ATHABASCA UNIVERSITY

MASTER OF SCIENCE IN INFORMATION SYSTEMS

SPATIAL ANALYSIS FOR CRANE LIFTING ON CONSTRUCTION SITES

BY

BAHAI SHENODA

An essay submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE in INFORMATION SYSTEMS

PROFESSOR: DR. LARBI ESMAHI

Athabasca, Alberta

May, 2008

© Bahai Shenoda, 2008
TABLE OF CONTENTS

LIST OF FIGURES ................................................................................................................. 4
LIST OF TABLES ...................................................................................................................... 5
ACKNOWLEDGMENT ............................................................................................................... 6
ABSTRACT ............................................................................................................................... 7
INTRODUCTION ...................................................................................................................... 8
CHAPTER I ............................................................................................................................... 11
Heavy Lift Planning in Construction ...................................................................................... 11
1- Overview of Heavy Lifting in construction sites ............................................................... 11
2- Manual Heavy Lift Planning ............................................................................................. 17
3- General System for Heavy Lift Planning .......................................................................... 19
   3.1 What should the computerized heavy lift planning system do? .............................. 19
   3.2 Tools for Heavy Lift Planning ................................................................................... 20
   3.3 Main factors affecting heavy lift planning ............................................................... 22
   3.4 Design of a Heavy Lift Planning System ................................................................ 22
4- Heavy Lift Planning System (HELPS) ............................................................................ 23
   4.1- Scope of the HELPS System ................................................................................... 24
   4.2- Selecting System’s Environment ............................................................................. 25
   4.2- What Can Walkthru Do and what Can’t? ................................................................ 26
4.3- HELPS System Architecture ...................................................................................... 27
   4.3.1 Walkthru's User Interface ................................................................................... 29
   4.3.2 Customizing and Improving Walkthru ............................................................... 31
5- DISCUSSIONS AND CONCLUSION ............................................................................. 35
CHAPTER II ............................................................................................................................ 38
Cooperative Crane Lifts ......................................................................................................... 38
Introduction .............................................................................................................................. 38
1- Issues to be addressed in Cooperative Cranes ............................................................... 39
   1.1 Degrees Of Freedom (DOFs) in Cooperative Cranes ................................................ 39
   1.2 Movement of Two Cooperative Cranes .................................................................... 41
   1.3 Cooperation between Cranes (Load Line’s Angle) .................................................. 42
   1.4 Collision among Cooperative Crane Manipulator System and Environment ........ 43
   1.5 What is the Best Lifting Path? .................................................................................. 43
   1.6 Modeling Conditions ............................................................................................... 44
2- SYSTEM DESIGN ............................................................................................................. 46
   2.1 Design Strategy ......................................................................................................... 46
   2.2 Design Tasks ............................................................................................................. 48
   2.3 Implementation Tasks ............................................................................................... 52
3- CONCLUSION AND DISCUSSIONS ............................................................................. 53
CHAPTER III .......................................................................................................................... 55
Automated Path Planning for Mobile Cranes ......................................................................... 55
1- Introduction ....................................................................................................................... 55
2- Path Planning in Robotic and Crane Manipulators .......................................................... 56
   2.1 Space Representation ............................................................................................... 57
   2.2 Finding Path ............................................................................................................. 58
   2.3 Representing Workspace using Configuration Space (C-space) ............................ 58
3- Data Flow Diagram (DFD) For Generic Automated Path Planning System........... 60
4. Automated Lift Planning System for Single Lifting Crane .................................. 61
   4.1 Development of Configuration Space (C-space) for Cranes ............................ 62
   4.2 Path Searching Criteria .................................................................................. 64
   4.3- System Implementation .............................................................................. 65
5. Automated Lift Planning Systems for Two Cooperative Lifting Cranes ............... 68
   5.1 Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search ... 68
   5.2 Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm... 74
6. CONCLUSION ................................................................................................. 80
CHAPTER IV ....................................................................................................... 85
Tools and Techniques for Automated Path Planning ............................................... 85
  Introduction ....................................................................................................... 85
  1. Path Planning Tools and Techniques................................................................. 86
  1.1 Graphical Tools.............................................................................................. 86
  1.2 Computational Tools..................................................................................... 86
  2. Ant Colony Optimization (ACO) Overview .................................................... 87
  2.1 What is ACO? ............................................................................................... 87
  2.2 Example using ACO algorithm in Artificial Ants ........................................... 88
3. Architecture of the Suggested Model ................................................................. 90
   3.1 Geometric Representation of the proposed Crane Lifting System ............... 90
   3.2 Degrees of Freedom of the Lifting Crane .................................................... 94
   3.3 Collision between Crane and Environment ................................................ 95
   3.4 The Ant System ............................................................................................ 95
   3.4.1 DFD for suggested lift planning system using ACO in C-space .............. 97
   3.5 Data Flow Diagram (DFD) For the Ant System ......................................... 103
   3.6 Principle of Grid Based ACO algorithm................................................... 103
4. DISCUSSIONS AND CONCLUSION ................................................................ 106
CHAPTER V ......................................................................................................... 109
CONCLUSION AND RECOMMENDATIONS ....................................................... 109
APPENDIX A ........................................................................................................ 116
  Hill Climbing .................................................................................................... 116
  A* Search ......................................................................................................... 116
Keywords ............................................................................................................. 118
REFERENCES ...................................................................................................... 119
## LIST OF FIGURES

