropagation step.

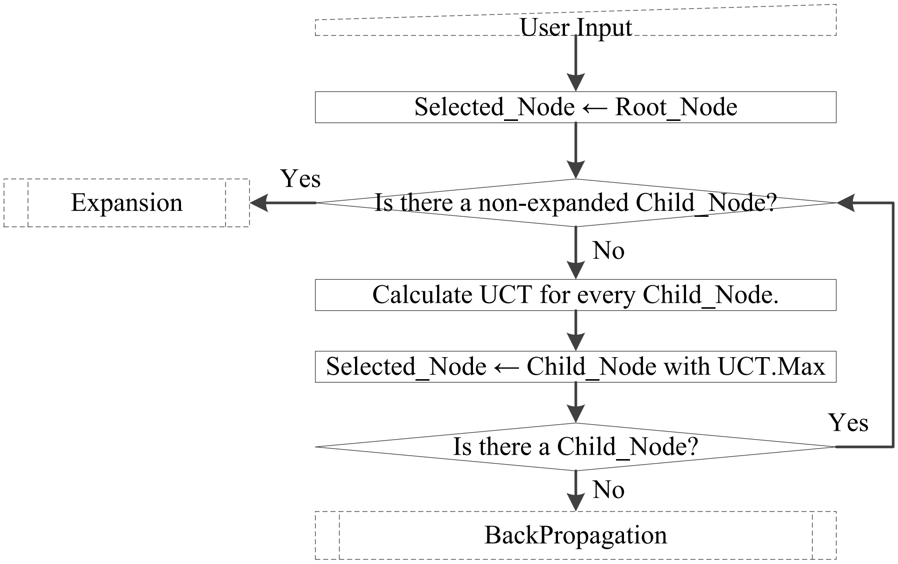


Figure 4.4: MCTS selection step flowchart

4.3.3 Expansion

The expansion policy stated in Section 4.2 is used here, that is, one node expansion per simulation. A child node of the selected node is added where it meets all the feasibility and constraint requirements discussed in Section 2.2. Figure 4.5 demonstrate how the expansion step is completed to ensure (i) that all project constraints are accounted for in the solution search, and (ii) all feasible options are given a chance of being part of the final solution.

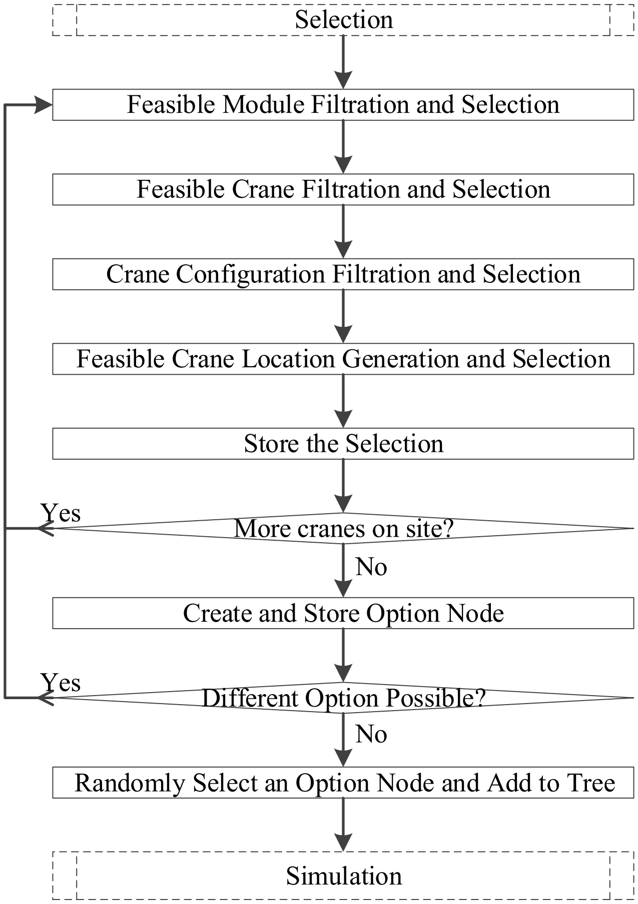


Figure 4.5: The detailed expansion procedure flowchart for module planning

4.3.4 Simulation

The simulation takes place from the lowest expanded node all the way to the end state of the problem. In the most basic form, the random default policy is utilized for simulation purpose. The feasible nodes at each level are randomly generated and selected. In order to enhance the optimality of the solution generated by the MCTS algorithm, a knowledge-based heuristic policy resembling the heuristic rules presented in Chapter 3 was utilized to improve the simulation process. While using the heuristic rule for the simulation policy resulted in a better initial solution, the slow performance of the program in simulating different branches of the tree resulted in fewer simulations in the specified time span. As will be discussed later, the proper level of heuristic knowledge implementation can be investigated in the future to enhance the program performance.

The flowchart in Figure 4.6 shows the step-by-step process that takes place during the simulation step. Starting from the newly added child node from the expansion step, a random solution is built by traveling down the tree choosing feasible nodes. The "Generate all feasible Simulated_Node.Childs" process consists of the same procedures which were explained in the previous section for the expansion step, and graphically depicted in Figure 4.6. However, the randomly selected nodes are not added to the tree structure. Once a complete solution is found, the cost of the simulated solution is calculated and the process moves to the backpropagation step.

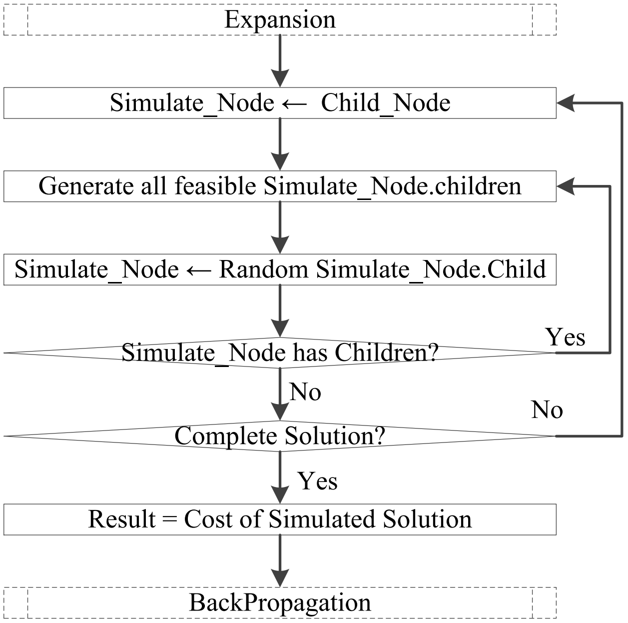


Figure 4.6: MCTS procedure flowchart for the simulation step

4.3.5 Backpropagation

As mentioned in Section 4.2, the average value of the simulation result and number of times the simulation has passed through a node are the most common statistics that are updated at different nodes during the tree search. Chaslot (et al. 2006) has also suggested that for the schedule optimization problem, the best solution found can be used and presented as the final solution for the problem. In this research, the average cost of the simulated result at each node is tracked and used in the UCT calculation. The simulation solution with the lowest cost is presented as the final solution to the module installation planning problem. The flowchart in Figure 4.7 demonstrates the overall steps completed as part of the backpropagation step.

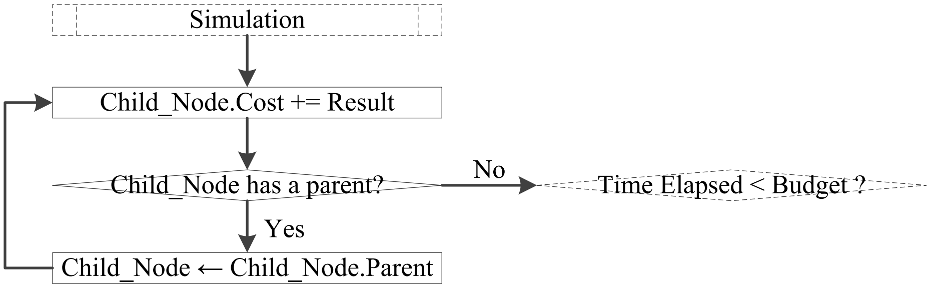


Figure 4.7: MCTS procedure flowchart for backpropagation step

The four steps above are repeated until the computation budget is reached, and the solution with the lowest cost value is presented as the final solution of the application of the MCTS algorithm to the problem. Algorithm 4.1 provides the pseudo code for the MCTS algorithm implemented.

```
While (within Time budget) Do
    selected_node ← root_node

    'The Selection Step Application
    While (selected_node ϵ Tree) Do
        Last_Node ← Selected_node
        Selected_node ← Selection_policy (Selected_node)
    End

    'Expansion Step
    New_Node ← Last_Node

    'Simulation Step
    Result ← Simulation policy (New_Node)

    'Backpropagation Step
    Current_Node ← Last_Node
    While (Current_Node ϵ Tree) do
        Backpropagation (Current_Node, Result)
        Current_node ← Current_node.parent
    End
End
Return Best_Solution

'Selection Policy Function
Function Selection_Policy (Selected_Node)
{
    UCT.Max = 0
    While (Child_Node ϵ Selected_Node)
        Child_Node.UCT = UCT equation
        IF Child_Node.UCT > UCT.Max Then
            UCT.Max = Child_Node.UCT
            Future_Node = Child_Node
        End IF
        Next Child_Node
    End
    Return Future_Node
}

'Simulation Function PolicyData
Function Simulation_Policy (New_Node)
{
    Simulate_Node ← New_Node
    While (! End of project) Do
        Simulate_Node ← Simulate_Node.Child
    End
    Result ← Solution.Cost
}
```

Algorithm 4.1: Pseudo code for MCTS implementation for Module Installation Planning (Based on Chaslot 2010)

4.4 Sample Case Study

In this section, the sample case study that was presented in Section 3.4 is used to demonstrate the advantages and potential of using MCTS in producing a good-quality module installation plan. First, the detailed steps required to apply MCTS in this case are shown for three iterations. Then, the different components of a software toolkit that was developed based on this MCTS implementation, as well as the type of results it produces, will be shown through this case study.

