A REAL-TIME CRANE SERVICE SCHEDULING DECISION SUPPORT SYSTEM (CSS-DSS) FOR CONSTRUCTION TOWER CRANES

by

AMIR ZAVICHI TORK
B.S. Ferdowsi University, Mashhad, Iran (2007)
M.S. Polytechnic University, Tehran, Iran (2010)

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Civil, Environmental and Construction Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2013

Major Professor: Kaveh Madani

ABSTRACT

The success of construction projects depends on proper use of construction equipment and machinery to a great extent. Thus, appropriate planning and control of the activities that rely on construction equipment could have significant effects on improving the efficiency of project operations. Cranes are the largest and most conspicuous construction equipment, widely used in typical construction sites. They play a major role in relocation of materials in horizontal and vertical directions on construction sites. Given the nature of activities relying on construction cranes in various stages of a project, cranes normally have control over the critical path of the project with the potential to create schedule bottlenecks and delaying the completion of the project. This dissertation intends to improve crane operations efficiency by developing a new framework for optimizing crane service sequence schedule. The crane service sequence problem is mathematically formulated as an NP-complete optimization problem based on the well-known Travel Salesman Problem (TSP) and is solved using different optimization techniques depending on the problem's size and complexity. The proposed framework sets the basis for developing near-real time decision support tools for on-site optimization of crane operations sequence. To underline the value of the proposed crane sequence optimization methods, these methods are employed to solve several numerical examples. Results show that the proposed method can create a travel time saving of 28% on average in comparison with conventional scheduling methods such as First in First out (FIFO), Shortest Job First (SJF), and Earliest Deadline First (EDF).

To My Lovely Family

Ahmad, Masoumeh, Maryam, and Mona

ACKNOWLEDGMENTS

First and foremost, I would like to express my deep gratitude to my advisor Dr. Kaveh Madani for his continuous help, encouragement, and support. He has always been an excellent advisor and an incredible friend. I really appreciate his effort and guidance to achieve the objectives of the presented work. This dissertation would never have been made possible without him. I would like to extend my gratitude to my Ph.D. committee members, Dr. Amr Oloufa, Dr. Petros Xanthopoulo, and Dr. Omer Tatari for their contributions, valuable suggestions and advices to improve the quality of the work. Finally, I would like to acknowledge Dr. Amir Behzadan who gave me the opportunity to attend UCF.

I am immensely grateful to my parents, Ahmad Zavichi and Masoumeh Daghighbin and my two beloved sisters, Maryam and Mona, who have always been a great source of motivation and support in every minute of my life.

I would also like to take this opportunity to thank all of my friends especially Shahrzad Jalali, Bardia Mehdizadeh, Masoud Malekzadeh, Marzieh Ghasemi, Amir Najafi Ardabili, and Bijan Roboubi for their sincere friendship and help during the past years. Finally, special thanks to Mr. Shahram Shahrestani for believing in me and giving me the opportunity to work at Port of Los Angeles.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 General Overview

Today, the necessity of completing the projects timely, in-budget, and with high quality requires proper use of construction equipment in order to ensure the project's success. Construction equipment alone places a great financial burden on the project and if not utilized efficiently will result in monetary loss. Among different classes of construction equipment, cranes are one of the most important lifting devices on the construction sites. Besides their expensive cost, they play a central role on construction sites, and often activities that rely on crane service fall on the project's critical path. This implies that monitoring and analysis of crane operation has the potential to enhance the project productivity.

Reducing the crane's travel time yields to a shorter crane cycle, and consequently shorter delays are expected in receiving the material by the crews waiting for crane service. This will increase total productivity of the crane operation as well as those in need of the crane service (Shapira, Rosenfeld, & Mizrahi, 2008). One of the major causes of fatalities in construction phase is the use of cranes or derricks during the lifting operation (Beavers, Moore, Rinehart, & Schriver, 2006). Therefore, reducing the crane operation time could also lead to a safer workplace as the risk of accident is significantly higher when the crane hook is moving. Previously, increasing the crane operation productivity has been investigated via two approaches: first, through optimizing facility layout in design phase by locating the tower crane and supply locations such that it reduces the total crane travel time. Second, through using adds-on technologies such as vision system, collision detection system, etc. to help the crane operator better navigate the motions of the end manipulator especially when the operator line of sight is obstructed.

Another potential way to reduce the crane travel time, which is the subject of this research, is to prioritize the crane service optimally in a tight working schedule when several requests are received simultaneously, while each request might have additional constraints such as priorities and deadlines. Traditionally, the operator uses his/her personal judgment or the help of an on-duty superintendent to schedule the material lifting tasks; however, due to human involvement, this manual scheduling would not lead to an efficient schedule necessarily.

1.2 Construction Crane Operations

Crane operation cycle consists of two work modes: stationary and dynamic. Stationary mode is experienced during loading or unloading, when the hook does not have any motion. The dynamic mode is experienced when the hook is moving, including hoisting (vertical), trolleying (radial) and slewing (circular) movements. The total time associated with the crane's dynamic mode comprises the crane's travel time in a working cycle. Operators commonly activate both trolleying (radial) and slewing (horizontal) movements simultaneously, which are actuated by a single lever (joy stick) followed by the hoisting (vertical) movement. The stationary mode of the crane cycle operation is often significant in low-rise and mid-rise construction while the dynamic mode is often significant in high-rise buildings.

Traditionally, on a construction job site, the tower crane operator is in contact with the working deck and ground via radio communication. The signal men improve the crane's operator limited line of sight by adopting a set of hand signals (will be described in more detail in section 1.4.3.1). Project Managers are in charge of every decision made during the project and as they are unable to handle the entire project, several superintendents are employed to monitor and coordinate the project in a hierarchical structure. Yet numerous important decisions are assigned to the laborers and operators without providing them with proper tools and information. For example, in tower crane operation, at any given time, requests might be sent to the crane operator to be fulfilled. In case of having several simultaneous requests, normally, the operator makes the decision of sequencing the requests, and most often, it is based on the FIFO (first-in-first-out) concept. At the best case scenario, requests are sent to the superintendent a day in advance and he schedules the crane operation for the following day. Then, the schedule will be given to the crane operator early in the morning and the operator works based on the pre-planned schedule for the rest of the day. If there are any items that might put anyone's safety at risk or cause delay to the project completion time, they will get immediate attention in coordination with the superintendent. In addition, parties or crews that fail to pre-plan their requested items for a timely submittal to the superintendent will have to wait until the next available crane time to get their items moved.

The current primitive crane management is also dependent and centralized on superintendent decisions that might be biased. Human judgment might be insufficient to develop to optimal plan to fulfill the requests. The current material handling operation results in longer than optimal operation time which can eventually alter the project's critical path in a tightly-scheduled project, heavily involving with crane operations. Neither the crane operator, nor the superintendent can enhance the crane scheduling and operation without utilizing a computational power. A partial schedule for an example construction site for a part of a day is shown in Table 1(Brandt & Robinson, 2012).

Table 1: An exemple of tower crane operation schedule

Job description	Time
Formworks lifting to 10th floor deck (remove rubbish from deck to ground with trips down)	7:00 to 8:00 a.m.
Pluming materials to 6th floor	8:00 to 8:30 a.m.
Chilled water pipes to 6th floor	8:30 to 9:00 a.m.
Locate rebar to 10th floor	9:00 to 9:45 a.m.
Raise column formwork	9:45 to 10:30 a.m.

Current methods for crane operation scheduling are manual, time consuming and do not guarantee the optimal process. Therefore, the need for a decision support system, to automate the crane service scheduling is eminent. Use of an integrated, central computational unit that receives the requests from different participant (crews) on the site and considers various constraints for each request could potentially increase the productivity of crane operation and consequently the tasks that are tightly related to the service that the crane provides. The following advantages could be considered for such a system:

- Transparency of the decision making process which is heavily dependent upon the on-duty superintendent ;

- Reduced the crane operation time by using a decision support system that utilizes structured reasoning approaches such as statistics and operations research in tandem with computer power to ensure the best possible practice for the work flow;
- Reduce dependency on the on-duty superintendent; and
- Reduced crew's waiting time to receive material

In short, the goal of this research is to develop a decision support system to reduce the dependability to assure optimality of crane operations. The new system is designed to be used in tandem with the existing technologies routinely used on the jobsites.

1.3 Problem Description

In heavy construction job sites with tight scheduling, where availability of material on installation points is directly related to the crane accessibility, low productivity of the crane has inverse impact on time and budget of the projects. Crane productivity is not only influenced by the crane operator's skill in navigating the crane but also influenced by the decisions he/she or the in charge superintendent needs to make to fulfill the tasks. Assume that the crane operator receives several simultaneous requests from crews to receive material from different supply locations in the jobsite. The crane operator would use his or her visual assessment and personal judgment or at best case, would use the help of an on-duty superintendent to decide the order of tasks to fulfill. This decision-making process could be biased upon certain activities or may be fulfilled in the order that the requests were received (the request which comes first will be served first (FIFO)). Because of human involvement, there are no conditions that lead consistently to the optimized working time for fulfilling the outstanding requests and as a result, longer operation time which can eventually -in a tight scheduled project which is involved heavily with crane operation- alter the project's critical path. The following small-sized problem shows the complexity of decision making when simultaneous requests are received.

1.3.1 Small Size Problem Illustration

The following small-sized example is used to illustrate the complexity of the crane service sequence problem (CSSP). Assume that only three requests have been sent to the crane operator to be fulfilled at a specific time period. Figure 1 shows a bipartite travel time graph in which the weights on connecting arcs are travel time associated to them. In addition, solid lines in Figure 1 represent outstanding material requests that must be fulfilled by the crane operator.

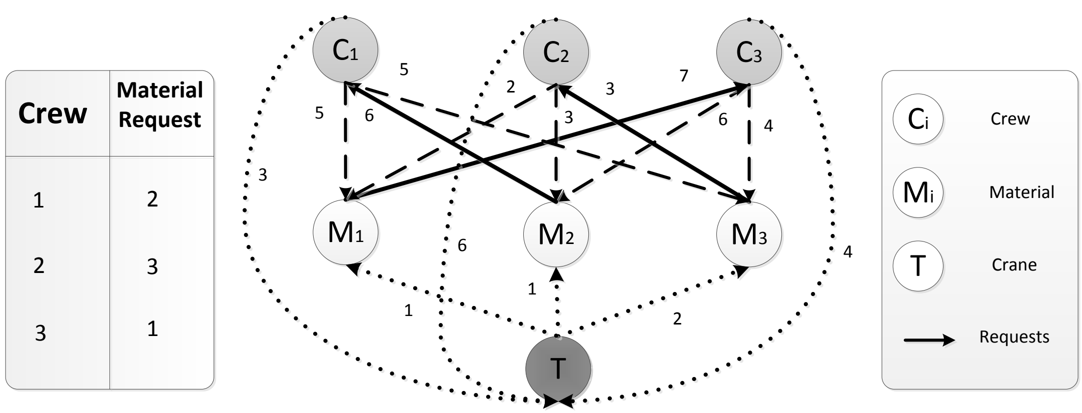

Figure 1: Travel time graph and service requests matrix for CSSP

As can be seen, crew 1 needs material from material storage 2; crew 2 needs material from material storage 3, and so on. Now the question for the crane operator is the order of nodes in bipartite graph that must be visited, starting from the crane's initial location and returning to the same location, in order to fulfill all outstanding requests and minimize the travel times simultaneously. For this small-sized problem, there would be 3! (permutation of requests) different ways to fulfill the requests as depicted in Figure 2. In this specific problem there is only one optimal solution with regards to the total travel time that is equal to 27. Since the permutation grows significantly with the number of requests, the challenge is to design a robust computational and automated method to determine the optimal sequence of tasks that yields the minimum completion time.

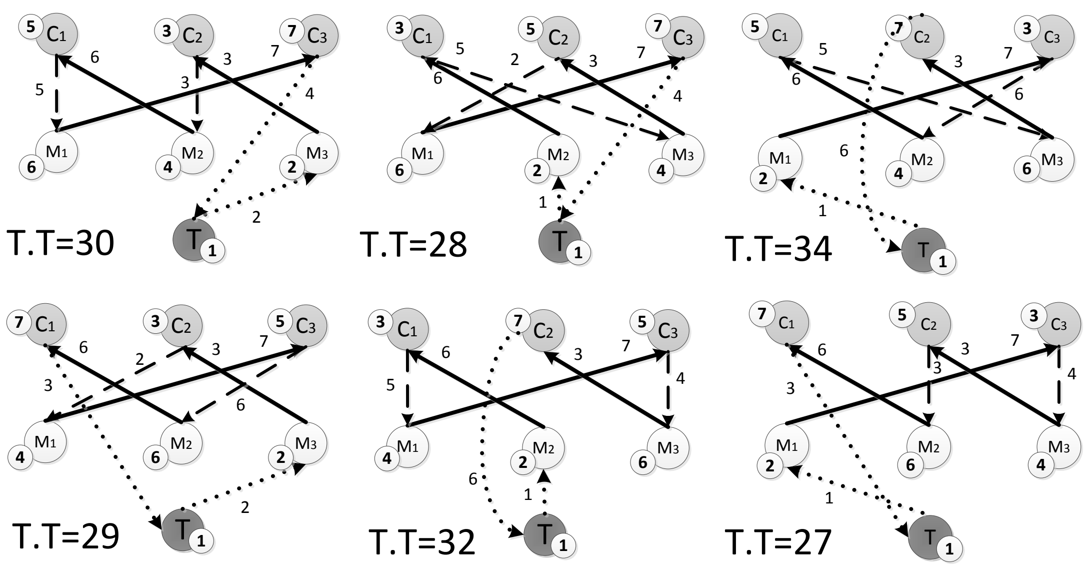

Figure 2: Possible request fulfillment sequences and their total travel time (T.T) for the CSSP example

Noted that, in CSSP each crew may or may not send a crane service request to receive material from a specific storage area at a certain time.

1.4 Crane Operation Literature Review

1.4.1 Hoisting Time or Hook Travel Time

Often, the objective of the crane optimization problem is to reduce cost of the crane operation through minimizing the path the hook traverses in order to fulfill certain tasks. Therefore, first step in such studies is to define the hook travel time between initial and target nodes. Predicting hook travel time for crane-dependent activities enable site managers to improve utilization of the crane activities. Accuracy in prediction of the crane travel time leads to a better scheduling and planning of construction projects especially where the crane plays a critical role in the project. In modeling the crane operation time, two approaches have been used by researchers so far: analytical and statistical models (Leung & Tam, 1999).

1.4.1.1 Statistical

In statistical model, the main quantifiable factors in crane's hook travel time is identified based on the expert knowledge from field studies, and a regression model that examines the contribution of different variables will be developed. Leung and Tam (1999) used multiple linear regression models to predict the hoisting time for the crane operation. Tam et al. (2002) developed a nonlinear neural network to predict the relationship between identified factors as dependent variable and transportation time as independent variable. Tam and Tong (2003) used artificial neural network to predict hoisting times combined with genetic algorithm to get an enhanced tower crane and supply points locations among possible locations while demand points were fixed, considering having a limited type of materials.

1.4.1.2 Mathematical or analytical

The number of factors considered in analytical method is limited compare to the statistical prediction model. The factors that have been used by researchers are loading and unloading geographical positions, crane specifications such as trolley speed in different directions (vertical, angular and radial) and a few factors to consider site layout and operator skills. To the best knowledge of authors Zhang et al. (1999) developed a mathematical model for the first time using Cartesian coordination of the supply, demand and crane locations. Since 1996, this mathematical model has been remained almost intact and is used in different researches with no or minor changes (Huang, Wong, & Tam, 2011; C. M. Tam, Tong, & Chan, 2001; Zhang et al., 1999).

1.4.2 Crane Location Optimization

Assume a construction site with one in-service crane; several crews are waiting to receive material representing demand nodes, and several warehouses or material supply locations representing supply nodes at the job site. The general approach in crane location optimization is that the demand locations are determined, and based on the site layout, structural design and spatial constraints, potential crane and supply locations will be defined. Among different alternatives, the optimal supply and crane locations will be determined using different optimization methods in order to minimize the total crane travel time. Zhang et al. (1996) used stochastic simulation model to optimize the location of a single tower crane. They predicted requests frequency based on the work schedule as an input for their simulation model. Zhang et al. (1999) used the same approach for location optimization of a group of tower cranes. They first classified the tasks into different groups based on their closeness to the cranes, and then Monte-Carlo simulation is used for each crane to find the crane location. Tam et al. (2001) used genetic algorithm to find the optimized location for supply points and one crane location among permissible points by minimizing the transportation time. Mathematical approach was used for hook travel time in the aforementioned research. Tam and Tong (2003) used artificial neural network to predict hoisting times combined with genetic algorithm to get an enhanced tower crane and supply locations among possible locations while demand locations are fixed, considering having a limited type of materials. Huang et al. (2011) used analytical method for calculating the hoisting time and formulated the problem as a mixed-integer-linear programming and used commercial package LINGO to solve the problem.

Table 2: Researches in tower crane hoisting time prediction and location optimization in litreture

Hoisting Time Prediction	Objectives	Solution Methods	Citations
Stochastic	Predicting hoisting time	linear regression model (SPSS)	Leung and Tom (1999)
	Predicting hoisting time	Nonlinear neural network(s), general regression neural network & group method of data handling	Tam, Leung, Liu (2002)
	Hoisting time prediction & tower crane location optimization	Artificial neural network &genetic algorithm	Tam and Tong (2003)
Mathematical	Single tower crane location optimization	Monte Carlo simulation	Zhang , Harris & Olomolaiye (1996)
	Group of tower crane location optimization	Mathematical formulation	Zhang, Harris, Olomolaiye & Holt, (1999)
	Supply location optimization around tower crane	Genetic algorithm	Tam, Tong, & Chan, (2001)
	Tower crane and supply locations optimization	Mixed integer programming	Huang, Wong, & Tam,(2011)

1.4.3 Planning of Physical Crane Motion

Cranes are among the most expensive piece of equipment in many construction projects as well as freight terminal operations, shipyards, and warehouses. Despite their wide range of application, a vast majority of cranes still in use do not feature the advanced automation and sensor technologies. A typical crane operator uses visual assessment of the jobsite conditions which may be enhanced through a signalperson on the ground.

Technologies have been used to improve crane coordination, in navigating the motions of the end manipulator (i.e. crane hook) and other body parts (e.g. boom, jib, trolley) from the moment a load is picked up until it is delivered to the desired location, is categorized into three sections: *vision enhancement* to provide the crane operator with a better view from lifting location and probably eliminate the need for a signal person, *Automatic or semi-automatic navigation* to ensure a smooth maneuvering and *motion planning & collision avoidance* to provide operator with an efficient path and improve job-site safety by avoiding collision incidents.

1.4.3.1 Vision enhancement

Although the tower crane operators have a bird's eye view of the job-site, they are yet unable to view the entire working area clearly due to the distance between crane operator and the lifting point that may reach even to a couple of hundreds feet's. In addition to distance that exacerbates the operator vision, obstacles also obstruct the operator's line of sight (refers to blind spot) that necessitate the use of vision enhancement tools. As the construction process progresses, obstructions are increasing which adds to the visibility problem. Job site conditions such as poor lighting also limit crane operator vision. The aforementioned problem has been partially addressed using signal person, and adoption of a set of hand signals to facilitate the communication. In some cases that the operators do not have a direct line of sight to the signal person due to existence of a large obstruction, and thus using a third person (tag man) in a location where both signal person and crane operator can see him/her is necessary. He/she is responsible to convey the signals from signal person to the crane operator (Everett & Slocum, 1993b). This practice has miscommunication issues which may lead to low productivity, safety problems and finally accident if not addresses properly.

To reduce the disadvantages of the current practice, Everett & Slocum (1993a) used a video camera which was installed at the tip of the mobile crane's boom pointing straight down with an angle adjustment mechanism, and a monitor in the crane operator's cab enabled with zooming function to enhance the crane operator's vision (CRANIUM). They reported a saving of 16-21% in crane's travel time by using this system. More recently Shapira et al. (2008) reported a practice of mounting a live video system on tower cranes in order to tackle this problem as well. The system had a moving trolley that could move horizontally along the crane jib with a high-resolution autofocus video camera, and a control screen installed in the cab. The mean saving time for the total travel time was reported 14-29%, while the saving percentage for total crane cycle time (including loading and unloading) was 11-26%.

1.4.3.2 Automatic or semi-automatic Navigation

In many construction sites, cranes are involved in a repetitive operation to lift material between two specific locations (e.g., concrete mixer and casting area). In this case, a significant portion of the cycle time is spent on maneuvering the crane's hook manually. One of the potential uses of automation is to release workers mind from repetitive tasks since performing repetitive tasks over a long period would lead to ignorance, distraction, and carelessness, which were identified as the most common causes of defects (Josephson & Hammarlund, 1999). With the same objective, Rosenfeld (1995) developed a prototype using automatic navigation in crane's maneuvering. The prototype was built and tested on an overhead gantry crane in laboratory environment. The system had the capability of memorizing different pre-planned benchmarks and the operator was able to interrupt as needed. Total travel time saving was reported to be within the range of 15 to 50%. Subsequently, Rosenfeld and Shapira (1998) examined the feasibility of utilizing semi-automatic control devices on the existing tower cranes and their potential to enhance the productivity. They also conducted a cost-benefit analysis and alleged that *"as the cost of high-technology components continues to decline and that of construction labor continues to rise"*, therefore installing semi-automatic navigation systems on existing tower cranes are technologically feasible and economically possible. Their system was tested on a small scale tower crane model and on an indoor 5-ton full scale electric overhead traveling crane. The observed saving in travel time using this system was reported 15-40%.

1.4.3.3 Motion Planning and Collision Avoidance

To consider the constraints imposed by the workspace where the crane operates, and take into account the possibility of collisions between crane and surrounding objects, path planning methods are borrowed from mechanical engineering and computer science to address this problem (Spong, Hutchinson, & Vidyasagar, 2006). In motion planning, construction cranes are considered as a multi-degree-of-freedom robotic manipulator, and the problem is to determine a path between initial and end positions while avoiding collision with objects in its workspace[1]. After the path is selected, *inverse kinematic* methods are often used to find the required crane motions. The *inverse kinematic* problem is to determine the value of joints variables given the end-effector position and orientation. On the contrary, *forward kinematic* is the inverse problem which is to determine the end-effector position and orientation in terms of joint variables.

---

[1] The workspace of a manipulator is total space that the end effector can cover as manipulator executes all possible motions.

Forward kinematic always has a unique solution whereas inverse kinematic may or may not have a solution. Even if a solution exists, it may or may not be unique. Several researchers have done elaborative studies in this field (AlBahnassi & Hammad, 2012; S. Kang & E. Miranda, 2006; Lei, 2011; Olearczyk, 2010; Sivakumar, Varghese, & Ramesh Babu, 2003).

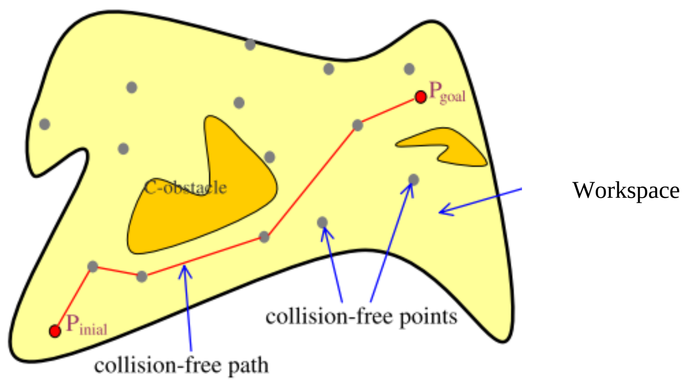

Figure 3: Motion planning in workspace (ShihChung Kang & Eduardo Miranda, 2006)

Table 3 shows the summary of studies that has been done in physical crane motion planning section.

Table 3: Summary of studies in planning of physical crane motion section

Category	Objective	Solution Methods	Citations
Vision enhancement	Travel time reduction/safety	Mounting camera system (mobile crane, 16-21%)	Everett & Slocum (1993)
	Travel time reduction/safety	Mounting camera system (tower crane, 14-29%)	Shapira, Rosenfeld & Mizrahi (2008)
Semi/automatic navigation	Travel time reduction	Navigation system (indoor overhead, 15-50%)	Rosenfeld (1995)
	Feasibility study	Navigation system (small scaled tower crane model and indoor overhead crane, 15-40%)	Rosenfeld & Shapira (1998)
Motion planning, and collision avoidance	Travel time calculation considering obstacles	Motion planning & inverse kinematic	Kang & Miranda (2006)
	Collision avoidance/ safety	Motion planning & simulation	AlBahnassi & Hammad (2012)

1.5 The gap in body of knowledge

The missing link between these two bodies of research *(optimization of crane layout pattern, planning of physical crane motions)* that have been discussed so far is the need for a tool that helps the crane operators to decide the sequence of fulfilling service requests send by working crews on the jobsite that yields to maximum production rate, and minimum operations time and consequently cost. This phase is happening before the operator starts actuating the crane. This gap of knowledge has been identified in the present research and is referred to the *decision-making phase*. In this phase, the operator prioritizes crane service requests and create a job sequence list given constraints such as idle times of working crews, significance of ongoing crew’s tasks, task deadline, and total resource idle times. Figure 4 shows the schematic overview of major areas with high potential for automation in crane operations. *Crane Layout Pattern* and *Planning of Physical Crane Motion* were addressed in the literature review sufficiently.