**Chapter I:**
- Fig-1 Manual Locating of Cranes and Planning for Lifting Path 14
- Fig-2 Overall Logic for Manual Heavy Lift Planning 18
- Fig-3 HELPS System Architecture 28
- Fig-4 Layout of Monitoring Window 33
- Fig-5 Motion Control For Long Loads 34

**Chapter II:**
- Fig-1 Degrees of Freedom for Lattice Boom Crane 41
- Fig-2 Simplified geometry of two cooperative cranes 45
- Fig-3 Benchmark Problems 48
- Fig-4 Position representation in configuration space and real space 48

**Chapter III:**
- Fig-1 Two DOF manipulator in 2D space 59
- Fig-2 Path Configurations in C-space for a two-DOF manipulator 60
- Fig-3 DFD for Automated Crane Lift Planning using C-space 61
- Fig-4 C-space diagram for a 3-DOF crane 63
- Fig-5 Visualization Platform (AutoCAD) 73
- Fig-6-a GA Search Program Flow (Coordination Phase) 76
- Fig-6-b GA Search Program Flow (Coordination and Collision Phase) 77
- Fig-7 Path String with 15 Intermediate Locations 79

**Chapter IV:**
- Fig-1 ACO Example with Artificial Ant 90
Fig-2 tessellated representation of an object 92
Fig-3 Structure of the Grid-Based Ant Colony Algorithm 96
Fig-4 DFD for suggested Crane Lift Planning using Ant Colony Optimization Algorithm in C-space 98
Fig-5 Flow Chart of the Ant Colony Algorithm 101
Fig-6 DFD for suggested Crane Lift Planning using Ant Colony Optimization Algorithm in Data Grid (Tessellated) Model 103
Fig-7 Example Ant Colony Algorithm in a 2D Grid 105

LIST OF TABLES

Chapter III: Table-1 Limiting Values of Movement for Different Arms of Crane Manipulator 79
ACKNOWLEDGMENT

I want to express my gratitude to all academic and staff at Athabasca University for their support through the McIS program that I am taking where I gained intensive academic and practical experiences in many different areas in Information Systems. This experience have significantly improved my technical skills and allowed me to produce this paper.

I gratefully acknowledge the support from Prof. Larbi Esmahi, who is the supervisor for this essay. His patience, generous support, advice, guidance and encouragements during my work in this paper, helped me to make it happen.
ABSTRACT

Efficient utilization of working space in construction sites is one of the main challenges that face construction projects, especially when lifting cranes are necessarily needed to move construction materials and equipments form one point to another within the construction workspace. Failure to making a good use of construction sites and resolving workspace conflicts might cause crucial problems during construction which results in a delay of the construction projects, reduce productivity, and/or cause accidents that threaten labor’s safety (Guo 2002). Building a crane lifting model became an urgent demand that must be fulfilled by construction engineers to model the workspace requirements for identifying and mitigating spatial conflicts related to crane operations, reasoning spatio-temporal behaviors of cranes and coordinating them within a dynamic construction environment across time, and optimally allocating workspace resources.