The Sample case study consisted of 8 modules, and 9 available crane locations. Figure 3.5 shows the site layout for the module installation. All the project inputs shown in Section 3.1 are entered into database. Based on the number of modules and number of options available to install each module, there are more than 500 million solution combinations to be considered in planning the installation of all 8 modules. The success of the MCTS in finding a good quality solution can be measured by the percentage of the solution space it searches and tests before the solution is finalized.

4.4.1 MCTS Process Details

In this section, the detailed processes that take place in the MCTS are shown for three iterations. Then, the MCTS software developed for solving the module planning problem is showcased.

The detailed algorithm for MCTS processes are as follow:

1.0 Start.

2.0 User Input:

  2.1 Project ID: {1}.

  2.2 Crane Clearance: {12}, this is only used when there are multiple cranes on site.

  2.3 Module list for installation: {M1, M2, M3, M4, M5B, M5T, M6B, M6T}.

  2.4 Crane Selection: {Crane 745}.

  2.5 Input previously installed module list: {Null}.

Iteration #1

3.0 Selection Step:

  3.1 Selected Node: {Node 0}, Root node.

  3.2 Check whether the selected node has non-expanded children: {(M1, C1, L1), …}. Since this is first iteration, none of the child nodes have been expanded.

  3.3 Select "Yes", and move to the Expansion Step.

4.0 Expansion Step:

  4.1 Feasible Module Filtration and Selection: {M1, M2, M3, M4, M5B, M6B} are feasible modules, and M1 is selected randomly. Note that M5T and M6T cannot be selected because the bottom-top module precedence relationship requirements are not satisfied for these two modules.

  4.2 Feasible Crane Filtration and Selection: {Crane 745} is the only crane assigned to this project.

    - 4.3 Feasible Crane Configuration and Selection: {C1, C2} are feasible crane configurations. C1 is selected randomly.
    - 4.4 Feasible Crane Location Generation and Selection: {L1, L2}. L1 is selected randomly.
    - 4.5 Store the Selection: (M1, 745, C1, L1).
    - 4.6 Check whether there are additional cranes assigned to the project that can be used at the same time: No; there is only one crane selected for this project.
    - 4.7 Create a new potential node: (M1, 745, C1, L1)
    - 4.8 Check whether a different node option is feasible: Since there is no node previously added to the tree, multiple other node options can be generated.
    - 4.9 Select yes and repeat Steps 4.1 to 4.8 until all potential nodes are generated: (M1, 745, C1, L2), (M1,745, C2, L1), …
    - 4.10 Randomly select a Child Node and add to the Tree Structure: (M1, 745, C1, L1).
    - 4.11 Go to the Simulation Step.
- 5.0 Simulation Step:
    - 5.1 Assign the Child Node as the Simulated Node: (M1, 745, C1, L1).
    - 5.2 Repeat Steps 4.1 to 4.9 where the Selected Node is the Simulated Node: (M2, 745, C1, L3), (M5B, 745, C1, L1), …
    - 5.3 Assign one of the nodes generated in Step 5.2 to the Simulated Node: (M2, 745, C1, L3).

    - 5.4 Repeat Steps 5.1 to Steps 5.3 until the Simulated Node does not have a Child Node.
    - 5.5 Check whether a complete solution is found: If no, repeat Steps 5.1 to 5.4.
    - 5.6 If a complete feasible solution is found, calculate the cost of the solution and then move to the Backpropagation Step. The cost functions are defined to evaluate each solution based on number of distinct crane locations, number of crane relocations, number of crane reconfigurations and the crane moving distance required to execute the simulated solution.
- 6.0 Backpropagation Step:
    - 6.1 Update the cost value for the Child Node and add one to the simulation number (n) for the Child Node.
    - 6.2 Check whether the Child Node has a Parent Node: Every Child Node will have one Parent Node, except the Root Node.
    - 6.3 If yes, assign the Parent Node as a Child Node and repeat Steps 6.1 to 6.3.
    - 6.4 If no, move to next step.

Figure 4.8 shows the steps presented above graphically for Step 3 to Step 6. In the first iteration, one node is added to the tree at Level 1. The above process is repeated at least 30 times before nodes are added at the second level or lower. Figure 4.9 demonstrate the state of the tree after completing 30 iterations. Table 4.1 shows what each node represents and the cost obtained by simulating each node after the

30th iteration of MCTS process. Please note that these costs values are random numbers between the predicted project cost boundaries for demonstration purposes only.

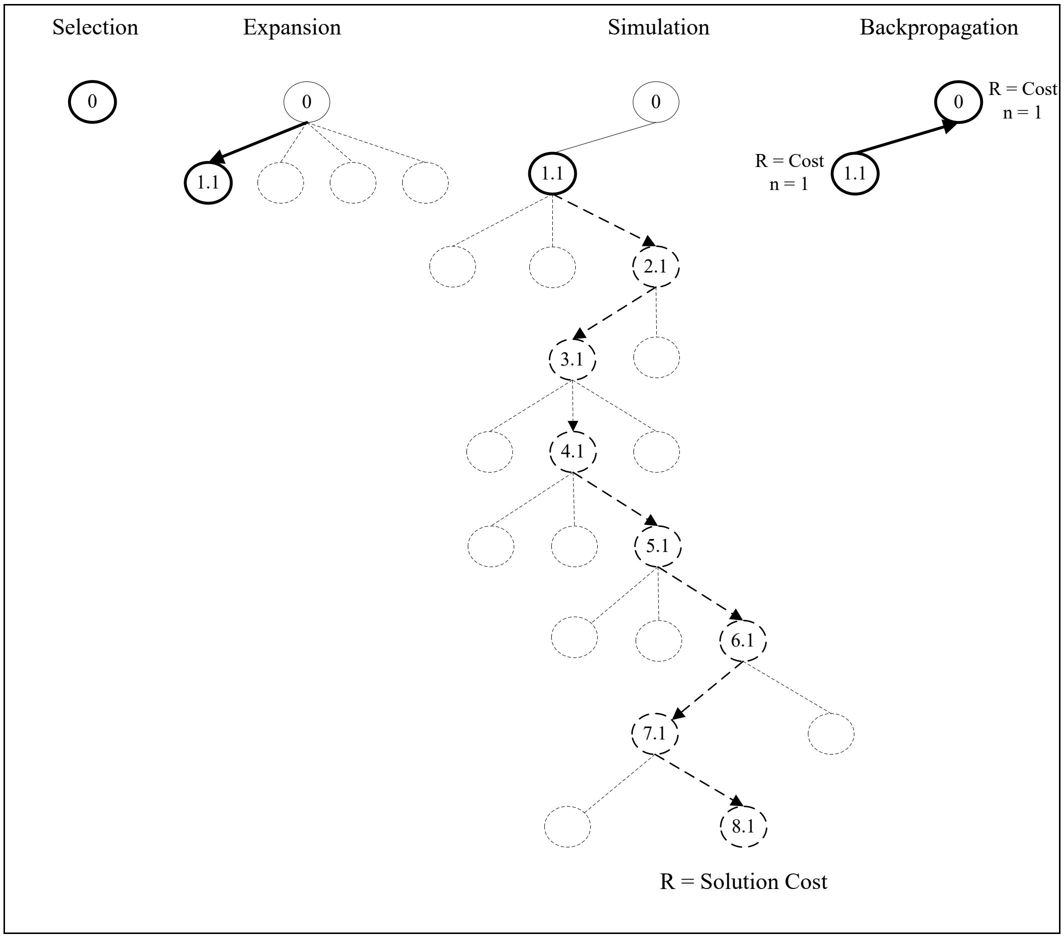


Figure 4.8: Graphical demonstration of the first MCTS iteration for the 8-module case study

