HEURISTIC APPROACHES FOR CRANE SCHEDULING IN SHIP BUILDING

By

Charlie Hsiao Kuang Wen

A Thesis
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirement
for the Degree of Master of Science
in Industrial Engineering in the
Department of Industrial and Systems Engineering

Mississippi State, Mississippi

August 2008

HEURISTIC APPROACHES FOR CRANE SCHEDULING IN SHIPBUILDING

By

Charlie Hsiao Kuang Wen

Approved:

Sandra Duni Eksioglu Assistant Professor of Industrial and Systems Engineering (Director of Thesis)	Allen G. Greenwood Professor of Industrial and Systems Engineering (Committee Member)
Burak Eksioglu Assistant Professor of Industrial and Systems Engineering (Committee Member)	Stanley F. Bullington Professor of Industrial and Systems Engineering Director of Graduate Studies in Industrial and Systems Engineering
Sarah A. Rajala Dean of Bagley College of Engineering	

Name: Charlie Hsiao Kuang Wen

Date of Degree: August 9th, 2008

Institution: Mississippi State University

Major Field: Industrial Engineering

Major Professor: Dr. Sandra D. Eksioglu

Title of Study: HEURISTIC APPROACHES FOR CRANE SCHEDULING IN SHIPBUILDING

Pages in Study: 75

Candidate for Degree of Master of Science

This study provides heuristic approaches, including an ant colony optimization (ACO) inspired heuristic, to solve a crane scheduling problem that exists in most shipyards, where cranes are a primary means of processing and handling materials. Cranes move on a network of tracks, thus, blocking of crane movements is an issue. The crane scheduling problem consists of two major sub-problems: scheduling problem that determines the best overall order in which jobs are to be performed; the assignment problem that assigns cranes to jobs. The proposed heuristic consists of an Earliest Due Date sorting procedure in combination with an ACO assignment procedure that aims to satisfy the objectives of minimizing makespan while maximizing crane utilization. Test data sets of various sizes are generated and the results of the proposed approach are compared to other developed heuristics. The proposed approach outperforms others in both objective measures and obtains solutions in a timely manner.

DEDICATION

To Dr. Daniel Hung Yueh Wen and Betty Heng Chiang

ACKNOWLEDGEMENTS

Special thanks to Dr. Sandra Eksioglu for her guidance and advice during the course of thesis writing and research. Also, special thanks to Dr. Allen Greenwood for his guidance in this research project and also for providing financial support. Additionally, special thanks to Dr. Burak Eksioglu for his technical advice and introduction to the field of heuristic optimization.

Expressed gratitude is also due to the faculty members of the Department of Industrial & Systems Engineering who provided additional mentoring and encouragement during the course of the graduate program. In particular, those faculty members are Dr. Mingzhou Jin, Dr. Stanley Bullington, Dr. John Usher, and Dr. Lesley Strawderman.

Lastly, special thanks to Dr. Stanley Bullington for reminding us all that “your thesis will not write itself!”

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Background

Scheduling problems are prevalent across industries in today's competitive business environment. With resources and time becoming more and more scarce, scheduling problems often present challenges for many personnel that are responsible for solving these problems. Furthermore, with dependencies within jobs and resources becoming more complex, the problem becomes even more difficult to solve. Scheduling problems are faced daily and play an important role particularly in the service and manufacturing industries. In these industries, time is crucial to the operational efficiency and effectiveness and can be carefully managed if scheduling problems are addressed with good methods. For service industries, the waiting time or service time for each customer is a key factor of operating a successful service. For manufacturing industries, the time to complete a certain set of jobs or time spent by jobs as work-in-process are important elements for success. Therefore, scheduling methods that output quality schedules that achieve desired objectives are always in demand. Since there are various objectives in scheduling problems, solution approaches are tailored to fit the desired objective. In addition, scheduling problems are usually large in size with numerous variables and constraints. Hence, scheduling problems are often approached with problem specific heuristics that are unique.

Within the realm of scheduling problems, there are several problem settings. The single-machine scheduling problem considers several jobs being processed by a single machine. The parallel machines scheduling problem considers several machines that can process all jobs in the problem with each job being processed once by a single machine. The flowshop scheduling problem considers several jobs being processed by a series of machines, with all jobs having the same processing sequence. The job shop scheduling problem considers several jobs that have to be processed by various resources before completing with each job having its own unique routing sequence. Open shop scheduling problems considers a job shop problem with the processing sequence of each job being unimportant. Among the common measures of performance in scheduling problems, the maximum completion times of all jobs, or makespan, is most commonly used. Others include lateness, tardiness, and flowtime. Lateness differs from tardiness in that lateness can have negative values, which indicate a job was completed before its due date. In other words, a negative lateness value is synonymous with earliness. On the other hand, tardiness can only take positive values, which indicate a job was completed after its original due date.

1.2 Problem Description

In this research, a crane scheduling problem in a ship manufacturing environment is investigated. A manufacturer of military and commercial ships and vessels based in coastal Mississippi has expressed concerns about the utilization and scheduling of cranes at the final assembly area. Cranes are the major source of transporting materials within the erection area; therefore, the performance of cranes is crucial to the operational efficiency and effectiveness of the entire shipyard.

An example of an erection area is shown in Figure 1 on the follow page. The erection area consists of several East-West and North-South tracks which provide the only travel medium for cranes. The nodes of the network represent areas where cranes stop to perform lifts or to switch tracks. The black nodes represent turn nodes where cranes are present only to make a turn or to pass. Arcs of the network represent tracks that cranes travel on to reach lifts. The rectangular and square shapes on the network are areas where lifts occur. Interference among cranes becomes an issue because cranes on the network can block others as they perform lifts and move about on the network.

The manufacturer believes that they do not have adequate cranes to meet lift requirements and that current scheduling strategies compromise the utilization of the cranes. Furthermore, they are concerned by the amount of time required to complete a set of lifts. As a result, a scheduling optimization heuristic algorithm needs to be developed to reconcile the issues faced by the shipbuilder. Unlike the traditional scheduling problem faced in other manufacturing environments, the crane scheduling problem involves the interference among the machines themselves. Furthermore, the problem addressed in this study involves more than determining the order of the jobs to be performed; it also involves determining the optimal assignment of machines (cranes) to the jobs so that they satisfy the shipbuilder's objectives.

Since the problem at hand involves two major parts, the proposed methodology also consists of two major parts. The first is to determine the sequence of the lifts to be performed on an overall perspective, but not with respect to each crane. This sequence will prioritize the lifts based on certain attributes, such as priority indices, due dates, or processing times. This is essentially the scheduling portion of the problem. The second part of the proposed methodology finds assignments of cranes to jobs that achieve the objectives. This is the assignment portion of the problem.

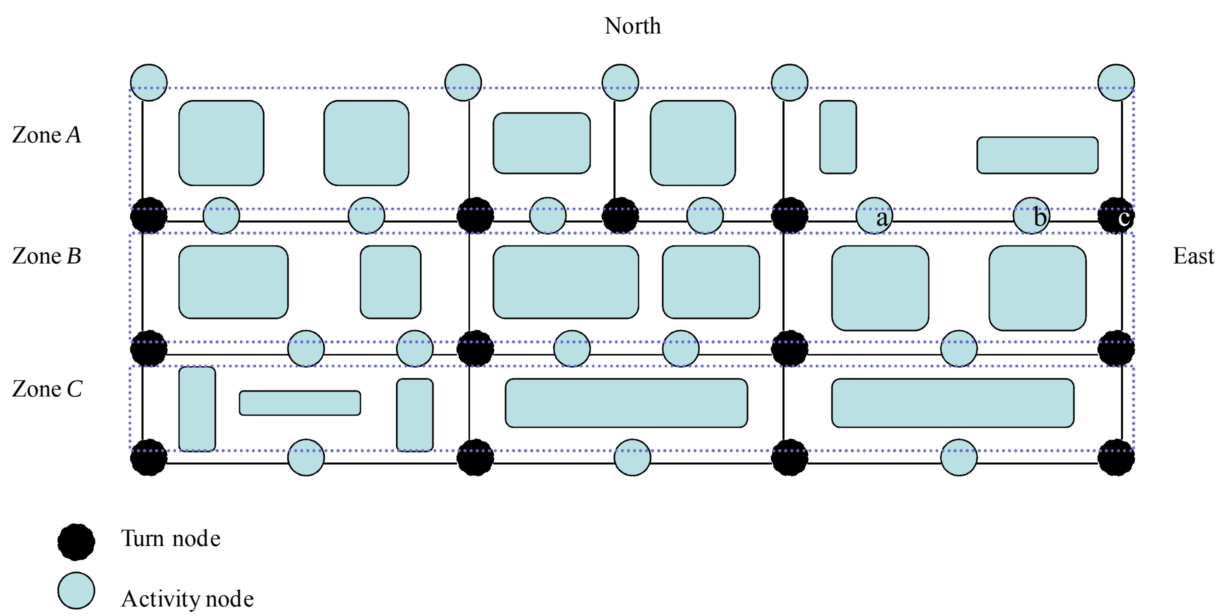

Figure 1

Sample Crane Track Network

The scheduling portion of the problem cannot be accurately categorized into traditional types of scheduling problems. The problem type that most closely fits is the parallel machines problem. However, for the problem at hand, not all cranes can perform all lifts. Still, for a particular lift, there are multiple cranes that can be used to complete the lift. Also, lifts are completed once they have been "processed" once by cranes. So, the scheduling portion of the problem can be labeled as a variation or extension of the parallel machines scheduling problem.

1.3 Objectives

The major objectives of this research are to minimize the maximum completion time (makespan) of all lifts and to maximize the utilization of the cranes. Upon initial glance, these may seem to be two separate objectives. Further investigation into the problem, however, shows that by minimizing makespan, crane utilization is in turn maximized. Crane utilization is calculated by dividing the total amount of time units a crane is actually performing lifts by the makespan. Since makespan is the denominator of the utilization equation, a decrease in makespan will automatically increase crane utilization. Furthermore, since the time it takes to perform a particular lift is the same for each crane, that is, lift durations are constant; that means $\sum_{i=1}^{N} LiftDuration(i)$ is a constant.

The equation for calculating crane utilization for a particular crane is as follows:

$$\sum_{j=1}^{M} Crane\,Utilization(j) = \frac{\sum_{i \ni N_j} LiftDuration(i)}{CT_{\max}} \qquad (7)$$

Where $CT_{\max}$ is the makespan and $LiftDuration(i)$ is the lift duration of lift *i* that is performed by crane *j*. N_j is the set of lifts that is performed by crane *j*. One can intuitively see that if makespan is decreased and the sum of the lift durations performed by the crane remains constant, then crane utilization is increased. Therefore, by minimizing the makespan of a set of lifts, the second objective of maximizing overall crane utilization is also achieved. Moreover, by minimizing makespan, time spent by the cranes that does not correspond to performing lifts (idle time, traverse time, time blocked) is also minimized. Therefore, the focus of the research becomes minimizing the makespan of a set of lifts.

1.4 Detailed Problem Statement

The problem at hand consists of two major components: a scheduling problem that involves determining the order or sequence of lifts to be scheduled. The second component of the problem is a variation of the generalized assignment problem that focuses on assigning cranes to lifts with the objective of minimizing the total cost incurred for each assignment. The verbal and mathematical formulations of both problems are presented in this section.

In words, the scheduling problem can be formulated as the following:

Minimize:

(Maximum completion time of all lifts)

Subject to:

Each lift get scheduled

The assumption is made in the formulation that all lifts are available to be performed at time = 0.

The mathematical formulation of the scheduling problem is as follows:

Min Z

Subject to:

$$\sum_{j}^{N} X_{ij} = 1 \qquad \forall\ i \text{ in lifts set} \tag{1}$$

$$\sum_{i}^{N} X_{ij} = 1 \qquad \forall j \tag{2}$$

$$\sum_{j}^{N} C_{ij} X_{ij} \leq Z \qquad \forall\ i \text{ in lifts set} \tag{3}$$

where X_{ij} is a binary decision variable that takes the value 1 if lift i is scheduled in position j and takes the value 0 otherwise. C_{ij} is the completion time of lift i when scheduled at the j-th position. N is the number of lifts in the lifts set. Z is a variable that represents the maximum completion time or project makespan.

In words, the generalized assignment problem can be formulated as the following:

Minimize:

(Total time to perform all lifts)

Subject to:

Each lift requires one crane option be assigned to it

Each crane may perform up to the number of lifts in a given set

The mathematical formulation of the generalized assignment problem is as follows:

Min Z

Subject to: $$\sum_{j=1}^{M} X_{ij} = 1 \qquad \forall\, i \text{ in lifts set} \qquad (4)$$

$$\sum_{i}^{N} X_{ij} \geq 0 \qquad \forall\, j \text{ in cranes set} \qquad (5)$$

$$\sum_{i}^{N} C_{ij} X_{ij} \leq Z \qquad \forall\, j \text{ in cranes set} \qquad (6)$$

where X_{ij} is a binary decision variable and C_{ij} is the completion time of lift *i* when crane *j* is assigned to it. *N* and *M* stand for the number of lifts and cranes in their respective sets. *Z* again is a dummy variable that represents makespan and is only introduced for formulation purposes.

In the formulations presented above, it is assumed that C_{ij} is known and deterministic. For the scheduling problem, C_{ij} is a value in a N x N matrix which has to be calculated before the linear programming model can be executed. By definition, the values in the matrix represent the minimum completion time of scheduling lift *i* in position *j*. Therefore, all possible lift scheduling combinations before position *j* have to be evaluated. Even for problems with small number of lifts, calculations for the C_{ij} values are cumbersome. As the number of lifts increase, the problem size grows exponentially and calculating C_{ij} values become intractable.