This paper is a literature review discussion for the work that has been done in the past as an attempt to address the different issues related to the use of crane lifting in construction sites.
INTRODUCTION

The need for lifting cranes in construction sites arises when dealing with heavy equipments, prefabricated construction components, and materials which are transported to the construction site and installed using specialized lifting methods and equipment. Also heavy lifts are considered in construction and maintenance of industrial facilities where installation and replacement of heavy plant machines and equipment is mainly required. In this paper a heavy lift is considered when its weight ranges between 30 and 800 tons, which requires detailed planning to be performed and needs drawings to explain the lift plan.

Any mistake in designing and/or carrying out the lifting plans may result in accidents, injuries, loss of life, damage of critical equipment, schedule delays and unpredictable construction costs. Therefore very careful planning is required for every heavy lift to identify and limit the involved risks. It has been found practically that it is a very complex task to develop a reliable heavy lift plan, since it involves not only the lift and site of operation but also all other supporting tasks of the lift such as transporting of load, assembly of the crane, position and location of the crane, etc. It may require months of manual planning to reach the level of confidence needed to perform some lifts. Typically Lift planning is performed in an iterative/trial-and-error way in order to reach a plan that satisfies all requirements.

However, due to the change of information available in the planning stage as the project progresses, it is important that lifting planners consider the information change and
incorporate these changes in their plans, which makes it a tedious and expensive process in order to prepare an accurate life plan using manual methods. In this case repeated draft drawings or constructed physical models need to be performed by planners to be confident that a chosen plan will work, where there is no easy way to examine and document alternative crane selections and lift plans. Additionally, developing a workable plan that does not investigate the tolerances of the plan is a common practice followed by planners. This means that any changes in the lifting parameters will require re-planning under a time constraint which increases the potential of errors and unsafe lift plans or delaying the lift activity.

The advances in computing technology and its associated visual software tools makes it easy for users to get a real time interaction with computer-generated 3D graphic images. Since heavy lift planning mainly relies on visualization, the productivity and reliability of planning heavy lifts will be improved by utilizing the computer power which will allow lifting planners to test the feasibility and tolerances of different lifting paths in no time by means of simulation of the critical parameters of the lift in a virtual environment.

The objective of this essay is to present and discuss the work that has been done previously in this area using computer tools to:

1- Develop the requirements for different planning tools and
2- Develop a prototype for a crane lift planning system using similar or different development tools.
The discussions of the previous works presented in this paper include:

1- Development of Computerized HEavy Lift Planning System for Crane Lifts (HEIPS) by “Koshy Varghese, Parmanand Dharwadkar, John Wolfhope & James T. O’Connor” 1997

2- Automated Path Planning for Mobile Crane Lifts by “H. Raghunatha Reddy & Koshy Varghese”

3- Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search by “PL. Sivakumar; Koshy Varghese; and N. Ramesh Babu”

4- Collision Free Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm by “M. S. Ajmal Deen Ali; N. Ramesh Babu; and Koshy Varghese”

In this essay three (3) chapters have been dedicated for discussing these four (4) systems listed above where summary and detailed discussions were presented. Another chapter (Chapter 4) is developed as my contribution to this area of research, where a proposal of an architecture for a crane lifting model that utilizes one of the most sophisticated path optimization algorithms “Ant Colony Optimization (ACO) algorithm” in order to reach the shortest lifting path. The last chapter (chapter 5) presents discussions and conclusion along with a section about future work in this area that comes at the end of chapter 5.
CHAPTER I

Heavy Lift Planning in Construction

1- Overview of Heavy Lifting in construction sites

Because of the high cost and the complexity of heavy lifting process, we need to verify whether the construction site and the working conditions allow executing this task or not, in advance. This makes it necessary to perform planning for heavy lifting in construction sites and find the best lifting paths if they are available. The following eight (8) factors (criterions) are identified as the main factors to determine the feasibility of performing heavy crane lifts on construction sites:

1. **Is Crane available?** Finding a crane with the required lifting capacity is very important although it is non-technical factor but it is very important factor since months in advance had to be considered to make reservations for such cranes. Therefore before planning the technical details of the lift, planners must know which cranes are available to carry out the lift.