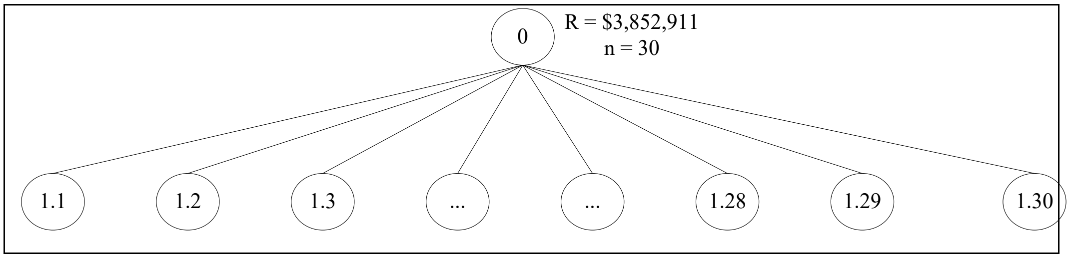


Figure 4.9: MCTS status after 30 iterations

Table 4.1: Node status after 30 iterations for sample case study

Node	Node Details	Number of Simulation (n)	Simulated Cost	UCT
0	-	30	\$3,852,911	-
1.1	(M1, 745, C1, L1)	1	\$109,552	1.96E-5
1.2	(M1, 745, C2, L1)	1	\$150,377	1.71E-5
1.3	(M1, 745, C1, L2)	1	\$121,221	1.87E-5
1.4	(M1, 745, C2, L2)	1	\$114,778	1.91E-5
1.5	(M2, 745, C1, L1)	1	\$104,842	2.00E-5
1.6	(M2, 745, C1, L2)	1	\$144,629	1.73E-5
1.7	(M2, 745, C1, L3)	1	\$153,490	1.69E-5
1.8	(M2, 745, C2, L3)	1	\$145,628	1.73E-5
1.9	(M2, 745, C1, L4)	1	\$148,262	1.72E-5
1.10	(M2, 745, C2, L4)	1	\$117,869	1.89E-5
1.11	(M3, 745, C1, L3)	1	\$119,378	1.88E-5
1.12	(M3, 745, C1, L4)	1	\$128,975	1.82E-5
1.13	(M3, 745, C1, L5)	1	\$90,893	2.14E-5
1.14	(M3, 745, C2, L5)	1	\$168,739	1.64E-5
1.15	(M3, 745, C1, L6)	1	\$99,515	2.05E-5
1.16	(M3, 745, C2, L6)	1	\$148,975	1.71E-5
1.17	(M3, 745, C1, L7)	1	\$167,958	1.64E-5
1.18	(M3, 745, C1, L8)	1	\$89,240	1.16E-5
1.19	(M4, 745, C1, L7)	1	\$140,894	1.75E-5
1.20	(M4, 745, C2, L7)	1	\$131,986	1.80E-5
1.21	(M4, 745, C1, L8)	1	\$99,538	2.05E-5
1.22	(M4, 745, C2, L8)	1	\$124,632	1.85E-5
1.23	(M5B, 745, C1, L1)	1	\$149,570	1.71E-5
1.24	(M5B, 745, C2, L1)	1	\$165,728	1.65E-5
1.25	(M5B, 745, C1, L2)	1	\$122,303	1.86E-5
1.26	(M5B, 745, C1, L3)	1	\$90,917	2.14E-5
1.27	(M5B, 745, C2, L3)	1	\$154,012	1.69E-5
1.28	(M5B, 745, C1, L9)	1	\$127,187	1.83E-5
1.29	(M6B, 745, C1, L1)	1	\$138,794	1.76E-5
1.30	(M6B, 745, C1, L9)	1	\$83,029	2.25E-5

The 31st iteration would be as follow:

Iteration #31

3.0 Selection Step:

3.1 Selected Node: {Node 0}, Root Node.

3.2 Check whether the selected node has nonexpanded children: Null; all child nodes of this parent were expanded in Iterations 1 to 30.

3.3 Select “No”, move to the calculation of UCT.

3.4 UCT values for each node from 1.1 to 1.30 are calculated using Equations 4.1 and 4.2. For Node 1.1:

$$\overline{X_j} = \frac{1}{109522} = 0.00000913$$

$$UCT = 0.00000913 + 2 \times 0.000002 \times \sqrt{\frac{2 \times \ln 30}{1}} = 0.0000196$$

UCT calculation results for all 30 nodes are shown in Table 4.1.

3.5 Node 1.30 has the largest UCT value and is selected.

3.6 Check whether Node 1.30 has any child: Yes, there are lower level nodes that have Node 1.30 as a parent.

3.7 Check whether Node 1.30 has any nonexpanded Child Node: Yes, since none of the Node 1.30 children is are added to the tree structure yet.

3.8 Select “Yes” (see Figure 4.4) and move to the Expansion Step.

4.0 Expansion Step:

4.1 Feasible Module Filtration and Selection: {M1, M3, M4, M5B, M6T} are feasible modules. M1 is selected randomly. Note that M5T cannot be selected because of the bottom-top module precedence relationship, and M2 cannot be selected because of the neighbor module precedence relationship.

4.2 Feasible Crane Filtration and Selection: {Crane 745} is the only crane assigned to this project.

4.3 Feasible Crane Configuration and Selection: {C1, C2} are feasible crane configurations. C1 is selected randomly.

4.4 Feasible Crane Location Generation and Selection: {L1, L2}. L1 is selected randomly.

4.5 Store the Selection: (M1, 745, C1, L1)

4.6 Check whether there are additional cranes assigned to the project that can be used at the same time: No, there is only one crane selected for this project.

4.7 Create a new potential node: (M1, 745, C1, L1)

4.8 Check whether a different node option is feasible: Since there is no child previously added to Node 1.3, multiple other node options can be generated.

4.9 Select "Yes" and repeat Steps 4.1 to 4.8 until all potential nodes are generated: (M1, 745, C1, L2), (M1,745, C2, L1), …

4.10 Randomly select a Child Node and add to the tree structure: (M1, 745, C1, L1)

4.11 Go to Simulation Step.

5.0 Simulation Step:

5.1 Assign the Child Node to the Simulated Node: (M1, 745, C1, L1)

5.2 Repeat Steps 4.1 to 4.9 where the Selected Node is the Simulated Node: (M6T, 745, C1, L1), (M5B, 745, C1, L1), …

5.3 Assign one of the nodes generated in Step 5.2 to be the Simulated Node: (M6T, 745, C1, L1)

5.4 Repeat Step 5.2 to 5.3 until all simulated node children have been generated.

5.5 Check whether a complete solution is found: If no, repeat Steps 5.1 to 5.4.

5.6 If a complete feasible solution is found, calculate the cost of the solution and move to the Backpropagation Step.

6.0 Backpropagation Step:

6.1 Update the Cost value for the Child Node and add one to the simulation number (n) for the Child Node.

6.2 Check whether the Child Node has a Parent Node: every Child Node needs to have one Parent Node, except the Root Node.

6.3 If "Yes," assign the Parent Node as a Child Node and repeat Steps 6.1 to 6.3.

6.4 If no, move to the next step.

Figure 4.10 demonstrates Step 3 to 6 graphically for Iteration 31. The above processes are repeated until the computation budget (time) allocated to solve the problem is over. As the iterations progress, branches are added to the tree structure based on the previous iteration results. Depending on the value selected for the constant C_p in Equation 4.1, the amount of exploration in new areas of the tree versus exploitation of previously expanded tree nodes is determined.

In order to take the full advantages of MCTS process, it is implemented in a software toolkit. In Section 4.4.2, the solutions to this sample 8-module problem are explored using the developed program.

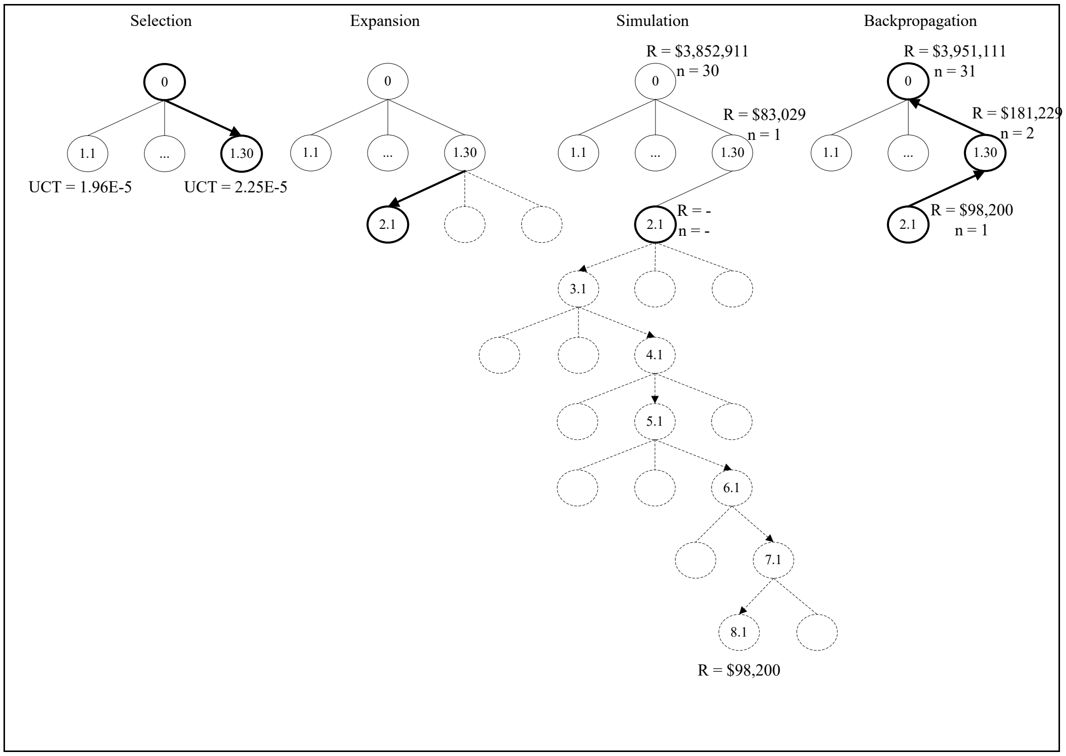
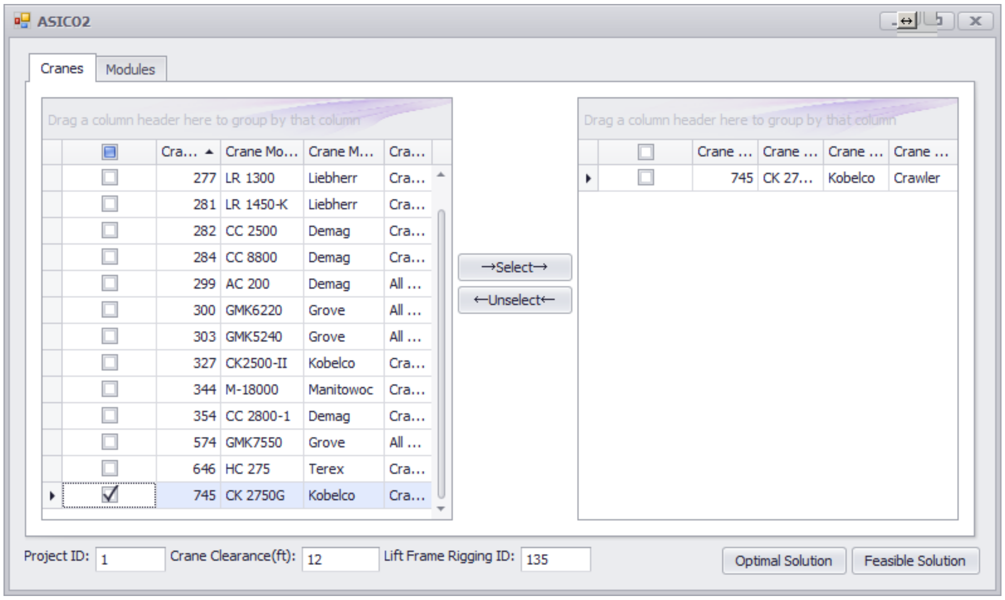