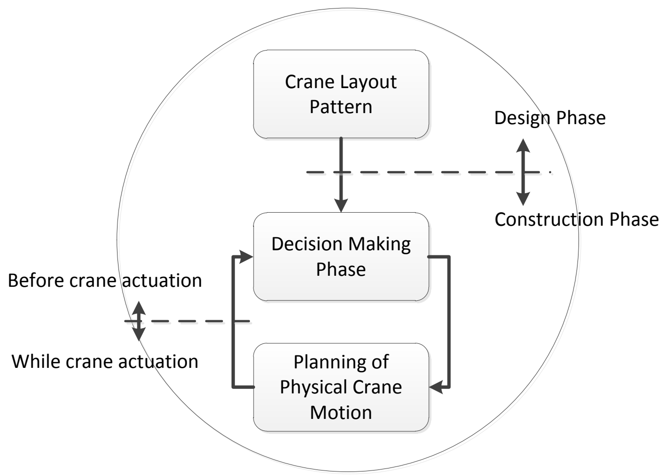

Figure 4: Schematic Overview of Crane Automation Process

The direction of arrows in Figure 4 relates to the flow of information during the crane operation life span in a job site. The *Crane Layout Pattern* phase occurs before the project start date, in which the main objective is to find the optimized location for tower crane(s) as well as supply locations with regards to the current or future demand locations. As soon as the crane operation starts and requests for crane operation arise as the project progress, two other phases of crane automation process (*Decision Making Phase* and *Planning Physical Crane Motion*) are working to fulfill the requests. For example, given a list of outstanding requests to be fulfilled, in *Decision Making Phase* the optimal order of fulfillment will be calculated, and based on that the crane is moving toward the specified node using the *Planning of Physical Crane Motion phase*.

Figure 5 shows an overview of different crane operation related researches. In this section we focus on creating the decision making phase.

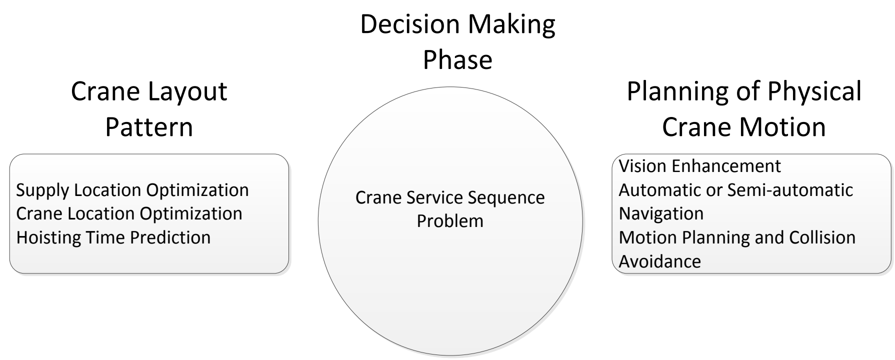

Figure 5: Crane Operation Automation Potential Section

1.6 Research Objectives

In order to address the limitation in the current state of knowledge, the overall goal of this research is to:

Design a robust decision making tool to aid the crane operator directly to fulfill the requests considering the dynamic and evolving construction work condition in order to yield the minimum completion time and consequently increase the efficiency of the project.

In order to achieve the main objective, the following milestones are pursued:

- Study the required parameters for tower crane travel time prediction and develop an applicable yet easy to use model as a basis for other goals of the research.
- Investigate the feasible Operations Research methods to formulate the Crane Service Sequence Problem (CSSP).

- Use mathematical modeling to obtain optimal solution for the problem.
- Develop heuristic methods to appropriately address the generalized CSSP considering additional constraints such as deadlines and priorities which may arise; and evaluate the performance of the algorithms.
- Devise and implement a decision support system which utilizes the algorithms that have been constructed in order to help the scheduler.

1.7 Methodology

This study has identified a potential way to decrease tower crane operation time through prioritizing the crane service sequence in construction job site by using optimization methods. In order to efficiently achieve the goal of this research, three main objectives are addressed:

Objective 1*: modeling the tower crane travel time:* one of the crane operation optimization's objectives is to reduce the crane operation's cost through minimizing its transportation time. Therefore, the first step in crane operations improvement is to define the hook's travel time, traversed between the initial and target nodes. The hook travel time prediction for crane-dependent activities enables the site manager to improve utilization of crane activities. Accuracy in travel-time prediction model leads to a better scheduling and planning of construction projects, especially in projects in which crane plays a critical role. Tower cranes are often located in a central place in construction job sites and because of this special configuration, a mathematical modeling based on polar coordinates system is developed for travel time calculations. Modifications have been added to the current body of knowledge to be better representative of the real-world applications.

***Objective 2**: crane service sequence optimization:* in this section, improving tower crane operation efficiency is investigated through prioritizing the concurrent requests. It was discussed in the problem description that numerating all possibilities (brute-force method) is not possible especially when the number of requests increase. Therefore, an exact method which is a deviation of a well-known combinatorial optimization problem, "Traveling Salesman Problem", is proposed to find the optimal solution among all alternatives. The Polar coordinates system is used to construct the travel time matrix between supply and demand locations in the job site.

***Objective 3**: considering deadline and significance of ongoing tasks (priority) as additional constraints for crane service sequence problem:* the CSSP complexity is increased if additional constraints such as priority and deadline exist in the operation. Priority between requests often happens for interrelated activities when some tasks have precedence over others. In addition, requests might have deadline when an activity starts by a specific time based on the project schedule and thus the material must be available by that time. Dealing with additional constraints simultaneously and ensuring the least travel time would not be possible for the crane operator without using a robust computational tool. Because of these extra constraints (priority and deadline) a penalty based genetic algorithm is proposed, which considers constraints simultaneously and guarantees a decent sequence within an acceptable computational time.

1.8 Dissertation Outline

This dissertation is organized into five main chapters. Chapter 1 was a general overview of the current crane operation methods followed by a discussion of problem complexity. In chapter 2, an innovative method is devised to convert crane service sequence problem to a well-known combinatorial optimization problem. The solution to this optimization problem is the optimal sequence with minimized crane travel time. Chapter 3 describes a special genetic algorithm designated to provide an efficient output in a practical computational time for the crane service sequence problem when requests having additional constraints. Chapter 4 discusses the details of the decision support system developed and implemented in this research. Finally, chapter 5 documents the results and presents a comprehensive summary of the dissertation, the main contributions of the research, and recommendation for future studies. The information presented in the chapters of this dissertation is supplemented by additional coding details in several appendixes.

CHAPTER 2: ENHANCED CRANE OPERATIONS IN CONSTRUCTION USING SERVICE REQUEST OPTIMIZATION

Cranes play a major role in relocation of materials in horizontal and vertical directions on construction job sites. They are among the high-demand equipment in construction operations due to their key role in providing material for downstream. Thus, they have been the focus of research on using new technologies as well as utilizing scheduling and optimization methods to improve crane operation efficiency. This chapter develops a service requests sequence optimization model for tower crane operations efficiency improvement. The suggested model uses integer programming and modifies the classical Traveling Salesman Problem (TSP) formulation for optimizing construction tower crane operations. Numerical examples demonstrate the mean saving time of 20-30% in total travel time depending on the number of simultaneous requests.

2.1 Introduction

Today, with the necessity of timely, on-budget and high quality operations in construction projects, effective use of construction equipment is essential to successful completion of projects. Construction equipment alone places a great financial burden on projects and can cause economic losses if not utilized efficiently. Cranes are among the costly construction equipment, playing an important role in construction sites, especially in high-rise building projects. Activities that depend on cranes are usually on the project's critical path. Thus, improving crane operations can enhance project performance significantly.

Construction cranes are classified into tower and mobile cranes. Tower cranes are popular due to their high horizontal and vertical reachability, as well as their small footprint, especially in dense areas around the world (Shapira, Lucko, & Schexnayder, 2007). Crane operation cycle consists of two work modes: stationary and dynamic. Stationary mode is experienced during loading or unloading when the hook does not have any motion. Dynamic mode is experienced when the hook is moving, including hoisting (vertical), trolleying (radial) and slewing (circular) movements. The total time associated with the crane's dynamic mode comprises the crane's travel time in a working cycle.

In this paper, we investigate the impact of prioritizing the crane-service sequence on the overall crane's travel time using a Traveling Salesman Problem (TSP)-based optimization model tailored specifically for construction tower crane operations. This model can assist on-site managers and crane operators in reducing crane travel time through crane-service sequence optimization. It should be noted that reducing crane travel time yields to a shorter crane cycle and consequently shorter delays for downstream crews in receiving the material which increases total productivity of crane operations as well as those activities in need of crane services (A. Shapira et al., 2008).

2.2 Background

Automating, planning and scheduling crane operation in order to improve total operation efficiency is of major interest due to the fact that cranes are the most instrumental material handling and lifting equipment in building construction. Their importance is not only due to their high cost but also due to the central role they play in transporting material on project sites.

Previous research on crane operation improvement mostly falls into two categories: crane layout pattern optimization and physical crane motions planning (A. Zavichi & Behzadan, 2011).

Crane layout pattern optimization deals with finding the optimum crane's location among available alternatives in order to satisfy criteria such as balancing workload and reducing crane total operation time, or minimizing spatial conflicts between cranes and other moving resources on the site. Zhang et al. (Zhang et al., 1996) used a Monte-Carlo simulation model to optimize the location of a single tower crane. In another study, Zhang et al. (Zhang et al., 1999) performed a location optimization for a group of tower cranes. Tam et al. (C. M. Tam et al., 2001), Tam and Tong (C. M. Tam & Tong, 2003), and recently Huang et al. (Huang et al., 2011) used various optimization techniques to optimize the locations of a single tower crane location and several supply points, keeping demand locations fixed.

Physical crane motions planning aims to develop methods and tools to help the crane operator navigate the motions of crane from the load pick up to delivery. The technologies and methods which have been used to improve physical crane motions planning can be categorized into three categories: vision enhancement to provide a better view from job site for the crane operator and eliminate the need for a signal person (Everett & Slocum, 1993a; A. Shapira et al., 2008); automatic or semi-automatic navigation to ensure a smooth maneuvering between loading and unloading locations (Rosenfeld, 1995; Rosenfeld & Shapira, 1998); and motion planning and collision avoidance to provide a path between loading and unloading locations while avoiding collision with objects surrounding the crane (S. Kang & E. Miranda, 2006; Lei, 2011; Olearczyk, 2010; Sivakumar et al., 2003).

Another potential way to improve the crane operation efficiency is minimizing the distance and time the crane travels through appropriate ordering of the sequence of locations the crane hook must visit in order to fulfill the service requested by the crews on a job site. A brute-force/exhaustive search method with the capability of analyzing up to 15 simultaneous requests was proposed by Zavichi and Behzadan (A. Zavichi & Behzadan, 2011). This method addressed the problem of determining the sequence of items to be relocated from their existing locations to their newly assigned locations using a tower crane, such that the total travel time is minimized. With the same objective, this paper presents a general mathematical model based on the well-known Traveling Salesman Problem (TSP) method for sequencing crane service requests without any limitation in the number of requests in order to minimize the total crane travel time and consequently to reduce total idle time for on-site crew and equipment.

2.3 Crane Service Sequencing Problem (CSSP)

Low efficiency of crane operations has inverse impacts on time and budget of the project. Crane operations efficiency is not only influenced by crane operator skills in navigating the crane but also by decisions the operator makes during operations. When there are several concurrent service requests, the crane operator uses his personal judgment, or uses the help of an on-duty superintendent to determine the sequence of crane activities. This decision-making process could be biased towards certain activities or operator's judgment, or may be simply based on the order of requests received, i.e., the requests that come first will be served first (FIFO). Lack of an optimal method for minimizing the working time for fulfilling the requests can result in a longer operation time and alteration of the project's critical path.

Crane service sequence problem can be best described using a graph, i.e. collection of vertices and connecting edges, associated with travel times. Figure 6 shows a site layout using a graph consisting of crew (C) and material (M) nodes in a construction site in which material is delivered from m storage areas to n working crews based on their requests using a central crane. Each crew node sends its requests to the crane operator to receive certain materials. The operator should then decide the order of request fulfillment, trying to minimize the total operation time as an objective with respect to different constraints such as the due time and precedence of tasks.

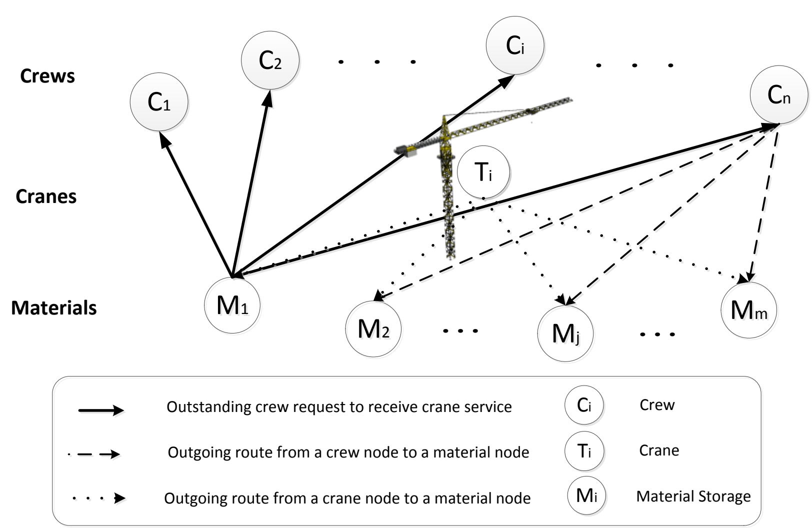

Figure 6: Graphical illustration of the site layout

Assuming that a subset of crews (*w* out of *n*) request material delivery; there are *w* alternatives for the decision maker (crane operator) to pick from as the first delivery service point. Once the first target crew node is chosen, the operator must load the requested material by this crew and deliver it to this node. If there is no new request, the process continues with *w-1* requests until all outstanding crane service requests are fulfilled. Given that the operator is free to choose any order of deliveries, there are a total of *w*! (permutation of *w*) possible ways to fulfill all requests. In order to minimize the overall travel time, the operator needs to find the optimal delivery sequence, which is not possible without serious computations even in small problems. For example, consider a small crane service sequencing problem (CSSP) depicted in Figure 7. This problem includes one crane, three request (crew) nodes, and three material loading nodes. The travel time associated with crane movement between each two nodes are shown next to the arcs connecting the nodes. Solid lines represent the outstanding crew requests (e.g. crew 1 needs material from material storage 2 and crew 2 needs material from material storage 3), dashed lines indicate the crane travel routes from crew nodes to material nodes, and dotted lines are outgoing routes from crane nodes to material nodes. In this problem, there are 3! (permutation of requests) possible movement sequences to fulfill the requests. Figure 8 shows sample sequences, each with a different total travel time. The crane operator must decide the order of locations to be visited in order to fulfill all outstanding requests while minimizing the total travel time. This problem has only one optimal solution with total travel time of 27 time units. For this small problem, the optimal order can be found through enumeration. However, the possible sequences grow significantly with increase in the number of requests. Thus, finding the optimum delivery order with the least travel time through enumeration (brute-force search) as done in previous research (A. Zavichi & Behzadan, 2011) is not mathematically efficient.

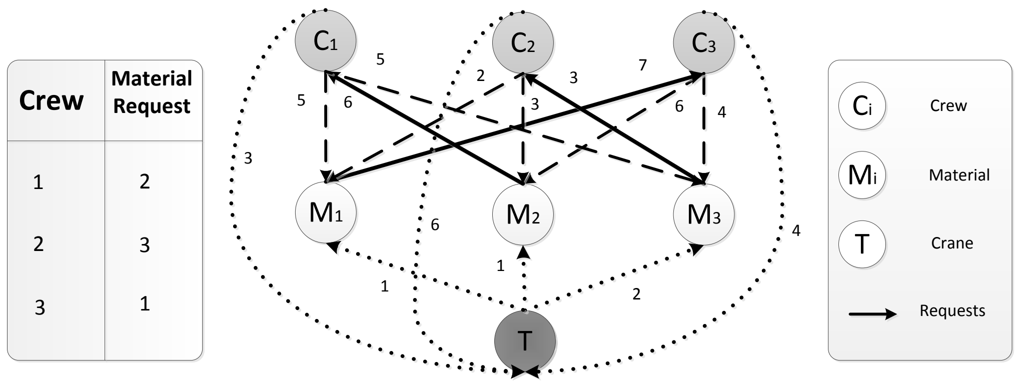

Figure 7: Travel time graph and service requests matrix for the example CSSP

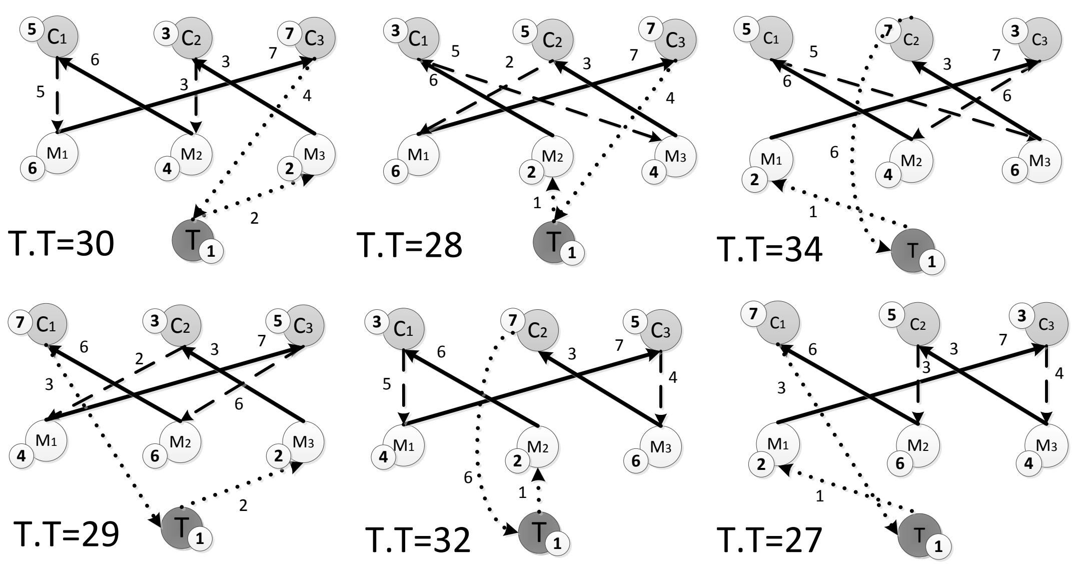

Figure 8: Possible request fulfillment sequences and their total travel time (T.T) for the CSSP example

In practice, different heuristics might be used in order to obtain the near-optimal operations sequence in absence of crane operations optimization tools. Three heuristic rationales for ordering the deliveries include fulfilling requests based on the first-in first-out/served (FIFO) method, fulfilling the nearest neighbor's request next or the nearest neighbor first (NNF) method, and fulfilling the request with the shortest travel time next or the shortest job first (SJF) method. While these methods can improve the crane operations efficiency to some extent, they do not generally result in the shortest path and least completion time and thus do not guarantee an optimal solution (Gutin & Punnen, 2004). Thus, the main objective of this paper is to develop an optimization method for solving the CSSP. To examine the efficiency of the heuristic order sequencing methods, crane travel time based on these methods is compared to the travel time using the optimization model proposed in this study. This will help identify the best heuristic method for order sequencing in absence of optimization model in the field.

2.4 Method

In principle, CSSP is similar to the Travelling Salesman Problem (TSP) –a well-known combinatorial optimization problem. In TSP, the salesman starts from an initial location, visits a prescribed set of cities, and returns to the original location (tour) in such a way that the total distance travelled is minimized and each city is visited only once (Gutin & Punnen, 2004). TSP is one the most notorious problems in Operation Research for being easy to explain but hard to solve (Wolsey, 1998). This problem is an NP-complete problem that has become representative of difficult combinatorial optimization problems. In TSP, starting at city 1, the salesman has

$n-1$ choices for the second city to visit and $n-2$ for the next choice, and so on. Thus, there are $(n-1)!$ possible tours in case of asymmetry (when the distance from city i to j is not equal to distance from city j to i) and $(n-1)!/2$ possible tours in case of symmetry.

CSSP can be formulated as a TSP, assuming that each request (starting from a material node and ending in a crew node) is a city and travel time associated with each link is the distance between two cities connected by the link. Based on this approach, the crane hook should fulfill requests by visiting request nodes and return to its initial position once no other outstanding request is left. This converts the CSSP to an asymmetric TSP, in which the cost (travel time) of moving from i to j is different from the cost of moving from j to i.

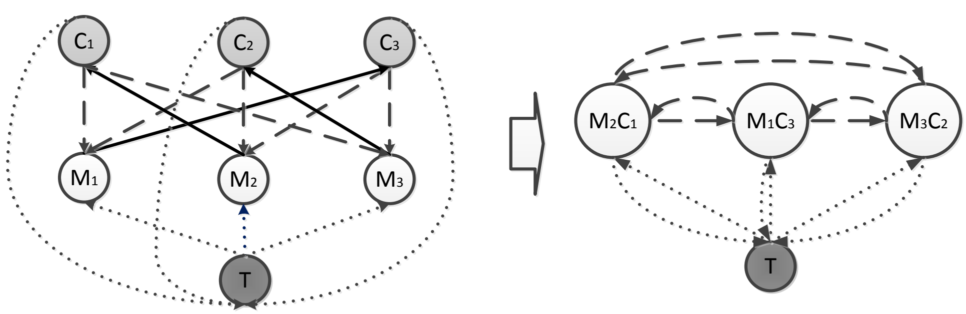

Figure 9: Transforming a sample CSSP (left) to a TSP (right)

Figure 9 shows how a sample CCSP can be converted to a TSP based on the suggested approach. The following points must be noted about the resulting TSP:

- Unlike TSP, cost (distance) is not only associated with the edges, but also with the nodes in the resulting TSP. The time associated with an edge relates to travel time from a delivery node to a new request node while the time associated with each node represents the travel time from a material node to the target delivery node requesting that material. The optimization problem targets minimizing the total time of traveling on the edges while the node-specific times are constant and not sensitive to routing options.
- Unlike TSP, in CCSP the crane (corresponding to the salesman in TSP) can visit one node (crew or material) more than once.
- CCSP is inherently asymmetric regardless of the initial graph characteristics.

The first step in solving CSSP is to develop a travel time (cost or distance) matrix (C) associated with the connecting arcs in the original CSSP problem (prior to conversion). This matrix is referred to as the location travel time matrix, which is a square ($n \times n$) matrix in the following form:

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & C_{ij} & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}$$

where n is the number of nodes in the graph and C_{ij} reflects the time that it takes for the crane's hook to travel from node i to node j.

2.4.1 Hook Travel Time Calculation

Reducing crane operations cost involves minimizing the transportation time of the crane for which a reliable estimation of hook's travel time is required. Statistical and analytical models (Leung & Tam, 1999) have been applied for crane travel time estimation. In statistical models, the main driving variables (e.g. loading and unloading locations, crane trolley velocities (vertical, angular and radial), site conditions, and operator's skill level) are identified based on knowledge from field studies and a regression model is used to examine the correlations between these variables and travel time. Leung and Tam (Leung & Tam, 1999) used multiple linear regression models and Tam et al. (C. M. Tam et al., 2001) developed a nonlinear neural network model to predict the relationship between the driving factors, as independent variables, and transportation time, as a dependent variable. In analytical models, the number of variables is limited compared to the statistical model. Zhang et al. (Zhang et al., 1996) developed an analytical model for tower cranes using the Cartesian coordinates of the supply, demand and crane locations. Since 1996, this mathematical model has remained almost intact and has been used in different studies (Huang et al., 2011; C. M. Tam & Tong, 2003; C. M. Tam et al., 2001; Zhang et al., 1999). Similarly, a polar coordinates system is used in this study in order to build the location travel time matrix.

2.4.1.1 Modeling transportation time using a polar coordinates system

Figure 10 shows a polar coordinates system with pole C and polar axis X, where C is the crane's base location $(r_{Cr_k}, \theta_{Cr_k}, z)$ and X is an arbitrary fixed direction from which other angles are measured. Cartesian locations (x, y, z) is used to calculate the radial distance (r) and angle (θ) using Equations 5 and 6:

$$\mathrm{r} = \sqrt[2]{\mathrm{x}^2 + \mathrm{y}^2} \tag{1}$$

$$\theta = \tan^{-1}\left(\frac{\mathrm{y}}{\mathrm{x}}\right) \tag{2}$$

where $\tan^{-1}\left(\frac{\mathrm{y}}{\mathrm{x}}\right)$ is interpreted as the two-argument inverse tangent which takes into account signs of x and y to determine the quadrant which θ lies in.