2. **Is construction site accessible by crane(s) and load(s)?** It is clear that the huge size heavy lifting cranes and the large/heavy loads components have to reach the construction site, so planning the access to the site must take place. Obstacles such as overpasses and power lines requires co-operation between the state transport authorities with assistance from the manufacturer, special transport contractors, and the planner, where both planner
and transport contractor are held responsible for ensuring that the crane and load can reach the site, in remote areas.

3. **Is lifting area accessible by crane(s) and load(s)?** Accessibility of lifting area determines whether it is possible for cranes and loads to travel between the site’s entrance or its storage locations and the location where the lift operations will take place. This requires planners to consider:

   1. Size of vehicles
   2. Steering parameters
   3. Layout of the site
   4. Location of overhead and underground elements
   5. The construction schedule in order to determine
      
      a. A suitable location to assemble the crane (if necessary) and
      b. A path to move the crane and load to the lift area.

Where any of these five (5) parameters can be changed to make sure that an assembly area is available and the access is possible.

4. **Is location ready to perform lifting?** Crane and loads must be located in areas where pick and place locations can be easily reached within the limits of the crane’s lifting and movement. Variables such as the layout of the lift area, sequence of construction, crane and load characteristics, determine where to place the crane. Planners can modify any of these parameters to allow suitable placement.
The diagram in figure-1 below shows how to manually determine the possible crane placement locations, and find the convenient lifting paths. In this case an engineering plan of the construction site including the lift area must be prepared identifying the loads to be lifted, pick and place locations. The diagram contains several arcs with different radius between the pick and place locations, and the arc overlaps define the region where the lifting crane can reach both pick and place locations, operating from one point. The possible lifting path is selected based on avoiding all different obstacles surrounding the selected location.

Luffing motion of the boom during the lift should also be avoided, and then the safest and most direct path from a given location is selected as a good path candidate. Later if the capacity, clearance, or ground support tests are found exceeding the safe limits, then the selected location and path might be considered unfeasible.
If multiple loads need to be lifted from the same lift area, crane has to be located in a central position to make all the lifts without moving or repositioning the crane.

Also in other scenarios, there could be lifts that might require moving the crane while carrying the load. In such cases, details of the pick location, moving path, and place location have to be carefully planned.

5. **Are clearances adequate along lifting path?** Clearances that planners should consider when they plan for crane lifting are: clearance from obstacles along the lifting path, clearance between the crane’s boom and load, and clearance between crane and load to the site obstacles. These clearances should be identified, checked and tested in order to
ensure that they satisfy the clearance criterion. If they don’t satisfy then changing parameters such as: crane, lift path, placement location, construction sequence, or module size, can be done by the planner reach a satisfying scenario.

Heavy lift planning routinely requires evaluating the clearance between the boom and the load. When the boom makes the largest angle to the ground, it reaches worst-case clearance location position along the lift path.

It has been realized that lifting planners deal with tedious and very time consuming job since they always have to manually (mathematically or with graphical tools) generate different lifting paths, reposition the crane, remove obstacles, or select new crane, re-evaluating the clearance, each time the clearance criterion is not satisfied.

6. **Are loads within crane capacity?** Determining whether the crane can move the load along the lifting path without exceeding the crane’s capacity is very essential, otherwise another crane, placement location, lift path, or module size must to be selected. Here are the steps that planners used to follow in order to perform planning for lifting paths:

- Several locations are selected by the planner to calculate capacities as s/he think are necessary to identify the highest radius required along the lift path.
- Worst-case locations are isolated, and the crane capacities are searched in the manufacturer capacity tables, where the capacity of the crane are specified for a given counterweight configuration given a boom length and radius.
- Interpolation is used to find the capacity for intermediate radius that is not specified in the tables.
- Radius and capacity (based on the used boom length) are calculated for each location.

It might look easy to calculate the capacity for a given condition; however the whole planning process is repetitive, where the process of repeatedly calculating capacities makes it very time consuming and may lead to human errors.