The generalized assignment problem is an easy problem to solve. However, the C_{ij} values in the assignment problem encounter the same problem as the scheduling problem discussed previously. For the assignment problem, C_{ij} is a value in an N x M matrix where values depend on the assignments of previous lifts. For the reasons mentioned above, a heuristic solution approach is most appropriate given the problem characteristics and settings.

CHAPTER II

LITERATURE REVIEW

A review of the literature was performed to gain an understanding of the current state-of-the-art and also to see if any insights or similar problems settings can be found for the crane scheduling problem presented in this study. Currently in literature, there are no approaches that exactly address the problem at hand. Most of the literature on crane scheduling is focused on port terminal operations where containers are loaded and unloaded from ships to yard areas or transport trucks. While considering different in problem settings, the objective function of most of the works in literature on crane scheduling are concerned with minimizing the makespan or total completion time of jobs, which is similar to the problem at hand. The remainder of this chapter summarizes the literature based on problem characteristics.

2.1 Crane Scheduling Considering Blocking or Interference

Ng [1] considers the blocking of cranes on a bi-directional axis in which the crane can only travel in two directions on a single track. An integer programming model is introduced with the objective function of minimizing the total completion times of all cranes.

Lim et. al [4] define spatial constraints in which the cranes cannot cross over each other when performing lifts, and also, a minimum distance between cranes is established to ensure cranes have enough flexibility to perform the lifts. A dynamic programming model is developed to maximize the total throughput of the system under the spatial and inter-crane requirements.

Ahuja et al. [5] developed a space-time network for a train dispatching problem where the horizontal and vertical axes represent time and space, respectively, and intersections in the graph denotes potential blocking or interference on a single track railway system. An integer programming model is developed for this problem with the objective of minimizing the delay time of all trains in the system.

Cai et al. [13] proposed a greedy search heuristic for the single-track railway scheduling problem considering realistic constraints that can be incorporated during run time.

2.2 Sequencing of Container Transfers

Kozan and Preston [2] focused on minimizing the sum of the setup and traveling times for all containers by using a mixed integer programming model and branch-and-bound techniques. The problem reduces to a job scheduling problem with n jobs and m machines.

Gambardella et al [3] developed a network flow model where nodes represented cranes or ships and are connected by arcs whose capacities depend on the allocated resources. The flows of the model are the containers to be moved and the objective is to find the best set of capacities of the arcs. The problem is solved with a mixed-integer program along with a branch-and-bound search.

Kim and Park [6] studied the quay crane scheduling problem in port container terminals to determine the sequence of discharging and loading operations that minimizes the completion time of the ship operation. Branch and bound and GRASP (greedy randomized adaptive search procedure) methods were proposed to find near optimal objective function values.

Bish [7] considers the problem of locating yard cranes in a storage area and determining the sequence of locations served by each yard crane to minimize the makespan in a multiple-crane-constrained vehicle location and scheduling problem (MVSL).

2.3 Scheduling of Quay Cranes

Quay cranes are relatively immobile cranes that are located on the quayside and are used for loading and unloading of containers to and from vessels. Peterkofsky and Daganzo [8] studied the static quay crane scheduling problem with the objective of minimizing aggregate vessel delay cost. They proposed various heuristic approaches and exact algorithms to solve the quay crane scheduling problem. Daganzo [9] also studied the impact of different quay crane scheduling rules on the throughput of the terminal and ship delays.

2.4 Scheduling of Container Handling Equipment

Since cranes are only one type of material handling equipment that is used in terminals and yards, the current literature on the scheduling of other handling equipment is also considered. Kim [11] focused on the problem of routing a straddle carrier, transport equipment that handles containers, to transport export containers to a loading vessel. A mixed integer program was proposed to determine the number of containers to be picked up in each yard slot.

2.5 Automated Guided Vehicle Systems (AGVS)

Literature related to automated guided vehicle systems, AGVS, is considered since there are similarities to the way cranes traverse in this study. AGVS are unmanned vehicles that travel uni-directionally or bi-directionally on a path network to transport material from one location to another on a factory floor. Gaskins and Tanchoco [12] studied the flow path design for AGVS to determine the optimal flow path which minimizes the total travel of loaded vehicles. The problem is formulated as a binary integer program and solved using a direct search algorithm. The authors, however, did not consider vehicle blocking or network congestion.

2.6 Crane Scheduling in Manufacturing Environments

Literature involving scheduling cranes in manufacturing environments was also reviewed since the problem in this study involves utilizing cranes in a manufacturing environment.

Matsuo et al. [14] studied a crane scheduling problem in a computer integrated manufacturing environment where a single crane acts as the only source of material handling equipment to transfer work-in-process between processing stations. Crane activities of a processed job pickup and an unprocessed job drop-off in the same tank are combined into one trip. A cyclic scheduling approach along with a network flows based a heuristic algorithm is proposed to obtain good solutions.

Ge and Yih [15] studied the problem of crane scheduling with time windows in circuit board production lines, where each job has to be processed for no less than its minimum required time and no more than its maximum processing time. The problem is formulated in a flow-shop setting and an incomplete branch-and-bound heuristic algorithm is presented for real-time scheduling and its performance is checked by computer simulation. Both works mentioned above considers only a single crane traveling on a single track between workstations; therefore blocking or interference is not considered.

As can be seen in a review of the current literature, the problem at hand has unique characteristics. Those major characteristic are that cranes can travel in all directions on several numbers of tracks and can block or interfere with other cranes that are on the same path.

2.7 Ant Colony Optimization

A review of the literature was also performed on the area of ant colony optimization and optimization heuristics that are inspired by the foraging behavior of ants. Ant colony optimization (ACO) is selected due to its ability to yield quality solutions to various combinatorial optimization problems. Within the routing category, the traveling salesman problem and vehicle routing problems have been solved using ACO methods. In the assignment category, quadratic assignment, graph coloring, and generalized assignment problems have been studied. Additionally, scheduling problems have also been studied, including job shop, open shop, flow shop, total tardiness, and total weighted tardiness [24]. The following sections contain a review of the literature regarding ant colony optimization within the assignment and scheduling domains.

2.7.1 Ant Colony Optimization for Scheduling Problems

There is a plethora of studies on the application of ant colony optimization to flowshop scheduling problems.

Shyu et al. [19] represented the flowshop problem as a weighted directed graph $G(V,E)$, with each edge (i,j) having an associated weight $w(i,j)$. As a result, this representation reduces the problem's objective function to that of the TSP problem. Furthermore, T'kindt et al. [18] translates the flowshop scheduling problem to that of a lexicographical scheduling problem while Ying and Liao [17] associate it with a disjunctive graph with a set of nodes, directed arcs, and set of disjunctive undirected arcs. The set of nodes stands for all of the processing operations acted upon the jobs. The set of directed arcs corresponds to the precedence relationships between the processing operations of a single job while the set of undirected arcs represents the machine constraint of operations belonging to different jobs. Variations of the flowshop scheduling problem were studied among the works mentioned above. Shyu et al. [19] considered a two-machine flowshop scheduling problem with the objective of minimizing total completion time while taking into account set-up times. T'kindt et al. [18] focused on a two-machine bicriteria flowshop problem with the objectives of minimizing total completion time and makespan. Rajendran and Ziegler [16] proposed two algorithms for minimizing total flowtime or completion time for the benchmark problems developed by Taillard [21] whereas Ying and Liao [17] concentrated on the general $n/m/P/C_{max}$ problem with n jobs to be processed on m machines while finding a permutation of jobs that will minimize makespan (C_{max}). Prior to Rajendran and Ziegler's [20] algorithms for minimizing total flow time, the tandem also developed ACO algorithms for minimizing makespan [16].

Besides flowshop scheduling, ACO has been applied to other scheduling problems as well. Gutjahr and Rauner [22] used an ACO algorithm for a dynamic regional nurse-scheduling problem where a limited pool of of nurses have to meet the demands of regional hospitals while bounded by hard and soft constraints such as nurses' and hospitals' preferences, working schedules and patterns, and costs. The results of the ACO algorithm proposed by Gutjahr and Rauner [22] obtained more favorable results when compared with a simple greedy approach. Dowsland and Thompson [23] used an improved ant colony algorithm that was originally developed for the graph coloring problem (ANTCOL) to solve the examination scheduling problem. The problem entails allocating a set of exams to certain number of timeslots so that there are no scheduling conflicts for any student.

2.7.2 Ant Colony Optimization for Assignment Problems

In addition to scheduling problems, ACO approaches have also been applied to solve assignment problems. Lee et al. [27] developed an immunity-based ant colony optimization algorithm for the weapon-target assignment (WTA) problem where the objective is to minimize the expected damages of assets of own forces. The algorithm combines an immune system inspired heuristic which performs local search functions after solutions are obtained by the ACO heuristic. The problem is modeled where ants select weapons and assigns them to targets based on pheromone and heuristic values.

Demirel and Toksarı [25] developed an ant colony optimization algorithm with an additional simulated annealing local search process to solve a quadratic assignment problem. Similarly, a population based hybrid ant colony system (PHAS) heuristic with probabilistic pheromone trail modification rules is proposed by Ramkumar and Ponnambalam [26] to solve machine layout problems formulated as quadratic assignment problems.

As seen in the review, there are many studies that apply ant colony optimization to various combinatorial optimization problems. Based on these works, ant colony optimization appears to be an effective method for finding quality solutions when compared to other heuristic optimization approaches. Therefore, ant colony optimization is a candidate approach for the crane scheduling problem at hand.

CHAPTER III

MODELING OF CRANE OPERATIONS

3.1 Introduction

Before a solution approach to the problem can be developed, crane operations in the erection area needs to be modeled. Information regarding cranes and lifts were gathered from the manufacturer to gain a better understanding of everyday operations in the shipyard. The manufacturer has categorized lifts into four categories or priority classes. The priority indices, ranging from 1 to 4, indicate the importance of the individual lift towards the completion of the entire project. Also, historical data was collected from the manufacturer to provide the average lift percentages and lift durations for each priority class. The findings are presented in the following table:

Table 1

Average Lift Percentages and Durations

Lift Type	Priority	Occurrence	Avg. Duration (hrs)
Super Lift	1	10%	24
Erection	1	10%	16
Turn	2	10%	12
Land In Bay	3	10%	8
Move	3	10%	8
Stack	3	10%	12
Support	4	40%	1

In addition to the lifts information, the manufacturer provided data on the cranes. Currently, 13 cranes are used in the erection area to perform lifts with various tonnage capacities. A detailed list of available cranes along with their tonnage capacities is provided in Table 2.

Table 2

Cranes and Capacities

Crane Identification Number	Capacity
Gantry	Extra Large
1	Large
2	Large
3	Large
4	Large
5	Small
6	Medium
7	Medium
8	Medium
9	Medium
10	Large
11	Large
12	Large

3.1.1 Gantry Crane

The gantry is quite different from the other cranes. Besides its much larger capacity than the others, it actually travels on separate exclusive tracks. A gantry crane consists of two vertical beams that support the crane and provide the horizontal movement of the crane. Gantry cranes lift objects by utilizing a hoist that is situated on a horizontal beam on top of the two vertical supporting beams. Given the structure of gantry cranes, they are often used in shipbuilding where the crane straddles the ship and massive components and parts are moved and lifted to and from the ship. A picture of a gantry crane is provided in Figure 2.

Figure 2

Sample Gantry Crane

In this case, the gantry crane travels on tracks that are not used by the other cranes. Additionally, there are special operational guidelines regarding Gantry, such as other cranes cannot pass under it while it is performing a lift, but they can freely pass through if they pass Gantry on either end. A sample network containing Gantry is presented in Figure 3.

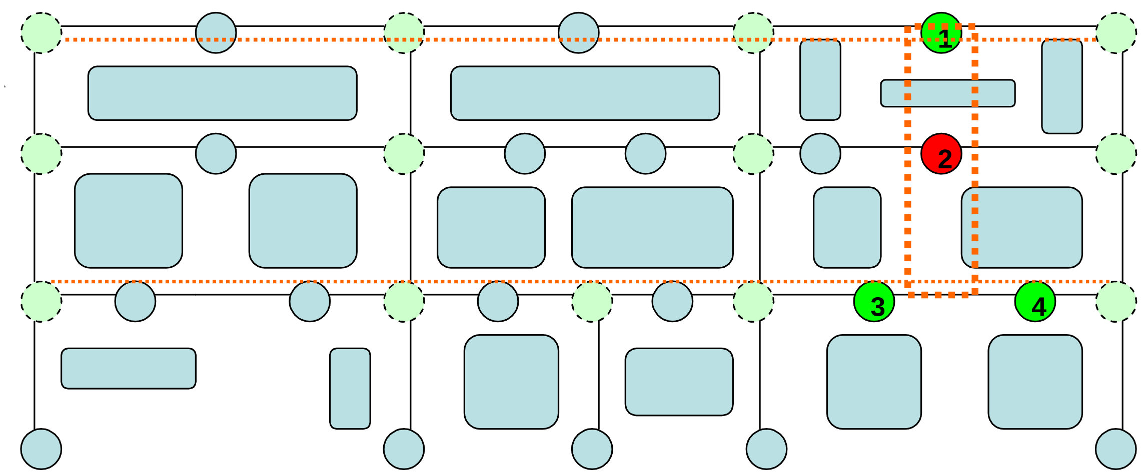

Figure 3

Sample Gantry Network

The horizontal dotted lines represent the Gantry tracks. Note that they are just inside the regular tracks and run parallel with the regular tracks in the two zones. To illustrate the operational guidelines of Gantry, consider the situation represented in Figure 3. While Gantry is performing a lift at nodes either 1 or 2, cranes can freely pass through node 1 and also between nodes 3 and 4. However, cranes may not pass through node 2 since that will be passing under Gantry. Additionally, lifts may not be performed at node 1 while Gantry is performing a lift in the highlighted area. In other words, node 1 is only available for cranes to pass through while Gantry is occupying the highlighted region on the network. However, for nodes 3 and 4, cranes can be present to either perform lifts or to pass. The highlighted region in the sample Gantry network figure is the approximate size of Gantry.