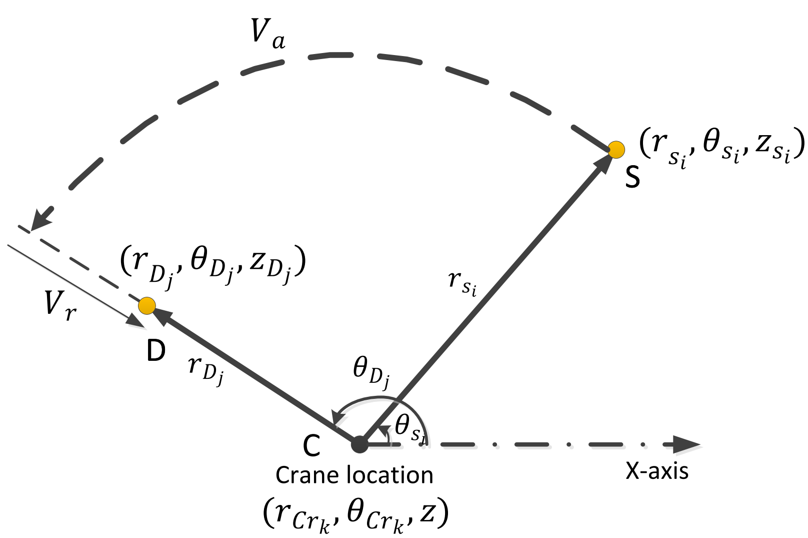

Figure 10: Polar crane coordinates

Trolley velocities in each direction can be based on crane manufacturing specifications (radial $\mathrm{V_r}(\mathrm{m/min})$, angular $\mathrm{V_a}(\mathrm{rpm})$, and vertical velocity $\mathrm{V_v}(\mathrm{m/min})$). Radial $(\mathrm{T_r^{(i,j)}})$ and angular ($T_a^{(i,j)}$) components of the hook's travel time between two locations i and j are calculated using Equations 3 and 4, respectively:

$$T_r^{(i,j)} = \frac{\left|r_{s_i} - r_{D_j}\right|}{V_r} \tag{3}$$

$$T_a^{(i,j)} = \frac{\left|\theta_{s_i} - \theta_{D_j}\right|}{V_a} \tag{4}$$

To rectify the possible underestimation of Zhang et al.'s model(Zhang et al., 1996) in estimating the vertical component of hook's travel time, in problems with difference between supply and node elevations (Figure 11), an extra motion (minimum hoisting height) is added to both sides of travel arcs (both loading and unloading locations). The minimum hoisting height depends on the type of material (e.g., loading steel bars needs more hoisting height than loading materials using a bucket), site topography, obstructions on the job site, and safety factors. Figure 6 shows how the minimum hoisting height for loading and unloading points can vary on job sites with and without elevation differences. Given that the minimum hoisting height is traversed two times, the vertical component of travel time can be calculated using Equation 5:

$$T_v^{(i,j)} = \frac{\left(\left|S_i^z - D_j^z\right| + 2\times\ h\right)}{V_v} \tag{5}$$

where h is the minimum hoisting height.

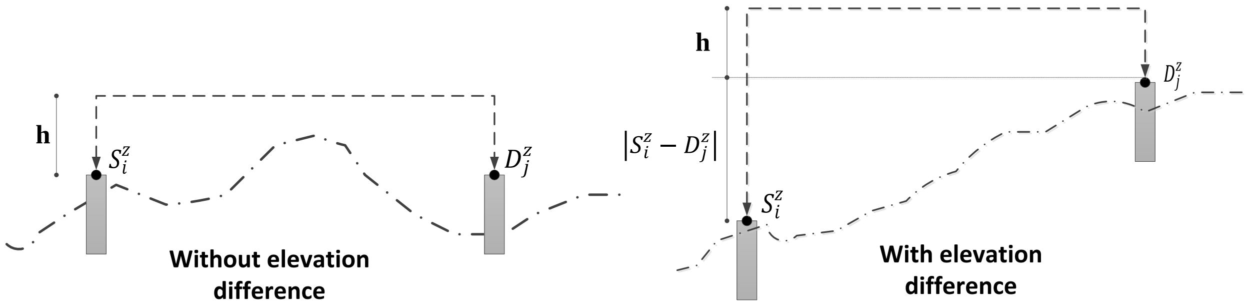

Figure 11: Hoisting height for supply and demand without/with elevation difference

Three parameters are used to account for operator's skill (α and β) and the site conditions (γ). α is the degree of overlap in radial and angular movements, i.e., to what extent the operator can simultaneously move the hook in both radial and angular directions. Travel time in horizontal plane can be calculated using Equation 6:

$$\mathrm{T_h^{(i,j)}} = \max\left\{\mathrm{T_r^{(i,j)}}, \mathrm{T_a^{(i,j)}}\right\} + \alpha \cdot \min\left\{\mathrm{T_r^{(i,j)}}, \mathrm{T_a^{(i,j)}}\right\} \tag{6}$$

Parameter β is used to take into account the operator's skill in simultaneous movement of the hook in horizontal and vertical planes. Travel time can be increased due to working site conditions such as weather conditions, existence of obstacles and different safety issues. Parameter γ is used to account for working site conditions (Huang et al., 2011). Total travel time, which is the combination of horizontal and vertical movement times, can be calculated using Equation 7:

$$\mathrm{T_{(i,j)}} = \gamma \cdot (\max\left\{\mathrm{T_h^{(i,j)}}, \mathrm{T_v^{(i,j)}}\right\} + \beta \cdot \min\left\{\mathrm{T_h^{(i,j)}}, \mathrm{T_v^{(i,j)}}\right\}) \tag{7}$$

where $0 \le \alpha \le 1$, $0 \le \beta \le 1$, $1 \le \gamma \le \infty$.

α, β, and γ are continuous positive numbers. Lower values of α and β reflect a higher degree of simultaneity in movements in two directions. Lower values for parameter γ reflects more convenient operations conditions, e.g. value of one is used for normal weather conditions in an open area without on-site obstructions. Values of α and β and γ need to be estimated based on observations of the construction job site. Using Equations 1-7 the location travel time matrix (*C: (c_{ij})*) can be developed.

2.5 CSSP Formulation

CSSP can be represented by a directed graph $G = (V, A)$, where V is the set of n vertices (representing requests) and A is the directed arc set. The mathematical formulation of the optimization model to solve the CSSP is as follows:

$$Minimize \quad \sum_k \sum_l p_{kl} y_{kl} \tag{8}$$

$$Subject\ to \quad \sum_{l:l \neq k} y_{kl} = 1 \qquad \forall\, k, l \in V\ , (k, l) \in A \tag{9}$$

$$\sum_{k:k \neq l} y_{kl} = 1 \qquad \forall\, k, l \in V\ , (k, l) \in A \tag{10}$$

$$\sum_{\mathrm{k} \in \mathrm{S}} \sum_{\mathrm{l} \in \bar{\mathrm{S}}} y_{kl} \geq 1 \qquad \mathrm{S} \subset \mathrm{V},\ 2 \leq |\mathrm{s}| \leq \mathrm{n} - 2\ ,\ (k, l) \in A \tag{11}$$

$$y_{kl} \in (0, 1) \qquad \forall\, k, l \in V\ , k \neq l \tag{12}$$

where $P: p_{kl}$ is the service request matrix associated with A.

The suggested optimization model determines the minimum cost (travel time) circuit fulfilling each request once and only once. Such a circuit is known as a *tour* or *Hamiltonian circuit* (or *cycle*) (Laporte, 1992). In this problem, a binary variable y_{kl} is associated with every arc (k, l), and is set equal to 1 if and only if arc (k, l) is used in the optimal solution $(k \neq l)$. In other words, $y_{kl} = 1$ if the crane hook goes directly from request node k to request node l, and $y_{kl} = 0$ otherwise (constraint 12). Constraints 9 and 10 are degree constraints which specify that every vertex is incident of one outgoing arc (constraint 9) and one ingoing arc (constraint 10). The solution considering only constraints 9 and 10 might lead to a disconnected solution (subtour) that needs to be excluded from the set of solutions. To eliminate solutions which consist of subtours (i.e., tours on subsets of less than n vertices), an additional constraint (constraint 11) is needed. S is a subset of V vertices and $|\mathrm{s}|$ is the cardinality of S. In addition, $\bar{\mathrm{S}}$ is a complement of S. Constraint 11 is only valid when $2 \leq |\mathrm{s}| \leq n - 2$ and prevents the solution to contain two or more disjoints subtours.

The CSSP travel time matrix (service requests matrix ($P: p_{kl}$)) is a dynamic matrix composed of the location travel time matrix (*C: (c_{ij})*) combined with the requests at a given time. The former matrix reflects the travel time between request nodes. This matrix is inherently asymmetric ($\mathrm{P}^{\mathrm{T}} \neq \mathrm{P}$).

As mentioned before, CSSP must be converted to an asymmetric TSP here. However, asymmetric TSPs are complex to solve. One way of solving an asymmetric TSP is to double the size of the distance matrix by replacing every node in the graph with two nodes (Jonker & Volgenant, 1983), having added nodes represent dummy cities. The links between each node and its corresponding duplicated dummy node is associated with very low travel costs ($-\infty$). This assures that a real node and its corresponding dummy node are passed through after each other in the final sequence. The original distances given in the service requests matrix ($P: p_{kl}$) are used for distances between the nodes and the duplicated dummy nodes, where paths start from real nodes and end in the duplicated dummy nodes. The distances between real nodes and between all duplicated nodes are assumed to have very large costs (+∞) since there is no path between them. This procedure transforms the asymmetric matrix to a symmetric one. As an example, the following asymmetric matrix for a TSP with four nodes (left) can be converted to a symmetric matrix (right) through the explained procedure.

$$\begin{pmatrix} 0 & d_{12} & d_{13} & d_{14} \\ d_{21} & 0 & d_{23} & d_{24} \\ d_{31} & d_{32} & 0 & d_{34} \\ d_{41} & d_{42} & d_{43} & 0 \end{pmatrix} \Longleftrightarrow \begin{pmatrix} +\infty & \begin{matrix} -\infty & d_{21} & d_{31} & d_{41} \\ d_{12} & -\infty & d_{32} & d_{42} \\ d_{13} & d_{23} & -\infty & d_{43} \\ d_{14} & d_{24} & d_{34} & -\infty \end{matrix} \\ \begin{matrix} -\infty & d_{12} & d_{13} & d_{14} \\ d_{21} & -\infty & d_{23} & d_{24} \\ d_{31} & d_{32} & -\infty & d_{34} \\ d_{41} & d_{42} & d_{43} & -\infty \end{matrix} & +\infty \end{pmatrix}$$

Once an asymmetric TSP is converted to a symmetric one, the minimum transportation cost and its associated travel route can be found using the proposed optimization model.

2.6 Numerical Evaluation

To underline the utility of the suggested optimization model, its performance is compared to three conventional heuristic scheduling methods, namely the FIFO, Shortest Job First (SJF), and Nearest Neighbor First (NNF) algorithms. The FIFO algorithm involves no intelligence and the crane operator processes the requests based on the received order. Based on the SJF algorithm, a request with the smallest estimated travel time has the highest priority to be fulfilled. Based on the NNF algorithm, crane operator should serve the next closest request after each delivery.

2.6.1 Computational Experiment: A Predefined Facility Layout Case

A numerical example with six material (supply) locations and eight demand (crew) locations is considered here. In this problem (Figure 12) one central tower crane is in charge of transporting materials. For simplicity, the initial tower crane hook location is assumed to be at $(0,0,0)$ and other locations are determined with respect to that. Table 4 lists the material and crew coordinates in this problem. Specifications of a Heavy-Load 4000 HC 100 Liebherr tower crane are used for crane velocities: $\mathrm{V_v} = 25\ m/min$, $\mathrm{V_a} = 0.6\ revolution/min$ and $\mathrm{V_r} = 60\ m/min$. α and β are assumed to be 0.25 and 1, respectively, based on previous studies (Huang et al., 2011; C. M. Tam et al., 2001; Zhang et al., 1999). γ is set equal to 1, assuming normal site conditions (Huang et al., 2011). Using the transportation model described in section 3.4.1 the hook travel time between the nodes can be calculated to develop the location travel time matrix.

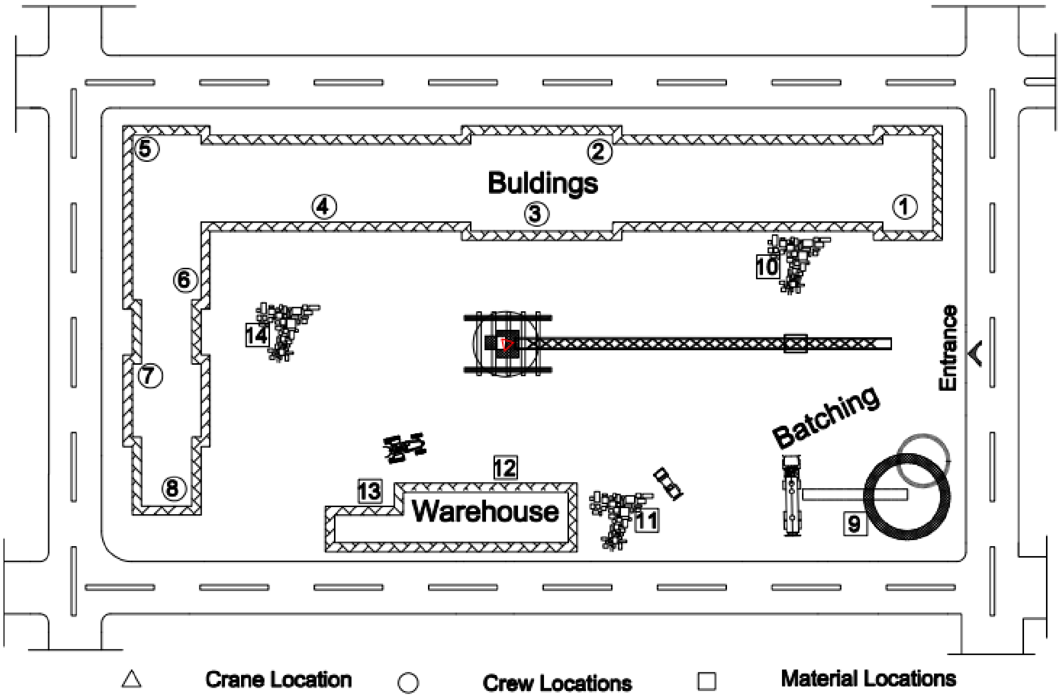

Figure 12: Site layout

Table 4: Coordinates of supply and demand locations

Material location (Supply)	Location (x,y,z)	Crew location (Demand)	Location (x,y,z)
1	(76,-39,0)	1	(86,29,10)
2	(57,16,0)	2	(20,41,5)
3	(30,-39,0)	3	(6,29,3)
4	(0,-27,0)	4	(-40,30,12)
5	(-30,-32,0)	5	(-79,42,4)
6	(-55,1,0)	6	(-70,13,5)

Material location (Supply)	Location (x,y,z)	Crew location (Demand)	Location (x,y,z)
		7	(-77,-7,0)
		8	(-72,-32,0)

To evaluate the performance and utility of the proposed optimization model for solving CSSP with different sizes, the numerical example is solved for 11 different sizes or numbers of requests (i.e., 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, and 1000 requests). Using a uniform distribution, random requests are generated for each CSSP with a given size. To ensure the generated problems are random, 100 CSSPs are generated and solved for each problem size, making the total number of solved problems 1100 (100×11). Each the 1100 CSSPs are then solved using the suggested optimization model as well the three heuristic crane operation algorithms. Mean and standard deviation of total travel times are determined for problem sizes using all scheduling methods, i.e. FIFO, SJF, NNF, and CSSP optimization. The optimization model is solved using CONCORDE, a symmetric TSP exact solver (Applegate, Bixby, Chvatal, & Cook, 2011) in integration with MATLAB.

Average savings in total travel time for different number of requests under different scheduling methods with respect to the travel time under FIFO method are presented in Table 5. To facilitate the comparison, total travel time based on the FIFO scheduling method is set as the baseline and other scheduling methods are compared to this baseline. Results show that the intelligence added to the sequence processing reduces average travel time by 6%, 25%, and 27% using the SJF, NNF, and the optimal scheduling methods in comparison to FIFO, respectively. As can be seen, the proposed optimal scheduling method outperforms other methods. The NNF heuristic rule also provides a satisfactory sequence scheduling without a need for optimization. Standard deviations for all problem sizes were less than 7% of the mean travel time. In addition, standard deviation decreased as the number of requests increased.

Table 5: Average time saving under different scheduling methods (SJF, NNF, and optimization) for the pre-defined site layout example

Number of requests	Time saving relative to the FIFO method (%)			Run time (Sec.)
	SJF	NNF	Optimization	
10	2%	14%	18%	0.36 ± 0.03
20	1%	18%	22%	0.41 ± 0.05
30	3%	20%	24%	0.71 ± 0.83
40	3%	22%	25%	0.77 ± 0.19
50	3%	23%	26%	0.97 ± 0.66
100	6%	26%	28%	3.2 ± 2.8
200	9%	28%	29%	13.4 ± 19.5
300	11%	29%	30%	26.6 ± 41.48
400	12%	29%	30%	59.03 ± 90.6
500	12%	30%	31%	100.3 ± 138.6
1000	6%	31%	32%	732.6 ± 1316.4

Table 5 also reports the mean computational time and its standard deviation for each size of problem. Small computational times show the applicability of the suggested crane service sequence optimization method in practice. Computation times increased exponentially as the number of requests increased. High variation in computational times is due to the problem structure variability, mainly due to the inclusion of the subtour elimination constraint in the model.

To evaluate the significance of the optimized results compared to the FIFO approach, the t-test was performed and the significance level (p-value) was calculated. The significance-level for all problem sizes was less than 10^{-15}. This suggests that the results are significantly different, rejecting the null hypothesis ($\mu_{FIFO} = \mu_{Optimal}$) for all problem sizes, which shows the consistency of the method in finding the optimal travel time.

2.6.2 Computational Experiment: A Random Site Layout

To show the independency of the optimization model performance to the site layout, the predefined (deterministic) site layout of section 5.1 is replaced with a random site layout. To test the performance of the sequencing methods in case of random site lay out, for each problem random nodes, scattered around a central tower crane with a 70-meter operation radius, are generated using a uniform distribution (Figure 13). Each random site layout is assumed to have 50 nodes with random coordinates and elevation differences from 0 to 10 meters with respect to the hook's initial idle position. Once random nodes are generated for a CSSP, random request pairs (i, j) are generated using a uniform distribution. Each request (i, j) pairs a supply node (i) with a demand node (j), where both i and j belong to the set of 50 generated random nodes. In this case, each node in the random site can serve both as a supply or demand location, depending on the randomly generated request pair. Similar to the previous case, the problem is solved 100 times for 11 different sizes. Each of the 1100 problems has a unique randomly generated layout with 50 nodes. Values of the other variables (crane velocities, operator skill parameter, etc.) are the same as the problem illustrated in the previous section.

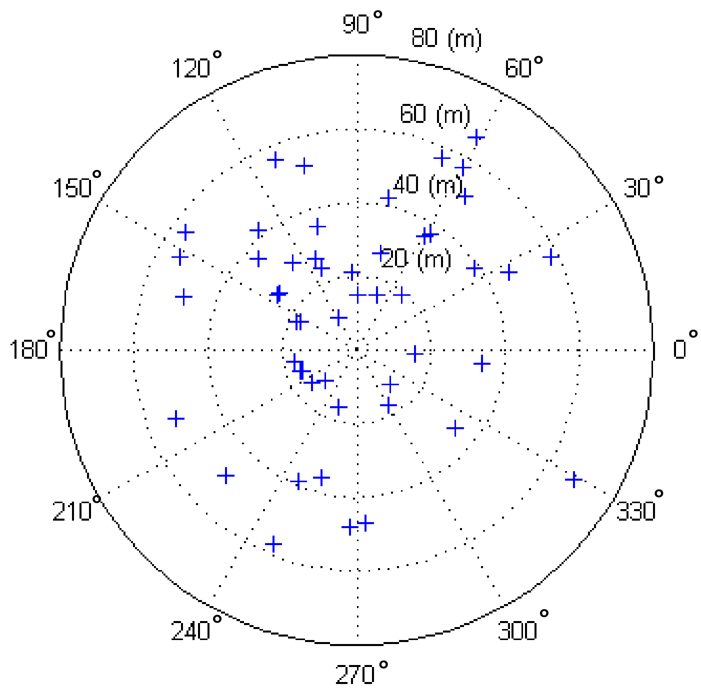

Figure 13: Random node coordinates around a central tower crane

Table 6 presents the results of the analysis for the random site layout problem. Similar to the previous example with predefined site layout, standard deviations are within an acceptable range (10% in this case). Performance of the SJF method is not significantly better than the FIFO method, making it an inferior scheduling method. On average, optimization results in time saving of 27%. This value is 24% for the NNF method as the best heuristic scheduling method. Time savings increase for all methods as the problem gets larger. Similar to the previous case, the computational time is very small for small sizes of problem, making the method applicable in practice. In general, the obtained results for the random site layout are consistent with the results in the previous case, indicating the robustness of the performance of the suggested optimization model.

Table 6: Average time saving under different scheduling methods (SJF, NNF, and optimization) for the random site layout example

Number of requests	Solution Algorithm			Run time (Sec.)
	SJF saving (%)	NN saving (%)	Optimal saving (%)	
10	1%	13%	18%	0.61 ± 0.05
20	-1%	16%	21%	0.74 ± 0.09
30	1%	19%	23%	0.87 ± 0.19
40	0%	21%	24%	1.13 ± 0.7
50	1%	22%	26%	1.58 ± 1.23
100	1%	25%	28%	7.85 ± 11.6
200	2%	28%	30%	41.53 ± 51.84
300	2%	29%	30%	125 ± 228.6
400	3%	30%	31%	280 ± 536.3
500	4%	30%	31%	325.3 ± 670.1
1000	5%	31%	32%	3156.1 ± 2134

2.6.3 Sensitivity to Input Parameters

To further examine if the model output is sensitive to the input parameters (V_v,V_a,V_r,α,β,γ), a one-way sensitivity analysis is conducted. In one-way sensitivity analysis, only one parameter changes at a time while other parameters remain constant and the impact of the change on the model output is examined. For simplicity, the size of the problem is fixed to 100 requests. The general modeling procedure is the same as the previous case with random site layouts. Each time, one input parameter of the model is changed by a given amount within a meaningful range, and the mean value of time saving percentage for 100 trials is recorded. Figure 14 shows the sensitivity analysis results. Each graph shows how the average time saving percentage with respect to the FIFO method varies by changing the value of one of the input parameters. While the model output is sensitive to the input parameters, the linear relationship between the input parameters and the time saving suggests that the model performance is not affected by input values and the output changes are consistent with the input changes (e.g. higher velocity results in higher time saving).

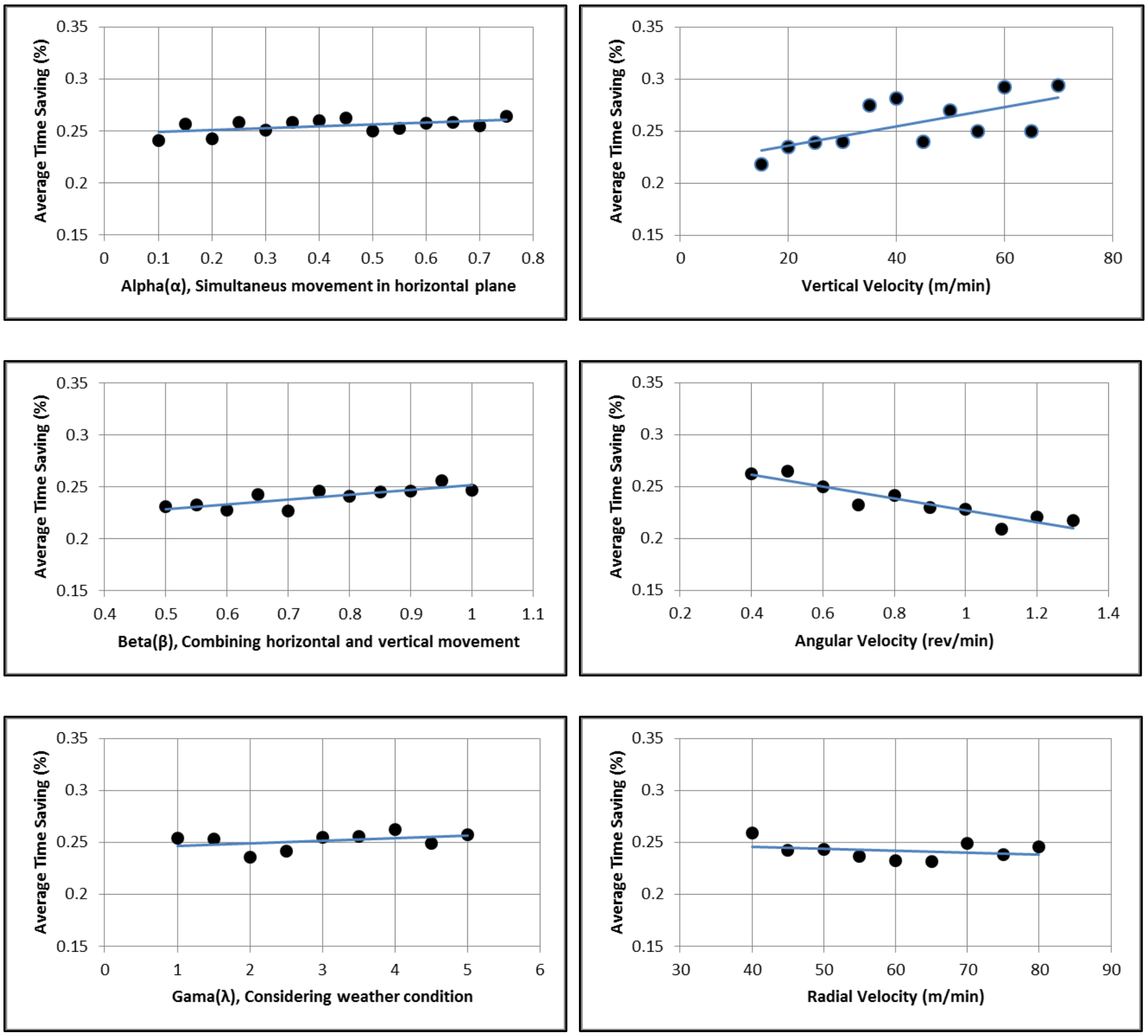

Figure 14: Sensitivity analysis results

2.7 Conclusions

Using an optimization technique to improve efficiency of the crane operations via prioritizing job requests was proposed in this paper. An exact combinatorial optimization method, which is a modification of the “Traveling Salesman Problem (TSP)”, was proposed for minimizing construction crane travel time by optimal ordering of crane movement sequences.