7. **Can crane be supported during lift?** Ground support for crane lifting includes:

- Ensuring that there are no underground objects such as drains etc.
- Determining the suitable type of foundation based on the soil physical and mechanical characteristics, of the lifting site where maximum forces transmitted during the lift are calculated.
- Designing the crane foundations based on the lifting loads and the soil type which might vary from basic wooden timbers distributing the load to piles under the crane’s outriggers, to solid concrete block foundations for very heavy lifts over poor soil.
- Crane foundations are very important and often require specialists to do the design. If ground support in a selected location is not good enough and it is not possible to build a suitable foundation, planners could look for another location for the crane, resize the load, or select another crane to handle the inadequate ground support scenario.
8. *Can crane be removed from lift area?* This factor defines whether the crane is free enough in the lifting area and there is enough space to disassemble the crane and remove it from the lift area, or not.

### 2- Manual Heavy Lift Planning

The eight (8) lift evaluation criteria discussed above, can only help to determine whether the lift is possible, but they do not determine the optimal lifting operation.

Fig-2 below shows the logic for manually developing a crane lifting plan which involves the entire criterions discussed earlier. The diagram clearly shows the iterative approach that planners have to go through for developing a set of workable plans to execute the lift. If a criterion is satisfied in a given scenario, the next criterion is tested; otherwise planners change the scenario by varying appropriate parameters and re-evaluate the new scenario for criteria that are affected by the change and so on...
Start
Select Initial Scenario
Is crane available?
Yes
Can crane/load access project site?
No
Next Potential Scenario
Select Initial Scenario
Yes
Can crane/load access lift area?
Yes
Can crane be located to pick and place loads?
Yes
Is clearance adequate along lift path?
No
Are loads within crane capacity?
Yes
Can crane be supported at location?
Yes
Can crane be removed?
Yes
Record Scenario & Lift Plan
No
Try Other Scenarios
Stop
Try Another Location
Try Other Locations
Record Scenario & Lift Plan
Try Other Scenarios
Try Another Location
Stop

Fig-2 Overall Logic for Manual Heavy Lift Planning [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 97]
3- General System for Heavy Lift Planning

What is really need to be achieved, is the development of a heavy lifting system utilizing the computing power and automates the iterative and tedious job that manual planners used to do. This would protect the process from human errors and save time and money since it does not rely on the planner’s long years of experience.

3.1 What should the computerized heavy lift planning system do?

A heavy lift planning system should offer custom engineered heavy lifting solutions to planners, where safety and flexibility are very important. It should provide planners with some of the key decision variables such as crane selection and crane locations. It also should provide lifting planners with simulation tool that simulates the actual lifting and placing operations while monitoring the parameters critical to the success of the whole lifting process. A heavy planning system would basically include the following features:

- Environment to simulate and visualize the lifting scenarios.
- Allow controlling the natural degrees of freedom of equipment.
- Allow Integration of both graphic images with non-graphic data.
- Ability to detect physical interferences to the crane and load during the lift.
- Provide an easy modification of planning variables to perform what-if analysis.
- Allow the save or archive of lift cases.
- User friendly and easy to maintain.
3.2 Tools for Heavy Lift Planning

1- **Drawings and templates**: [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997] available and inexpensive tools but require lots of time for manual work which results in reduction of productivity

2- **Scaled solid models**: [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997] is excellent visualization tool, but it is expensive to build, difficult to modify and hard to transport

3- **Computer-aided design (CAD)**: [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997] not widely used by crane lift planners, it saves time in redrawing lift configurations; but it does not provide any help in the actual analysis.

**Disadvantages of these tools** [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

1- Limited visualization capabilities

2- Only the spatial attributes of the problem are presented, where non-spatial information such as crane capacity and construction sequence have to be repeatedly obtained from other sources

3- It is difficult to modify variables in order to carry out what-if analysis.

Therefore there is a real need to develop more efficient lift planning tools to improve productivity and accuracy of the planning activity. The obvious approach to build such efficient tools is to utilize the computing power to perform these time-consuming lift planning tasks.
Also there are construction companies that are specialized in the development of computerized lift planning tools that are limited to the internal use within the company such as:

1- **Computer-Aided Rigging Software**: developed in *Brown & Root* by a group consisting of engineering professionals and software developers who are developing a system called *Computer-Aided Rigging* (CAR). This software is developed using Microstation Development Language (MDL), with a database of cranes and the corresponding loading charts. This system, allows users to enter the dimensions of the load and radius of the lift, then it will select a set of cranes that can be used for the job along with a drawing of the lift configuration. It also provides a 3D view of the lift configuration and a playback animation. However the software does not provide a real time control of the crane, also the graphic image of the crane is not dynamically linked to the crane capacity charts. These limitations does not allow for an easy interactive simulation of the lift plan. [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