3.2 Model Assumptions

In order to accurately model crane operations, several assumptions are made and are described below.

- The time for each crane to travel from node to node is 15 minutes, regardless of the actual distance that separates two nodes. Distances between nodes on the network are fairly similar and travel time is not significant.
- One time unit (T) in the model is equivalent to 15 minutes.
- Cranes are not allowed to take paths that require more than two turn nodes on the network. This is an established rule that the manufacturer follows. Due to this restriction, paths that require more than two turn nodes will not be considered.
- Each crane will spend two hours at a turn node while switching from track to track.
- Maximum allowed traverse time on a particular path is 4 hours (T = 32). Any path taking longer than 4 hours will be excluded.

- Cranes will remain at its last lift location until it is assigned to perform another lift. In doing so, initial locations of cranes at the start of the heuristic will not include turn nodes.
- All lifts are ready to be performed at the beginning of run-time, or when T = 0.
- Lift locations are uniformly distributed throughout the network area. Every node has the same percentage of lifts that occurs there.
- There is no relationship between lift location and the priority index of a lift. Certain areas on the network will not be dominated by a particular priority index.
- Lifts information such as tonnage requirement, location, and due date are known inputs to the problem.
- If a crane is blocked, it will remain at its current location until the blocking has concluded. The crane will not travel on a particular path if it is currently blocked.
- Once a crane has started a lift, it cannot stop and perform another lift (i.e., no preemption allowed).
- The initial location of Gantry is located on the far West side of the erection area. Gantry will remain at its initial location until it has been assigned a lift.
- The time for Gantry to travel between regions is also fifteen minutes (T = 1).
- Two cranes of large capacities are required for all priority one lifts (Super and Erection lifts). Combination of tonnages is acceptable.
- If a crane meets or exceeds the tonnage requirement of a lift, that crane is available to perform the lift.

3.3 Crane Path Table

A crane path table documents all possible paths from each node to every other node on the network. Due to restrictions placed on crane paths, the number of paths between two nodes is limited. In most cases, there exists only one path between two selected nodes. The nodes are numbered so that they can be easily identified. As mentioned earlier, some nodes on the network are merely turning nodes where no actual lifts occur. Therefore, the nodes that are in the 'Origin' and 'Destination' columns contain only nodes where actual lifts occur. A sample of the crane path table is provided in Table 3.

Table 3

Crane Path Table

Origin	**Destination**	**Time**	**Path**							
1	1	**0**	-							
1	2	**21**	3	105	6	7	8	109	4	
1	3	**1**	3							
1	4	**20**	3	105	6	7	8	109		
1	6	**9**	3	105						
1	8	**11**	3	105	6	7				
1	10	**2**	3	5						
1	11	**18**	3	105	6	107				
1	12	**20**	3	105	6	7	8	109		
1	13	**11**	3	5	10	114				
1	15	**11**	3	5	10	114				
1	17	**13**	3	5	10	114	15	16		
1	19	**4**	3	5	10	14				
1	20	**13**	3	5	10	14	19	122		
1	21	**20**	3	105	6	107	11	16		
1	21	**20**	3	5	10	114	15	116		
1	21	**22**	3	5	10	14	19	122	23	124
1	23	**13**	3	5	10	14	19	122		

The values in the ‘Time’ column are calculated based on the assumptions listed earlier. These values represent the time units required for a crane to travel from the origin node to the destination node. Node numbers that are greater than 100 in the table denote a turn node. To obtain the actual node numbers of turn nodes, simply subtract 100 from the value listed in the table. For example, for the first path listed between nodes 1 and 21 in Table 3, node 5 is denoted as a turning node with a node number of 105.

A similar table is constructed for Gantry, but instead of time and paths, the table is concerned with the nodes included and nodes blocked when Gantry is at a specific region on the network. Since the width of Gantry spans across multiple zones, multiple nodes can be reached for lifts when Gantry is at one position on the network. Similarly, multiple nodes may also become unavailable for other cranes. A sample Gantry regions table is provided in Table 4.

Table 4

Sample Gantry Regions Table

Region	**Nodes Included**	**Nodes Blocked**
11	35,36,37,38,39	36,37,38
10	32,33	33
9	29,30,31	30,31
8	25,26,27	26
7	22,23,24	23,24
6	19,21	21
5	14,15,16,17,18	15,16,17
4	10,11,12	11
3	5,6,7,8,9	6,7,8
2	3,4	-
1	1,2	-
0	Origin	-

3.4 Crane and Node Status Matrices

Two matrices are constructed to keep track of the status of cranes and nodes: the Node-Time matrix and the Crane-Time matrix. The Node-Time matrix is a binary-valued matrix which has the occupancy information on all nodes for all time periods. A '1' denotes a node is occupied with a crane while a '0' denotes a node is vacant. Hence, the Node-Time matrix is a (number of nodes) X (number of time periods) matrix. The Node-Time matrix will be used to determine if a path is feasible for cranes to pass through, make a turn, or perform a lift. The Node-Time Table is initialized at the start of run-time to reflect the initial locations of the cranes and at any given period for any given node, its occupancy status is either '0' or '1'. A sample Node-Time matrix with four cranes and four time periods is shown in Figure 4. For example, node 2 is occupied in times periods 1 and 2.

$$
\begin{array}{ccc}
 & & \textbf{\textit{Time}} \\
 & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\
\textbf{\textit{Node}} & &
\end{array}
$$

Figure 4

Sample Node-Time Matrix

The Crane-Time matrix contains the status of each crane for all time periods. The matrix is a (number of cranes) X (number of time periods) matrix which will be used for all performance calculations regarding the cranes. At any given time period for any given crane, a status indicator will take on one of the following values: 1 = idle, 2 = travel, 3 = performing lift, 4 = blocked, 5 = waiting for second crane to arrive on priority one lifts. At the beginning of run-time, all cranes' status indicators are set to 1 (idle), which also means every crane is available to perform a lift. A sample Crane-Time matrix with four cranes and four time periods is presented in Figure 5. For example, crane 2 is performing a lift in time periods 1 and 2 (status = 3).

$$
\begin{array}{cc}
 & \textbf{\textit{Time}} \\
\textbf{\textit{Crane}} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}\begin{bmatrix} 1 & 2 & 2 & 3 \\ 3 & 3 & 1 & 1 \\ 2 & 2 & 5 & 3 \\ 1 & 1 & 1 & 2 \end{bmatrix}
\end{array}
$$

Figure 5

Sample Crane-Time Matrix

CHAPTER IV

ALGORITHM DEVELOPMENT

4.1 Introduction

Once crane operations have been modeled, heuristic scheduling algorithms are developed. Regardless of the approach, inputs to the algorithm and the desired outputs remain the same across all methods. Inputs are clearly the lifts and cranes and their respective attributes. The outputs are a proposed schedule of lifts, crane assignments for all lifts, and performance measures. An overview of the algorithm is provided in Figure 6.

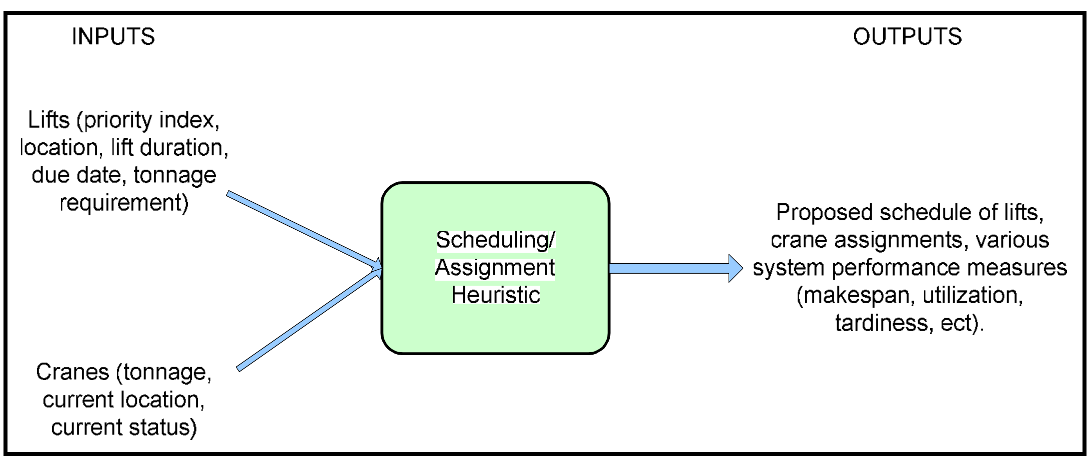

Figure 6

Algorithm Inputs and Outputs

4.2 Test Data Generation

Since actual data is not available from the manufacturer, test data are needed to analyze the performances of the heuristic approaches. The data sets are randomly generated. Six different sets of problems containing various total number of lifts are generated (6, 10, 30, 50, 75, and 100 lifts) with each set containing five problem scenarios. Therefore, the total number of problem cases is thirty (6 sets X 5 problem scenarios per set). As a result, there are five problem scenarios with 6 lifts, five problem scenarios with 10 lifts, five problem scenarios with 30 lifts, and so on. For each case, lift location, lift duration, tonnage requirement, and due date are randomly generated. Lift location is generated based on the uniform distribution where all nodes have an equal probability to host a lift. Before lift durations are generated, statistical measures are calculated using the triangular distribution and are presented in Table 5 below:

Table 5

Lift Duration Statistics in T Units (T = 15 minutes)

Priority	Min (a)	Avg. (c)	Max (b)	Mean	Variance	Std Dev
1	64	96	176	112.00	554.67	23.55
2	32	48	64	48.00	42.67	6.53
3	32	37.3	42.6	37.30	4.68	2.16
4	2	4	16	7.33	9.56	3.09

The following equations are used to calculate the statistical measures in Table 5.

$$Mean = [(a + b + c)] / 3 \tag{7}$$

$$Variance = \frac{a^2 + b^2 + c^2 - ab - ac - bc}{18} \tag{8}$$

$$Standard\ deviation = [\frac{a^2 + b^2 + c^2 - ab - ac - bc}{18}]^{1/2} \tag{9}$$

The minimum, average, and maximum times are estimates based on historical data provided by the manufacturer. As can be seen from the Table 5, there is a large gap between the lift durations of priority 1 lifts and the other priority indices. Even between priority 1 and 2 lifts, priority 1 lifts take twice as long as priority 2 lifts on average. Additionally, when comparing variances and standard deviations, lift durations of priority 1 lifts are far more variable than others.

Based on the calculated statistics in the table, the NORMINV() function within Microsoft Excel is used to generate lift durations. The NORMINV() function requires three arguments: probability, mean, and standard deviation. The probability argument is a uniform randomly generated number between 0 and 1, U[0, 1], while the mean and standard deviation arguments are values from Table 5. The distribution for the Priority 4 lifts can generate negative durations therefore, the duration of a priority 4 lift will be regenerated if such an event occurs.

The lift types for each problem set are generated according the historical data provided by the manufacturer in Table 1. That is, priority 1 lifts make up 20% of all lifts, priority 2 lifts make up 10% of all lifts, priority 3 lifts make up 30% of all lifts, and finally, priority 4 lifts make up 40% of all lifts. Therefore, a problem instance with 100 lifts would have twenty priority 1 lifts, ten priority 2 lifts, and so on.

Time horizons are assigned to the different test cases according to the number of lifts. Time horizons are chosen for each test set after some initial test runs of the base heuristic and are loosely based on the number of priority one lifts. Time horizons are chosen so that the heuristic has a fair chance of completing all lifts within the allotted time horizon. However, they are also set so as to give the heuristic a fair chance of failing to complete all lifts by the allotted time. Consequently, sets with 100 lifts have a longer time horizon than sets with 75 or 50 lifts. Time horizons for each data set are provided in Table 6.

Table 6

Time Horizons in T Units

Number of Lifts	6	10	30	50	75	100
Time Horizon	128	256	640	1280	1920	2560

All due dates for lifts are generated using the following equation:

$$\textbf{Due Date} = \alpha(\text{Planning horizon} - \text{Lift duration}) + \text{Lift duration} \quad (10)$$

where $\alpha = U(0, 1)$

As $\alpha \rightarrow 1$, due dates become larger and easier to meet. When α=1, due date is simply the planning horizon. As $\alpha \rightarrow 0$, due dates become smaller and harder to meet. When α=0, the due date is simply equal to the lift duration.

The test data problems are generated in a manner which limits the randomness between subsets in different number of lifts. For example, a test problem with 6 lifts is generated at first, and then a test problem with 10 lifts is generated that includes the test problem with 6 lifts. The test problem with 10 lifts basically keeps all of the lifts from the 6-lifts problem and additional lifts are generated to reach 10. The additional lifts are generated in accordance with the lift type percentages discussed earlier. For example, to complete a 30-lift test problem generation, all lifts from the 10-lifts test problem are retained and 20 more lifts are generated in accordance with the lift percentages. Therefore, the additional lifts include four priority 1 lifts, two priority 2 lifts, six priority 3 lifts, and eight priority 4 lifts. This method is continued until the 100 lifts problem is generated. This approach allows the number of lifts to increase without invoking unnecessary noise.