The suggested optimization model results in 20-30% saving in travel time in comparison with the conventional First-In-First-Out approach in fulfilling the requests. The model's performance is not highly sensitive to input parameters and different jobsite specifications. The small run time of the optimization model makes it useful in practice, helping reduce crane operations and crane-related activity costs considerably. The developed model optimizes the crane travel time only, which is the significant portion of crane cycle operations, especially in high rise construction and when loading and unloading nodes are not close.

Similar to any other modeling study, this study had some limitations and simplifying assumptions that can be addressed in future studies. Here, the travel time between two nodes was considered to be deterministic while travel time can vary in practice. Future studies can consider stochastic travel times. Given that the time savings increase with increased travel time resulting from elevation differences, future studies can investigate the effects of larger elevation differences (more than 10 meters) on travel time. This study assumed that each loaded bucket can be sent to one target location only, i.e., the crane hook cannot visit multiple demand nodes after being loaded. Future studies might relax this assumption. To make the developed proof-of-concept model more practical, task deadline, sequence priority, and intermittent requests can be added to the problem formulation. While in this study travel time was assumed to be independent of the load, future studies can evaluate the effects of material weight on travel time. Finally, given that crane operations efficiency is strongly tied to project duration and cost, future studies might consider evaluating this connection.

CHAPTER 3: CONSTRUCTION TOWER CRANE SERVICE SEQUENCING PROBLEM WITH DEADLINE

Tower cranes are the centerpiece equipment in high-rise building construction and can be the bottleneck of the construction operations. Inefficient planning and scheduling of crane operations not only have a significant negative impact on crane operation duration, but also on the pending activities that rely on crane services. Therefore, effective crane service scheduling can have considerable effects on construction project time and operations efficiency. Previous research has shown that optimal sequencing of crane movements, using the Traveling Salesman Problem (TSP)-based crane operations optimization model, can result in 18-32% saving in the overall crane's travel time. However, this model overlooks the value of crane's travel sequence optimization in problems with service deadlines. Crane service deadlines have significant importance in construction operations efficiency, especially when constructions activities that rely on crane service are on the project's critical path and any delay in accomplishing these activities could increase the total project duration. This paper explores the value of crane sequence optimization in scheduling problems with deadline. For this purpose, the TSP-based crane operations optimization model is revised to include the service deadline constraint. The developed NP-complete optimization problem is then solved using a heuristic Genetic Algorithm (GA)-based optimization method. The developed solution method is able to solve the optimization problem with minimal violations of the deadlines and the results show an average saving of 21-38% in crane's total travel time when deadlines are considered.

3.1 Introduction

Efficient management of construction equipment is essential to successful and timely accomplishment of construction projects. Among a variety of equipment used in typical construction sites, cranes are the largest and the most conspicuous. The nature of activities that a construction crane is involved in (e.g., lifting, transporting material) gives it a potential to create schedule bottlenecks, control the critical path of the project, and create delays in project completion (Rosenfeld, 1995). Therefore inefficient planning and operation of cranes can have major implications for success of construction projects (Wakisaka, Furuya, Inoue, & Shiokawa, 2000).

Since crane operation efficiency has impacts not only on crane's operation time but also on the timing of the activities that rely on the crane, reducing crane's operation time has been interesting to construction management researchers. Two main approaches have been promoted for improving crane operation efficiency: technology advancement and site layout optimization.

Everett & Slocum (1993) reported 16-21% reduction in crane cycle duration by installing a video system on a mobile crane to enhance the operator vision at the lifting point (Everett & Slocum, 1993a). Rosenfeld (1995) proposed a semi-automatic control system with the ability to navigate on pre-planned paths autonomously and memorize multiple benchmarks resulting in 15-50% saving in crane cycle (Rosenfeld, 1995). Rosenfeld and Shapira (1998) could shorten crane cycle by 6% through minimizing the maneuvering in loading and unloading zones with application of a semi-automatic navigation system (Rosenfeld & Shapira, 1998). The saving was reported to be tangible as crane operation efficiency is strongly tied to overall jobsite operation efficiency. Along the same line, Shapira et al. (2008) used a high resolution wireless video camera system that can be navigated from several zoom modes accessible by the operator. They reported 14-29% saving in travel time and 11-26% saving in total crane cycle (A. Shapira et al., 2008).

Optimal facility layout design and planning has also significant effect on crane travel time. To transport material efficiently, the crane must be well located with respect to locations of loading and unloading nodes. Zhang et al. (1996) used a Monte-Carlo simulation model to optimize the location of a single tower crane (Zhang et al., 1996). In another study, Zhang et al. (1999) could achieve 10-40% saving in the total hook travel time by optimizing the locations of a group of tower cranes (Zhang et al., 1999). Tam et al. (2001) used a genetic algorithm method to optimize the tower crane and material supply locations, resulting in 18% saving in crane travel time (C. M. Tam et al., 2001). Huang et al. (2011) used mixed integer linear programming to optimize tower crane location and material supply locations in high-rise buildings. Their results show 7% improvement in total crane transportation time compared to alternative location optimization solutions (Huang et al., 2011). More recently Zavichi et al. (2013) presented a new approach to reduce the crane travel time by optimizing the sequence of locations that the crane has to visit in order to fulfill a set of requests on a job site. They reported an average saving of 27% in the crane's total travel time by optimal sequencing of the requests in comparison to heuristic scheduling methods normally used in practice (A Zavichi, Madani, Xanthopoulos, & Oloufa, 2013). Their numerical modeling experiments did not include crane job deadlines. Nevertheless, given the difference in importance of different crane activities during operations, operators need to optimally sequence crane movements with respect to the associated deadlines. To underline the value of crane movement optimization in practical problems this paper extends the scope of previous work by including crane job deadlines into crane job sequence optimization model formulation.

3.2 Problem Statement

Figure 15 shows a graph-based representation of a construction job site. This figure consists of a set of vertices and pairwise edges with a central tower crane (T), surrounding material (M), and crew (C) locations. Connecting edges represent the simplified crane travel paths between pairs of vertices (M, C, and T). Each connecting edge is associated with a travel time indicating the time the crane requires to travel between the two ends of the edge.

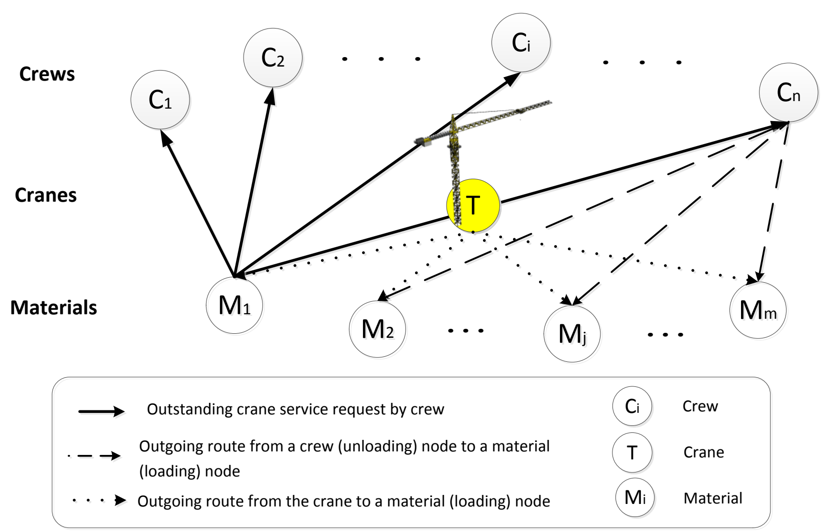

Figure 15: Graph-based representation of a construction site layout

Assume that several crews request material delivery while there is only one in-service crane to fulfill the requests. Allocating the crane service, as a scarce resource, to various activities with respect to minimizing the total crane travel path constitute the Crane Service Sequence Problem (CSSP). This paper deals with a special case of CSSP in which there is only one supply location for each requested material, which is the common case in construction.

Zavichi et al. (2013) formulated Traveling Salesman Problem (TSP)-based optimization model for solving the CSSP with no crane activity deadlines (A Zavichi et al., 2013). TSP is the well-known problem of finding the shortest closed path (tour) by visiting n given locations exactly once and returning to the initial position. In mathematical terms, TSP is the problem of finding the optimal nodes sequence among permutation ($P = (i_1 i_2 i_3 . . . i_n)$) of the integers from 1 through n that minimize the value of $C_{i_1 i_2} + C_{i_2 i_3} + C_{i_3 i_4} + . . . + C_{i_n i_1}$, where i_1 through i_n represent the nodes and $C_{\alpha\beta}$ represents the costs between each pair of nodes (Flood, 1956). TSP is an NP-complete problem, requiring a considerable computational capacity due to a very large set of alternative solutions.

CSSP can be converted through a polynomial transformation into an asymmetric TSP (ATSP) by the approach suggested in the previous research (A Zavichi et al., 2013). Based on this approach, each request (starting from a material node and ending in a crew node) is considered to be a node, assuming that travel time is the cost of switching between the requests (Figure 16). This reduces the CSSP to the problem of finding the optimum sequence of requests to be fulfilled to minimize the makespan. In the reduced problem, each request comprises a material and crew node, and thus, cost is not only associated with traveling between the nodes, but also with node visits. Therefore, the travel time incurred in the nodes is added to the outbound arcs from the nodes. The resulting pairwise travel times are asymmetric, meaning that the cost (travel time) of moving from R_i to R_j is different from the cost of moving from R_j to R_i. The merit of this transformation is that the methods developed for the ATSP can be directly applied to the CSSP without any major modifications (A Zavichi et al., 2013).

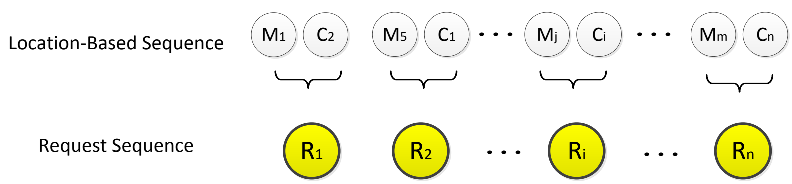

Figure 16: CSSP to ATSP conversion

3.3 Formulating CSSP with Deadline

The CSSP with deadline can be presented using graph theory: consider a directed graph $G = (V, A)$, where V is a set of n request nodes, and A is the set of directed arcs connecting the request nodes. The arc's travel time (t_{ij}) is the cost of switching between the request nodes. In addition, for each node in the graph $(i \in V)$, a processing time (p_i) and a deadline (d_i) are given. The processing time (p_i) represents the elapsed time between the arrival and the departure at node i and it corresponds to the travel time to fulfill a request starting from a material node and ending in a crew node. Each request is comprised of material and crew nodes. The deadline for a request node is the latest time the material must be received by the crew. The duration corresponding to arcs (t_{ij}), processing time (p_i), and due dates (d_i) are considered to be nonnegative here. Figure 17 shows a graphical illustration of the CSSP with deadlines.

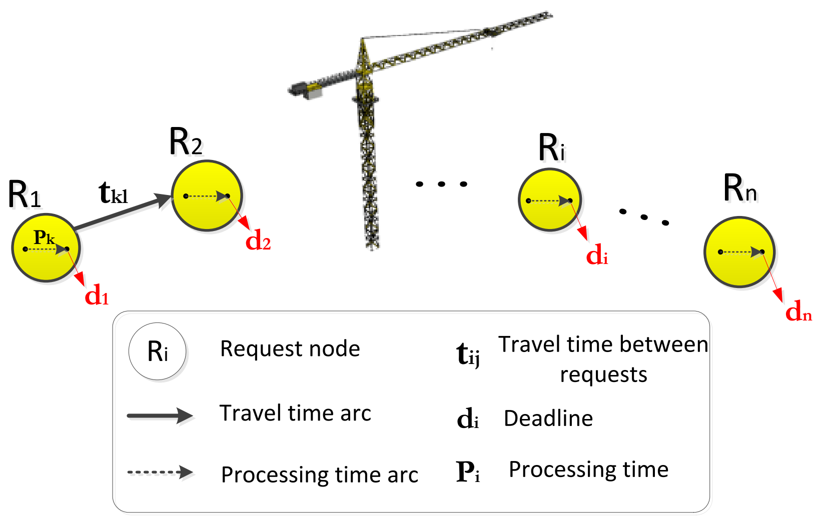

Figure 17: Graphical illustration of CSSP with deadlines

The crane service sequence problem with deadline can be formulated as an integer programming problem as follows:

$$\text{(ATSP)} \quad \mathit{Minimize} \quad \sum_i \sum_j t_{ij} y_{ij} \tag{13}$$

$$\mathit{Subject\ to} \quad \sum_{j:i \neq i} y_{ij} = 1 \qquad \forall\, i,j \in V\,,\,(i,j) \in A \tag{14}$$

$$\sum_{i:i \neq j} y_{ij} = 1 \qquad \forall\, i,j \in V\,,\,(i,j) \in A \tag{15}$$

$$S_i + p_i + t_{ij} - (1 - x_{ij})M_{ij} < S_j \tag{16}$$

$$S_i + p_i < d_i \tag{17}$$

$$y_{ij} \in (0,1) \qquad \forall\, i,j \in V \;\;, i \neq j$$

where M is a big constant value. The proposed formulation uses a binary variable $\mathrm{y_{ij}}$ for each arc $(\mathrm{i,j})$, which equals to one if the arc is traversed in the optimal sequence and to zero otherwise. Constraints (14) and (15) are degree constraints which specify that every vertex is an incident of one ingoing and one outgoing arc. Constraint (16) ensures that the arrival time in a given request node cannot be smaller than the arrival time in the request node visited immediately before plus the cost of traveling between the two requests and the associated processing time. This constraint ensures that the times are increasing along any path, and therefore, no subtour cycles can be generated in the solution (Dash, Günlük, Lodi, & Tramontani, 2012). Finally, constraint (17) checks if the solution respects the request's deadline or not.

In this formulation, duration corresponds to arcs (t_{ij}) and processing time (p_i) is extracted from the location travel time matrix (C: (c_{ij})). The location travel time matrix is a symmetric square matrix in the following form where n is the number of nodes in the graph (A Zavichi et al., 2013):

$$\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & C_{ij} & \vdots \\ c_{n1} & \cdots & c_{nn} \end{bmatrix}$$

where c_{ij} is the time that it takes for the crane's hook to travel between the node i to node j in Figure 15 (A Zavichi et al., 2013).

The trolley's travel time between nodes i and j is composed of three components: radial ($\mathrm{T}_{\mathrm{r}}^{(\mathrm{i,j})}$), angular ($\mathrm{T}_{\mathrm{a}}^{(\mathrm{i,j})}$), and vertical ($\mathrm{T}_{\mathrm{v}}^{(\mathrm{i,j})}$) travel times. The possibility of simultaneous movements in different directions in space is considered by introducing two random variables, α and β, degree of simultaneity indicators for movements in horizontal and vertical planes, respectively. The travel time is considered to be a linear combination of horizontal and vertical travel times as $T_{(i,j)} = (max\left\{T_h^{(i,j)}, T_v^{(i,j)}\right\} + \beta \cdot min\left\{T_h^{(i,j)}, T_v^{(i,j)}\right\})$, where $T_{(i,j)}$ is the total travel time between node i and j, and $\mathrm{T}_{\mathrm{v}}^{(\mathrm{i,j})}$ and $\mathrm{T}_{\mathrm{h}}^{(\mathrm{i,j})}$ are the vertical and horizontal components of travel time, respectively. Similarly, the horizontal travel time is considered to be a linear combination of the radial and angular movements as $T_h^{(i,j)} = max\left\{T_r^{(i,j)}, T_a^{(i,j)}\right\} + \propto \cdot min\left\{T_r^{(i,j)}, T_a^{(i,j)}\right\}$, where$\mathrm{T}_{\mathrm{r}}^{(\mathrm{i,j})}$ is the travel time for the radial movement and $\mathrm{T}_{\mathrm{a}}^{(\mathrm{i,j})}$ is the tangent travel time. Radial, angular and vertical travel times are computed assuming constant velocities in each direction using $T_r^{(i,j)} = \frac{\left|r_{s_i} - r_{D_j}\right|}{V_r}$, $T_a^{(i,j)} = \frac{\left|\theta_{s_i} - \theta_{D_j}\right|}{V_a}$, and $\mathrm{T}_{\mathrm{v}}^{(\mathrm{i,j})} = \frac{\left(\left|\mathrm{Z}_{\mathrm{S_i}} - \mathrm{Z}_{\mathrm{D_j}}\right| + 2 \times \mathrm{h}\right)}{\mathrm{V_v}}$ (A Zavichi et al., 2013). In the vertical travel time equation, h is the "minimum hoisting height, to account for the additional height traversed to move the material depending on loaded material type, site topography, obstruction and safety factors, and other conditions at both ends of the travel path. Radial $\mathrm{V_r}(\mathrm{m/min})$, angular $\mathrm{V_a}(\mathrm{rpm})$, and vertical $\mathrm{V_v}(\mathrm{m/min})$ velocities can be obtained from the in-service crane manufacturing specifications. S_i and D_j correspond to supply and demand locations as the starting and ending nodes of the trolley for a given request. r, θ, and Z are the location coordinates in a cylindrical coordination system (Figure 18).

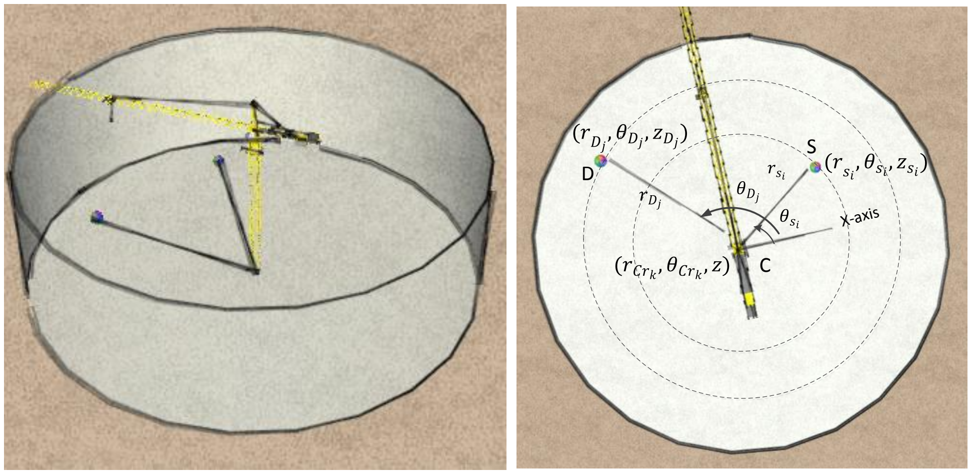

Figure 18: Cylinder coordination system elements used for travel time estimation

3.4 Solution Algorithm

So far, it was illustrated how the CSSP with Deadlines can be formulated as an ATSP with deadline. Given the mathematical complexities of both TSP and ATSP, one cannot prescribe an efficient global optimal solution method as the suitability of the method can vary depending on different characteristics of the problem (size, possibility of subtours, etc). Similar studies in literature are different exact and heuristic algorithms to solve TSP and ATSP with time window constraints. Time window is a time range in the form of $[a_i, b_i]$ that request delivery in CSSP or city visiting in TSP and ATSP is expected. Deadline is when a_i is assumed to be zero.

Christofides et al. (1981) and Baker (1983) used the branch and bound technique for time-constrained TSP for small size problems with up to 50 nodes. Their algorithms performed well on least overlap time windows and when the time window constraints were “moderately tight”.

Langevin et al. (1993) presented a two-commodity flow formulation for TSP suited for time window constraints. The resource in this formulation was time and the numerical experiment reported success with solving problems with up to 60 nodes. Dumas et al. (1995) used a dynamic programming approach in conjunction with reducing the search space based on the time window structure. Successful solution for problems with up to 200 nodes was reported. For asymmetric TSP with time windows, Ascheuer et al. (2001) used several heuristic methods with branch-and-cut and showed that problems with up to 50-70 nodes can be solved optimally. They were able to solve a problem with up to 250 nodes. Recently, Dash et al. (2012) and Baldacci et al. (2012) used branch-and-cut algorithm and dynamic programming and solved TSP problems with up to 233 vertices. In general, capability of different solution algorithms in finding the exact optimal solution highly depends on the structure of the time windows. Normally, the TSP with deadlines is difficult to solve, when more than 50% of the nodes have time constraints (Ascheuer et al., 2001) or when the time windows are wide (Dumas et al., 1995). As NP-complete and complex problems, TSPs have been also solved using metaheuristic methods. For example, Carlton and Barnes (1996) used a tabu search heuristic approach with penalties to solve a TSP with time window. Gendreau et al. (1998) introduced a generalized insertion heuristic method to build an optimal travel route and improve the route by repetitive removal and reinsertion of all vertices. Recently, Ohlmann and Thomas (2007) used a variant of simulated annealing with a variable penalty function to solve the TSP with time windows.

In many real world situations, the deadline constraints can be violated to some extent. For example, in CSSP, requests are made by different parties that might not be aware of the other requests in the system, creating deadline conflicts. In such a situation, there exists no feasible solution to fulfill all requests by their deadlines. Therefore, here request deadlines are considered as soft deadlines, which might be violated. The overall objective is to minimize the total travel time and meet the deadlines to the extent possible. A Genetic Algorithm (GA)-based solution algorithm is developed here to provide an efficient solution to the CSSP with deadlines.

The underlying notion to use GA as an optimization method is to change a population of candidate solutions into a fitter population using genetics-inspired operators (Holland, 1975). Candidate solutions for TSP-related problems are permutations of non-repeating sequence. Conventional GA operators are incapable of producing valid and legal offspring (Üçoluk, 1997). Therefore, an asexual reproduction scheme is used to create valid candidate solutions. Under this scheme, the GA searches for a good ordering of requests, while construction of the feasible solution is handled by the greedy heuristic. This type of GA operators is relatively simple to implement and they do not generate any infeasible solution. So, there is no need for further modifications for the offspring generated in the evolutionary process. In addition, this algorithm works for asymmetric as well as symmetric TSP. The suggested algorithm is outlined in Figure 19:

CSSPD GA (N_{pop}, N_{iter})
//Initial Population Generation
1. For i=1: to N_{POP}
2. **Generate** Initial Population using **nearest neighborhood** and **shortest due date first** algorithm
3. End
//Sequence Search
4. Repeat
5. **Evaluate** the fitness value for each chromosome (total distance) considering the penalty function
6. **Group** randomly each 4 chromosomes: N_g
7. **Determine** the fittest individual in each group **//selection**
8. **For** i=1 to N_g
9. **Reproduce** 3 new chromosomes by applying **//asexual reproduction**
10. Inversion Operator
11. Swap Operators
12. Insertion Operators
13. **End**
14. **Add** the newly generated chromosomes to the population
15. Until the number of iteration (**N_{iter}**) is reached
16. Return the fittest chromosome in the population

Figure 19: A penalty-based GA for solving CSSP with deadlines

The solution method is initiated by generating N_{pop} (user-defined parameter) initial solutions using two local search heuristics: nearest neighborhood first (NNF), and the earliest deadline first (EDF). NNF search prioritizes fulfilling the closest requests first while EDF tries to fulfill the requests with the earliest upcoming deadlines. Both algorithms (illustrated in Figure 20), on average, produce a healthier population with a better convergence rate, in comparison to a random population (lines 1 to 3).

Nearest Neighbor First ([t_{ij}])
//t_{ij} : Service Request Matrix
1. Start from crane's idle position
2. Repeat
3. **Find** the nearest unfullfilled request
4. **Fulfill** the request
5. Until no outstanding requests is left
6. Return to the first position

Earliest Deadline First ([t_{ij}],[d_j])
//t_{ij} : Service Request Matrix
//d_j: Deadline Matrix
1. Start from crane's idle position
2. Repeat
3. **Find** an unfulfilled request with the earliest deadline
4. **Fulfill** the request
5. Until no outstanding requests is left
6. Return to the first position

Figure 20: Pseudo codes for the NNF and EDF heuristic methods

The main loop of the algorithm (lines 4 to 15) iteratively tries to minimize the fitness value. In each generation, the initial population is randomly divided into N_g group ($N_g = N_{pop}/4$) (line 6). In each group, the selection operator chooses the fittest chromosome (line 7) and passes it to the asexual reproduction module (lines 8 to 13) to create a new fitter population. The procedure is continued until the termination condition, or specified number of iteration (N_{iter}) is met (line 15).