Figure 4.10: Graphical demonstration of Iteration 31 of the MCTS process for an 8-module example case study

4.4.2 MCTS Implementation

The benefits of the MCTS can be realized when enough iterations have taken place to explore different branches of the tree. In order to fully implement and utilized the MCTS process described in previous sections, the above processes are implemented and a software toolkit is developed. In this section, different program features as well as the optimality of the solution that can be obtained using the toolkit for the sample case study installation are demonstrated.

Figure 4.11 shows the first input screen of the program, which allows the selection of the crane for the project. In this window, based on the source database, the user can make a crane selection based on the list of available cranes provided. The user can also specify a specific clearance requirement between cranes on site, if multiple cranes are being utilized at the same time.



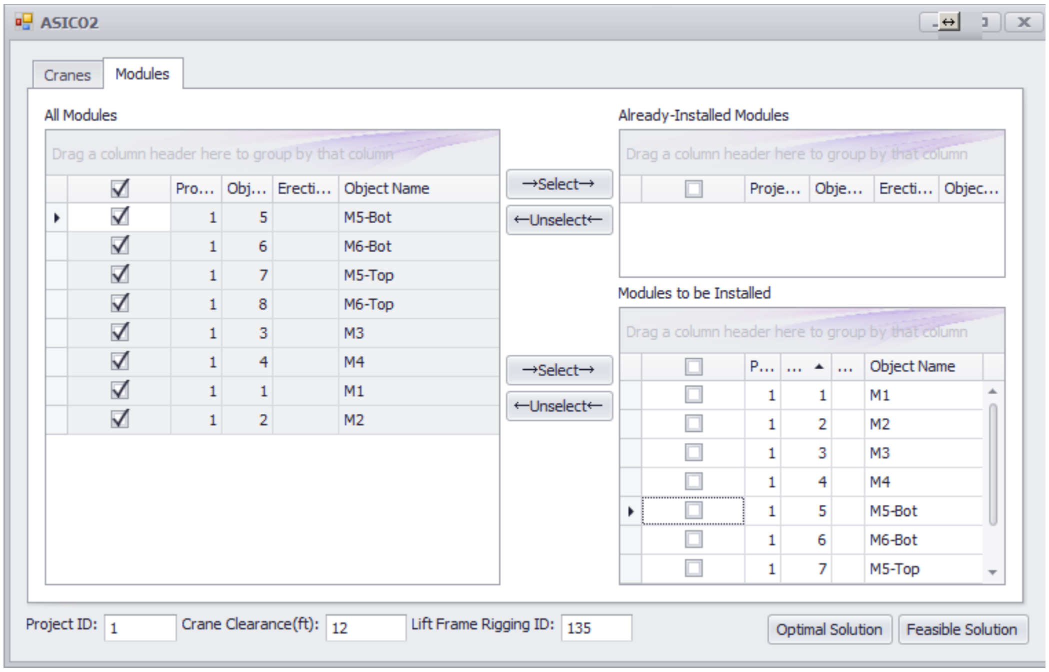
	☐	Cra...	Crane Mo...	Crane M...	Cra...
	☐	277	LR 1300	Liebherr	Cra...
	☐	281	LR 1450-K	Liebherr	Cra...
	☐	282	CC 2500	Demag	Cra...
	☐	284	CC 8800	Demag	Cra...
	☐	299	AC 200	Demag	All ...
	☐	300	GMK6220	Grove	All ...
	☐	303	GMK5240	Grove	All ...
	☐	327	CK2500-II	Kobelco	Cra...
	☐	344	M-18000	Manitowoc	Cra...
	☐	354	CC 2800-1	Demag	Cra...
	☐	574	GMK7550	Grove	All ...
	☐	646	HC 275	Terex	Cra...
▸	☑	745	CK 2750G	Kobelco	Cra...

	☐	Crane ...	Crane ...	Crane ...	Crane ...
▸	☐	745	CK 27...	Kobelco	Crawler

Project ID: 1
Crane Clearance(ft): 12
Lift Frame Rigging ID: 135

Figure 4.11: Selection window in the MCTS program

In the next screen, under the "Modules" tab, the already installed modules as well as the module to be installed can be selected using the control buttons. Figure 4.12 shows a screenshot of the interface. In this example, it is assumed that no previous module is installed on site. Once the modules are properly selected, the MCTS algorithm can be initiated by clicking on the "Optimal Solution" button in the bottom right corner of the screen.



All Modules

	☑	Pro…	Obj…	Erecti…	Object Name
▸	☑	1	5		M5-Bot
	☑	1	6		M6-Bot
	☑	1	7		M5-Top
	☑	1	8		M6-Top
	☑	1	3		M3
	☑	1	4		M4
	☑	1	1		M1
	☑	1	2		M2

Already-Installed Modules

	☐	Proje…	Obje…	Erecti…	Objec…

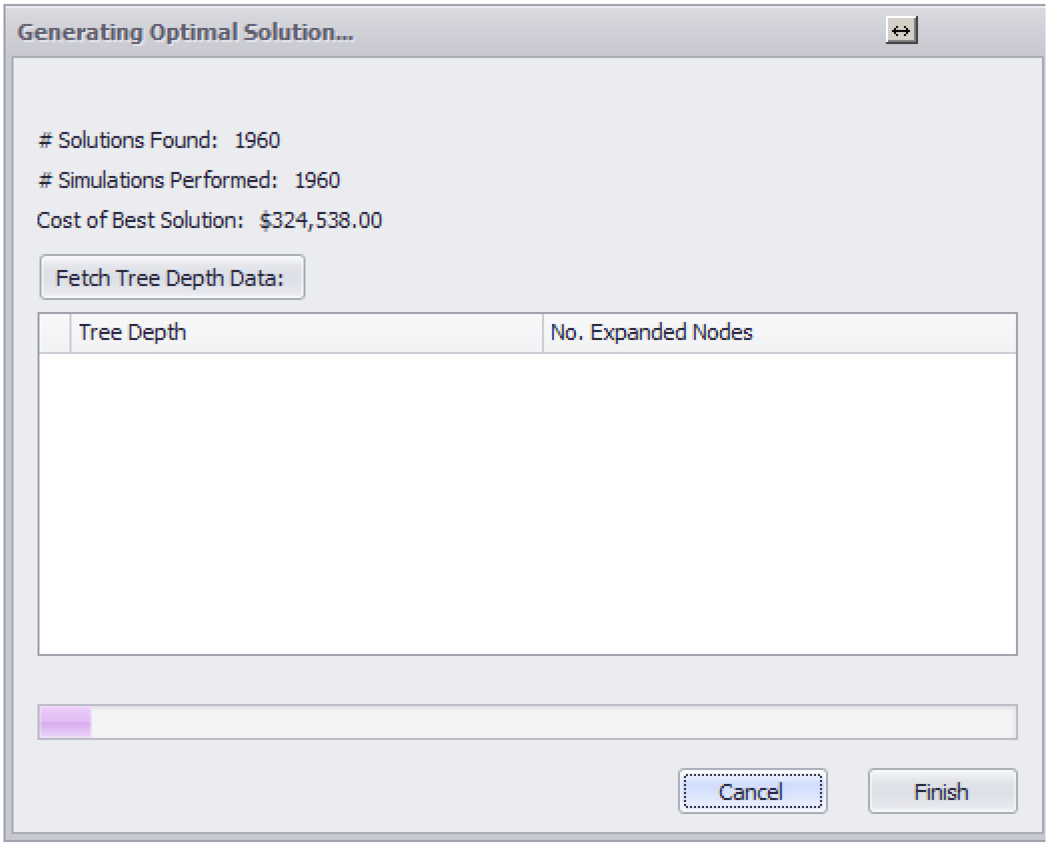
Modules to be Installed

	☐	P…	… ▲	…	Object Name
	☐	1	1		M1
	☐	1	2		M2
	☐	1	3		M3
	☐	1	4		M4
▸	☐	1	5		M5-Bot
	☐	1	6		M6-Bot
	☐	1	7		M5-Top

Project ID: 1
Crane Clearance(ft): 12
Lift Frame Rigging ID: 135

Figure 4.12: Module selection window in MCTS program

As soon as the MCTS algorithm starts to search for a solution for the problem, another screen appears, showing the status of the search. Figure 4.13 shows a screenshot of the new "Generating Optimal Solution" window where the number of iterations performed, the number of feasible solutions found, and the cost of the best solution found is shown. Note that the cost functions are designed to reflect the optimum solution with the objective of minimizing the number of crane locations, relocations, and reconfigurations during the project. These objective cost functions must be properly defined for each activity, based on cost details of the project, to provide an actual cost estimate for the solution scenario.