Because the planning horizon varies according to the number of lifts in a problem set, due dates of the lifts cannot be the same across all sets. Therefore, rather than due dates remaining the same, alpha remains the same across a set and new due dates are adjusted without compromising the consistency of the generated test data. Two problem instances are provided in Appendix A.

The only crane parameter that is randomly generated is the initial location of cranes on the crane-track network. The uniform distribution is again used so that all nodes on the network have an equal probability of hosting an initial crane location. However, as per the assumptions described in the previous chapter, all initial locations of cranes are at nodes where lifts actually occur (not at turning nodes). If a turn node is randomly generated to be an initial location, the node number will be increased until a non-turn node is found. For example, if node 5 is a turn node and is randomly generated to be an initial location, node 6 will replace node 5 if it is not a turn node.

All test data are kept in text files and are stored in comma delimited format. Samples of the crane and lift test data are provided in Appendix B.

4.3 Base Heuristic

A base heuristic is developed to evaluate the current operating procedures used by the shipbuilder. For the scheduling portion of the problem, the manufacturer currently uses a priority-based sorting procedure. Lifts are sorted in ascending order with respect to their priority indices such that lifts with low indices (high priority) are placed at the top of the list. Therefore, priority 1 lifts are found at the top of the list since they are considered to be high-priority lifts. For the assignment procedure, the shipbuilder currently uses a greedy approach to search for cranes to perform lifts. The greedy approach identifies candidate cranes and then selects a crane or cranes (in the case of priority 1 lifts) that yield the minimum completion time for that particular lift. This approach is suspected to be myopic since it only considers the current lift and does not consider the effects of the current lift's assignment on later lifts and the overall system.

The algorithm is coded in Visual Basic for Applications (VBA) using Microsoft Excel. Microsoft Excel is deemed to be a good host application since lift and crane information can be easily displayed and read in a row-column format. Furthermore, the ability to make quick calculations and organize numerous spreadsheets makes Excel attractive.

Initially, all lift and crane information are loaded into spreadsheets. All information contained in the comma delimited text files can be seen in the spreadsheet. Then, lifts are sorted according to their priority indices and then by their latest start times. In other words, within each priority index, an additional sorting procedure occurs which places lifts with soon latest start times above others that have later latest start times.
The flow chart that describes the base heuristic is shown in Figure 7.

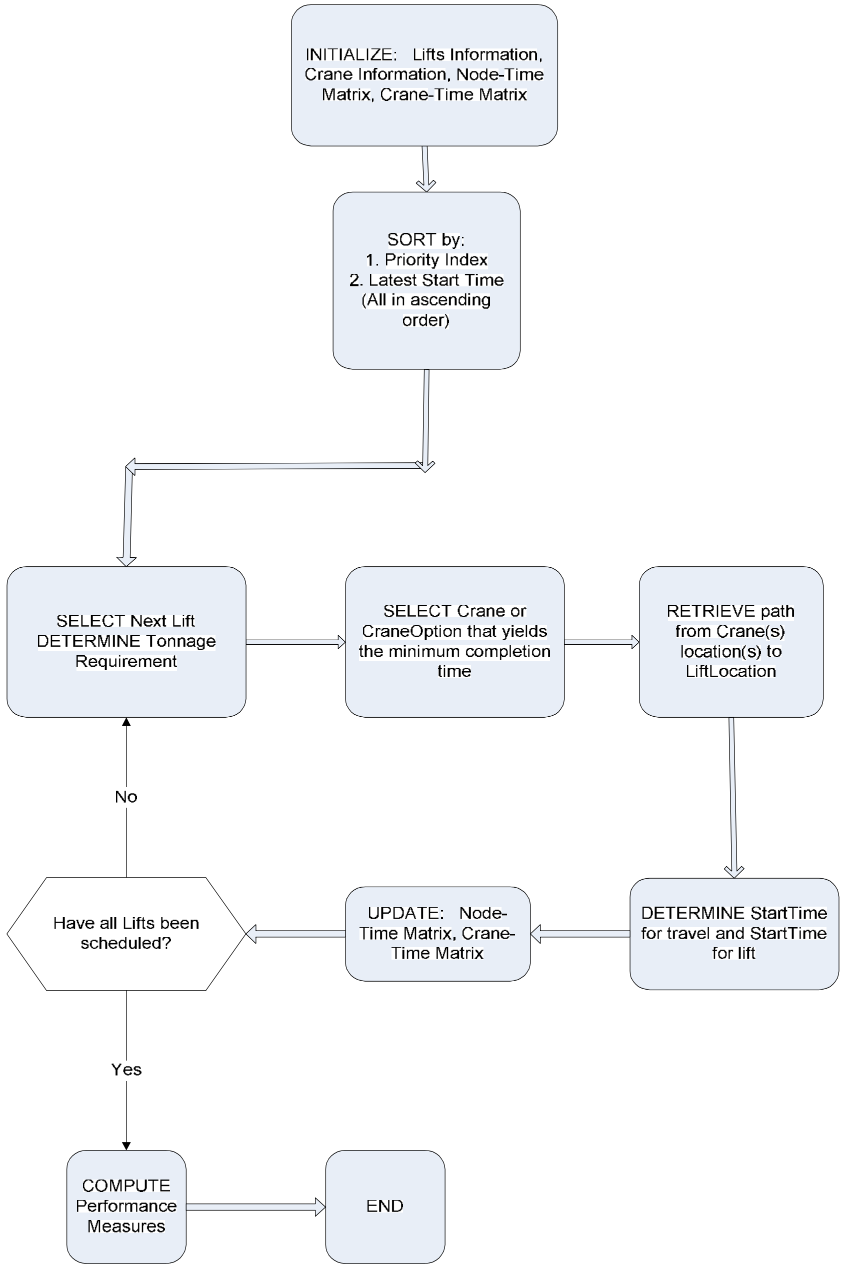

Figure 7

Base Heuristic Flowchart

4.4 Earliest Due Date (EDD)

An Earliest Due Date (EDD) procedure is introduced for the scheduling portion of the heuristic. The EDD procedure replaces the priority-based sorting procedure that is used in the base heuristic as described in the previous section. Instead of prioritizing lifts by their priority indices, lifts are prioritized by their due dates, with short due dates receiving top priority. In an event where there is a tie between lifts, priority indices act as the tiebreaker, with high priority indices winning over low priority indices. The EDD procedure has been shown to minimize maximum tardiness and the number of tardy jobs in a given set. Although the objective of the research is not minimizing maximum tardiness or number of tardy jobs, the EDD procedure may have a positive impact on makespan. Even if the EDD procedure cannot improve makespan, it can improve performance measures regarding lateness and tardiness. Additionally, the EDD procedure will be combined with an assignment procedure that may improve makespan, which may yield quality results in terms of both tardiness and makespan.

4.5 Ant Colony Optimization (ACO)

As stated in a previous chapter, Ant Colony Optimization (ACO) is identified to be a candidate solution method for the problem in this research since it is evident from current literature that ACO can yield quality solutions to scheduling and assignment problems. ACO is a heuristic technique that is inspired by the natural behavior of ants when they are searching for food from their colony. As ants travel in search of food, they deposit chemical pheromones on the ground. When an ant finds food, it also leaves a trail of pheromones on its way back to the colony so that when other ants from the colony searches for food, they will follow the path already taken by an ant that has found food. There may exist multiple paths to the food source, therefore, pheromone intensity plays an important role in the selection of paths. Since pheromone evaporates as time progresses, long paths eventually die out and new paths will have to be explored. Short paths, however, retain their pheromones for longer periods of time and will be used more frequently by other ants to find food. Additionally, pheromone intensities of short paths are reinforced more frequently. Ultimately, a shortest (best) path to the food source will be established and will be used by all ants of the colony. In nature, one can often observe ants who have found food travel in a single-file line from their colony to the food source in a very organized fashion. On the other hand, ants who have not found food usually travel on scattered and seemingly random paths. Figure 8 further illustrate the path selection procedure based on pheromone intensities:

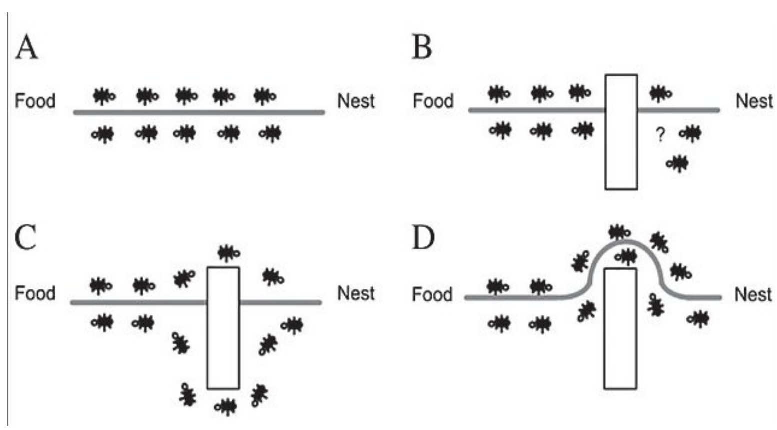

Figure 8

ACO Path Selection Example

By placing an obstacle between the nest and the food as seen in (B) in Figure 8, two paths are created in which ants can select from. Initially, ants will travel on both paths as seen in (C). But as time progresses, the shorter path's pheromone level will intensify due to reinforcement and will eventually emerge as the best path.

In ant colony optimization, artificial ants are constructed as agents and will move to different locations based on a probabilistic function. After the agents have visited all locations or constructed a solution, they will update the trail intensities by artificially depositing pheromones on the paths they have traveled. To mimic the natural behavior of the system, paths which yield better objective function values will have stronger trail intensities and be more attractive for future agents. This means that ants who found better solutions will be allowed to deposit greater amounts of pheromone on the paths they just traveled. The general structure of the ACO algorithm is shown in Figure 9.

Step 1:	Initialize the pheromone values and the parameters
Step 2:	**While** the terminating condition is not met Select the 1st lift from the ordered lift list **Do Until** all lifts have been assigned Select crane option based on pheromone intensity Update the availability of the selected crane Move to next lift in the list **Loop** Update the pheromone intensities using a pheromone updating rule **Loop**
Step 3:	Return the best solution found

Figure 9

ACO General Algorithm Structure

In step 1, the pheromone value and algorithm parameters are initialized and set. In step 2, crane options are selected based on pheromone intensity and the procedure continues until all lifts have been assigned a crane option. When all lifts have been assigned, pheromone intensities for the crane options selected are updated. Finally, in step 3, the algorithm terminates based on some specified terminating condition and the best solution is returned.

The ACO algorithm is applied to the assignment portion of the problem instead of the scheduling portion since it appears that the assignment of cranes to lifts plays a more important role than how the lifts are sequenced. Since all jobs are ready to be performed at time zero, a lift that is in the middle of the list could still be started at an early time. Also, since blocking on the track network is a major issue and affects the completion time of lifts, selecting crane options that avoid blocking becomes vital to reducing makespan. Therefore, the ACO algorithm is designed to handle the crane assignment portion of the problem.

In order to apply the concepts of ACO, the assignment problem is represented graphically with nodes representing lifts and arcs connecting nodes representing crane options. So, for a given problem set with N lifts, there are a total of $N + 1$ nodes and numerous connection arcs. An ant starts its solution building path at the first node on the graph. Next, the ant selects a crane option (an arc) which has the highest pheromone level. This selection process continues until the END node is reached, at which point, the trip is concluded and a solution is built. Once a solution is built, pheromone levels are updated for the arcs used in the trip and makespan is calculated. In the algorithm, an iteration corresponds to a solution set. However, the solutions may not be unique between different iterations. In other words, there may be a case where multiple iterations produced the same set of solutions. Figure 10 provides a graphical representation of the assignment problem. The arcs are identified using the crane ID – tonnage format.

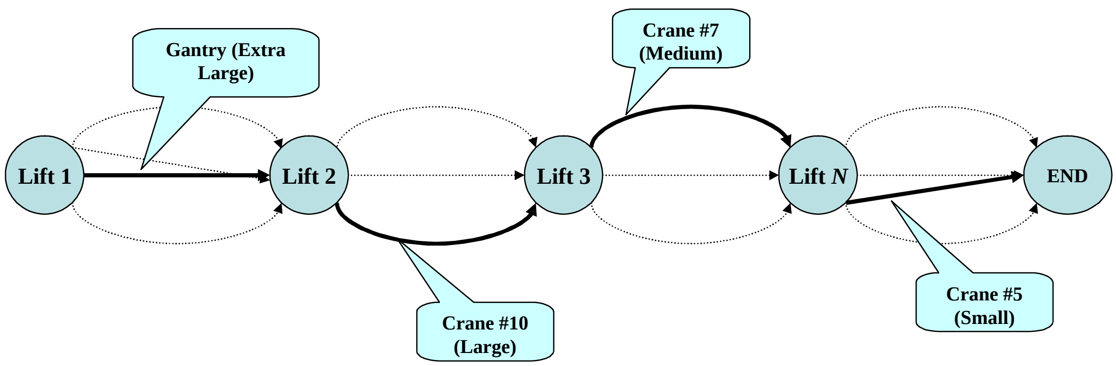

Figure 10

ACO Graphical Representation of Assignment Problem

Pheromone intensities are updated using the following equation:

$$\tau_{ij} = (1-\rho)\tau_{ij} + \Delta\tau_{ij} \tag{11}$$

where r is the evaporation parameter that takes on values between 0 and 1, and τ_{ij} is the pheromone intensity of assigning crane j to lift i. $\Delta\tau_{ij}$ is calculated after each iteration and is simply the inverse of the makespan achieved in that iteration (1/makespan). So, it is intuitive from the equation that small values of makespan lead to larger pheromone intensity updates.