The asexual reproduction module randomly modifies individual chromosomes (Chatterjee, Carrera, & Lynch, 1996). Operators used in the algorithm include the inversion, swap, and insertion operators (Yu, Chen, & Li, 2011). The inversion operator chooses two random genes of the chromosome and reverses the order of the genes between them (Melanie, 1999). The swap operator, uses the 2-opt heuristic method and substitute the value of two randomly chosen genes (Croes, 1958). Finally, the insertion operator inserts the value of a randomly selected gene into another location in the string. Figure 21 shows the schema of the mutation operators.

The fittest chromosome is the one with the least fitness value. Fitness value is computed using Equation (18):

$$f_{(R_i)} = \sum_{k=1}^{n-1} t_{k,k+1} + \sum_{k=1}^{n} p_k + \alpha . \sum_{k=1}^{n} lv \tag{18}$$

where $t_{k,k+1}$is the switching time between requests in a given sequence ($< R_1, R_2, \dots, R_{n-1}, R_n >$), p_k is the request processing time, and lv is the penalty function associated with request to impose penalty when deadline is violated (penalty value is set to zero if the request is fulfilled before the deadline.) α is a penalty coefficient that can be adjusted by user. For request nodes in the sequence, penalty value (lv) is computed using Equation (19) in which s_k represents arrival time at the node, p_k is processing time, and d_k is the deadline for the corresponding node. Penalty value is set to zero if the request is fulfilled before the deadline; otherwise is the penalty of deviation from deadline.

$$lv_k = (s_k + p_k) - d_k , \ \begin{Bmatrix} lv_k = 0, if\ lv_k \leq 0 \\ lv_k = lv_k, if\ lv_k > 0 \end{Bmatrix}, \forall\ k \in V \tag{19}$$

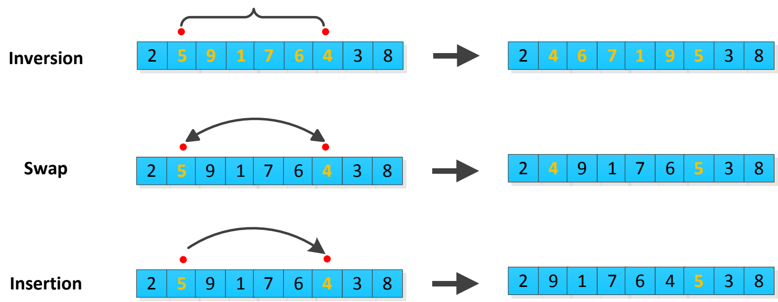

Figure 21: Asexual reproduction operators

3.5 Experimental Settings

Following Zavichi et al. (2013) to test the performance of the presented GA method for service sequencing with deadlines, a random site lay out is considered. In this layout, random nodes are generated around a central tower crane with a 70-meter operation radius using a uniform distribution scattered. Each random site layout is assumed to have 50 nodes with random coordinates and elevation differences from 0 to 10 meters with respect to the hook's initial idle position. Once random nodes are generated, random request pairs (i, j) are generated using a uniform distribution. Each request (i, j) pairs a supply node (i) with a demand node (j), where both i and j belong to the set of 50 generated random nodes. In this case, each node in the random site can serve both as a supply or demand location, depending on the randomly generated request pair. This is the case when the tower crane hauls some material to a location (e.g. concrete) and brings back some other material (e.g. formwork) on the return path and thus one location serves as both demand and supply nodes.

In order to develop the location travel matrix, specifications of a Heavy-Load 4000 HC 100 Liebherr tower crane are used for crane velocities: $\mathrm{V_v} = 25\, m/min$, $\mathrm{V_a} = 0.6\, revolution/min$ and $\mathrm{V_r} = 60\, m/min$. α and β are assumed to be 0.25 and 1, respectively, based on previous studies (Huang et al., 2011; C. M. Tam et al., 2001; Zhang et al., 1999). γ is set equal to 1, assuming normal site conditions (Huang et al., 2011). Finally, the GA parameters used for the experimental settings are as follows: initial population is set to 200 ($N_{pop} = 200$) and the termination criterion is set to 1000 iterations ($N_{iter} = 1000$).

3.5.1 Performance Evaluation

To evaluate the reliability of the proposed GA method in finding the optimal solution to the CSSP, its performance is first compared against the exact optimal solution method of Zavichi et el. (2013). This is done by elimination deadline violation penalty from the fitness function (setting the penalty coefficient (α) to zero). The numerical example is solved for 11 different sizes of CCSP, each with a different numbers of requests (i.e., 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, and 1000). For a given size problem, the exact algorithm and the GA is run for 100 times with random site layout and requests using a uniform distribution and the “best”, “average”, and “worst” outputs are reported in Table 7.

Table 7 shows the comparison results in terms of the difference of the GA solution with the exact optimal solution in percentages. “Best”, “Average”, and “Worst” percentages, respectively show the minimum,, average, and maximum difference of the GA solution with the exact optimal solution for the numerical CSSP with no deadline penalty. The maximum difference in the optimal solution is 2.7%, reflecting the reliability of the proposed GA method in finding the optimal solution to CSSP.

Table 7: Relative difference of the GA-based optimal solution with the exact optimal solution for different problem sizes

Number of requests	10	20	30	40	50	100	200	300	400	500	1000
Best (%)	0	0	0	0.057	0.436	0.754	0.712	0.83	0.781	0.665	0.759
Average (%)	0.09	0.708	1.055	1.313	1.356	1.367	1.402	1.397	1.33	1.228	1.076
Worst (%)	1.662	2.06	2.573	2.742	2.554	2.126	2.025	2.203	2.371	1.949	1.472

3.5.2 Algorithm Validation

To further evaluate the efficiency of the proposed GA, its performance was tested by applying to 19 different well-known ATSP benchmark problems, with from 17 to 443 nodes, for which optimal solutions are available at TSPLIB (http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/). Similar to the previous test, the penalty component of the fitness function was set to zero during the validation tests. Results indicated that the proposed GA method has an acceptable performance, providing solutions with at most 10% difference with the optimal solution. One important observation during validation tests was that the performance of the algorithm is more sensitive to the structure of the problem than the number of nodes. For example, for "Kro124p" benchmark problem with 100 nodes, the GA solution had the maximum relative difference (9.84%) with the optimal solution. However, for larger problems the relative difference was lower.

3.5.3 Setting the Penalty Coefficient

The critical value of the penalty coefficient was found through a sensitivity analysis. A one-way sensitivity analysis was conducted for an arbitrary number of requests (here, 100 requests) to evaluate the effect of the penalty weighting on fitness value. In each sensitivity analysis run, the penalty coefficient was changed, and the mean value of the fitness value for 100 trials was recorded while random variables (e.g., requests and deadline) and other site-specific variables (e.g., crane velocities, operator skill parameter) remained the same for all different coefficient values. Penalty coefficient was selected to be positive integers out of the 0-10 range Figure 22 shows how the crane's operation Fitness Value declines when the deadline penalty coefficient increases. For penalty coefficients of greater than 2 ($\alpha \geq 2$) the travel time remains unchanged, showing that the performance of the algorithm is insensitive to the magnitude of the penalty coefficient if α is large enough. Based on the findings in this step α was set to 2 for the rest of the analysis.

Figure 22: Penalty coefficient effect on crane operation time

3.5.4 Comparison Against Heuristic Scheduling Methods

To evaluate utility of the proposed GA-based crane operations scheduling model in solving CSSP with deadline (CSSPD), its performance was compared to other alternative scheduling methods using a numerical experiment. The alternative scheduling methods are normally based on heuristics and can be used in absence of decision making tools. Two of the heuristic scheduling methods applicable to CSSPD are “First-in-First-out (FIFO)” and “Earliest Deadline First (EDF)”. Based on the former method, requests are fulfilled with respect to the chronological order of received requests. Based on the latter method the request with the highest fulfillment priority is the one with the earliest deadline.

The numerical experiment involved solving problems with 11 different numbers of requests (i.e., 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, and 1000 requests). Using a uniform distribution, random requests are generated for each CSSPD with a given size. Each of the generated problems is then solved with GA-based optimization model as well as the two other heuristic methods.

Deadlines were assigned randomly to the demand node of every request, considering every request consists of supply and demand nodes. Since in practice, only a portion of requests may have deadlines associated, assigned deadlines are in $[0, 2n \times \bar{X}]$ interval, where n is the number of requests and $\bar{X}$ is the average travel time between requests. Given that $2n \times \bar{X}$ is larger than the average time to finish a set of requests, it implies that only a portion of requests have deadline.

To encapsulate the randomness of the problem, 100 CSSPs were generated for each 11 different numbers of requests (i.e., 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, and 1000 requests) and Mean and standard deviation of total operation times were determined for different sizes of problem using the three scheduling methods described earlier.

Table 8 shows the results of the numerical experiment under different scheduling methods with respect to FIFO operation fitness value. To facilitate the comparison, total operation fitness value based on FIFO scheduling method is set as the baseline and other scheduling methods are compared to this baseline. The results clearly show that the FIFO method is the worst method among the three methods tested here. This is because this method involves no intelligence and has no way of addressing deadlines, resulting in a longer operations fitness value. Therefore, this method was disregarded in further investigation.

Table 8: Total crane operations fitness value based on the FIFO, EDF, and GA-based optimization methods for different sizes of CSSPD"

Requests' numbers	Crane Operation Fitness Value	
	EDF	GA
10	11%	32%
20	34%	48%
30	47%	60%
40	48%	60%
50	47%	63%
100	39%	56%
200	32%	52%
300	29%	48%
400	34%	60%
500	41%	61%
1000	36%	58%

Table 9 compares the results of the GA-based scheduling and EDF methods. The proposed GA-based scheduling method shows an average saving of 29% in the total operation fitness value compared to the EDF method. To evaluate the significance of the results, the t-test was performed and the significance levels (p-values) were calculated, showing that the results were significantly different. So, the null hypothesis ($\mu_{\text{FIFO}} = \mu_{\text{Optimal}}$) was rejected for all of problem sizes, showing the consistency of the method in finding the optimal travel time.

Table 9: Comparison of the GA-based scheduling method with resoect to the EDF method

Number of requests	Crane Operation Fitness Value		
	Saving	P-Value	Computational Time
10	24%	$< 10^{-15}$	10.78 ± 0.13
20	21%	$< 10^{-15}$	12.31 ± 0.1
30	25%	$< 10^{-15}$	13.65 ± 0.13
40	24%	$< 10^{-15}$	14.78 ± 0.14
50	30%	$< 10^{-15}$	16.23 ± 0.18
100	27%	$< 10^{-15}$	23.37 ± 0.17
200	29%	$< 10^{-15}$	39.26 ± 0.18
300	35%	$< 10^{-15}$	56.73 ± 0.14
400	38%	$< 10^{-15}$	76.75 ± 0.19
500	35%	$< 10^{-15}$	95.61 ± 0.24
1000	34%	$< 10^{-15}$	202.44 ± 0.29

3.5.5 Sensitivity to the CSSSPD Structure

To further examine how the problem structure with respect to the time deadlines affects the performance of the proposed optimization method in comparison to the EDF method, a one-way sensitivity analysis was conducted for an arbitrary number of requests (here, 100 requests).

Requests were generated randomly using a uniform distribution. Random deadlines were generated for $[{}^{1}/_{10}\, n \times \bar{X}, 2n \times \bar{X}]$ interval in which ${}^{1}/_{10}\, n \times \bar{X}$ representing tight deadlines and $2n \times \bar{X}$representing wide deadlines. Figure 9 shows how the total crane operation fitness value (in percentage of the operation fitness value based on the EDF method) changes as deadline ranges get larger. This figures shows that when deadlines are very tight, the GA-based method's performance is very close to the EDF method. This is because that in case of tight deadlines the EDF method produces a decent solution, i.e. serving the request with the earliest deadline next. When deadlines get wider the GA-based optimal scheduling method can result in higher time savings by developing the enhanced sequence of request fulfillments.

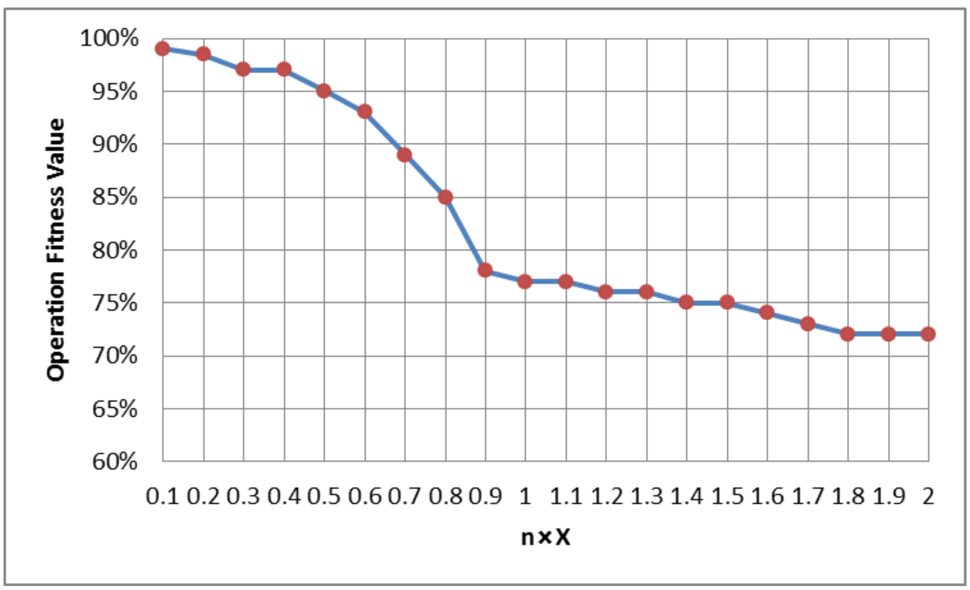

Figure 23: Sensitivity of crane operational time (in percentage of operational time based on the EDF method) to the magnitude of deadline ranges

3.6 Conclusion

This chapter presented a metaheuristic approach to solve the crane service sequence problem with deadlines (CSSPD) in construction operation. The problem was formulated using a well-known combinatorial optimization problem, i.e., the "Traveling Salesman Problem", and deadline constraints were added to the formulation. The main objective was to follow the construction schedule as closely as possible and thus the deadline constraints were treated as soft constraints. The proposed algorithm results in on average 29% saving in crane operation fitness value compare to the best manual approach available when service deadline exists. The small run time of the model makes it useful in practice, helping reduce crane operations and crane-related activity costs considerably.

Similar to any other studies, this research had some limitations that can be addressed in future studies. This study assumed that deadlines are soft constraints and thus can be violated to some extent. This assumption can be relaxed by developing an exact solution for the crane service sequencing with deadline and treat deadlines as hard constraints. Another simplifying assumption is the travel time model that has been considered deterministic and independent of the load. The impact of an accurate travel time model can be investigated in future research.

CHAPTER 4: A DECISION SUPPORT SYSTEM FOR REAL-TIME TOWER CRANE SCHEDULING

This paper introduces a new Decision Support System (DSS) for Crane Service Sequencing Problem (CSS) to help crane operators to schedule jobs in response to on-site's requests. DSS-CSS incorporates several mathematical models, including the travel time prediction model, the exact crane service sequence problem (CSSP) solver, and the CSSP with deadline heuristic solver. DSS-CSS is equipped with a user-friendly graphical user interface (GUI) with data collection and data analysis modules to produce a service sequence with the minimal crane cycle time. The crane operation schedule is continuously updated in real-time based on the latest conditions of the system (e.g. current state of the crane, new requests, delay in operations, urgent service requests). Real-time optimal sequencing is shown to be effective in reducing the total crane travel time by 20%, resulting in a considerable improvement of the efficiency of operations and economic benefits.

4.1 Introduction

Construction activities are classified into three general categories: value adding (productive), non-value adding (waste), and value supporting (contributory) activities (Han, Lee, Fard, & Peña-Mora, 2007; Park, Yang, & Lee, 2012). Value adding activities are the necessary intermediate tasks to accomplish the main project goal. For example, in building a foundation, placing bars and pouring concrete are among the value adding activities. Non-value adding activities are those that do not add any value to the end product and one has interest in eliminating them (Koskela, 1992). Examples include the waiting time to receive materials, excessive inventory materials, and breaks. Value supporting activities are in between the value adding and non-value adding activities. The interest is in optimizing these activities as they indirectly facilitate the realization of the project goal. Inspection, cleaning, and transporting material are among the contributory (value supportive) activities. Serpell et al. (1995) found that productive activities account for 35-55% of construction time while contributory and non-contributory activities account for 24-34% and 18-31%, respectively (Figure 24). The contributory and productive activities are necessary to accomplish the project goal. However, excessive contributory activities decrease productivity. Thus, the optimal level of contributory activities must be determined (Park et al., 2012). Among the construction contributory activities, transportation is the largest element, being responsible for about 50% of contributory activities. In addition, waiting time to receive material, accounts for 37% of non-contributory activities (Figure 25) (Serpell, Venturi, & Contreras, 1995).

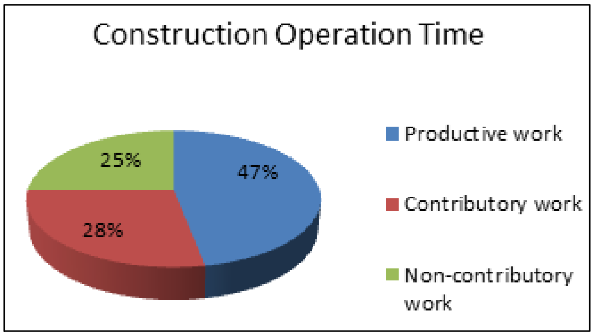

Figure 24: Construction operation time components (Serpell et al., 1995)

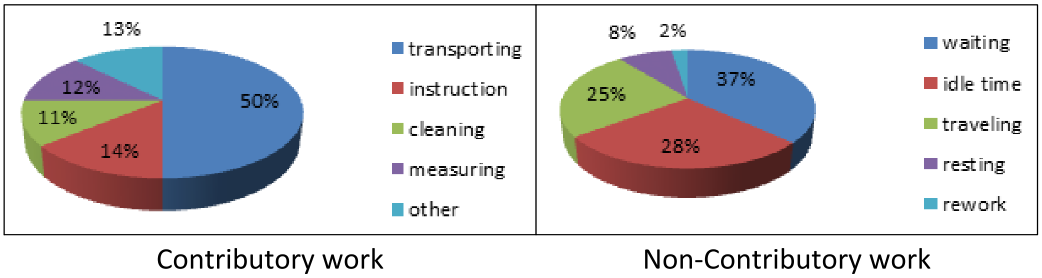

Figure 25: Contributory and non-contributory works time components

Optimization of travel time has direct and indirect advantages. Not only it reduced a significant portion of contributory work, but also it reduces the idle time of crew and equipment, significantly improving and enhancing the project performance (Park et al., 2012). Transportation can be divided into two categories: off-site transportation, dealing with material transportation from the manufacturer to the construction site, and on-site transportation, which is responsible material delivery to the construction crew. When crane is in charge of on-site transportation, enhancing tower crane operation's efficiency can significantly help with improving contributory activities and minimizing the non-contributory activities. This paper presents a site-level material supply planning and scheduling decision support system (DSS) for improving the efficiency of construction tower cranes.

Tower cranes are the centerpiece equipment in high-rise building construction on which the productivity of construction activities relies on. Therefore, they can turn into the bottleneck of the operations if not managed carefully. As the number of crane service requests increases, determining the optimized sequence of locations that the tower crane visits in order to fulfill the outstanding requests becomes complicated. In addition, request significance, precedence, and deadline can make the problem much more complex.

4.2 Background

Reducing the crane operation time has been one of the research areas in the construction management field. Facility layout optimization, i.e. determining the optimal location of tower cranes and supply locations is one of the methods for reducing the crane's total travel time. Zhang et al. (1996) used a Monte-Carlo simulation method to determine the optimal crane location, reducing the crane's horizontal travel time by 20-40% (Zhang et al., 1996). Zhang et al. (1999) developed a mathematical method to optimize the locations of a group of tower cranes on the job site, resulting a 10-40% saving in the hook's travel time (Zhang et al., 1999). Tam et al. (2001) investigated reducing crane operation time through optimizing the locations of supply nodes around a central tower crane using a genetic algorithm (C. M. Tam et al., 2001). Huang et al. (2011) optimized the tower crane and material supply locations using mixed-integer linear programming (Huang et al., 2011).

In addition to site layout optimization, reducing the crane travel time has also been pursued using add-on technologies such as vision enhancement systems (Everett & Slocum, 1993a; Lee et al., 2012; A. Shapira et al., 2008) as well as collision detection systems (S. Kang & E. Miranda, 2006; Lei, 2011; Sivakumar et al., 2003), which help facilitating crane navigation, especially when the operator's line of sight is obstructed. Rosenfeld and Shapira (Rosenfeld & Shapira, 1998) examined the feasibility of retrofitting an existing tower crane with a semi-automatic system, installed on an in-door overhead crane. 15-40% saving in maneuvering time was reported by these researchers. Shapira et al. (2008) reported the application of a mounting live video system on tower cranes in order to improve crane operations efficiency, resulting in 14-29% saving in the total travel time (aviad Shapira et al., 2008). Table 10 summarizes the previous research on reducing crane operation time.

Table 10: Past research on crane operation time estimation and improvement

Objective(s)	Method	Time saving (%)	Citation
Predicting crane hoisting time	Linear regression model	NA	Leung and Tom (1999)
Single tower crane location optimization	Monte-Carlo simulation	20-40%	Zhang et al. (1996)
Group tower crane location optimization	Monte-Carlo simulation	10-40%	Zhang et al. (1999)
Supply location optimization around tower crane	Genetic algorithm-based optimization	18%	Tam et al. (2001)
Tower crane and supply locations optimization	Mixed integer programming	7%	Huang et al. (2011)
Reduce travel time and increase safety	Camera system utilization	14-29%	Shapira et al. (2008)
Decreasing maneuvering time	Navigation system utilization	15-40%	Rosenfeld & Shapira (1998)

The objective of this Chapter is to design a functional crane operations decision support system (DSS) which integrates data collection and processing into a physical crane model in order to achieve the highest possible productivity and to facilitate planning crane operations in real time. This DSS lets the crew place their requests and track their request processing. The DSS reduces the responsibility of the superintendent in the process by using a computational module which expedites the operation and reduces the inventory waiting time.

4.3 Crane Service Sequencing Decision Support System (CSS-DSS)

Decision Support Systems (DSSs) are described as interactive computer-based systems that gather information from multiple sources, compile useful information, and facilitate users' judgment and decision making in a structured environment (Ecker, Gupta, & Schmidt, 1997). Human judgment can result in sub-optimal results, especially when decision making involves multiple steps and requires extensive amount of information (Druzdzel & Flynn, 2003). DSSs use structured reasoning approaches such as statistics and Operations Research in conjunction with computer power to solve complex problems. The term "interactive" refers to the crucial component of every DSS as they must be user friendly and easy to use in practice.

Choosing the best sequence among available alternatives (CSSP) is a hard combinatorial optimization problem (A Zavichi et al., 2013). It is almost impossible for an unaided decision maker using his/her intuition to choose the best sequence of alternatives. Therefore, there is a need for a DSS that uses the jobsite information and deploys optimization methods can be used for scheduling by the superintendent or directly at the crane's operator can significantly improve the efficiency of crane operations. Such a DSS is developed here.

Figure 26 shows the structure of the proposed crane decision support framework. This system consists of three major units: information, request collection, and computational units, illustrated in the following sections.

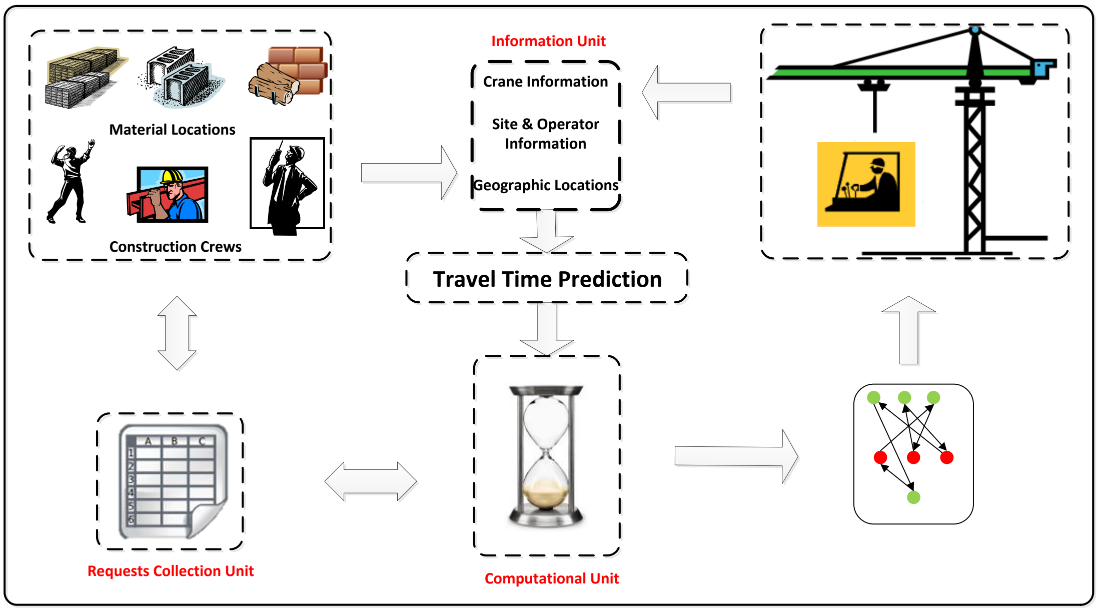

Figure 26: Schematic of the proposed request sequence optimization system

4.3.1 Information Unit

Information unit is the core of the DSS and collects information from multiple sources. Input information consists of several components: crane specifications, geographical locations of material and crews on the job site, crane operator skills, and site conditions. The travel time prediction model, incorporated in the computational unit, uses this information for service sequence optimization. Components of the information unit are described in the following sections.