Generating Optimal Solution...

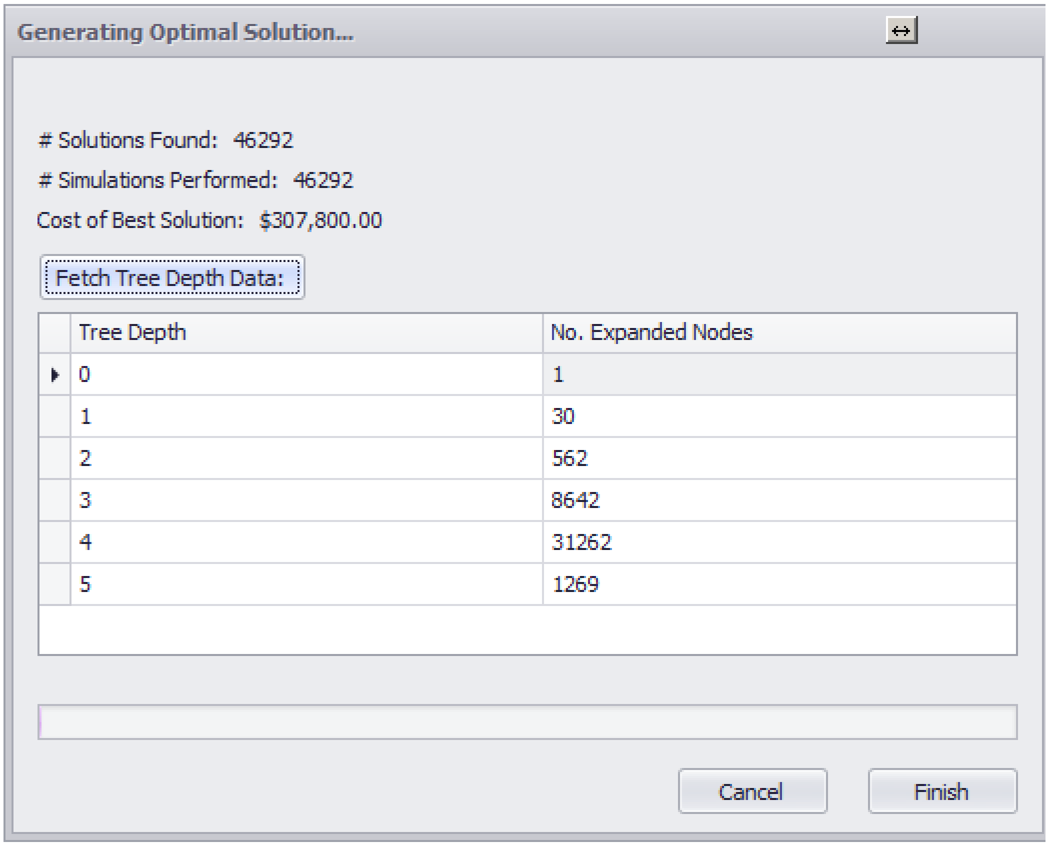
Solutions Found: 1960

Simulations Performed: 1960

Cost of Best Solution: $324,538.00

Figure 4.13: MCTS program search update window in real time

Further, the status and depth of the search tree can be examined by clicking on "Fetch Tree Depth Data". This allows access to the depth of the tree search that has been explored by the MCTS algorithm in real time. Figure 4.14 lists the number of nodes that have been expanded at each tree level after about 46,000 iterations of the MCTS Process.



Solutions Found: 46292

Simulations Performed: 46292

Cost of Best Solution: $307,800.00

Tree Depth	No. Expanded Nodes
0	1
1	30
2	562
3	8642
4	31262
5	1269

Figure 4.14: MCTS program update window showing the tree search depth update

The MCTS program was able to achieve the same solution optimality as the one developed using heuristic knowledge (see Chapter 3) for the case study involving the installation of 8 modules. It took anywhere from between 8,000 to 87,000 MCTS process iterations before the program was able to find the optimum solution for this problem. This can be explained by the randomness incorporated into the MCTS algorithm. Further, while all of these solutions ranked the same based on objective functions (number of crane relocations, reconfigurations, and locations used), these solutions were not identical. Table 4.2 shows a sample of an optimum solution found by the MCTS program for this sample case study.

Table 4.2: Example of installation plan found using MCTS for sample case study

Installation Iteration #	Module	Crane Configuration	Crane Location
1	M3	C1	L7
2	M4	C1	L7
3	M2	C1	L1
4	M1	C1	L1
5	M5B	C1	L1
6	M6B	C1	L1
7	M6T	C1	L1
8	M5T	C1	L1

4.4.3 MCTS Solution Validation

As in Section 3.4.4, in order to validate the output plan produced by the MCTS program, an animation for installing the modules in accordance with the formulated installation plan is created, reviewed, and scrutinized in order to ensure that the crane-module technological constraints are fulfilled. Then, the optimality of the solution is reviewed. Since the problem size of this case study is small, it can be confirmed that there is no better solution available.

The enforcement of the constraints explained in Section 2.2.1 is first checked based on the provided solution. Figure 4.15 shows the enforcement of neighbor module precedence relationships. When the crane moves to L1, M2 is installed during Installation Iteration 3 in order to satisfy the requirement. Figure 4.16 demonstrates the enforcement of the bottom-top precedence relationship. As shown in screenshots from the animation, the bottom module, M6B, is installed during Installation Iteration 6 before the corresponding top module M6T is installed during Installation Iteration 7.

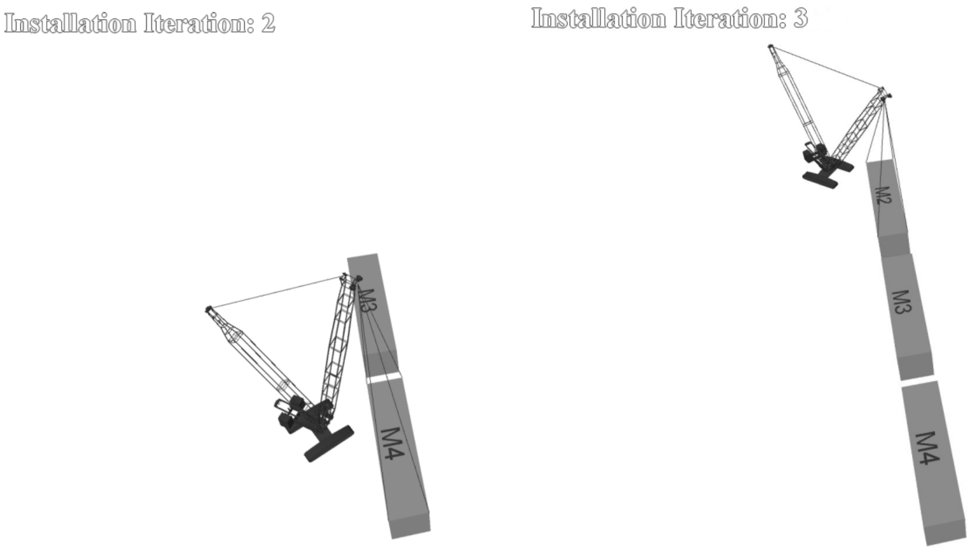


Figure 4.15: Validation of the neighbor module precedence relationship

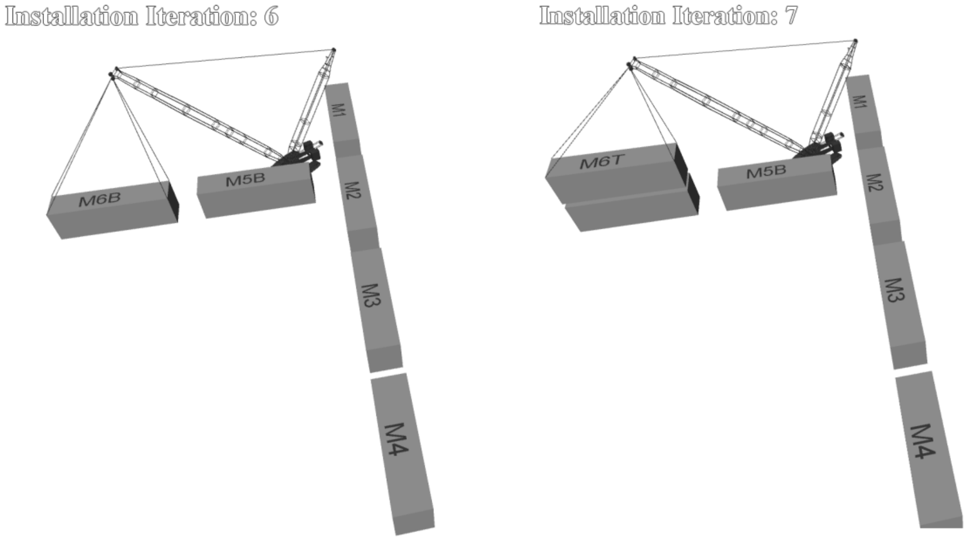


Figure 4.16: Validation of the `bottom-top precedence relationship

After verifying that the solution satisfies all the constraints, the optimality of the solution is investigated. In order to check whether there is a better solution to this small sample problem, three strategic questions, similar to those in Section 3.4.4, were proposed and answered. First, is it possible to install all 8 modules using only one crane location (compared to the current solution, which requires two locations)? Second, is it possible to have fewer crane movements between different module installations? Third, is it possible to use any crane location with a shorter straight travel distance?