CHAPTER V

EXPERIMENTAL RESULTS AND ANALYSIS

5.1 Introduction

The algorithms are evaluated based on experiments conducted with the model of crane operations and the test data sets defined earlier r. Initial experimentation is conducted on a partial crane track network approximately two-thirds of its actual size. Likewise, an abbreviated crane list is used for the initial runs to evaluate the heuristic approaches, i.e., two large cranes, one medium crane, one small crane, and Gantry. Partial network and crane list are used in initial experiments to reduce run time of the algorithm. Table 7 summarizes the experiments.

Table 7

Heuristic Approaches

Approach	Scheduling	Assignment
Base Heuristic	Priority	Greedy
EDD/Greedy	EDD	Greedy
EDD/ACO	EDD	ACO

5.1.2 Performance Measures

To evaluate the quality and effectiveness of the heuristics, various performance measures are considered. In addition to lift makespan and crane utilization, measures of lateness and tardiness are also used. The performance measures are defined as follows:

- *Makespan* – maximum completion time in a given set of lifts.

- *Average Lateness* –sum of total lateness in a given set of lifts divided by the number of lifts in that set. For a particular lift, lateness is calculated by subtracting the due date from completion time. Therefore, lateness can be negative, which indicates the lift was finished prior to its due date.

- *Maximum Lateness* –maximum lateness in a given set of lifts.

- *Number Tardy* –number of tardy lifts in a given set of lifts. Tardiness, unlike lateness, is always positive and is essentially any nonnegative lateness. Number of tardy lifts is synonymous with number of late lifts.

- *Average Tardiness* –sum of total tardiness for a set of lifts divided by the number of lifts in the set.

- *Percent Tardy* –number of tardy lifts in a given set divided by the number of lifts in the set. Calculated by *Number Tardy* divided by number of lifts.

Other crane performance measures used to evaluate the heuristic approaches include:

- *Percent Waiting* –percent of time a crane waits for the second crane to arrive at the lift location during a priority 1 lift. This only applies to cranes that are able to perform priority 1 lifts.

- *Percent Idle* –percent of time the crane is idle. This measure is calculated by dividing the units of time a crane is idle by the makespan.

- *Percent Blocked* –percent of time the crane is blocked by another crane on a selected path. This measure is calculated by dividing the units of time a crane is blocked by the makespan.

- *Travel Utilization* –percent of time a crane is traveling. This measure is calculated by dividing the units of time a crane is traveling by the makespan.

- *Actual Utilization* –percent of time a crane is actual performing a lift. This measure is calculated by dividing the total time a crane spends performing lifts by the makespan. Actual utilization is synonymous with crane utilization described in the research objectives in section 1.4.

As can be seen in the descriptions above, makespan plays an important role in crane performance and utilization. Every crane performance measure is based on makespan, which again signifies the relationship between makespan and crane performance in this research problem.

5.2 Base Heuristic

Recall, the base heuristic sorts lifts based on priority indices and makes crane assignments greedily. Since the base heuristic models the current operating procedures used by the shipbuilder, it is expected that the results are would be close to current performance levels. Table 8 contains a summary of results after all test sets have been run:

Table 8

Summary of Results (Base Heuristic)

# of Lifts	Makespan	Avg. Tardy	Max Tardy	%Tardy
6	130.40	41.60	74.00	30.00%
10	220.20	78.41	133.40	22.00%
30	643.60	118.11	279.40	18.67%
50	1039.20	193.62	437.80	13.60%
75	1429.00	272.97	671.40	18.40%
100	1768.00	391.61	959.40	17.00%

Table 8 contains averages of the five problem instances within each test set. The unit for makespan and tardiness is T, which again is 15 minutes. As expected, as the number of lifts increase, the performance of the system worsens. This can be seen in areas of makespan, average tardiness, and maximum tardiness. In addition, crane performance results are provided in Table 9 for the base case.

Table 9

Crane Performance (Base Heuristic)

Crane ID	Capacity	Waiting	Idle	Blocked	Travel	Avg. Utilization
Gantry	Extra Large	0.00%	1.96%	0.00%	0.57%	97.46%
1	Large	0.30%	63.84%	6.38%	9.40%	20.07%
2	Large	0.25%	87.89%	1.45%	1.28%	9.11%
5	Small	0.00%	54.11%	13.51%	12.00%	20.37%
6	Medium	0.00%	55.16%	7.69%	13.54%	23.60%
7	Medium	0.00%	53.54%	7.55%	13.07%	25.85%
8	Medium	0.00%	86.32%	5.40%	2.42%	5.87%
10	Large	1.39%	85.21%	2.13%	1.00%	10.27%
11	Large	2.88%	87.16%	1.03%	0.71%	8.21%

Similar to Table 9, the values reported are averages of all problem instances. The low utilizations of the large capacity cranes can be attributed to the fact that Gantry is performing most of the priority 1 lifts. In fact, Gantry is working for almost the entire duration of makespan, yielding a high utilization of 97.46%. Consequently, Gantry also has the lowest percentages among all cranes in being idle, blocked, and traveling.

Results from the base heuristic serve as a baseline of current operating conditions at the shipyard. As the results suggest, there are opportunities for improvement in system and crane performance and the heuristic itself.

5.3 EDD/Greedy

As mentioned previously, the Earliest Due Date approach replaces the priority index sorting procedure with a due-date based sorting procedure. The assignment procedure remains the same, the greedy approach. Results from the EDD/Greedy approach are presented in the Table 10.

Table 10

Summary of Results (EDD/Greedy)

# of Lifts	Makespan	Avg. Tardy	Max Tardy	%Tardy
6	140.60	19.77	20.20	23.33%
10	225.60	12.30	11.40	8.00%
30	650.80	48.70	57.60	7.33%
50	1058.60	14.27	21.00	4.00%
75	1450.20	35.20	56.80	2.93%
100	1771.60	18.10	30.40	2.20%

As expected, performance measures regarding tardiness improved dramatically over the base case. There are significant reductions in all categories of tardiness, with an average 80% decrease in average tardiness across the data sets. Figure 12 illustrates the improvement in average tardiness:

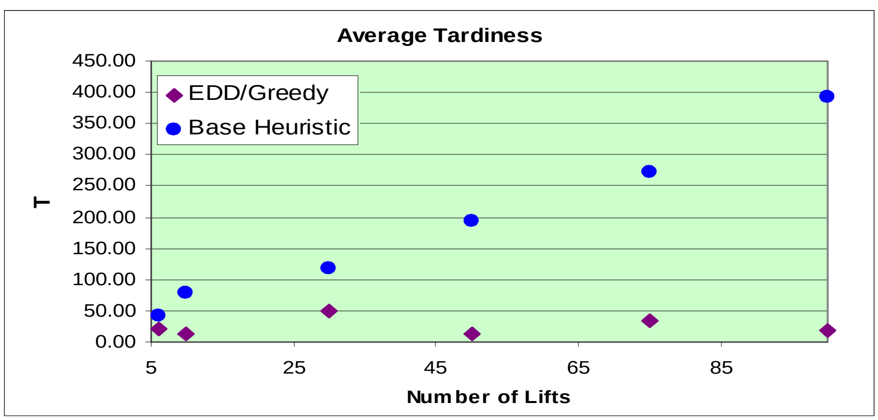

Figure 11

Average Tardiness (EDD/Greedy vs. Base Heuristic)

As can be seen in Figure 11, average tardiness is limited to fewer than 50 time units for all test sets when using EDD while average tardiness seems to increase linearly as the number of lifts increase for the base heuristic. Despite the great improvements in the area of tardiness, the EDD/Greedy approach actually performed worse than the base heuristic in terms of makespan. Even though makespan only slightly increased with the EDD/Greedy method, the approach is not suitable since the objective of the research is to reduce makespan. Since reduction in makespan is not achieved using this heuristic approach, increases in crane utilization are not expected

5.4 EDD/ACO

In this case, the greedy crane assignment procedure is replaced with the ant colony optimization based heuristic. The ACO parameter settings are: $\rho = 0.25$ (evaporation rate) and $\tau_o = 0.01$ (initial pheromone level). Twenty iterations were run and a summary of the results from the best iteration is provided in Table 11.

Table 11

Summary of Results (EDD/ACO)

# of Lifts	Makespan	Avg. Tardy	Max Tardy	%Tardy
6	124.80	9.30	11.20	20.00%
10	211.30	20.75	24.25	20.00%
30	535.00	18.17	20.50	7.50%
50	744.25	17.50	24.50	3.50%
75	1056.50	10.02	9.75	2.67%
100	1357.80	30.38	54.00	3.75%

As the results show, the combination of the earliest due date sorting procedure and the ant colony optimization based assignment procedure yields improved results in terms of makespan and tardiness when compared to the base heuristic.

Figure 12 illustrates the improvement in makespan achieved by using EDD/ACO:

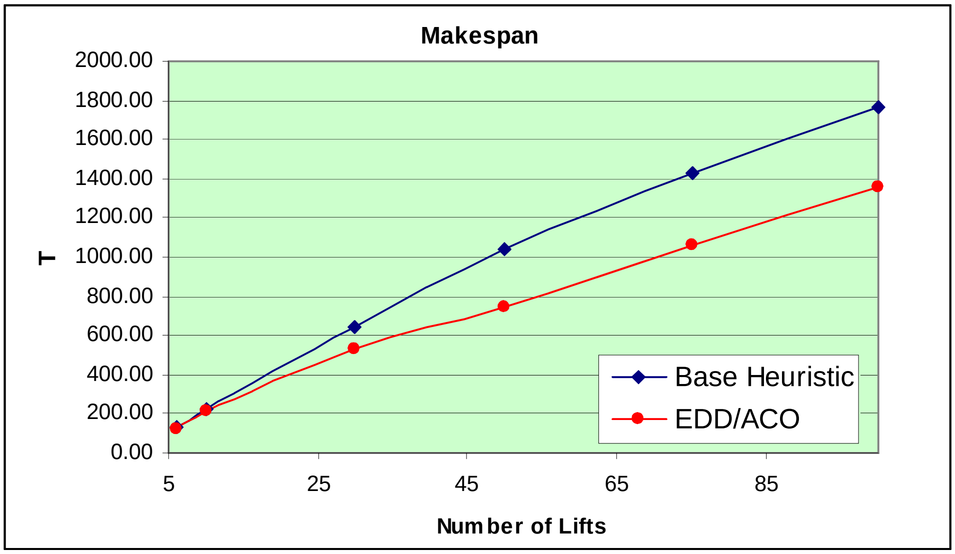

Figure 12

Makespan (EDD/ACO vs. Base Heuristic)

As the number of lifts in the test set increases, the reduction in makespan also increases. A reduction in makespan of approximately 25 to 30% is seen for sets with 50 or more lifts. The gap in makespan between the two methods is expected to increase as the number of lifts extends beyond 100.

The EDD/ACO approach also results in improvements in the tardiness measures when compared to the EDD/Greedy approach. The ACO assignment approach obtained a lower average tardiness value for three of the six test sets when paired with EDD as compared with the greedy assignment approach. Comparison in average tardiness between the two methods is shown in Figure 13. The shape of the figure may be misleading as the time scale is small. Average tardiness of both approaches is actually stable when graphed on a larger time scale.

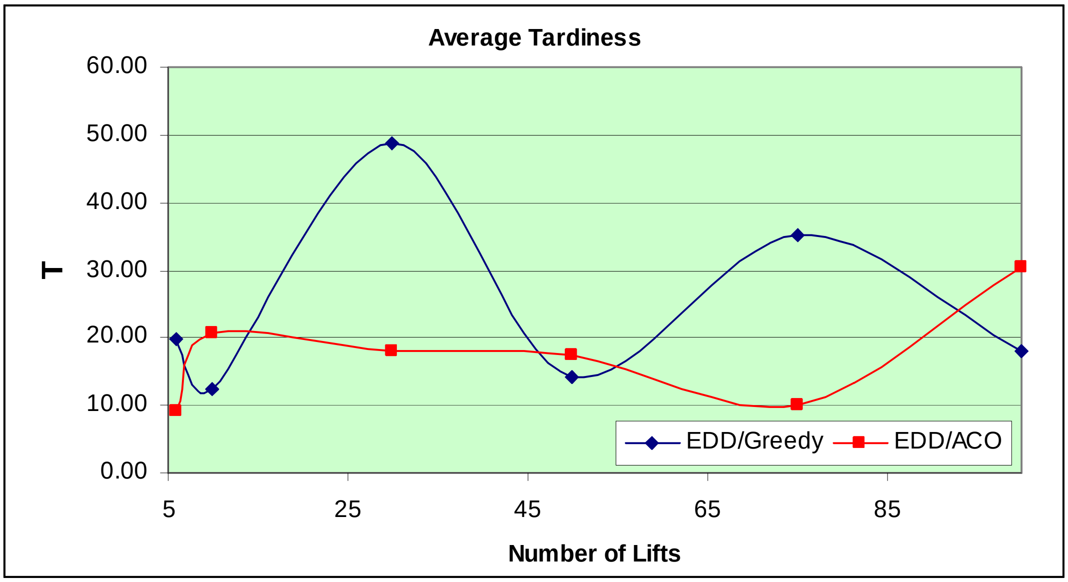

Figure 13

Average Tardiness (EDD/ACO vs. EDD/Greedy)

Based on the experimental results, EDD/ACO performs better than the other two approaches, both in terms of makespan and tardiness. EDD/ACO performs better in both measures when compared to the base heuristic while the ACO procedure performs better than the greedy procedure in makespan and tardiness for half of the test sets when the assignment procedures are paired with the EDD sorting procedure.