4.3.1.1 Crane Specification Information

Crane's hook movement evaluation is important to estimate the material transportation time. At most three movements are required to relocate the trolley between two points. The winch motor lifts the load and moves the trolley vertically. Slewing motor moves the jib in the angular direction, and in case the points are not in the same working radius the trolley runs along the jib to move the load in and out (radial direction) to reach the target location. A velocity is associated with each direction. The radial $V_r(m/min)$, angular $V_a(rpm)$, and vertical $V_v(m/min)$ velocities can be obtained based on the operating crane manufacturing specifications. This input module is shown in Figure 27.

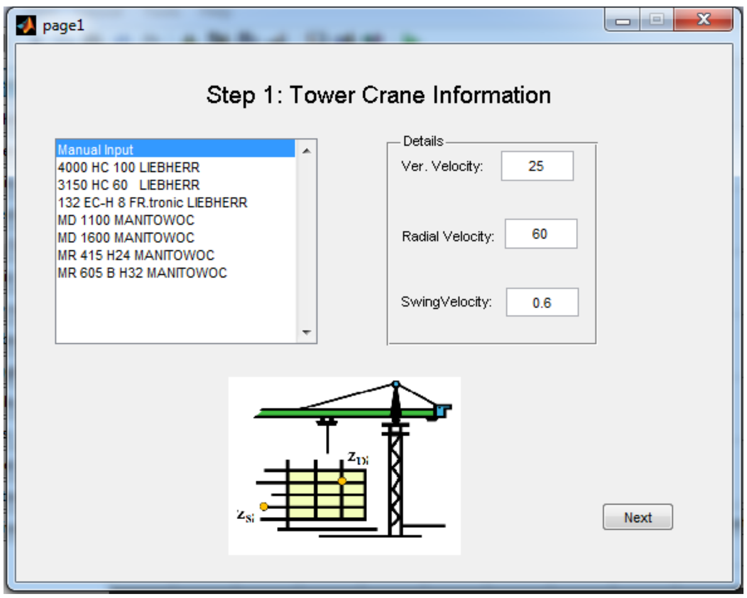

Figure 27: Crane specifications input module

4.3.1.2 Geographical Location Information

In addition to crane specifications, the travel time prediction model requires supply and demand locations on the job site. Due to the dynamics of construction projects, supply and demand locations might be changing over time this possibility must be considered by the DSS. Two approaches, grid system and sensory geographical location information, are proposed to be used for determining the material storage (supply) and crew (demand) positions on the job site.

4.3.1.2.1 Grid System

Based on this method (Figure 28), potential supply and demand locations are considered as discrete points in space and are specified with 3D coordinates. An identifying tag is assigned to each node in the grid system that is saved along its coordinates in the information unit. No intrinsic discretion is defined for supply and demand locations at this stage, and thus each of the nodes can function both as supply or demand nodes. In addition, some nodes in the grid can be identified as “banned nodes” due to safety issues or other reasons (solid diamonds in Figure 28), that the crane is prohibited to visit.

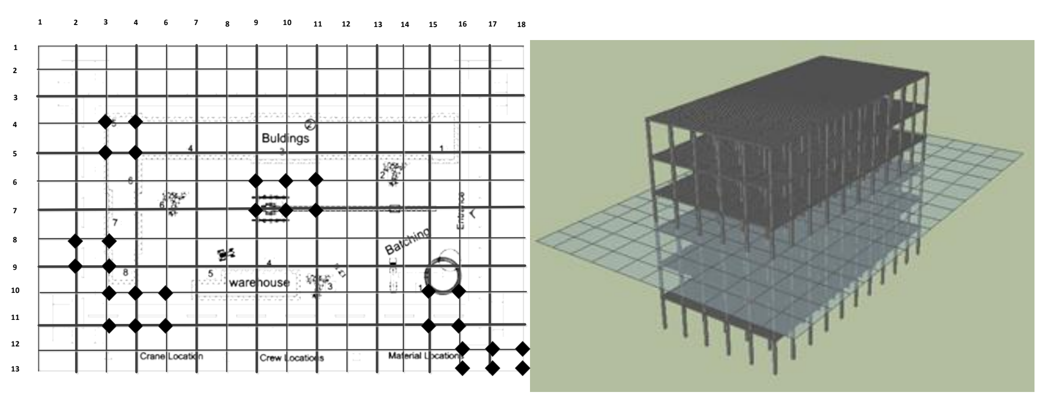

Figure 28: 2D (left) and 3D (right) representations of potential supply and demand nodes using a predefined grid system

4.3.1.2.2 <u>Geographical Location Information</u>

Based on this method a sensor data acquisition system, such as a network of Real Time Kinematic (RTK) (Figure 29), is used that transfers supply and demand locations in real time. For this purpose, each mobile crew platform must be equipped with an RTK antenna, positioned relative to the RTK base stations.

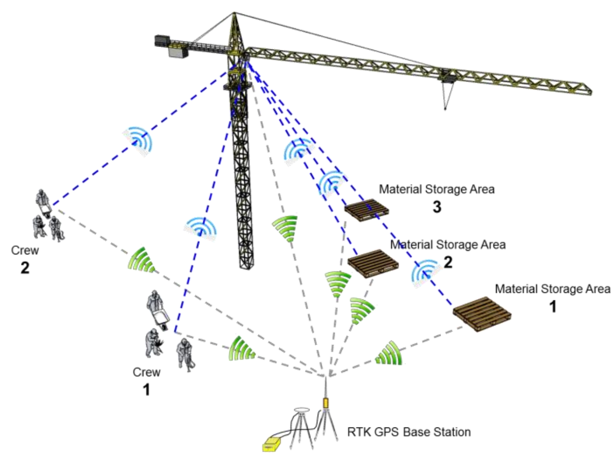

Figure 29: Schematic representation of the data acquisition method

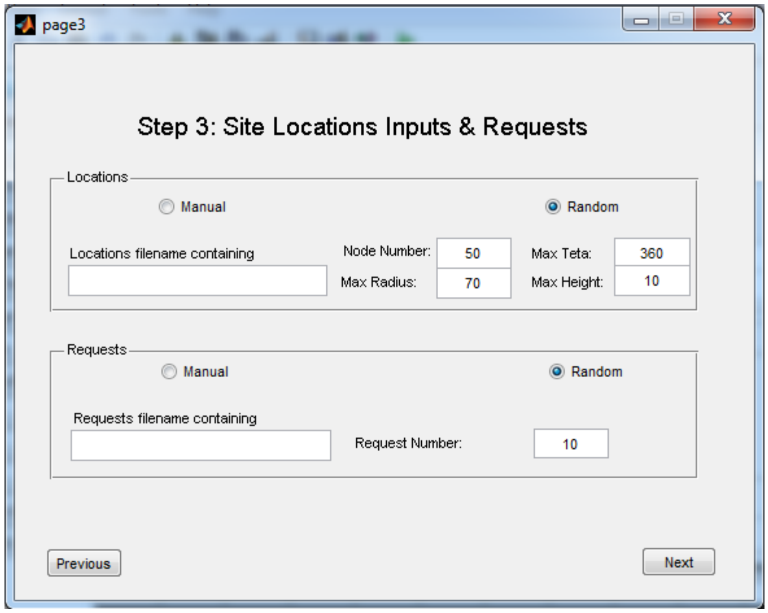

Figure 30: Supply and demand locations input module

The grid system is efficient when the site is large in size and the crane has to visit many locations. The sensor-based data acquisition system is useable when the accuracy of locations is important and the crews are moving frequently. In this case, 3D coordinates are introduced using the site location input module (Figure 30).

4.3.2 Site and Operator Information

Crane travel time is influenced by operator skill and different site conditions. Three parameters are used to account for the operator skill ($\propto$ & β) and different site conditions (γ) (A Zavichi et al., 2013). $\propto$ is the degree of simultaneity in radial and angular directions. In addition, β is used to take into account the simultaneity degree in horizontal and vertical plane. Total travel time can be increased due to undesirable site conditions such as adverse weather, existence of obstacles and safety issues. To account for working site conditions, a numerical parameter (γ) is used.

Parameters ∝ and β are continuous positive numbers less than or equal to one. ∝ equal to one implies that the radial and angular movements are actuated separately; β equal to one implies sequential horizontal and vertical movements. In contrast, when ∝ and β get closer to zero, the movements find larger overlap in time. γ is one for normal weather conditions in an open area without obstructions on site, and greater than one otherwise. The values of ∝ and β and γ need to be calibrated by observed data obtained from the construction job site. These parameters are entered using the module shown in Figure 31.

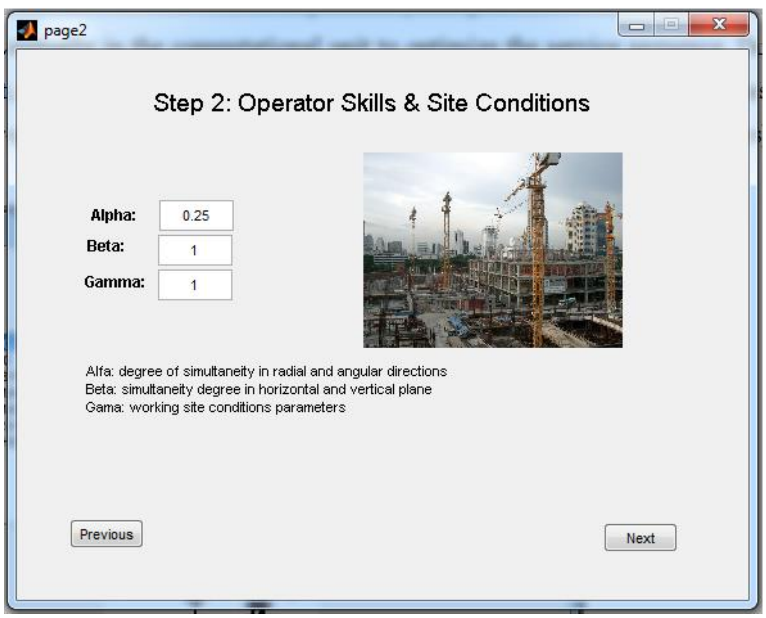

Figure 31: Operator skills and site conditions parameteres input module

By using the information provided by the user through the modules shown in Figure 27, Figure 30, and Figure 31, the travel time between all pairs of defined nodes on the job site can be calculated (A Zavichi et al., 2013) and be used by the computational unit in optimizing the service sequence.

4.3.3 Request Collection Unit

This unit is responsible for collecting the requests that are sent by crews to receive materials along with the information associated with their requests, such as the significance of their requests and the deadline to receive the materials. Requests are comprised of a pair of nodes as (a, b). The first component (a) is the material location or departure node that the hook starts from. The second component (b) is the crew location that the crane operator sends the material to. The request table is updated and re-optimized after each task is complete.

4.3.3.1 Dynamic Request Updating

Requests in system are updated, if necessary, after the crane operator fulfills each request. An update is required in two cases: 1) to consider for the newly intermittent requests and their associated conditions (e.g. priority and deadline); and 2) to consider the feedback received from the past performance. This feedback is crucial because it helps with continuous revision of the service schedule to address the past misjudgment due to lack of accurate information. For example, travel times between nodes are considered deterministic in the time model, so the optimization module is assumed to have a perfect foresight in the future. However, travel time is a stochastic variable in practice and can be affected by a range of factors. To reduce the effect of travel time uncertainty on the optimal solution, the schedule is re-optimized after each task is complete based on the latest conditions of the system (current time, current location of the hook, new demands, new emergency tasks, etc.). If needed, the sequence of tasks will be changed to ensure that the optimal schedule is followed and different constraints are met. For example, to meet the deadline for task c, which was supposed by follow task b, the operator might need to skip task b when a big delay in task a (preceding task b) has left insufficient time to accomplish both tasks b and c before the deadline.

Using the scheduling interface (Figure 32) the crane operator/scheduler can continuously update the service schedule based on the latest conditions of the system to ensure that the schedule does not deviate from the optimal path.

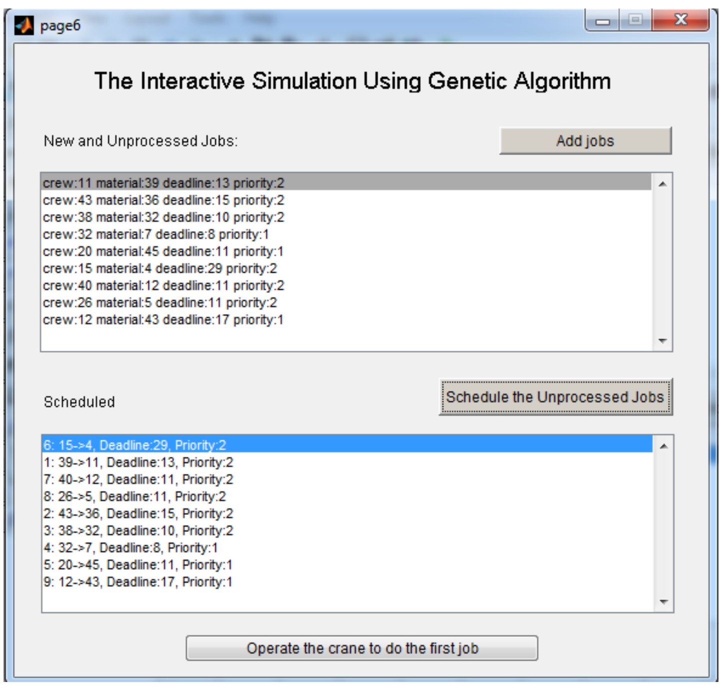

Figure 32: Interactive scheduling user interface

4.3.4 Computational Unit (Optimization Engine)

This unit uses the latest data from the travel time prediction model and request collection unit and develops the efficient sequence of crane services in order to minimize the overall crane's travel time. Each crew sends one or multiple crane requests to the automated DSS. At any given time, the DSS can suggest the most time-efficient sequence of all requests based on the spatial positions of crews and material storage areas, crane location, configurations and specifications, deadlines, and urgency levels. The service request table is updated continuously based on the conditions in the job site (new requests, cancellations, etc.).

4.3.4.1 Solution Algorithm

The problem of the crane service sequence can be reduced to a variation of Traveling Salesman Problem (TSP) by considering each request (pair of supply and demand locations) as a city in the TSP model (A Zavichi et al., 2013). TSP has very practical applications in real world and thus is one of the widely studied problems in combinatorial optimization. Consider having n cities. The goal of TSP is to find the shortest circuit possible between cities to travel and visit each city exactly once and return to the home city (Lawler, Lenstra, Kan, & Shmoys, 1985). The scheduling complexity increases when constraints such as deadline and significance of ongoing tasks are added to the problem (Chapter 3). Researchers have used heuristic algorithms to obtain near-optimal solutions for TSP (Bansal, Blum, Chawla, & Meyerson, 2004; Choi, Kim, & Kim, 2003; Fogel, 1988; Yu et al., 2011). The CSSP search engine uses a hybrid penalty-based genetic algorithm (GA) technique to optimize the requests' sequence in order to minimize operation time with respect to deadlines and priority of requests.

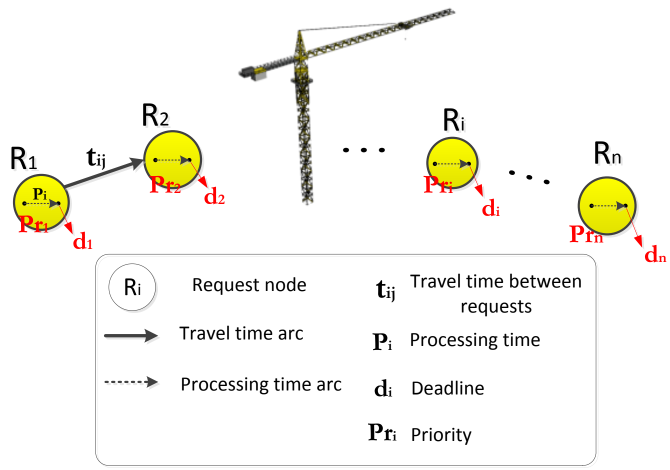

Figure 33: Transforming the crane service problem into ATSP

Figure 33 shows the schematic overview of the crane service sequence problem (CSSP). The primary objective of this problem is to minimize the total travel time operation. The secondary objective is to minimize the violated deadlines.

CSSP can be defined on a graph $G = (R, T)$ where R is set of requests defined as nodes and T is the arc set. Each arc is associated with a ravel time (t_{ij}), which is the time of switching between requests. In addition, a service time (p_i) that is the time for fulfilling a request (traveling from a supply location to a demand location) is associated with every node. p_i and t_{ij} are extracted from the travel time prediction unit output once the requests are entered to the DSS. Each node ($i \epsilon R$) might also have a deadline (d_i) and a priority (pr_i). A deadline is the latest possible time to fulfill a request and assumed to be a positive value. Priority reflects the urgency level of a request. Here, priority numbers are integers between 1 and 3. A higher priority number corresponds to a higher priority service. Including priority in the DSS is especially useful when dealing with intermittent requests that need urgent service. Deadlines are normally specified by the service requestors (might require approval from a supervisor, depending on the management structure on the site) and priorities are normally given by the scheduler or superintendent. Priorities and deadlines are additional constraints that make CSSP different from a typical asymmetric TSP. Given the complexity of this NP-complete optimization problem, GA is used as the optimization solution method. Each solution is conceived as a sequence of requests and is encoded as a string of genes called chromosomes (Holland, 1975). Each chromosome encompasses all genes without repetitions (path representation) (Greco, 2008; Murata, Ishibuchi, & Tanaka, 1996) (Murata et al., 1996). The evolutionary algorithm process is illustrated in Figure 34.

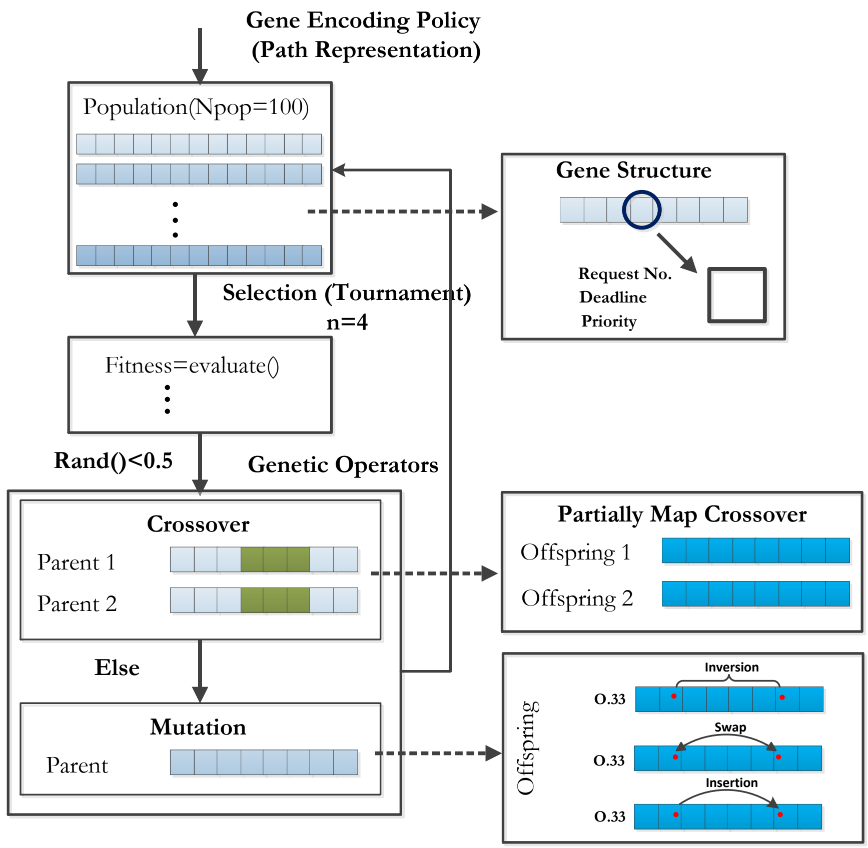

Figure 34: Overview of the solution algorithm

The solution method is initiated by generating N_{pop} random solutions (chromosomes). Healthier chromosomes are selected to be used as parents to produce offspring for the next generation. The quality of an individual is measured with an evaluation function that considers total travel time as well as the level of respect for the deadline and priority constraints. The goal for the scheduler is to minimize total time of the operation along with minimizing the deviation from constraints. So, the objective function is as follows:

$$\min f(\sigma) = f(\text{Total Operation Time, Deadline Violation}) \qquad (20)$$

where Total Operation Time is the total time for fulfilling the requests without considering any constraints; Deadline Violation is defined as deviation from the deadlines.

To account for the priorities, priorities for all requests in the system are stacked according to their priority number form highly urgent (3) to normal (1). Then, the GA is run for each urgency level considering the two major objectives: minimizing the total travel time and minimizing deviations from deadlines. Given a request route sequence $< R_1, R_2, \dots, R_{n-1}, R_n >$, the objective function is defined in which the deviation from deadline is penalized with penalty:

$$Min\ \sum_{i=1}^{n-1} t_{i,i+1} + \sum_{i=1}^{n} p_i + \alpha . \sum_{i=1}^{n} lv_i \qquad \forall\ i \in V \qquad (21)$$

where the first component is the total travel time for switching between requests; the second component is sum of the requests' processing times. Sum of the two first components is the total time to fulfill requests (Figure 33). Finally, the third component is sum of all penalties for possible violation of the deadline constraints in a sequence. α is the penalty coefficient that can be customized by the user. For request nodes in the sequence, penalty value (lv) is computed using Equation (22) in which s_k represents arrival time at the node, p_k is processing time, and d_k is the deadline for the corresponding node. Penalty value is set to zero if the request is fulfilled before the deadline.

$$lv_k = (s_k + p_k) - d_k, \ \begin{Bmatrix} lv = 0, if\ lv \leq 0 \\ lv = lv, if\ lv > 0 \end{Bmatrix}, \forall\ k \in V \qquad (22)$$

CSSP aims at finding the minimum value for the objective function and therefore, the reciprocal of the solution value represents the fitness value: $f_{(i)} = {}^{1}/_{f_{(R_i)}}$. The longer the time of operation to fulfill the request, the smaller the fitness value will be. In each population, selection operator chooses the mates to be used in crossover and mutation operator to produce offspring for the next generation. Generally, it is recommended that the chromosomes with higher fitness value (or lower objective function value) have higher chance of selecting in the next generation (Tavakkoli-Moghaddam, Makui, Salahi, Bazzazi, & Taheri, 2009). Common applicable selection methods for CSSP include tournament selection, roulette wheel, and rank-based roulette wheel selection. Tournament selection is shown to be efficient and is used in this paper (Noraini & Geraghty, 2011). In tournament, n chromosomes (tournament parameter) are chosen randomly from the population to compete against each other. The winner which is the chromosomes with highest fitness function value will be included in the next generation. Larger tournament size reduces the diversity of the population. However larger diversity reduces convergence speed.

Crossover combines genes of two chromosomes to produce offspring that have features from both parents in the next generation. Common crossovers are Partially Mapped Crossover (D. E. Goldberg, Lingle, R, 1985), Order Crossover (Davis, 1985), Order Based Crossover (D. E. Goldberg, 1989), and Cycle Crossover (Oliver, Smith, & Holland, 1987). Partially mapped crossover (PMX) has been proven as an effective with fast convergence rate operator (Larranaga, Kuijpers, Murga, Inza, & Dizdarevic, 1999; Lawler et al., 1985) and thus is chosen to be used in here. PMX is functioning as following:

PMX starts with two randomly selected parents, and determines two positions randomly along the chromosome (crossover points). Genes between these two crossover points are directly inherited by the offspring. Since changing genes directly between two crossover points lead to illegal chromosomes with missing or repeated genes, PMX follows the following procedure: each gene between the crossover points in one parent is directly mapped onto the position held by this element in the other parent. The gene value that has been occupied by the mapped elements is transferred to the location that has the mapped element value in it to retain the legality of the chromosomes. This procedure will continue until all elements between the crossover points are mapped. Figure 35 shows PMX procedure with crossover for the first three genes (Fogel, 1988; Starkweather, McDaniel, Mathias, Whitley, & Whitley, 1991; Üçoluk, 1997).