Answer to the first question: No single crane location would provide a complete coverage for all module installations. As a result, a second location is required to provide the coverage for installing all modules in this problem.

Answer to the second question: No, considering that the crane installs all 8 modules from 2 distinct locations, one crane movement (i.e., from L7 to L1) is the minimum requirement, as given in the current solution.

Answer to the third question: It is necessary to distinguish between the straight path distance between crane locations and the actual path that the mobile crane travels on site. Currently, the straight path distance is used as a measure of the traveling distance, and the locations L1and L7 are in a straight line from North to South. Using these 2 crane locations for crane movement thus provides the shortest crane travel distance.

4.5 Program Limitations

The proposed MCTS implementation for module installation planning described above currently has limitations on the size of the problem for which it can be applied. While the program was able to find the best solution for a small 8 module case example within a few minutes, it performs very poorly on larger problems. For example, the attempt to solve the real case study presented in Section 3.4, with 78 modules, failed to provide a decent solution. Even after running the MCTS program for over 24 hours, the solution obtained was not satisfactory.

The shortcoming of the program in producing acceptable solutions for larger problems can predominantly be attributed to the amount of time that it takes to complete any single simulation. Due to large tree branching factors, the depth of the solution in the tree, and the number of constraints that need to checked and verified during each simulation, the MCTS was not able to sample from enough branches of the solution tree in order to efficiently guide the direction of the search. In the next section, several suggestions are provided for ways to enhance the MCTS program in order to make it feasible for use on larger problems in the future.

4.6 Future Research and Developments

As previously observed by researchers, the basic implementation of MCTS would commonly result in weak results, if the method is not enhanced for tackling the specific problem (Brown et al. 2012). Enhancement in selection policy and simulation policy can be explored in order to obtain better quality results for module installation planning problems. In this section, some of these potential enhancements that could be implemented in the future are reviewed.

4.6.1 Selection Policy Enhancement

The selection policy controls the balance between exploration and exploitation. The UCT selection policy (Kocsis and Szepesvari 2006) presented earlier has two parts, which balance the exploration and exploitation within tree by means of the C_p

value. Kocsis and Szepesvari (2006) and Chaslot (2010) have stated that the value of C_p needs to be tuned experimentally, based on the problem domain.

In addition, other selection policies can be tested and used to further improve the MCTS algorithm. The objective Monte-Carlo (OMC) selection policy (Chaslot et al. 2006), Probability to be Better than Best Move (PBBM) selection policy (Coulom 2006), and UCB1-TUNED selection policy (Gelly and Wang 2006) are other examples of selection policies that can be explored.

Finally, the selection strategy can be combined with expert knowledge to obtain good quality solutions for cases when the number of simulation iterations is lower than the potential available branching factors. For example, less promising crane locations can initially be excluded from the solution space based on expert knowledge, and then can be slowly added to the problem, based on the number of simulation iterations and the available computational budget. In the literature, this is referred to as progressive widening (Chaslot et al. 2006, Coulom 2006, Chaslot 2010). The same strategy can be applied to eliminate some of the module options and crane configuration options initially at each tree level, and these can be progressively added back to the search tree. The heuristic rules presented in Chapter 3 could be utilized as guiding principles in pruning and limiting options in terms of the crane locations, crane configurations and module selections that are considered at each level of the tree.

4.6.2 Simulation Policy Enhancement

Like selection policy enhancement, enhancements to simulation policy can lead to significant performance gain in MCTS (Chaslot 2010). A random simulation policy can be changed to a pseudo-random simulation policy by use of knowledge regarding where quality solution can be found in the solution space. As stated in Section 4.2, the full implementation of heuristic knowledge, based on Chapter 3, resulted in very slow program performance. For future research, it is suggested that different level of heuristic knowledge implementation be tested. It is highly desirable to choose a simulation policy that strikes a balance between improving the quality of the simulation outcome and the number of simulation iterations performed.

4.7 Conclusion

In this chapter, after reviewing the basic MCTS components, a detailed specification of the proposed MCTS structure for solving the module installation planning problem was presented. A sample case study involving the installation of 8 modules is used to demonstrate the detailed process, step by step, for three iterations. Then, a software toolkit is presented in which a MCTS specific to module installation planning is implemented. The software was used to solve a small case study.

While the software performed well in solving the small case study example within several minutes, there were difficulties in using the software to solve the larger case study, involving installation of 68 modules. Even after 24 hours, the MCTS program was not able to produce a good quality solution. The number of iteration per minute was low and the quality of the simulation result was not impressive.

In order to effectively apply the MCTS methodology to a larger problem, suggestions are made for future research. Investigating changes in the selection policy, utilizing a progressive widening strategy and properly utilizing domain knowledge during the simulation stage are possibilities for future research.

Chapter 5: Conclusions

5.1 Research Summary

Recent trends towards a higher level of modularization in industrial projects, and budget and schedule pressures during the construction process have driven efforts to scrutinize and research different areas of industrial project construction. Different areas of on-site module installation using heavy-duty mobile cranes have been subject of much research, as this is recognized as a high-cost task on the critical path of the project. The overall module installation planning and installation sequence determination is a complex process, and can be a daunting task when multiple constraints, requirements, and preferences must be taken into account simultaneously. Practitioners and planners can spend weeks trying to create a feasible installation sequence and plan that meets the project schedule requirements. Crane location, crane configuration and module installation sequences are determined, largely on a trial-and-error basis. In addition, multiple project constraint requirements, including bottom-top module precedence relationships, neighbor module precedence relationships, module blocking precedence relationships, and the minimum clearance between multiple crane on site must be checked before finalizing a project schedule. Developing a formalized framework and methodology for effectively planning module installation can significantly contribute to the successful planning and delivery of new industrial construction projects.

In this thesis, two new frameworks and approaches for module installation planning were developed and proposed. Each of these frameworks ensures that multiple project constraints are satisfied while project costs are minimized. In the proposed heuristic framework, a schedule is built iteratively, by ranking and selecting options based on different criteria and preferences while constraints are checked to ensure that only feasible options are selected. In the proposed MCTS-based framework, the search for a solution is based on a biased random operation. Planning toolkits have been developed based on these two frameworks for the purpose of module installation planning on industrial sites. The heuristic-based toolkit provides the planner the ability to iteratively (or automatically) build a complete site installation plan for each crane while all project constraints are checked and satisfied, and project cost and duration are minimized. The MCTS-based toolkit is used to demonstrate the potential of utilizing this novel method for module installation planning on a smaller (eight-module) case study. The above results were achieved by: (i) defining the problem and constraints in detail, (ii) developing heuristic rules based on the subject matter expert approach and problem characteristics, and (iii) recognizing the shortfalls of the proposed heuristic method and utilizing past developments and achievements in an MCTS-based method for planning module installation. These developments achieved the objectives set out in Chapter 1.

In Chapter 2, the module planning problem was described, and the multiple constraint criteria that need to be taken into account in module planning were explained and demonstrated in detail. Many cost factors and how they affect the overall crane operation costs were recognized and reviewed. Also, an overall picture of the available input information for the module planning problem was given, and the desired output result from any module installation planning algorithm was demonstrated. Chapter 2 also provided an overview of previous work and research in mobile crane operation planning on industrial sites. It was noted that the past research and advancement in site-wide module construction planning automation using mobile cranes is limited, and does not provide industry practitioners a solid framework or tool with which to plan module installation for new industrial projects. Finally, the module installation planning problem was compared to similar problems in operation research. The similarities and differences between module installation planning and established problems in operation research were discussed before the path to solve the problem was laid out.

In Chapter 3, a heuristic based module installation planning framework was presented, in which all project constraints and requirements identified in Chapter 2 are enforced while crane operation costs are minimized. Initially, the strengths and advantages of heuristic optimization in solving large complex combinatorial problems similar to the module installation planning problem were established. Then, a proposed methodology that built a complete installation plan in an iterative process was presented. In this method, the module installation sequence, crane configurations and crane locations were ranked based on multiple criteria. The step-by-step processes of the method were presented and explained using a detailed flowchart. Next, a small (eight-module) sample case study was used to demonstrate the detailed calculations and steps involved in solving the module installation problem. Finally, the methodology was automated using VBA for Access, and a software toolkit was developed in order to solve a larger size problem. A real case example was used to demonstrate the features of the developed tool. The result of both the simplified eight-module case study and real case study involving 78 modules were validated using subject matter experts, as well as visualization animation. It was established that the proposed heuristic framework can be used efficiently in addressing large module installation planning problems.