In terms of crane utilization, improved results are also expected for the EDD/ACO procedure since reductions in makespan are seen. Average crane utilizations and other measures are shown in Table 12.

Table 12

Crane Performance (EDD/ACO)

Crane ID	Capacity	Waiting	Idle	Blocked	Travel	Avg. Utilization
Gantry	Extra Large	0.00%	16.76%	0.00%	1.44%	81.85%
1	Large	5.08%	30.37%	11.40%	13.61%	39.81%
2	Large	8.67%	26.57%	10.72%	8.33%	45.96%
3	Small	0.00%	74.91%	10.51%	5.46%	9.37%
4	Medium	0.00%	59.17%	10.05%	10.65%	20.39%
5	Medium	0.00%	55.37%	16.60%	9.79%	18.49%
6	Medium	0.00%	47.50%	14.71%	14.19%	23.86%
7	Large	17.94%	23.14%	6.71%	5.54%	46.92%
8	Large	16.08%	24.01%	11.46%	8.34%	40.37%

As the results show, the ACO crane assignment procedure improves the overall average crane utilization as compared with the greedy procedure. The ACO assignment procedure yielded an overall average crane utilization of 36.34% as compared with 24.53% achieved by the greedy assignment procedure from the base heuristic. Consequently, the percentage of time cranes spent waiting, idling, being blocked, or traveling decreased as a result of increased utilization. As expected, cranes that meet tonnage requirements to handle priority one lifts exhibit the highest utilization values since priority one lifts dominate the other lift types in terms of lift durations. As a result, utilization values for Gantry and the large capacity cranes are higher than that of other cranes.

In summary, the EDD/ACO method offers a greatly improved solution approach to the problem at hand. It satisfies the research objectives of reducing lift makespan and increasing crane utilization. In addition to satisfying the research objectives, the EDD/ACO approach also returns good results in terms of tardiness (it limits the tardiness of lifts to fewer than 30 time units, on the average). Given these positive experimental results, improvements to the EDD/ACO approach are further investigated.

5.5 ACO Parameter Investigations

After obtaining promising results from the ACO methodology, further investigations are made into the parameters of the algorithm to see if different settings may impact the quality of solutions. Two parameters, r and t_o, are identified as key components to the algorithm, with r being the evaporation rate and t_o being the initial pheromone level. Experimental results shown in the previous section had the parameters set nominally at $r = 0.25$ and $t_o = 0.01$. To determine the impact of each parameter, a test problem is run 10 times with different settings for each parameter setting. Two additional values are selected for each parameter, with one above the nominal value and one below the nominal value. For the evaporation rate, additional parameter values are 0.1, and 0.50. For the initial pheromone level, the additional parameter values are 0.001, and 0.1. An instance of a 30-lift test problem is used for all 10 replications and results are provided in Table 13. The "+" sign indicates the parameter is set at the value that is above the nominal while the "-" sign indicates the parameter is set at the value below the nominal.

Table 13

Parameter Analysis (30 Lifts)

Replicate	r:+ to:-	r:- to:+	r:+ to:+	r:- to:-
1	460	425	381	495
2	401	428	506	460
3	464	378	491	437
4	461	510	485	382
5	425	454	395	478
6	457	435	363	477
7	469	445	512	493
8	449	521	383	513
9	444	472	378	415
10	366	475	421	333
Mean	439.6	454.3	431.5	448.3
Std Dev	33.10	42.32	59.93	56.89
Variance	1096.04	1791.56	3592.50	3237.12

A balanced two-way analysis of variance (ANOVA) test is conducted to reveal the effects of the two parameters and their interactions. The statistical software Minitab is used to conduct the ANOVA test. Results obtained from the data in Table 13 are shown Figure 14. (R and T denote evaporation rate and initial pheromone level, respectively).

ANOVA: Mspan versus R, T

Factor	Type	Levels	Values
R	fixed	2	-, +
T	fixed	2	-, +

Analysis of Variance for Mspan

Source	DF	SS	MS	F	P
R	1	2481	2481	1.02	0.319
T	1	11	11	0.00	0.947
R*T	1	497	497	0.20	0.654
Error	36	87455	2429		
Total	39	90444			

S = 49.2880   R-Sq = 3.30%   R-Sq(adj) = 0.00%

Figure 14

ANOVA Table (30 Lifts)

P-values in the ANOVA table are used to determine if that factor has a significant effect on makespan. If the p-value is less than alpha, which is set at 0.05, then the factor is deemed significant. However, as can be seen in the ANOVA table, p-values for the two parameters and their interaction are all less than alpha, which indicates the effects of the parameters and interaction between them are not significant.

To assess the model's adequacy, residuals of the experimental data are further examined. In the model, residuals are the differences between the actual observations and the averages for each parameter setting. In order to draw valid conclusions from the ANOVA test, residuals need to follow the normal distribution. Residual plots from Minitab are shown in Figure 15.

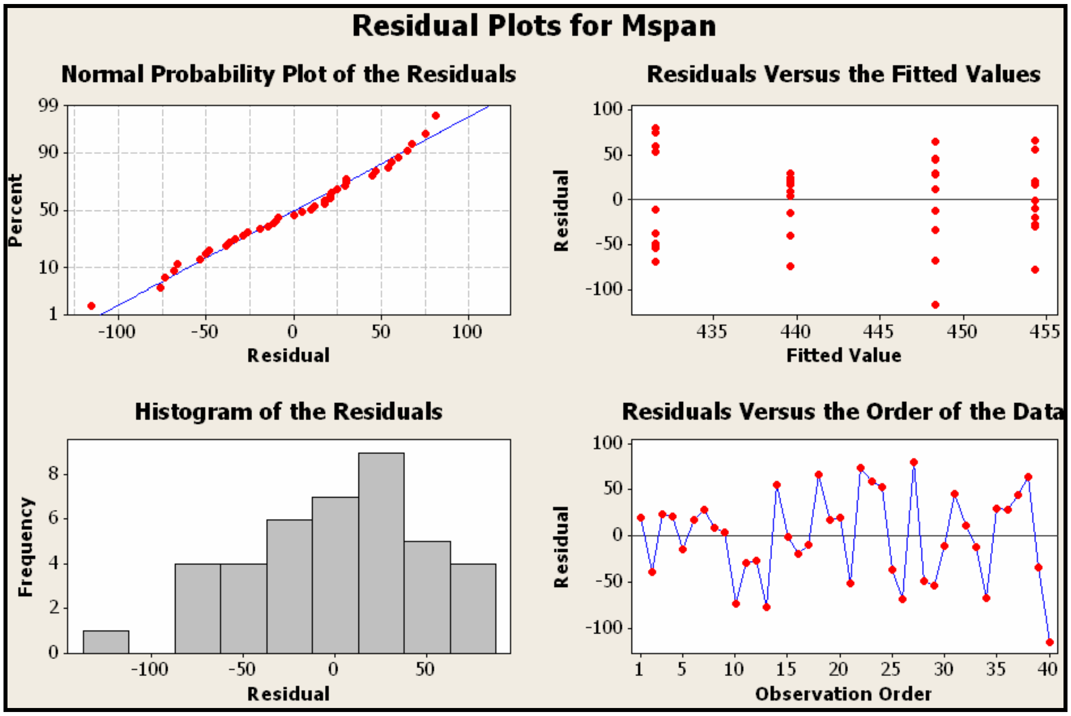

Figure 15

Residual Plots (30 Lifts)

From the normal probability plot shown in Figure 15, it appears that the residuals follow the normal distribution. Therefore, the ANOVA test and conclusions drawn from the test appear to be valid.

In order to further investigate the effects of the ACO parameters on the algorithm's performance, similar experiments were conducted on problem instances containing 75 and 100 lifts. It is unknown whether increasing the problem size will have any effect on the parameters. Results of the parameter analysis for the 75-lift problem are shown in Table 14.

Table 14

Parameter Analysis (75 Lifts)

Replicate	r:+ to:-	r:- to:+	r:+ to:+	r:- to:-
1	1102	1017	870	1090
2	1011	1006	1033	1040
3	1260	1044	1058	956
4	921	969	937	911
5	1020	972	887	988
6	963	999	974	1010
7	1059	1034	947	1039
8	1017	949	1069	835
9	1046	921	940	1041
10	969	954	1033	874
Mean	1036.8	986.5	974.8	978.4
Std Dev	93.95	39.85	70.45	82.62
Variance	8826.62	1588.72	4963.95	6826.48

Again, a two-way ANOVA test is conducted to determine if the effects of the parameters and their interaction are significant. The ANOVA table, as calculated by Minitab, is shown in Figure 16.

ANOVA: Mspan versus R, T

```
Factor  Type   Levels  Values
R       fixed       2  -, +
T       fixed       2  -, +


Analysis of Variance for Mspan

Source  DF      SS      MS     F      P
R        1    5452    5452  0.98  0.328
T        1    7263    7263  1.31  0.260
R*T      1   12285   12285  2.21  0.146
Error   36  199852    5551
Total   39  224852


S = 74.5080   R-Sq = 11.12%   R-Sq(adj) = 3.71%
```

Figure 16

ANOVA Table (75 Lifts)

Again, the p-values from the ANOVA table indicate that the ACO parameters and their interaction do not have significant effects on makespan when alpha is set at 0.05. Residuals from the model are examined to validate the statistical model's adequacy and are provided in Figure 17.

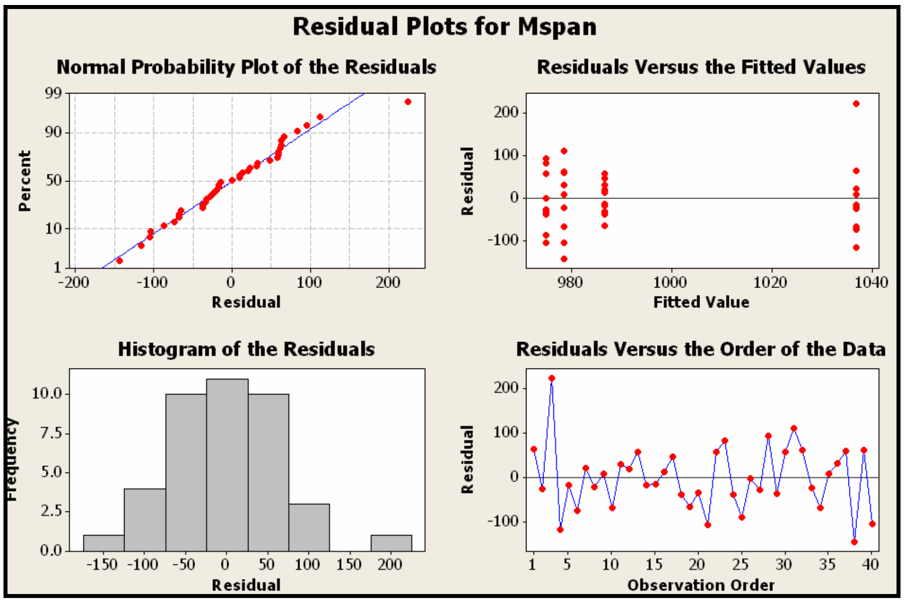

Figure 17

Residuals (75 Lifts)

Besides for one apparent outlier in the normal probability plot shown in Figure 17, the residuals seem to follow the normal distribution. As a result, conclusions drawn from the ANOVA table appear to be valid.

The problem size is increased once again and the parameter analysis for the 100-lift problem is conducted and the results are presented in Table 15.

Table 15

Parameter Analysis (100 Lifts)

Replicate	r:+ to:-	r:- to:+	r:+ to:+	r:- to:-
1	1377	1361	1101	1320
2	935	1224	1089	1142
3	1327	1311	1306	1189
4	1384	1205	1462	1337
5	1023	1317	1162	1099
6	1025	1205	1185	1121
7	1200	1349	1332	1441
8	1508	1448	825	973
9	1251	1135	739	1533
10	1200	1162	1259	1170
Mean	1223	1272	1146	1232
Std Dev	184.35	100.18	223.29	171.27
Variance	33987.35	10036.12	49861.52	29334.82

Similarly, the two-way ANOVA test is performed using Minitab and the results are presented in Figure 18. The p-values in the table again indicate that the ACO parameters and their interaction have no statistically significant effects on makespan when alpha is set to 0.05. Residuals are examined to validate the model's adequacy and the above conclusion. Residual plots, as constructed by Minitab, are provided in Figure 19.