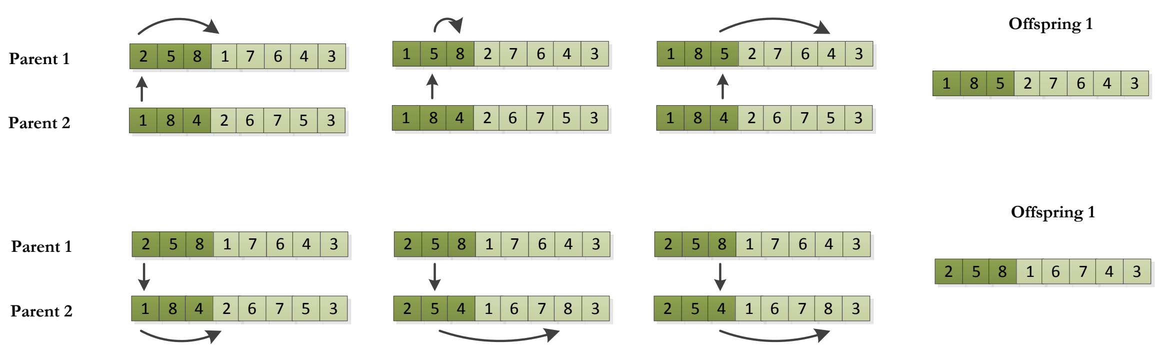

Figure 35: PMX with one crossover point

The mutation operator is the random modification on individual chromosomes and is used to enhance the diversity and provides a chance to escape from local optima. Operators used in this algorithm consist of inversion, swap, and insertion operators (Yu et al., 2011). The inversion operator chooses two random points in the chromosome and reverses the order of the genes between them (Melanie, 1999). The swap operator uses 2-opt heuristic to substitutes the value of two randomly chosen nodes in order to improve the constructed tour (Croes, 1958), and finally the insertion operator inserts the value of a randomly selected gene into another location in the string. Figure 36 shows the schema of the mutation operators.

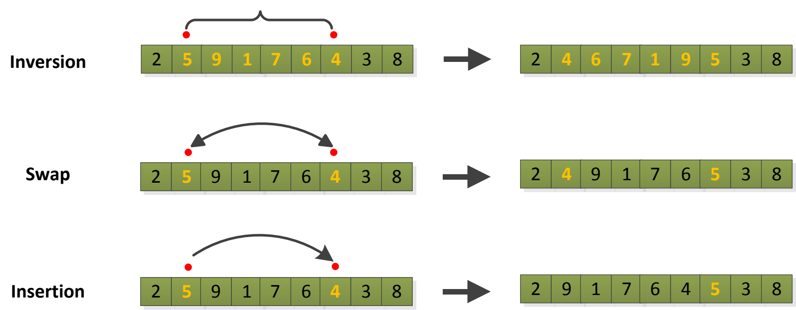

Figure 36: Mutation operators

4.4 Experimental Results

We carried out the numerical experiment on an Intel Core i5 processor (i5-520M / 2.4 GHz) with a 4 GB RAM, and application software was developed using MATLAB's graphical user interface (GUI). A prototype site layout with 50 randomly generated coordinates, using a uniform distribution, scattered around a central tower crane with 70 meter operation radius, is considered. The elevation of nodes varies from zero to ten meters representing the jobsite elevation difference. Requests are generated randomly using a uniform distribution. In the GA, size of population (N_{pop}) is set to 100. Tournament parameter is 4 and crossover probability ratio is 0.5. If crossover is selected, the top two fittest chromosomes are selected for reproduction in order to produce two offspring; or else for mutation only one chromosome is selected. In mutation, the probability of inversion, swap, and insertion is 0.33 and one offspring will be produced. This procedure will be repeated until 100 chromosomes are produced during the next generation. Finally, the number of iteration is set to 200 as the stopping criterion.

4.4.1 Computational Experiment: GA Performance Comparison with Conventional Heuristics Methods

In the first set-up, the performance of the GA algorithm is compared to the FIFO (First-in-First-Out) and Earliest Deadline First (EDF). In FIFO, requests are fulfilled based on their arrival time. In EDF, request with the earliest deadline will get priority over others and fulfilled first. For now, priority constraint is disregarded as none of the conventional scheduling methods consider deadline and priority constraints at the same time. Deadlines are assigned randomly to the demand node in every request -considering every request consists of a supply and a demand node- in a range of zero to $n \times \bar{X}$, where n is the number of requests and $\bar{X}$ is the average travel time between requests for the set of requests in the system. Penalty coefficient (α) is set to two (refer to Chapter 4 for choosing the penalty function coefficient calculation).

Fitness value for FIFO, EDF, and the proposed GA algorithm to solve crane service sequence problem are compared by solving the problem of 10 requests for 100 iterations. Figure 37 shows the comparison between the aforementioned scheduling methods. Vertical axis shows the fitness value which is the sum of the total operation times and deadline deviation penalty. Deadline deviation penalty is the deviated time from deadline, and is considered as the penalty that is added to the total operation time. Based on Figure 37, FIFO is by far the worst scheduling method when it comes to considering deadlines in the operation as it does not have any means to reflect deadline. EDF however is very good when the only constraint is deadline compared to the FIFO. The proposed GA outperforms the other two algorithms and presents the saving of 18% and 45% compared to the EDF and FIFO respectively.

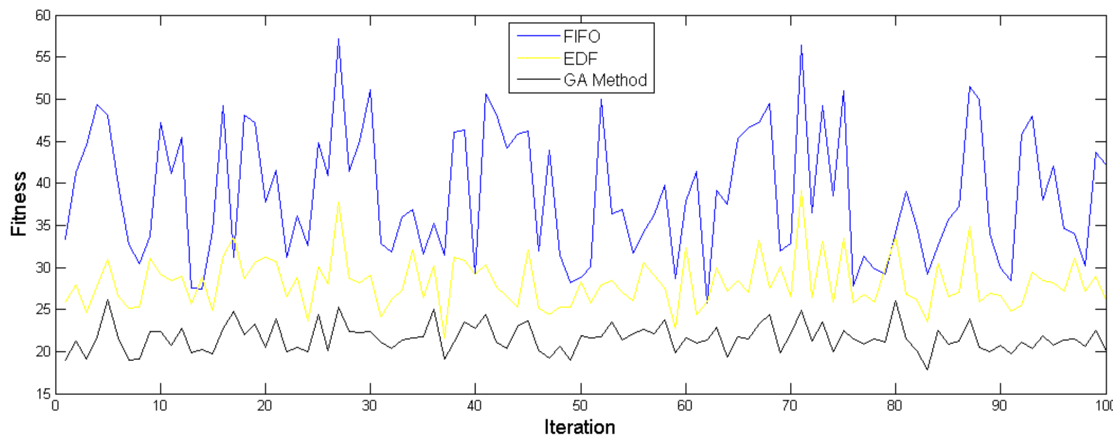

Figure 37: Comparison of the fitness value based on FIFO, EDF, and the proposed GA method

The fitness value, compared in Figure 37 consists of two portions: operation time and violation time. If we are interested in comparing the operation time portion between the scheduling methods, still about 15% saving is expected. In addition, no distinct significance can be seen between FIFO and EDF methods in terms of operation time. The results are depicted in Figure 38.

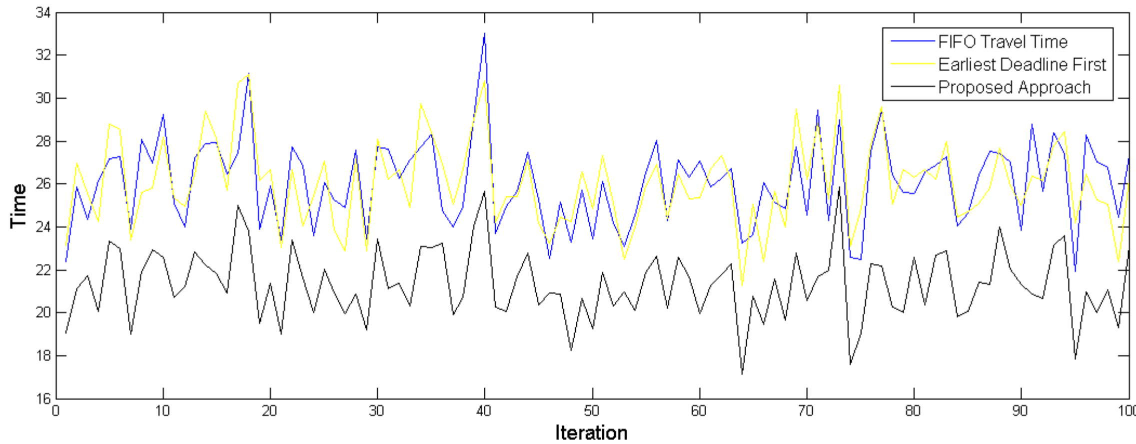

Figure 38: Operation time comparative result between FIFO, EDF, and GA

Another byproduct of implementing the proposed GA approach is reducing the number of deadline violations; however, the direct objective of the GA was to reduce the fitness value and not directly reducing the number of deadline violation. The average numbers of deadlines that are met are compared for three algorithms in Figure 39. As can be seen, FIFO is not recommended to be used while deadlines are associated to the requests. The proposed GA works closely compare to EDF in terms of the number of deadlines that are met. However, EDF is slightly better on average as its primary goal is to meet the deadlines.

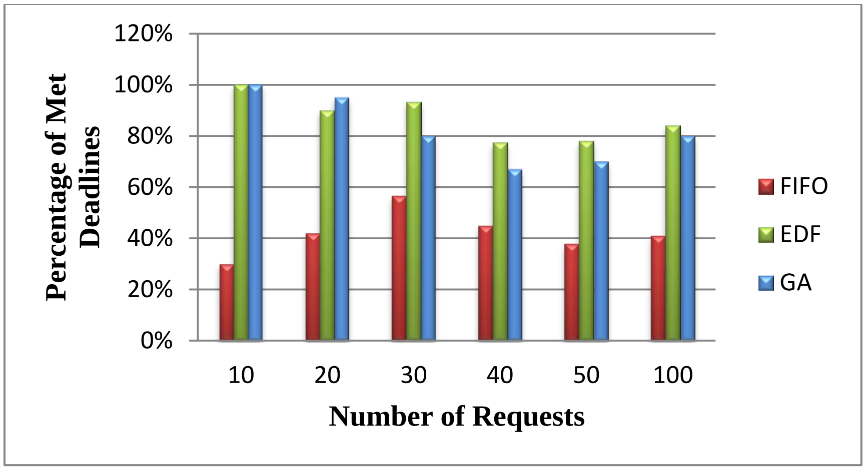

Figure 39: Deadline violations based on FIFO, EDF, and GA methods

4.4.2 Computational Experiment: Dynamic Sequencing

In this setup, the effects of considering intermittent requests in the system are considered and are compared to the situation in which the intermittent requests are not accepted until the current batch in the system is being fulfilled. The assumption is that the system will be updated for intermittent requests after each request is fulfilled. In other word, operator cannot leave the current request until it gets fulfilled.

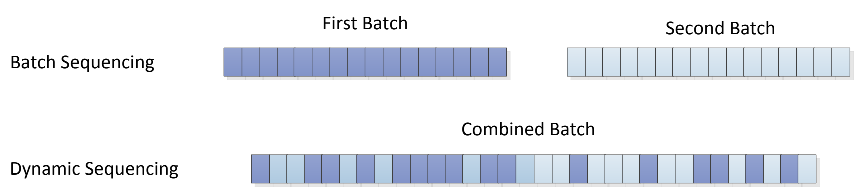

Figure 40: Dynamic sequencing versus batch sequencing scheduling

For this purpose, a primary batch is created randomly with n number of requests with their associated deadlines and priority. Another batch with the same characteristic is created as well.

Both batches are optimized separately with the proposed GA method. In addition, the time to fulfill batches sequentially is compared to the time when the intermittent requests are accepted. The problem is run for 100 iterations and the average operation time for both sets is compared. When the number of batch requests equals to 10, the comparison is shown in Figure 41.

Figure 41: Comparison of operation time based on batch scheduling and dynamic scheduling

In a short time period, the dynamics of request's sequencing might have adverse effect on the operation time; however, it will reduce the total operation time in the long run. In this case, on average, a saving of 20% is expected. In short period of time, an increase in the operation time is expected by considering the intermittent requests due to the fact that intermittent requests with deadlines will interrupt the efficient path that has been chosen by GA. Since the requests are addressed at the time they are received, fewer deadlines will be missed which results in less penalty accumulated which adds to the efficiency of the project in the long run.

Table 11 shows the results for 10 requests. Requests are classified based on their priorities and the requests with higher priorities are addressed earlier in the system. In each class, GA finds the best sequence based upon the objective of minimizing the time and reducing the deadline violation. The clock column shows the time at which the request is being fulfilled. If the request is fulfilled before the deadline, the deadline constraint is met and "ok" appears in the check column.

Table 11: A sample output for 10 requests

REQUESTS	NODES	PRIORITIES	DEADLINES	CLOCK	CHECK
1	38 -> 21	2	3	1.310185	OK
2	47 -> 14	2	4	3.85625	OK
3	25 -> 14	2	16	6.306111	OK
4	49 -> 36	2	17	8.842222	OK
5	12 -> 38	2	19	11.833472	OK
6	19 -> 32	1	6	12.703194	-
7	36 -> 25	1	9	14.937824	-
8	32 -> 33	1	12	16.702731	-
9	27 -> 37	1	19	19.031713	-
10	39 -> 38	1	27	21.327315	OK

4.5 Cost Benefit Analysis of the Proposed Optimization Method

Crane cycle time entails loading/unloading and travel times. Each portion's share in a crane cycle depends majorly on two parameters: (1) the type of the load and (2) the height of the structure under the construction (A. Shapira et al., 2008). In addition, other parameters such as weather, crane hoisting speed, crew productivity, and operator competence are also determinant in crane cycle time. In high-rise building, hoisting speed is the governing factor, and the type of the load is less important in determining overall cycle time.

Traveling portion of the crane cycle, which is the focus of this study, can vary significantly from a small percentage (e.g. 10%) in low-rise buildings with loading large/heavy framework panels to very high percentage (e.g. 80%) in high-rise buildings. This research mainly contributes to the high-rise buildings in which the percentage of traveling time out of the crane cycle time is high. However to be conservative, traveling portion is assumed 50% of the total crane cycle ($\rho = 0.5$). Based on the analysis presented in this research whether it is CSSP with deadline or without, on average, 30% of the traveling time portion can be saved by using this system ($\gamma = 0.3$).

In general, tower crane utilization in construction job sites is estimated to be 50% to 80% (Hasan, Bouferguene, Al-Hussein, Gillis, & Telyas, 2013; Kay, 2001; Rosenfeld & Shapira, 1998). Shortening the crane cycle not only has effect on increasing the crane productivity but also could lead to reducing the total project duration. This is mainly because the tower crane, during its busy schedule, very soon becomes the bottleneck of the project that will determine the project production rate, and thus enhancing the crane operation will not only increases the crane utilization but also expedites the work that are related to crane operations (e.g. workers that are waiting for material), and it leads to more efficient job site operations (A. Shapira et al., 2008). Utilization is considered 65% for calculation purpose ($u = 0.65$).

To summarize:

- $\rho = 0.5$: Mean percentage of tower crane traveling time (out of total crane cycles)

- $\gamma = 0.3$: Fraction of tower crane traveling time that is saved through using this crane sequence optimization
- $u = 0.65$: Mean tower crane utilization

By using the proposed system, saving time for the entire crane operation is on average 10% which is the product of three parameters (ρ, γ, u). This number is used for further economic analysis.

A similar study for economic feasibility of the crane operation is done by Rosenfeld and Shapira (Rosenfeld & Shapira, 1998). We are conducting the same analysis procedure with the updated cost of the operation. The costs are extracted from two sources: 1- California Department of Industrial Relations for hourly labor wages (Relations, 2013), and 2- Labor surcharge and equipment rental rates for California's Department of Transportation (Transportation, 2013). Saving in crane operation has impact mainly on two major parts. First, saving is in crane operation attributed to shortening crane cycle time. Second, higher labor productivity is expected due to shortening the crane operation. Other advantages such as: transparency in the operation process, eliminating the need for full time scheduler, expediting the total operation by reducing the waiting time in system are other benefits of this system that are conservatively ignored for financial analysis as there are no simple calculation to find their impacts.

Saving related to the crane operation costs can be divided into four categories:

1. Annual ownership cost: This is depreciated cost of buying a crane throughout its useful life span.
2. Operation cost: includes fuel, maintenance, license fees, and insurance.
3. Operator cost: This is the salary of the crane operator.
4. Labor cost: cost of the laborer working directly with the crane.

These elements are calculated in the following:

4.5.1 Annual ownership cost

Tower cranes are often owned by the contractor especially in the U.S. unlike mobile cranes that are often procured by rental companies and subcontractors (Shapira & Glascock, 1996). This is basically due to the fact that tower cranes stay longer in the project and rental rates and mobilizing costs are high and thus are not economic to rent (Rosenfeld & Shapira, 1998; Sullivan, Wicks, & Luxhoj, 2009). In order to calculate the time dependent annual cost of the tower crane, U.S. Army Corps of Engineers pamphlet is used as the reference (Engineers, 2011). The methodology described in the following is only applies to equipment that prime contractors or subcontractors either own or control. For this purpose equivalent uniform annual cost of ownership is calculated, and the salvage value, if any, that may be recovered at the end of asset useful life is deducted accordingly (Newnan, Eschenbach, & Lavelle, 2004):

$$annual\ ownership\ cost = P \times \left[\frac{i(1+i)^N}{(1+i)^N-1}\right] - F \times \left[\frac{i}{(1+i)^N-1}\right] \qquad (23)$$

Where:

P: Present value of crane or the purchase price

F: Future value or salvage price of the crane, is defined as the price for which the crane can be sold at a particular time in future

i: Nominal interest rate

N: Crane economic life in years, or duration which the investment is made

The first component in annual ownership cost is the annual equivalent payment for the crane purchase price during the crane's working life span. The second component is the annual equivalent credit for the predicted salvage or sell price that represent the depreciation of the equipment. Using the following parameters, annual ownership cost is estimated $51,155.

- P = $650,000; Tower cranes range in price from $200,000 to over $6 million each depending on size. For a good average purchase price, $650,000 is used as an average price (Carbeau, 2012).
- F = $130,000; residual value of heavy equipment such as cranes are mainly unknown (Lucko & Vorster, 2003). US Army Corps of Engineers (Engineers, 2011) suggested 20% of its initial purchase price as the salvage value of the tower crane.
- i =5.6 %; this value is extracted based on the historical data between 1962 to 2012 nominal interest rate using U.S. treasury market data base (System, 2013).
- N=20 years; 20 years is considered for useful life of tower crane (Engineers, 2011).

4.5.2 Operation cost

Major operation costs entail fuel, maintenance, license and insurance the will be estimated in the following annually:

- Annual fuel cost for a tower crane with an electric motor is calculated using the following formula (Engineers, 2011):

$$\frac{fuel\ cost}{hr} = fuel\ factor \times horsepower(HP) \times energy\ cost(\frac{kw}{hr}) \qquad (24)$$

Fuel factor is 0.24 for electrical tower crane and it takes into account the conversion rate for hp/kw in the formula. The horsepower is sum of the all the tower crane's engines horsepower (Table 12). Driving units (467KW) for heavy-load 4000 HC 100 LIEBHERR tower crane is used to calculate the horsepower using 1.34 as conversion factor. Energy cost range between $0.2 to $0.5 per kilowatt hour depends on the time of the year. Conservatively, $0.2/hr is used as the energy cost.

Annual average operating hours per year is determined by reducing the maximum normal working hours (40 hours per week, 52 weeks per year) to consider the adverse impact of: weather, equipment down time, holidays and etc. and is considered 1800 hrs./year for tower cranes (Engineers, 2011). The calculation yields to $54,067 annual fuel cost.

Table 12: Driving units for a 400 HC 100 LIEBHERR tower crane

Hoist gear	**Slewing gear**	**Trolley travel gear**
340 KW	6×15 KW	37 KW
	Total Power=	467 KW

- Service and maintenance is considered to be 6% of the purchase price ($39,000) (Rosenfeld & Shapira, 1998).

- Registration and insurance is estimated to be 2% of the purchase price ($13,000) (Rosenfeld & Shapira, 1998).

4.5.3 Operator cost

Operator cost is estimated to be $110,000 considering $65 hourly wage for tower crane operator (Relations, 2013).

Therefore, total cost for crane operation is sum of the aforementioned elements and is: $51,155+$54,067+$39,000+$13,000+$110,000 = $267,222. This leads to $148.5 per hour operating rate including the operator's salary. By using the request sequence optimization, 10% saving in total operation is expected which is $0.1 \times \$267{,}222 = \$26{,}722$ per year.

4.5.4 Reducing labor cost

Reducing the crane time and increasing the productivity of operation has direct effect on the laborer working directly with the crane operation. Four laborers are considered to be working in loading and unloading area and thus the annual saving for them is:

Saving= $4 \times 1800 \left(\frac{hr}{year}\right) \times 45 \left(\frac{\$}{hr}\right) \times 0.1(saving) = \$32{,}400/year$

Total saving of using this system is estimated conservatively to be $\$26{,}722 + \$32{,}400 = \$59{,}122/year$.

4.6 Conclusions

The temporary aspect of construction projects and the involved dynamics prevent one-time operations planning to be effective and hence continuous update of the operations schedule based on the latest conditions of the job site is essential to guarantee the success of the project. In this Chapter, a DSS was developed that helps the operator in crane operation scheduling. The real-time sequencing feature which allows for considering intermittent requests, showed to be 20% more efficient regardless of the scheduling method, resulting in significant economic savings.

CHAPTER 5: CONCLUSION

5.1 Introduction

The temporary aspect of construction projects and the involved dynamics might prevent efficient operations. Equipment operations and efficiency of individual pieces of equipment are major players in determining the overall project productivity and cost, and improper planning of them will lead to wasting the project resources. Cranes, by far, are one of the most expensive pieces of equipment in many construction projects as well as freight terminal operations. Since cranes play a major role in construction operations, expediting their operations would have direct impact on the operation efficiency.

The main motivation of this research was to overcome the existing problem in crane operation scheduling. Conventional scheduling in fleet management, including cranes, is manual, and very dependent to human judgment. Since human judgment combined with his/her ability to decision making can result in sub-optimal decisions, a decision support system (DSS) was developed to help the on-site scheduler or directly the crane operator in his/her routine operation.

5.2 Research Contributions

The following list summarizes the individual research challenges successfully addressed and described in the preceding chapters:

- Tower crane travel time prediction model: A mathematical tower crane travel time prediction model was developed as the basis for implementation of this research and was used repetitively for analysis purpose in Chapters 3, 4, and 5.

- An optimization technique to improve efficiency of the crane operations via prioritizing job requests was proposed in Chapter 3. An exact combinatorial optimization method, which is a modification of the “Traveling Salesman Problem (TSP)”, was proposed for minimizing construction crane travel time by optimal ordering of crane movement sequences. The suggested optimization model results in 20-30% saving in travel time in comparison with the conventional First-In-First-Out approach in fulfilling the requests. The optimization model’s performance is not highly sensitive to input parameters and different jobsite specifications. The small run time of the optimization model makes it useful in practice, helping with reducing the crane operations and crane-related activity costs considerably. The developed model optimizes the crane travel time only, which is the significant portion of crane cycle operations, especially in high rise construction and when loading and unloading nodes are not close.
- Adding a deadline constraint makes the CSSP problem more challenging. Therefore, in Chapter 4, to include this constraint, a penalty-based genetic algorithm was proposed. The deadline constraint was relaxed through a penalty function. Computational results showed an average saving of 27% in total travel time by using the proposed method compared to the best practices on construction sites. In addition to saving in crane travel time, this method was working acceptably in term of deadline violations.
- Chapter 5 developed a crane decision support system which integrates job site information and processing unit into a physical crane model in order to achieve the most time efficient service sequence scheduling. A very important feature of the proposed DSS is that the optimal task sequence is not calculated only once at the beginning of the

operation. Rather it is determined on a continous basis in order to take into account the evolving sequence of events in a typical project. A saving of on average 18% in long run is expected when intermittent request are accepted in the scheduling. In the same chapter priority of requests was also considered as an additional constraint and was addressed accordingly.

5.3 Limitations and Future Research

Similar to any other modeling study, this study had some limitations and simplifying assumptions that can be addressed in future studies. A general variant of the CSSP is when each supply material has several alternative locations. However, this research dealt only with a simplified version of the problem in which there is only one supply location for each material. Future studies can address this simplifying assumption. In the travel time prediction model, the travel time between two nodes was considered to be deterministic while travel time can vary in practice. Future studies can consider stochastic travel times. Given that the time savings increase with increased travel time resulting from elevation differences, future studies can investigate the effects of larger elevation differences (more than 10 meters) on travel time. This study assumed that each loaded bucket can be sent to one target location only, i.e., the crane hook cannot visit multiple demand nodes after being loaded. Future studies might relax this assumption. While in this study travel time was assumed to be independent of the load, future studies can evaluate the effects of material weight on travel time.