In Chapter 4, an artificial intelligence based Monte-Carlo Tree Search (MCTS) methodology was proposed, where the optimum result is searched for through biased sampling of the solution space. In this chapter, the basis and general framework of MCTS was initially presented. The four steps of the MCTS (i.e. Selection, Expansion, Simulation and Backpropagation Steps) were explained in detail. The applications of MCTS to similar problems were reviewed. It was noted that MCTS was previously used to solve production management problems, printing process scheduling, the travelling salesman problem, and job shop scheduling. In order to apply MCTS to module installation planning, a specific, detailed methodology and framework were established. The Upper Confidence Bound for the Tree (UCT) was used as the selection policy, and the reward of each solution was defined as the inverse of the cost of module installation. The method involved the expansion of one node in the tree for each simulation. Further, a random simulation policy was used to generate a feasible solution for each simulation iteration. The cost of each simulated module installation plan and the number of simulated solutions through each node were backpropagated and stored for the calculation of UCT values. It was decided that the lowest cost solution found during the Simulation Step would be kept as the final solution to the problem. The processes were explained using detailed flowcharts, and step by step sample calculations. The procedures were shown for a sample case study, involving the installation of 8 modules. A software toolkit was developed in the VB.Net framework. In this toolkit, the proposed MCTS methodology was implemented for module installation planning. It effectively solved the small case problem to optimality within a few minutes. The program, however, has yet to provide a good solution to the larger case study, involving the installation of 78 modules. Further research and development were proposed in the chapter, in order to improve the quality of the solution for the larger problem.

5.2 Research Contribution

This thesis proposes two frameworks for module installation sequencing and planning automation for industrial construction sites. A heuristic framework is developed based on the current practice of subject matter experts, the cost factors in crane operation, and an understanding of the problem structure. The heuristic-based methodology ensures all the project constraints are accounted for when preparing a module installation plan. The method builds an installation plan by iteratively selecting options that minimize crane operation cost. When implemented, this heuristic method is capable of producing good quality module installation plans for large projects within reasonable timeframes. It considerably reduces the time and effort required to plan module installation on industrial construction sites.

Further, an artificial intelligence based MCTS methodology is proposed to automate generating an optimum module installation plan. The general MCTS framework is customized to properly represent module installation planning. The detailed specifications and processes of the MCTS as it applies to the modular installation problem were presented. The implemented MCTS was also able to produce a good quality plan for the smaller sample case study. The advances presented in this thesis provide the foundation for the application of MCTS to module planning. Future research areas are suggested that can improve the search algorithm functionality for larger practical size case problems in module planning. The proposed MCTS methodology searches for plan optimality with minimum domain knowledge. It can easily adapt to project-specific preferences by changing the objective function.

Due to the complex nature and set-up of industrial construction sites, the two suggested frameworks ensure the feasibility of the plan while searching for optimality using the following strategies in the list below.

1. Bottom-top module precedence relationships of the modules are properly defined and incorporated into the plan generation algorithm.
2. Neighbor module precedence relationships are established between sets of modules, and installation sequences are determined taking this requirement into account.
3. Module-blocking precedence relationships are realized and the proper installation sequences and crane locations were chosen to ensure that the

installation can be completed without any clashes arising between previously installed modules and new installations.

4. A proper distance was allowed between two cranes working simultaneously.
5. Both the heuristic rules and the objective functions in MCTS were defined in such a way that the number of crane locations, number of crane relocations, number of crane reconfigurations and crane total moving distance were minimized.
6. In the heuristic methodology, subject matter expert knowledge and preferences could be easily taken into account by building the schedule on a step-by-step basis and allowing the user to make the final selection at the end of each installation sequence.
7. In the MCTS methodology, module installation sequences can be predefined and the tree can be built to respond to these predefined installation sequences.

The industrial contributions of this research include formalizing the module installation planning procedure, and developing a heuristic-based planning toolkit that can support practitioners in planning module installation on industrial sites as an effective decision support tool. The developed tool significantly reduces the effort of the practitioner involved in preparing the complete module installation plan. This is done by incorporating the established precedence relationship requirements and searching for the plan optimum where the number of crane foundations, relocations, and reconfigurations, and crane moving distances are minimized. By allowing the user to either prepare a complete schedule automatically, or build a schedule on a step-by-step basis, the knowledge and opinions of the subject matter expert can be efficiently utilized to improve the installation plan quality. Further, the ability of the tool to provide a cost value for the generated solution would allow the practitioner to carry out many what-if scenarios before finalizing the installation plan.

5.3 Recommendations for Future Research

The two methodologies presented in this thesis have different types of limitations that can be addressed in future research. The heuristic-based methodology is currently limited to the case of utilization of one crane on site. Also, it lacks the capacity to take into account the module rigging requirements when preparing the module installation plan. Further, the current heuristic framework is geared toward planning and sequencing the module installation, and does not address the scheduling challenges where specific installation dates are determined based on project date requirements.

The proposed MCTS methodology implementation was only able to solve a small case study efficiently. Due to large branching factors in the larger practical case example, the MCTS toolkit was not able to generate enough sample solutions in a reasonable time frame. In order to overcome the limitations for the heuristic-based and MCTS-based methodologies in planning module installation for practical case, the following suggestions are made for future research:

1. Expanding the heuristic-based methodology would allow the use of multiple cranes simultaneously on site.
2. The module rigging requirements could be added as a ranking criterion to the heuristic based framework.
3. To minimize number of crane location options for each module, a clustering technique could be utilized. The current grid options, provided as crane locations 3 feet apart, result in large branching factors, which the MCTS algorithm then needs to search through.
4. The selection policy for the MCTS algorithm can be fine-tuned in order to improve the searching procedure. The proposed UCT selection policy can be adjusted by changing the parameters to fit the problem better. Also, other selection policies can be investigated.
5. Knowledge-based rules can be incorporated in the selection policy to limit the size of the solution space that the MCTS algorithm needs to search. In Section 4.6.1, progressive widening was suggested as an example for the future enhancement of the MCTS algorithm.
6. The simulation policy used in the MCTS method can be enhanced by use of heuristic-based knowledge to improve the optimality of each simulation solution. The simulation policy should be a balance between the speed with which a solution can be simulated as well as the quality of the solution generated.

References

A1A Software 2009. 3D Lift Plan (software). <http://www.3dliftplan.com/> Accessed 03/ 2016.

Al-Hussein, M., Alkass, S., & Moselhi, O. (2001). "An algorithm for mobile crane selection and location on construction sites." *Construction Innovation*, *1*(2), 91-105.

Al-Hussein, M., Alkass, S., & Moselhi, O. (2005). "Optimization algorithm for selection and on site location of mobile cranes." *Journal of Construction Engineering and Management*, *131*(5), 579-590.

Baker, B. M., & Ayechew, M. A. (2003). "A genetic algorithm for the vehicle routing problem." *Computers & Operations Research*, *30*(5), 787-800.

Bektas, T., Erdogan, G., & Røpke, S. (2011). "Formulations and branch-and-cut algorithms for the generalized vehicle routing problem." *Transportation Science*, *45*(3), 299-316.

Bell, J. E., & McMullen, P. R. (2004). "Ant colony optimization techniques for the vehicle routing problem." *Advanced Engineering Informatics*, *18*(1), 41-48.

Boctor, F. F. (1990). "Some efficient multi-heuristic procedures for resource-constrained project scheduling." *European Journal of Operational Research*, *49*(1), 3-13.

Bouzy, B. (2005). “Associating domain-dependent knowledge and Monte Carlo approaches within a Go program.” *Information Sciences*, *175*(4), 247-257.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., & Colton, S. (2012). “A survey of Monte Carlo tree search methods.” *IEEE Transactions on Computational Intelligence and AI in Games*, *4*(1), 1-43.

Chaslot, G. (2010). “Monte-Carlo tree search.” *Maastricht: Universiteit Maastricht*.

Chaslot, G. M. J. B., De Jong, S., Saito, J. T., & Uiterwijk, J. W. H. M. (2006, January). “Monte-Carlo tree search in production management problems.” In *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence* (pp. 91-98).

Chaslot, G. M. J. B., Saito, J. T., Bouzy, B., Uiterwijk, J. W. H. M., & Van Den Herik, H. J. (2006). “Monte-Carlo strategies for computer go.” In *Proceedings of the 18th BeNeLux Conference on Artificial Intelligence, Namur, Belgium* (pp. 83-91).

Chen, S. M., Chen, P. H., & Chang, L. M. (2013). “A framework for an automated and integrated project scheduling and management system.” *Automation in Construction*, *35*, 89-110.

Chevallier, N., & Russell, A. D. (1998). “Automated schedule generation.” *Canadian Journal of Civil Engineering*, *25*(6), 1059-1077.

Chua, D. K., Nguyen, T. Q., & Yeoh, K. W. (2013). "Automated construction sequencing and scheduling from functional requirements." *Automation in Construction*, *35*, 79-88.

Coulom, R. (2006, May). "Efficient selectivity and backup operators in Monte-Carlo tree search." In *International Conference on Computers and Games* (pp. 72-83). Springer Berlin Heidelberg.