ANOVA: Mspan versus R, T

Factor	Type	Levels	Values
R	fixed	2	-, +
T	fixed	2	-, +

Analysis of Variance for Mspan

Source	DF	SS	MS	F	P
R	1	45698	45698	1.48	0.231
T	1	3572	3572	0.12	0.735
R*T	1	33756	33756	1.10	0.302
Error	36	1108685	30797		
Total	39	1191710			

S = 175.490   R-Sq = 6.97%   R-Sq(adj) = 0.00%

Figure 18

ANOVA Table (100 Lifts)

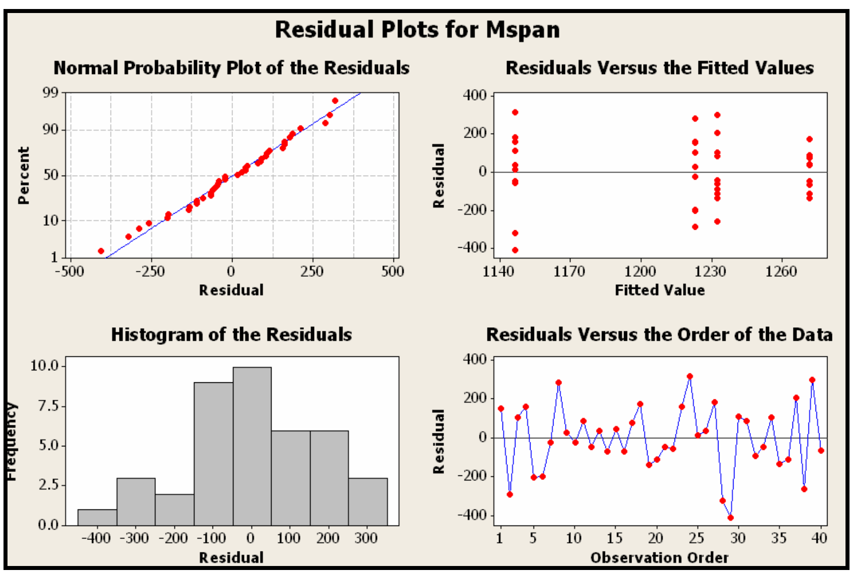

Figure 19

Residuals (100 Lifts)

The normal probability plot of the residuals shows that the residuals of the model appear to follow the normal distribution. Hence, adequacy of the model and the conclusions formed appear to be validated.

5.6 Full Crane Track Network and Crane List

So far, all experiments are conducted using a partial crane track network that is approximately two-thirds of the original size, and a partial set of cranes that is about 80% of the total number of cranes used in the shipyard. To obtain results that are consistent with the shipbuilder's final assembly area and its resources, experimental results are also collected using the full crane track network and crane list. Since the EDD/ACO heuristic has been shown to be the best approach to the problem, it is used to obtain results for the expanded network and crane list. All test data sets are regenerated to include the new node numbers from the expanded area of the network. The crane list is also updated to include all cranes as listed in Table 2.

Given that the parameter settings did not influence results, the heuristic was run with both ACO parameters at their nominal levels. Also, keeping both parameters at their middle levels is consistent with the experiments conducted for the partial network and crane list in section 5.4. Table 16 provides a summary of results for the EDD/ACO approach when it is applied to the full network and crane list.

Table 16

Summary of Results (EDD/ACO – Full Network and Crane List)

# of Lifts	Makespan	Avg. Tardy	Max Tardy	%Tardy
6	122.4	17.27	18.4	16.67%
10	147.8	1.60	1.6	4.00%
30	340	2.60	3.4	1.33%
50	568	16.53	17.4	2.40%
75	686.6	2.60	2.6	0.53%
100	972.4	24.55	37.8	2.40%

The results indicate that as the network and crane list is expanded, makespan and average tardiness continues to improve when compared to previous experimental results.

Figure 20 illustrates the difference between the makespans for the partial and full networks. Figure 21 illustrates the difference between the average tardiness for the partial and full networks.

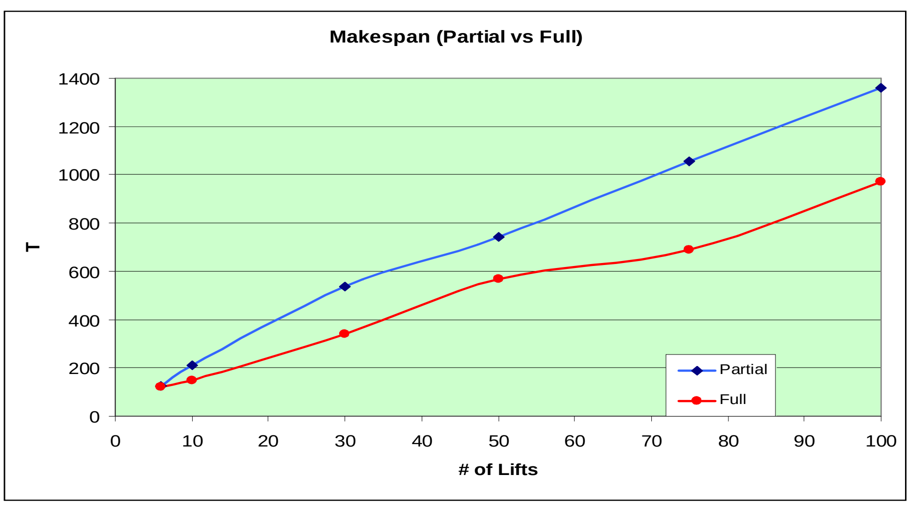

Figure 20

Makespan (Partial vs. Full)

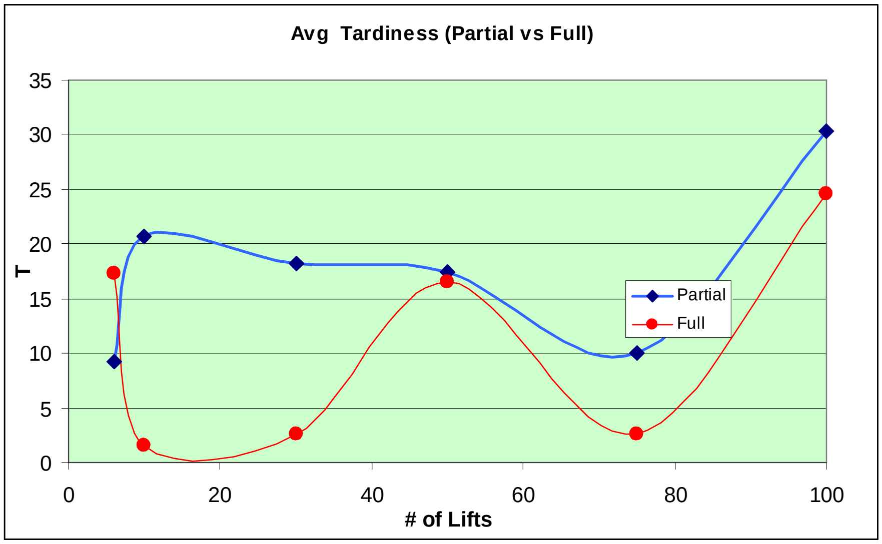

Figure 21

Average Tardiness (Partial vs. Full)

The improvement in makespan and average tardiness may be attributed to the fact that there are more resources to perform lifts, leading to more options for the heuristic to choose from. Additionally, out of the four cranes that are added to the list, three of them are eligible to perform priority 1 lifts, which are main drivers of makespan in the problem instances. Results of crane performance are shown in Table 17.

Table 17

Crane Performance (EDD/ACO – Full Network and Crane List)

Crane ID	Capacity	Waiting	Idle	Blocked	Travel	Avg. Utilization
Gantry	Extra Large	0.00%	23.66%	0.00%	1.29%	75.38%
1	Large	17.88%	33.10%	11.48%	5.69%	32.18%
2	Large	3.90%	33.47%	16.95%	9.30%	36.72%
3	Large	5.87%	19.55%	13.78%	11.48%	49.64%
4	Large	9.56%	58.09%	11.48%	3.71%	17.49%
5	Small	0.00%	77.41%	11.19%	3.10%	8.64%
6	Medium	0.00%	42.97%	17.97%	16.00%	23.39%
7	Medium	0.00%	55.91%	14.77%	8.62%	21.03%
8	Medium	0.00%	45.82%	11.29%	16.74%	26.48%
9	Medium	0.00%	57.09%	13.72%	11.16%	18.35%
10	Large	12.13%	43.70%	10.21%	3.09%	31.20%
11	Large	10.20%	48.37%	14.57%	3.09%	24.09%
12	Large	13.71%	64.10%	6.49%	1.52%	24.51%

The average crane utilization from the experiments with the full network and crane list as is 29.9%, as compared with 36.3% from the EDD/ACO implementation on the partial network and crane list. The decrease in average utilization of cranes may be attributed to there being more cranes. Gantry saw an approximate 6% decrease in average utilization from 81.85% to 75.38%. This decrease is most likely due to Gantry not having access to the new node locations that were added to the partial network; i.e. (Gantry is not a crane option if a priority one lift is located in one of the new node locations.)

5.7 Priority/ACO

The base heuristic is modified to see if it can yield results that are superior to the EDD/ACO heuristic. As stated previously, the base heuristic combines a priority-based sorting procedure with a greedy crane assignment procedure. In this modification, the greedy assignment procedure is replaced with the ACO assignment algorithm. Therefore, the modified base heuristic, Priority/ACO, sorts lifts based on their priority indices and then assigns cranes using the ACO algorithm. Like the base heuristic, lifts with low numeric indices (high priority) will be placed on the top. Experimental results are obtained from running test problem instances that contain the full crane track network and the full crane list and are summarized in Table 18.

Table 18

Summary of Results (Priority/ACO)

# of Lifts	Makespan	Avg. Tardy	Max Tardy	% Tardy
6	130.8	14.60	27.2	23.33%
10	148	35.70	54	14.00%
30	307.8	58.70	90.2	18.00%
50	587.8	232.67	405.6	12.40%
75	788.4	221.33	342.6	14.67%
100	1007.2	312.05	587.4	19.00%

Based on the results, as summarized in Figure 22, it does not appear that the Priority/ACO approach is better than the EDD/ACO approach. For problem cases with less than 75 lifts, the two approaches yield very similar results.

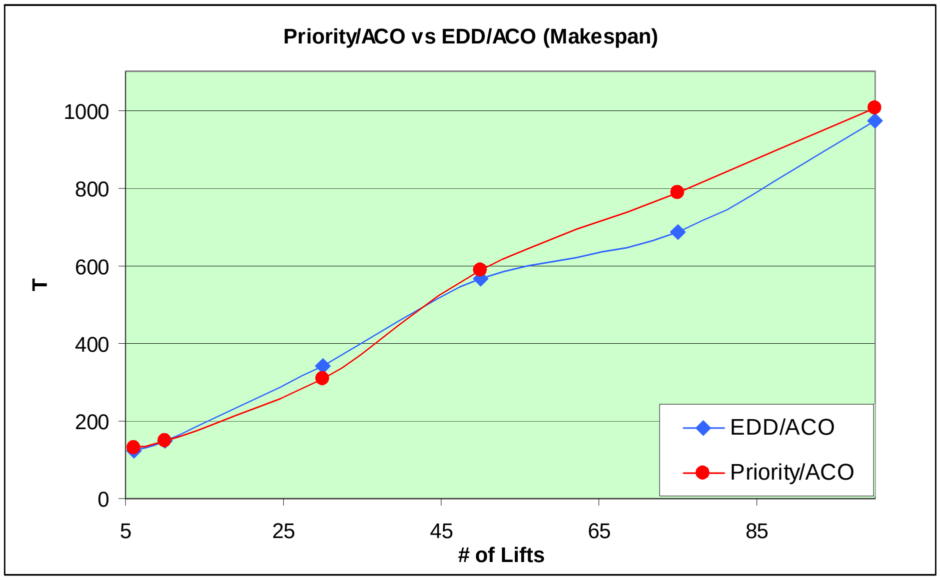

Figure 22

Priority/ACO vs. EDD/ACO (Makespan)

The EDD/ACO approach performs better in problem instances of 6, 50, 75, and100 lifts but the gap between the two approaches is small on the average. However, as expected, the Priority/ACO approach did not yield good results in terms of tardiness. In all tardiness performance measures, the Priority/ACO approach was worse than the EDD/ACO approach. Average tardiness for the two approaches is shown in Figure 23.

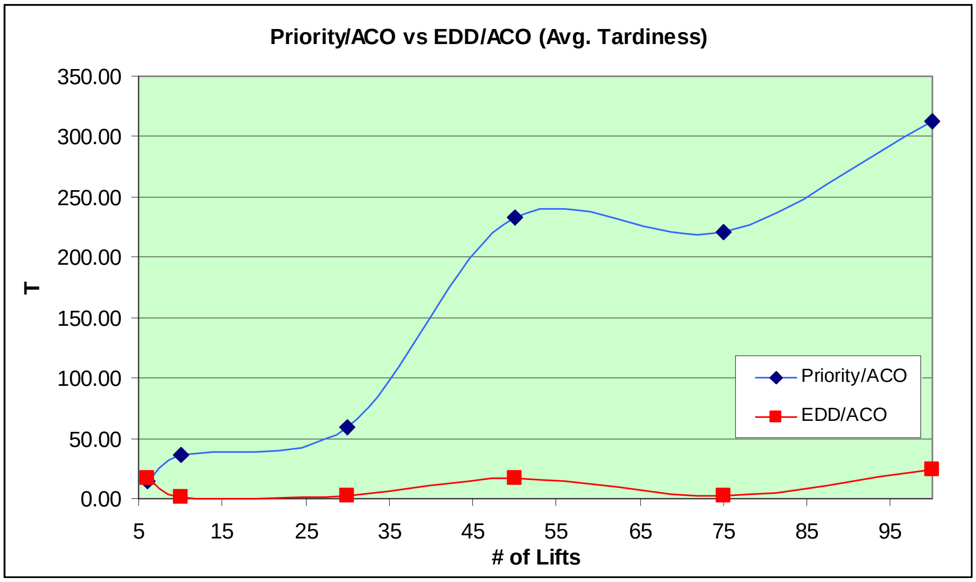

Figure 23

Priority/ACO vs. EDD/ACO (Average Tardiness)

As can be seen from the figure, there is a large gap in average tardiness when the problem size extends beyond 30 lifts. The same can be concluded for maximum tardiness and percentage of tardy lifts.