APPENDIX:
CRANE SERVICE SEQUENCING OPTIMIZATION BASE MODEL

Body Content

```
clear all;
%Number of Geographical Nodes on site
NodeNo=50;
%Number of requests [crews,material]
RequestNo=10;
%Initial Complete Travel Time Matrix
InitTravelMat=Travel_Mat(NodeNo);

FIFOTravelTimeMat=[];
SJFTimeMat=[];
GreedyTimeMat=[];
OptimalTimeMat=[];
FixedTravelTimeMat=[];
Time=[];
%%i is the number of iterations
for i=1:1
      [Requests,RequestsMat,Requestsok]=Requests_Gen(NodeNo,RequestNo,InitTra
      velMat);

      [FIFOTravelTime,FIFORequests,FixedTravelTime]=FIFO_Time(Requests,InitTr
      avelMat);
      FIFOTravelTimeMat=horzcat(FIFOTravelTimeMat,FIFOTravelTime);
      FixedTravelTimeMat=horzcat(FixedTravelTimeMat,FixedTravelTime);

      [SJFTime,SJFSequence]=SJF_Time(Requests,InitTravelMat);
      SJFTimeMat=horzcat(SJFTimeMat,SJFTime);

      [GreedyTime,GreedySequence]=Greedy_Time(Requests,InitTravelMat,Requests
      Mat);
      GreedyTimeMat=horzcat(GreedyTimeMat,GreedyTime);

      [OptimalTime,OptimalSequence,Distance,TimeElapsed]=Optimal_Time(Request
      sMat,Requestsok,Requests,InitTravelMat);
      OptimalTimeMat=horzcat(OptimalTimeMat,OptimalTime);

      Time=horzcat(Time,TimeElapsed);

end;

FIFOMean=mean(FIFOTravelTimeMat);
FIFOSD=std(FIFOTravelTimeMat);
OptimalMean=mean(OptimalTimeMat);
OptimalSD=std(OptimalTimeMat);
GreedyMean=mean(GreedyTimeMat);
GreedySD=std(GreedyTimeMat);
SJFMean=mean(SJFTimeMat);
SJFSD=std(SJFTimeMat);
[Hypothesis,PValue]=ttest2(FIFOTravelTimeMat,OptimalTimeMat);
TimeElapsedMean=mean(Time);
```

```
TimeElapsedSD=std(Time);
FixedMean=mean(FixedTravelTimeMat);
fprintf('*******************************\n')
fprintf('Optimal Mean is:%10.2f \t S.D. is: %10.2f ', OptimalMean,OptimalSD )
fprintf('\n\n Greedy Mean is:%10.2f \t S.D. is: %10.2f', GreedyMean,GreedySD)
fprintf('\n\n S.J.F. Mean is:%10.2f \t S.D. is: %10.2f', SJFMean,SJFSD )
fprintf('\n\n   FIFO Mean is:%10.2f \t S.D. is: %10.2f',FIFOMean,FIFOSD)
fprintf('\n\nSaving Percentage:')
fprintf('%10.2f ', (FIFOMean-OptimalMean)/(FIFOMean) )
fprintf('\nP-Value:')
fprintf('%10.2f ', PValue )
fprintf('\nTime Mean:%10.2f\t Time SD %10.2f ', TimeElapsedMean,
TimeElapsedSD )
fprintf('\nFixedTimePercentage:')
fprintf('%10.2f ', FixedMean/FIFOMean )
```

Travel Time_Function

```
function[InitTravelMat]=Travel_Mat(NodeNo)

% %input: NodeNumber(number of randomly generated nodes)
% %Outputs: Initial Travel Matrix for complete graph which is symmetric
% %Polar Coordination Generation

r=randi([10 70],NodeNo,1);
teta=randi([0 360],NodeNo,1);
z=randi([0 10],NodeNo,1);
PolarCoord=horzcat(r,teta,z);

%Polar Coordination Plot:
Polar(teta*pi/180,r,'+')

%Crane Specifications:
VerVelocity=25;   %vertical trolley velocity (m/min)
SwingVelocity=0.6; %revolution per minute
RadialVelocity=60; %(m/min)

MinHoistHeight=5;  %m

 for i=1:max(size(PolarCoord))
      for j=1:max(size(PolarCoord))
          RadialT(i,j)=abs(PolarCoord(i,1)-PolarCoord(j,1))/RadialVelocity;
          AngularT(i,j)=abs(PolarCoord(i,2)-
PolarCoord(j,2))/(SwingVelocity*360);
          VerticalT(i,j)=(abs(PolarCoord(i,3)-
PolarCoord(j,3))+(2*MinHoistHeight))/VerVelocity;
      end;
  end;
```

```
  %Site Parameters and operator skills
  Alpha=0.25;
  Beta=1;
  Gama=1;

  %Horizontal travel time
  HorizontalT=max(RadialT,AngularT)+Alpha*min(RadialT,AngularT);
  %Total travel time

InitTravelMat=Gama*(max(HorizontalT,VerticalT)+Beta*min(HorizontalT,VerticalT
));
```

Request Generation_Function

```
function
[Requests,RequestsMat,Requestsok]=Requests_Gen(NodeNo,RequestNo,InitTravelMat
)

%REQUESTS :[CREWS, MATERIALS]
%random crews
H=randi([2 NodeNo],RequestNo,1);
%random materials
F=randi([2 NodeNo],RequestNo,1);

Requests1=horzcat(H,F);

kf=0;
for i=1:max(size(Requests1(:,1)))
    if Requests1(i,1)~=Requests1(i,2)
        kf=kf+1;
        Requests(kf,1:2)=Requests1(i,1:2);
    end;
end;

Requestsok=vertcat([1,1],Requests);

%RequestMat is
for i=1:max(size(Requestsok));
    for j=1:max(size(Requestsok));
    RequestsMat(i,j)=InitTravelMat(Requestsok(i,1),Requestsok(j,2));
    end;
end;
```

First-In-First-Out Method-Function

```
function [FIFOTravelTime,FIFORequests,FixedTravelTime]=
FIFO_Time(Requests,InitTravelMat)

RequestsInRow=[];
for i=1:max(size(Requests(:,1)))
    for j=2:-1:1
        RequestsInRow=horzcat(RequestsInRow,Requests(i,j));
    end;
end;
RequestsInRow1=[1,RequestsInRow,1];

FIFOTravelTime=0;
for i=1:max(size(RequestsInRow1))-1
   FIFOTravelTime=
FIFOTravelTime+InitTravelMat(RequestsInRow1(1,i),RequestsInRow1(1,i+1));
end;
FIFORequests=RequestsInRow1;

FixedTravelTime=0;
for i=1:max(size(Requests(:,1)))

FixedTravelTime=FixedTravelTime+InitTravelMat(Requests(i,2),Requests(i,1));
end;
```

Shortest Job First Method_Function

```
function [SJFTime,SJFSequence]=SJF_Time(Requests,InitTravelMat)

for i=1:max(size(Requests(:,1)))
    RequestsSJF(i,1)=InitTravelMat(Requests(i,2),Requests(i,1));
end;
SJF=horzcat(Requests,RequestsSJF);
SJF1=sortrows(SJF,3);

SJF1InRow=[];
for i=1:max(size(SJF1(:,1)))
    for j=2:-1:1
        SJF1InRow=horzcat(SJF1InRow,SJF1(i,j));
    end;
end;
SJF1InRow1=[1,SJF1InRow,1];
SJFTime=0;
for i=1:max(size(SJF1InRow1))-1
   SJFTime= SJFTime+InitTravelMat(SJF1InRow1(1,i),SJF1InRow1(1,i+1));
```

```
end;
SJFSequence=SJF1InRow1;
```

Nearest Neigbor First Method_Function

```
function
[GreedyTime,GreedySequence]=Greedy_Time(Requests,InitTravelMat,RequestsMat)

RequestsMatGreedy=RequestsMat;

for i=1:max(size(RequestsMatGreedy))
       RequestsMatGreedy(i,i)=100;
  end;
   Current=[1];
   GreedyCost=[];
   I=1;
   RequestsMatGreedy(:,1)=100;
for i=2:max(size(RequestsMatGreedy))
    [C,I]=min(RequestsMatGreedy(I,:));
    Current=horzcat(Current,I);
    GreedyCost=horzcat(GreedyCost,C);
    RequestsMatGreedy(:,I)=100;
end;
Requestsok=vertcat([1,1],Requests);
CurrentInRow=[];
for i=2:max(size(Current))
    D=Requestsok(Current(1,i),:);
    CurrentInRow=horzcat(CurrentInRow,D(1,2),D(1,1));
end;
CurrentInRow=[1,CurrentInRow,1];
GreedyTime=0;
for i=1:max(size(CurrentInRow))-1
   GreedyTime= GreedyTime+InitTravelMat(CurrentInRow(i),CurrentInRow(i+1));
end;
GreedySequence=CurrentInRow;
```

CSSP Optimal Sequence_Function

```
function[OptimalTime,OptimalSequence,Distance,TimeElapsed]=Optimal_Time(Reque
stsMat,Requestsok,Requests,InitTravelMat)
tic

%CONCORDE only works with Integer Numbers so decimals are changed to
%integers by multiplying the RequestsMat *M1
```

```
%M1=10000 is a large number to make travel times, integers

RequestsMat=1000*RequestsMat;

%-10 is a low number to make sure that original node jumps to its dummy
%node, for more information refer to TSP in Wikipedia, it represents (-
infinity)
%cost matrix for Concorde must be positive integers

for i=1:max(size(RequestsMat))
    for j=1:max(size(RequestsMat))
        if i==j
            RequestsMat(i,j)=-9999;
        end;
    end;
end;

% a number represents (+infinity) to make ATSP to TSP, M2=100000, M1<M2

Distance= [9999*ones(max(size(RequestsMat))) RequestsMat';RequestsMat
9999*ones(max(size(RequestsMat)))];
Distance=Distance+9999;

for i=1:max(size(Distance))
    Distance(i,i)=0;
end;

TSPfile=fopen('amir3.tsp','w');
a='%6.0f';
for i=1:max(size(Distance))-1
    a=[a '%6.0f '];
end;
a=[a ' \n '];

fprintf(TSPfile,'NAME: %10.0f REQUESTS and 1
CRANE\n',max(size(Requests(:,1))));
fprintf(TSPfile,'TYPE: TSP\n');
fprintf(TSPfile,'COMMENT: %10.0f REQUESTS + 1
CRANE\n',max(size(Requests(:,1))));
fprintf(TSPfile,'DIMENSION: %10.0f\n',max(size(Distance)));
fprintf(TSPfile,'EDGE_WEIGHT_TYPE: EXPLICIT\nEDGE_WEIGHT_FORMAT:
FULL_MATRIX\nEDGE_WEIGHT_SECTION\n');
fprintf(TSPfile,a,Distance);
fprintf(TSPfile,'EOF');

fclose(TSPfile);

system('concorde amir3.tsp');

TSPSol=fopen('amir3.sol','r');
```

```
OptimalRouteRaw=fscanf(TSPSol,'%d');
fclose(TSPSol);

OptimalRouteRaw2=OptimalRouteRaw(2:max(size(OptimalRouteRaw)));
%*****************************
%calculating time using Distance Matrix: this is the same time that
%Concorde gives you

OptVarRoute=[OptimalRouteRaw2+1 ;1];
VarTime=0;
for i=1:max(size(OptVarRoute))-1
    VarTime=VarTime+Distance(OptVarRoute(i),OptVarRoute(i+1));
end;

%**************************
if (OptimalRouteRaw2(2)~=max(size(RequestsMat)))
    OptimalRouteRaw2(1)=[];
    OptimalRoute=flipud(OptimalRouteRaw2);
    OptimalRouteRaw2=[0 ; OptimalRoute];
end;

OptimalRoute01= OptimalRouteRaw2 + 1;
OptimalRoute06=[ OptimalRoute01; 1];

OptimalRoute1=[];
for i=1:max(size(OptimalRoute06))
    if mod(i,2)~=0
    OptimalRoute1=horzcat(OptimalRoute1,OptimalRoute06(i));
    end;
end;
OptimalRoute34=[];
for i=2:(max(size(OptimalRoute1))-1)
    for j=2:-1:1

OptimalRoute34=horzcat(OptimalRoute34,Requestsok(OptimalRoute1(i),j));
    end;
end;

OptimalRoute341=[1 OptimalRoute34 1];

TotalTravelTime=0;
for i=1:max(size(OptimalRoute341))-1

TotalTravelTime=TotalTravelTime+InitTravelMat(OptimalRoute341(i),OptimalRoute
341(i+1));
end;
OptimalTime=TotalTravelTime;
OptimalSequence=OptimalRoute341;
TimeElapsed=toc;
```

LIST OF REFERENCES

AlBahnassi, H., & Hammad, A. (2012). Near real-time motion planning and simulation of cranes in construction : framework and system architecture. *computing in civil engineering* 54-63.

Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2011). *The traveling salesman problem: a computational study*: Princeton University Press.

Ascheuer, N., Fischetti, M., & Grötschel, M. (2001). Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. *Mathematical Programming, 90*(3), 475-506.

Baker, E. K. (1983). Technical Note—An Exact Algorithm for the Time-Constrained Traveling Salesman Problem. *Operations Research, 31*(5), 938-945.

Baldacci, R., Mingozzi, A., & Roberti, R. (2012). New state-space relaxations for solving the traveling salesman problem with time windows. *INFORMS Journal on Computing, 24*(3), 356-371.

Bansal, N., Blum, A., Chawla, S., & Meyerson, A. (2004). *Approximation algorithms for deadline-TSP and vehicle routing with time-windows.* Paper presented at the Proceedings of the thirty-sixth annual ACM symposium on Theory of computing.

Beavers, J. E., Moore, J. R., Rinehart, R., & Schriver, W. R. (2006). Crane-related fatalities in the construction industry. *Constrcution Engineering and Managament*, *132*(9), 901-910.

Brandt, C., & Robinson, D. (2012). constrcution tower crane operation. In A. Zavichi (Ed.).

Carbeau, B. (2012). [Tower Cranes Price Range in North America].

Carlton, W. B., & Barnes, J. W. (1996). Solving the traveling-salesman problem with time windows using tabu search. *IIE transactions, 28*(8), 617-629.

Chatterjee, S., Carrera, C., & Lynch, L. A. (1996). Genetic algorithms and traveling salesman problems. *European journal of operational research, 93*(3), 490-510.

Choi, I.-C., Kim, S.-I., & Kim, H.-S. (2003). A genetic algorithm with a mixed region search for the asymmetric traveling salesman problem. *Computers & Operations Research, 30*(5), 773-786.

Christofides, N., Mingozzi, A., & Toth, P. (1981). State-space relaxation procedures for the computation of bounds to routing problems. *Networks, 11*(2), 145-164.

Croes, G. (1958). A method for solving traveling-salesman problems. *Operations Research, 6*(6), 791-812.

Dash, S., Günlük, O., Lodi, A., & Tramontani, A. (2012). A time bucket formulation for the traveling salesman problem with time windows. *INFORMS Journal on Computing, 24*(1), 132-147.

Davis, L. (1985). *Applying adaptive algorithms to epistatic domains.* Paper presented at the IJCAI.

Druzdzel, M. J., & Flynn, R. R. (2003). Encyclopedia of Library and Information Science, chapter Decision Support Systems: Marcel Dekker.

Dumas, Y., Desrosiers, J., Gelinas, E., & Solomon, M. M. (1995). An optimal algorithm for the traveling salesman problem with time windows. *Operations research, 43*(2), 367-371.

Ecker, K., Gupta, J. N., & Schmidt, G. (1997). A framework for decision support systems for scheduling problems. *European Journal of Operational Research, 101*(3), 452-462.

Engineers, U. S. A. C. o. (2011). construction equipment ownership and operating expense schedule (Vol. 5): Department of Army.

Everett, J. G., & Slocum, A. H. (1993a). Cranium - Device for Improving Crane Productivity and Safety. *Journal of Construction Engineering and Management-Asce, 119*(1), 23-39. doi: Doi 10.1061/(Asce)0733-9364(1993)119:1(23)

Everett, J. G., & Slocum, A. H. (1993b). CRANIUM : Device for improving crane productivity and safety. *Journal of Construction Engineering and Management*, 23-39.

Flood, M. (1956). The Traveling-Salesman Problem. *Operations Research, 4*(1), 61-75.

Fogel, D. B. (1988). An evolutionary approach to the traveling salesman problem. *Biological Cybernetics, 60*(2), 139-144.

Gendreau, M., Hertz, A., Laporte, G., & Stan, M. (1998). A generalized insertion heuristic for the traveling salesman problem with time windows. *Operations Research, 46*(3), 330-335.

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

Goldberg, D. E., Lingle, R. (1985). *Alleles, loci, and the traveling salesman problem.* Paper presented at the International Conference on Genetic Algorithms and Their Applications, Carnegie Mellon University, USA.

Greco, F. (2008). *Traveling Salesman Problem*: InTech.

Gutin, G., & Punnen, A. P. (2004). *The travelling salesman problem and its variations* (Vol. 12). Newyork: Springer.

Han, S., Lee, S., Fard, M. G., & Peña-Mora, F. (2007). *Modeling and representation of non-value adding activities due to errors and changes in design and construction projects.* Paper presented at the Proceedings of the 39th conference on Winter simulation.

Hasan, S., Bouferguene, A., Al-Hussein, M., Gillis, P., & Telyas, A. (2013). Productivity and CO2 emission analysis for tower crane utilization on high-rise building projects. *Automation in Construction, 31*, 255-264.

Holland, J. H. (1975). Adaption in natural and artificial systems.

Huang, C., Wong, C. K., & Tam, C. M. (2011). Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming. *Automation in Construction, 20*(5), 571-580. doi: DOI 10.1016/j.autcon.2010.11.023

Jonker, R., & Volgenant, T. (1983). Transforming asymmetric into symmetric traveling salesman problems. *Operations Research Letters, 2*(4), 161-163.

Josephson, P. E., & Hammarlund, Y. (1999). The causes and costs of defects in construction: A study of seven building projects. *Journal of Automation in Construction, 8*(6), 681-687.

Kang, S., & Miranda, E. (2006). Planning and visualization for automated robotic crane erection processes in construction. *Automation in Construction, 15*(4), 398-414. doi: DOI 10.1016/j.autcon.2005.06.008

Kang, S., & Miranda, E. (2006). Planning and visualization for automated robotic crane erection processes in construction *Automation in Constrcution*, 398-414.

Kay, J. N. (2001). Construction Work Study. Retrieved July 7th, 2013, from http://civcal.media.hku.hk/workstudy/resource/crane/default.htm

Koskela, L. (1992). *Application of the new production philosophy to construction*: Stanford university (Technical Report No. 72, Center for Integrated Facility Engineering, Department of Civil Engineering). Stanford, CA.

Langevin, A., Desrochers, M., Desrosiers, J., Gélinas, S., & Soumis, F. (1993). A two-commodity flow formulation for the traveling salesman and the makespan problems with time windows. *Networks, 23*(7), 631-640.

Laporte, G. (1992). The Traveling Salesman Problem - an Overview of Exact and Approximate Algorithms. *European Journal of Operational Research, 59*(2), 231-247. doi: Doi 10.1016/0377-2217(92)90138-Y

Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review, 13*(2), 129-170.

Lawler, E. L., Lenstra, J. K., Kan, A. R., & Shmoys, D. B. (1985). *The traveling salesman problem: a guided tour of combinatorial optimization* (Vol. 3): Wiley Chichester.

Lee, G., Cho, J., Ham, S., Lee, T., Lee, G., Yun, S.-H., & Yang, H.-J. (2012). A BIM-and sensor-based tower crane navigation system for blind lifts. *Automation in Construction, 26*, 1-10.

Lei, Z. (2011). A robotic approach to the analysis of obstacle avoidance in crane lift path planning *Master Thesis*. Edmonton, Alberta, Canada: University of Alberta.

Leung, A. W. T., & Tam, C. M. (1999). Prediction of Hoisting time for tower cranes fro public housing construction in Hong Kong. *Construction Management and Economics*, 305-314.

Lucko, G., & Vorster, M. C. (2003). *Predicting the residual value of heavy construction equipment.* Paper presented at the Proceedings of the 4th Joint International Symposium on Information Technology in Civil Engineering.

Melanie, M. (1999). An introduction to genetic algorithms. *Cambridge, Massachusetts London, England, Fifth printing*.

Murata, T., Ishibuchi, H., & Tanaka, H. (1996). Genetic algorithms for flowshop scheduling problems. *Computers & Industrial Engineering, 30*(4), 1061-1071.

Newnan, D. G., Eschenbach, T. G., & Lavelle, J. P. (2004). *Engineering Economic Analysis: Study Guide*: Oxford University Press.

Noraini, M. R., & Geraghty, J. (2011). Genetic Algorithm Performance with Different Selection Strategies in Solving TSP.

Ohlmann, J. W., & Thomas, B. W. (2007). A compressed-annealing heuristic for the traveling salesman problem with time windows. *INFORMS Journal on Computing, 19*(1), 80-90.

Olearczyk, J. (2010). Crane lifting operation planning and lifted object spatial trajectory analysis *Doctoral Dissertation*. Edmonton, Alberta, Canada: University of Alberta.

Oliver, I., Smith, D., & Holland, J. R. (1987). *A study of permutation crossover operators on the traveling salesman problem.* Paper presented at the Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application.

Park, M., Yang, Y., & Lee, H.-S. (2012). Floor-Level Construction Material Layout Planning Model Considering Actual Travel Path. *Journal of construction engineering and management, 138*(7), 905-915.

Relations, D. o. I. (2013). General prevailing wage determinations. Retrieved July 15 https://www.dir.ca.gov/OPRL/PWD/Southern.html

Rosenfeld, Y. (1995). Automation of existing cranes: from concept to prototype. *Automation in construction, 4*(2), 125-138.

Rosenfeld, Y., & Shapira, A. (1998). Automation of existing tower cranes: economic and technological feasibility. *Automation in construction, 7*(4), 285-298.

Serpell, A., Venturi, A., & Contreras, J. (1995). Characterization of waste in building construction projects. *Lean Construction*, 68-77.

Shapira, A., & Glascock, J. D. (1996). Culture of Using Mobile Cranes for Building Constrcution. *Journal of Constrcution Engineering and Management* 298-307.

Shapira, A., Lucko, G., & Schexnayder, C. J. (2007). Cranes for building construction projects. *Journal of Construction Engineering and Management-Asce, 133*(9), 690-700. doi: Doi 10.1061/(Asce)0733-9364(2007)133:9(690)

Shapira, A., Rosenfeld, Y., & Mizrah, I. (2008). Vision system for tower cranes. *Journal of Construction Engineering and Management-Asce, 134*(5), 320-332. doi: Doi 10.1061/(Asce)0733-9364(2008)134:5(320)

Shapira, a., Rosenfeld, y., & Mizrahi, i. (2008). vision system for tower cranes. *journal of construction engineering and management, 134*(5), 320-332.

Sivakumar, P., Varghese, K., & Ramesh Babu, N. (2003). Automated path planning of cooperative crane lifts using heurisitc search. *Journal of computing in civil engineering 17*(3), 197-207.

Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot Modeling and Control*. New York: John Wiley & Sons.

Starkweather, T., McDaniel, S., Mathias, K. E., Whitley, L. D., & Whitley, C. (1991). *A comparison of genetic sequencing operators.* Paper presented at the ICGA.

Sullivan, W. G., Wicks, E. M., & Luxhoj, J. T. (2009). *Engineering economy*: Pearson Prentice Hall.

System, F. R. (2013). Interest Rates. Statistical Releases and Historical Data Retrieved July 17, 2013 http://www.federalreserve.gov/releases/h15/data.htm#fn11

Tam, C., Leung, A. W., & Liu, D. (2002). Nonlinear models for predicting hoisting times of tower cranes. *Journal of computing in civil engineering, 16*(1), 76-81.

Tam, C. M., & Tong, T. K. L. (2003). GA-ANN model for optimizing the locations of tower crane and supply points for high-rise public housing construction. *Construction Management and Economics, 21*(3), 257-266. doi: 10.1080/0144619032000049665

Tam, C. M., Tong, T. K. L., & Chan, W. K. W. (2001). Genetic algorithm for optimizing supply locations around tower crane. *Journal of Construction Engineering and Management-Asce, 127*(4), 315-321. doi: Doi 10.1061/(Asce)0733-9364(2001)127:4(315)

Tavakkoli-Moghaddam, R., Makui, A., Salahi, S., Bazzazi, M., & Taheri, F. (2009). An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering, 56*(1), 241-248.

Transportation, C. D. o. (2013). Labor Surcharge and Equipment Rental Rates (cost of equipment ownership). from Department of Transportation http://www.dot.ca.gov/hq/construc/eqrr/Book_2013.pdf

Üçoluk, G. (1997). Genetic algorithm solution of the tsp avoiding special crossover and mutation.

Wakisaka, T., Furuya, N., Inoue, Y., & Shiokawa, T. (2000). Automated construction system for high-rise reinforced concrete buildings. *Automation in Construction, 9*(3), 229-250.

Wolsey, L. A. (1998). *Integer Programming*. New York: John Wiley & Sons.

Yu, Y., Chen, Y., & Li, T. (2011). *A new design of genetic algorithm for solving TSP*. Paper presented at the Computational Sciences and Optimization (CSO), 2011 Fourth International Joint Conference on.

Zavichi, A., & Behzadan, A. H. (2011). *A real time decision support system for enhanced crane operations in construction and manufacturing.* Paper presented at the Proceedings of ASCE International Workshop on Computing in Civil Engineering, ASCE, Miami, FL.

Zavichi, A., Madani, K., Xanthopoulos, P., & Oloufa, A. (2013). *TSP-based Model for on-site Material Handling Operations with Tower Cranes.* Paper presented at the 30th International Symposium on Automation and Robotics in Construction and Mining, Montreal, Canada.

Zhang, P., Harris, F. C., & Olomolaiye, P. O. (1996). A computer-based model for optimizing the location of a single tower crane. *Building Research and Information, 24*(2), 113-123. doi: Doi 10.1080/09613219608727511

Zhang, P., Harris, F. C., Olomolaiye, P. O., & Holt, G. D. (1999). Location optimization for a group of tower cranes. *Journal of Construction Engineering and Management-Asce, 125*(2), 115-122. doi: Doi 10.1061/(Asce)0733-9364(1999)125:2(115)