Dong, N., Fischer, M., Haddad, Z., & Levitt, R. (2013). "A method to automate look-ahead schedule (LAS) generation for the finishing phase of construction projects." Automation in Construction, 35, 157-173.

Gelly, S., & Wang, Y. (2006, December). "Exploration exploitation in go: UCT for Monte-Carlo go." In *NIPS: Neural Information Processing Systems Conference On-line trading of Exploration and Exploitation Workshop*.

Gendreau, M., Hertz, A., & Laporte, G. (1994). "A tabu search heuristic for the vehicle routing problem." *Management Science*, *40*(10), 1276-1290.

Ghiani, G., & Improta, G. (2000). "An efficient transformation of the generalized vehicle routing problem." *European Journal of Operational Research*, *122*(1), 11-17.

Ha, M. H., Bostel, N., Langevin, A., & Rousseau, L. M. (2014). "An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size." *Computers & Operations Research*, *43*, 9-19.

Haas, C. T., & Lin, K. (1995). “An interactive database system with graphical linkage for computer aided critical lift planning.” In *Proceedings of the 12th International Symposium on Automation and Robotics in Construction (ISRC), Warsaw, Poland* (pp. 313-324).

Han, S., Lei, Z., Bouferguène, A., Al-Hussein, M., & Hermann, U. (2014, January). “Integrated visualization and simulation for lifting operations of modules under congested environment.” In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 31, p. 1). Vilnius Gediminas Technical University, Department of Construction Economics & Property.

Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., & Lim, P. (1987). “Expert system for construction planning.” *Journal of Computing in Civil Engineering*, *1*(4), 253-269.

Hermann, U., Hendi, A., Olearczyk, J., & Al-Hussein, M. (2010). “An integrated system to select, position, and simulate mobile cranes for complex industrial projects.” In *Proceedings of Construction Research Congress* (pp. 267-276).

Hornaday, W. C., Haas, C. T., O'Connor, J. T., & Wen, J. (1993). “Computer-aided planning for heavy lifts.” *Journal of Construction Engineering and Management*, *119*(3), 498-515.

Kocsis, L., & Szepesvári, C. (2006, September). “Bandit based Monte-Carlo planning.” In *European Conference on Machine Learning* (pp. 282-293). Springer Berlin Heidelberg.

Kolisch, R. (1996). “Efficient priority rules for the resource-constrained project scheduling problem.” *Journal of Operations Management*, *14*(3), 179-192.

Kolisch, R., & Hartmann, S. (1999). “Heuristic algorithms for the resource-constrained project scheduling problem: Classification and computational analysis.” In *Project Scheduling* (pp. 147-178). Springer US.

Kolisch, R., & Hartmann, S. (2006). “Experimental investigation of heuristics for resource-constrained project scheduling: An update.” *European Journal of Operational Research*, *174*(1), 23-37.

Koo, B., Fischer, M., & Kunz, J. (2007). “A formal identification and re-sequencing process for developing sequencing alternatives in CPM schedules.” *Automation in Construction*, *17*(1), 75-89.

Laporte, G., Gendreau, M., Potvin, J. Y., & Semet, F. (2000). “Classical and modern heuristics for the vehicle routing problem.” *International Transactions in Operational Research*, *7*(4-5), 285-300.

Laporte, G., Mercure, H., & Nobert, Y. (1986). “An exact algorithm for the asymmetrical capacitated vehicle routing problem.” *Networks*, *16*(1), 33-46.

Lei, Z., Han, S., Bouferguène, A., Taghaddos, H., Hermann, U., & Al-Hussein, M. (2014). “Algorithm for mobile crane walking path planning in congested industrial plants.” *Journal of Construction Engineering and Management*, *141*(2), 05014016.

Lei, Z., Taghaddos, H., Han, S., Bouferguène, A., Al-Hussein, M., & Hermann, U. (2015). “From AutoCAD to 3ds Max: An automated approach for animating heavy lifting studies.” *Canadian Journal of Civil Engineering*, *42*(3), 190-198.

Lei, Z., Taghaddos, H., Olearczyk, J., Al-Hussein, M., & Hermann, U. (2013). “Automated method for checking crane paths for heavy lifts in industrial projects.” *Journal of Construction Engineering and Management*, *139*(10), 04013011.

Lin, K. L., & Haas, C. T. (1996). “An interactive planning environment for critical operations.” *Journal of Construction Engineering and Management*, *122*(3), 212-222.

Matsumoto, S., Hirosue, N., Itonaga, K., Yokoo, K., & Futahashi, H. (2010). “Evaluation of simulation strategy on single-player Monte-Carlo tree search and its discussion for a practical scheduling problem.” In *Proceedings of the International MultiConference of Engineers and Computer Scientists* (Vol. 3, pp. 2086-2091).

Mikulakova, E., König, M., Tauscher, E., & Beucke, K. (2010). “Knowledge-based schedule generation and evaluation.” *Advanced Engineering Informatics*, *24*(4), 389-403.

Morad, A. A., & Beliveau, Y. J. (1991). “Knowledge-based planning system.” *Journal of Construction Engineering and Management*, *117*(1), 1-12.

Navinchandra, D., Sriram, D., & Logcher, R. D. (1988). “GHOST: project network generator.” *Journal of Computing in Civil Engineering*, *2*(3), 239-254.

Olearczyk, J., Bouferguène, A., Al-Hussein, M., & Hermann, U. R. (2014). "Automating motion trajectory of crane-lifted loads." *Automation in Construction*, *45*, 178-186.

Olearczyk, J., Lei, Z., Ofrim, B., Han, S., & Al-Hussein, M. (2015, January). "Intelligent Crane Management Algorithm for Construction Operation (iCrane)." In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction* (Vol. 32, p. 1). Vilnius Gediminas Technical University, Department of Construction Economics & Property.

Osman, I. H. (1993). "Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem." *Annals of Operations Research*, *41*(4), 421-451.

Pisinger, D., & Ropke, S. (2007). "A general heuristic for vehicle routing problems." *Computers & Operations Research*, *34*(8), 2403-2435.

Pop, P. C., Matei, O., Sitar, C. P., & Chira, C. (2010, June). "A genetic algorithm for solving the generalized vehicle routing problem." In *International Conference on Hybrid Artificial Intelligence Systems* (pp. 119-126). Springer Berlin Heidelberg.

Pop, P. C., Pintea, C., Zelina, I., & Dumitrescu, D. (2009, April). "Solving the Generalized Vehicle Routing Problem with an ACS-based Algorithm." In *BICS 2008: Proceedings of the 1st International Conference on Bio-Inspired Computational Methods Used for Solving Difficult Problems: Development of Intelligent and Complex Systems* (Vol. 1117, No. 1, pp. 157-162). AIP Publishing.

Reddy, H. R., & Varghese, K. (2002). “Automated path planning for mobile crane lifts.” *Computer‐Aided Civil and Infrastructure Engineering*, *17*(6), 439-448.

Reddy, S.D., Varghese, K. and Srinivasan, N., 2007. “A computer-aided system for planning and 3D-visualization of multiple heavy lifts operations.” InProc., 24th Int. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*.

Rimmel, A., Teytaud, F., & Cazenave, T. (2011, April). “Optimization of the nested Monte-Carlo algorithm on the traveling salesman problem with time windows.” In *European Conference on the Applications of Evolutionary Computation* (pp. 501-510). Springer Berlin Heidelberg.

Runarsson, T. P., Schoenauer, M., & Sebag, M. (2012). “Pilot, rollout and Monte Carlo tree search methods for job shop scheduling.” In *Learning and Intelligent Optimization* (pp. 160-174). Springer Berlin Heidelberg.

Sargent, R.G. (2005, December). “Verification and validation of simulation models.” *In Proceedings of the 37th conference on Winter simulation*, 130-143. winter simulation conference.

Shah, R. K. (2014). “A new approach for automation of location-based earthwork scheduling in road construction projects.” *Automation in Construction*, *43*, 156-169.

Shaked, O., & Warszawski, A. (1995). “Knowledge-based system for construction planning of high-rise buildings.” *Journal of Construction Engineering and Management*, *121*(2), 172-182.

Taghaddos, H., AbouRizk, S., Mohamed, Y., & Hermann, U. (2010, May). “Simulation-based multiple heavy lift planning in industrial construction.” In *Proceedings of the 2010 ASCE Construction Research Congress (CRC)*.

Taghaddos, H., AbouRizk, S., Mohamed, Y., & Hermann, U. (2011). “Simulation-based auction protocol for resource scheduling problems.” *Journal of Construction Engineering and Management*, *138*(1), 31-42.

Waly, A. F., & Thabet, W. Y. (2003). “A virtual construction environment for preconstruction planning.” *Automation in Construction*, *12*(2), 139-154.

Westover, L., Olearczyk, J., Hermann, U., Adeeb, S., & Mohamed, Y. (2014). “Analysis of rigging assembly for lifting heavy industrial modules.” *Canadian Journal of Civil Engineering*, *41*(6), 512-522.

Williams, M., & Bennett, C. (1996, June). “ALPS: the automated lift planning system.” In *Computing in Civil Engineering* (pp. 812-817). ASCE.

Wu, I. C., Borrmann, A., Beißert, U., König, M., & Rank, E. (2010). “Bridge construction schedule generation with pattern-based construction methods and constraint-based simulation.” *Advanced Engineering Informatics*, *24*(4), 379-388.