Crane performance is shown in the following table.

Table 19

Crane Performance (Priority/ACO)

Crane ID	Capacity	Waiting	Idle	Blocked	Travel	Avg. Utilization
Gantry	Extra Large	0.00%	53.00%	0.00%	1.00%	46.25%
1	Large	9.06%	29.50%	7.75%	7.47%	46.47%
2	Large	6.91%	23.81%	12.94%	12.00%	44.59%
3	Large	8.12%	31.96%	19.40%	11.26%	29.51%
4	Large	6.31%	57.57%	9.13%	5.07%	22.17%
5	Small	0.00%	67.52%	20.57%	3.95%	8.21%
6	Medium	0.00%	37.42%	25.82%	14.01%	23.00%
7	Medium	0.00%	44.21%	22.14%	9.60%	24.30%
8	Medium	0.00%	31.99%	21.28%	17.08%	29.90%
9	Medium	0.00%	53.28%	22.00%	9.53%	15.44%
10	Large	5.96%	45.88%	6.62%	3.15%	38.65%
11	Large	7.37%	42.88%	4.97%	4.81%	40.22%
12	Large	4.42%	72.30%	2.98%	1.57%	18.98%

Average crane utilization from the Priority/ACO and EDD/ACO approaches are 29.8% and 29.9% respectively. It is not surprisingly that average crane utilization is similar between the two approaches since the makespans are similar.

Changes in how lifts are sorted do not really affect makespan but do affect other performance measures. The earliest due date sorting procedure is effective in limiting tardiness but ineffective in reducing makespan. The priority-based sorting procedure is ineffective in terms of tardiness but can obtain makespans as good as the earliest due date procedure when both are paired with the ACO assignment procedure. Since experimental results from the Priority/ACO approach did not show improvements, EDD/ACO is still considered as the best approach to the problem.

CHAPTER VI

SUMMARY AND CONCLUSIONS

6.1 Summary

In this research, a crane scheduling problem in a shipbuilding environment is studied. The primary objectives of the study are to reduce makespan, or time required to complete a set of lifts, and increase crane utilization. As it turned out, the objectives are complementary in that by achieving one objective, the other is also satisfied. Prior to developing scheduling and assignment approaches, crane operations at the shipyard are modeled and numerous test data sets are applied since actual data is not available from the shipbuilder.

The approaches are first tested using a partial crane track network and list of available cranes in order to reduce the time required to obtain initial results. The first approach, base heuristic, attempts to reflect the current operations at the shipyard, where lifts are sorted by their priority indices and then cranes are assigned greedily based on minimum completion times. The second approach, EDD/Greedy, replaces the priority sorting procedure with an earliest due date procedure. Although there are no improvements in makespan, results from the EDD/Greedy approach showed dramatic improvements in tardiness. The third approach, EDD/ACO, replaces the greedy assignment procedure with an ant colony optimization (ACO) algorithm. The results are promising since average makespan is reduced by approximately 20% while the improvements in tardiness from the EDD/Greedy approach are maintained.

Given that the EDD/ACO approach yielded favorable results, ACO parameter settings are investigated in order to determine if the settings affect the performance of the algorithm. Experiments were conducted to investigate the effects of parameters and their interactions. Based on those experiments, it is concluded that parameter settings do not have a significant statistical effect.

The EDD/ACO approach is further tested on a full crane-rack network that encompasses the entire assembly area and crane list. Experimental results show that the EDD/ACO approach continues to provide better makespan. However, decreases in average crane utilizations are observed; this is likely caused by the increase in the number of cranes while the number of lifts in the problem cases remains constant.

A fourth approach, Priority/ACO, is considered and tested using the full crane track network and list. This approach is essentially the base heuristic introduced in the beginning of the solution approach but with an ACO assignment procedure that replaces the greedy procedure. Results show that the Priority/ACO approach obtains makespans that are slightly larger on average than the EDD/ACO approach but cannot compete in terms of tardiness -- there is a large gap in average tardiness between the two approaches as expected.

6.2 Conclusions

Based on the cases considered, it is clear that the EDD/ACO approach is the best method for the crane scheduling problem in this operational environment. Not only does the EDD/ACO approach reduce makespan of completing lifts, it also limits average and maximum tardiness. Although not a primary objective of the study, limiting or reducing tardiness can be beneficial to any production scheduler or project manager. Since makespan is reduced by using EDD/ACO, average crane utilization is increased, thus satisfying the primary objectives of this research.

The proposed ACO algorithm is not sensitive to parameter settings; therefore, one can set parameters within the range tested and obtain quality results. Given that optimal solutions are unknown for the test problem cases, results from the EDD/ACO approach cannot be benchmarked. Moreover, the effect of problem size on the performance of the proposed method is unknown. Although it is certain that makespan increases as the problem size is increased, it is uncertain whether the performance of the proposed method worsens as problem size is increased.

The author believes that the main problem faced by the shipbuilder is blocking of cranes that are capable of performing priority one lifts. Since priority one lifts determine the makespan in a set of lifts, having cranes available to perform those lifts as quickly as possible is a key factor in the reduction of makespan. For this reason, Gantry should be utilized as much as possible given that it cannot be blocked and does not share the track network with the other cranes.

6.3 Future Research

The work developed in this study provides opportunities for future research efforts. An exact solution approach, that can obtain the optimal solution, would be very useful for determining true performance.

Furthermore, simulation models can be used to simulate the performance of the schedules that are generated by this approach. A simulation model of crane operations can provide visual insights into the problem and can be used to verify the results obtained by the heuristic approaches. The heuristic approaches can be coupled with a simulation model as the basis for a decision support system for the shipbuilder.

The concepts of ant colony optimization can be examined more closely to improve the algorithm, especially how pheromone levels are updated and how ants choose their next destinations. Also, the ACO algorithm could be used in the sorting procedure as well as the assignment procedure. The proposed ACO algorithm is quite basic compared to the ACO algorithms that have been developed for other scheduling and assignment problems

Different approaches and solution methods to the problem exist besides those covered in this research. One could focus how cranes are assigned to lifts and not worry about the sorting of lifts. On the other hand, how lifts are sorted may be the focus of the research. Even more, one could have different model assumptions as listed in this study to see if it has a positive influence on the primary objectives. In any case, the approach presented here is a solid starting point for solving the crane scheduling problem faced by shipbuilders.

REFERENCES CITED

[1] W.C. Ng., Crane scheduling in container yards with inter-crane interference, *European Journal of Operational Research*, 164, 64-78, 2005

[2] E. Kozan, P. Preston, Genetic algorithms to schedule container transfers at multimodal terminals, *International Transactions in Operational Research*, 6, 311-329, 1999.

[3] L. M. Gambardella, M. Mastrolilli, A. E. Rizzoli, M. Zaffalo, An optimization methodology for intermodal terminal management. *Journal of Intelligent Manufacturing*, 12, 521-534, 2001.

[4] A. Lim, B. Rodgrigues, F. Xiao, Y. Zhu. Crane scheduling using Tabu Search, *Proceedings of the 14th IEEE International Conference on Tools with Artificial Intelligence*, 2002.

[5] G. Sahin, R. K. Ahuja, C. Cunha, New approaches for the train dispatching problem, *Transportation Research B*, unpublished, 2005.

[6] K. H. Kim, Y-M. Park, A crane scheduling method for port container terminals, *European Journal of Operational Research*, 156, 752-768, 2004.

[7] E. K. Bish, A multiple-crane-constrained scheduling problem in a container terminal, *European Journal of Operational Research*, 144, 83-107, 2003.

[8] R. I. Peterkofsky, C.F. Daganzo, A branch and bound solution method for the crane scheduling problem, *Transportation Research*, 24, 159-172, 1990.

[9] C. F. Daganzo, Crane productivity and ship delay in ports, *Transportation Research Record*, 1251, 1-9, 1990.

[10] C. F. Daganzo, The crane scheduling problem, *Transportation Research Part B 23 B*, 3, 159-175.

[11] K.Y. Kim, K.H. Kim, A routing algorithm for a straddle carrier to load export containers onto a containership, *International Journal of Production Economics*, 59, 425-433, 1999.

[12] R.J. Gaskins, J.M.A. Tanchoco, Flow path design for automated guided vehicle systems, *International Journal for Production Research*, 25, 667-676, 1987.

[13] X. Cai, C.J. Goh, A.I. Mees, Greedy heuristics for rapid scheduling of trains on a single track, *IIE Transactions*, 30, 481-493, 1998.

[14] H. Matsuo, J.S. Shang, R.S. Sullivan, A crane scheduling problem in a computer integrated manufacturing environment, *Management Science*, 37, 5, 587-605, 1991.

[15] Y. Ge, Y. Yih, Crane scheduling with time windows in circuit board production lines, *International Journal of Production Research*, 33,5,1187-1199, 1995

[16] Rajendran, C., Ziegler, H., 2004. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research* 155, 426-438.

[17] Ying, Kuo-Ching, Liao, Ching-Jong., 2004. An ant colony system for permutation flow-shop sequencing. *Computers & Operations Research* 31, 5, 791-801.

[18] T'kindt, V., Monmarche, N., Tercinet, F., Laugt, D., 2002. An ant colony optimization algorithm to solve a 2-machine bicriteria flowshop problem. *European Journal of Operations Research* 142, 250-257.

[19] Shyu, S.J., Lin, B.M.T., Yin, P.Y., 2004. Application of ant colony optimization for no-wait flowshop scheduling problem to minimize the total completion time. *Computers & Industrial Engineering* 47, 181-193.

[20] Rajendran, C., Ziegler, H., 2005. Two ant-colony algorithms for minimizing total flowtime in permutation flowshops. *Computers & Industrial Engineering* 48, 789-797.

[21] Taillard, E., 1993. Benchmarks for basic scheduling problems. *European Journal of Operations Research* 64, 278-285.

[22] Gutjahr, J.W., Rauner, M.S., 2007. An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Computers & Operations Research* 34, 642-666.

[23] Dowsland, K.A., Thompson, J.M., 2005. Ant colony optimization for the examination scheduling problem. *Journal of the Operational Research Society* 56, 426-438.

[24] Dorigo, M., Stutzle, T., 2004. Ant colony optimization. Cambridge, MA: The MIT Press.

[25] Demirel, N.C., Toksarı, M.D., 2006 Optimization of the quadratic assignment problem using an ant colony algorithm. *Applied Mathematics and Computation* 183, 427-435.

[26] Ramkumar, A.S., Ponnambalam, S.G., Hybrid ant colony system for solving quadratic assignment formulation of machine layout problems. *2006 IEEE Conference on Cybernetics and Intelligent Systems*, 2006, 4017845.

[27] Lee, Z-J, Lee, C-Y, Su, S-F, 2002. An immunity-based ant colony optimization algorithm for solving weapon–target assignment problem. *Applied Soft Computing* 2, 39-47.

APPENDIX A

SAMPLE PROBLEM INSTANCES (30 AND 10 LIFTS)

Lift #	Priority	Duration	Alpha	DueDate
1	1	118.23	0.92	597.62
2	1	87.60	0.42	317.37
3	1	87.76	0.33	267.50
4	1	121.86	0.67	469.00
5	1	101.27	0.96	616.25
6	1	99.17	0.18	198.94
7	2	48.87	0.54	366.26
8	2	37.60	0.29	212.06
9	2	62.71	0.34	259.01
10	3	37.15	0.54	360.19
11	3	36.21	0.89	571.86
12	3	36.65	0.88	566.57
13	3	36.93	0.88	567.85
14	3	37.76	0.28	207.60
15	3	38.02	0.99	633.77
16	3	33.70	0.49	331.32
17	3	38.25	0.13	114.61
18	3	41.46	0.40	280.42
19	4	5.32	0.99	631.00
20	4	8.73	0.90	579.50
21	4	6.43	0.05	37.36
22	4	7.02	0.54	350.78
23	4	6.95	0.34	222.21
24	4	10.67	0.73	470.13
25	4	6.76	0.30	198.37
26	4	6.64	0.85	547.75
27	4	7.36	0.71	456.76
28	4	5.38	0.56	357.83
29	4	10.68	0.52	335.27
30	4	10.00	0.71	454.63

Lift #	Priority	Duration	Alpha	DueDate
1	1	118.23	0.92	244.81
2	1	87.60	0.42	157.65
3	2	48.87	0.54	160.08
4	3	37.15	0.54	154.42
5	3	36.21	0.89	231.20
6	3	36.65	0.88	229.31
7	4	5.32	0.99	252.44
8	4	8.73	0.90	232.30
9	4	6.43	0.05	18.61
10	4	7.02	0.54	142.24

APPENDIX B

SAMPLE CRANE AND LIFTS TEST DATA

CraneTestData
ID, Tonnage, Location

002,200,3
003,200,19
009,39,25
034,97.5,30
035,100,21
036,100,33
038,300,23
039,300,8

LiftsTestData
Lift#, Priority, LatestStart, Location, Tonnage, Duration, DueDate
----
1,1,41,23,500,75,116
2,1,78,1,200,49,127
3,2,29,25,200,39,68
4,3,33,13,100,37,70
5,3,97,13,39,5,102
6,3,64,32,39,4,68
7,4,84,30,39,5,89
8,4,39, 20,100,7,46
9,4,17, 21, 39, 2,19
10,4,17,8,39,8,25