2- **Automated Lift Planning System (ALPS)**: developed in *Bechtel* by a group of developers who are using a visualization environment to develop a system called *Automated Lift Planning System* (ALPS). The software allows designing the assembly process for a given load, selecting the suitable crane, and depicting the lift environment for making decisions on the lift parameters. [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]
• **Multiple heavy lift Planning System**: [S. Reddy D; Varghese K.; Srinivasan N., 2007]
  this system has been developed in AutoCAD 2007 environment that has been
  customized using VBA (Visual basic for Applications) API to create the multiple lift
  planning environments. The system maintains two (2) databases: one is graphic
  database to store crane models; and the second database is non-graphic database that
  stores the load charts. The system provides a friendly user interface with high
  performance, and offers many features such as crane selection, crane locations. The
  system allows 3D simulation of multiple lifts and 4D representation of the
  construction site.

• Finally the **HELPS (Heavy Left Planning System)** system that has been developed
  and introduced by Koshy Varghese, Parmanand Dharwadkar, John Wolfhope &
  James T. O’Connor [Varaghese K.; Dharwadkar P.; Wolfhope J.; O’Connor J. T.,
  1997]. This system will be discussed in detail in section (4) below.

### 3.3 Main factors affecting heavy lift planning

The main factors affecting heavy lift planning are the eight (8) factors discussed earlier
at the beginning of this chapter in the **Overview** section.

### 3.4 Design of a Heavy Lift Planning System

Typically a heavy planning system utilizes computer and information systems technology
to perform the job. Such a system would consist of three (3) main components as follow:

1- **Visualization Component**: enables to generate graphical models (in 2D or 3D or
  more), representing the working environment, where the geometry of the construction
site, obstacles, loads and cranes are modeled. Tools such as AutoCAD and Microstation are open environments having their own programming language, and they generate output graphical files.

2- **Translation Component**: enables to convert the output files of the visualization tool which are graphical files in visual format into a readable form (usually text format) that can be used by the simulation tool.

3- **Simulation Component**: provides simulation tasks for the load to be lifted and traveled according to different scenarios between pick and place locations and for different types of loads. The translated graphical mode, the construction scheduling file, and the load definition files are typical inputs to the simulation process. The simulation tool should provide a convenient interface to select the crane and manipulate its Degree Of Freedom (DOF) and virtually operate the crane on the graphical model of the site, where planners can visualize the lift path between the pick and place locations. Typically simulation process results in producing an optimized lifting paths in respect to each lifting scenario.

These three (3) components can be integrated into three (3) programming modules **Input Module, processing Module and Output Module**, that present a heavy planning system, as discussed in details in the following section.

**4- Heavy Left Planning System ( HELPS )**

This section provides detailed description on the HELPS system as presented in

[Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Conner J. T., 1997]
4.1- Scope of the HELPS System

The HELPS tool handles only three (3) factors out of the eight (8) factors (criteria) discussed in the Current Lifting practice section discussed earlier. These three (3) criteria are:

1- Location where the load to be lifted.
2- Lift path clearances based on site geometry, obstacles, and DOF of the crane equipment.
3- Capacity of crane equipment during lift.

Because of their interdependency and their commonly share in all heavy lifts, these three criteria are selected in this design.

To provide lifting planners with simulation tool that simulates the actual lift while monitoring the parameters critical to the success of the lift, a heavy planning system would basically include the following features [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]:

- Environment to visualize the lift situation.
- Allow controlling the natural degrees of freedom of the lifting crane.
- Allow Integration of both graphic images with non-graphic data.
- Ability to detect physical interferences to the crane and load during the lift.
- Provide an easy modification of planning variables to perform what-if analysis.
- Allow the save or archive of lift cases.
- User friendly and easy to maintain.
Selecting a suitable environment, assess its features, and conceptually determining how the required features could be incorporated are the first step that need to be taken toward developing the system.

4.2- Selecting System’s Environment

To develop the system, the option was to select one approach of two. The first approach suggests utilizing the CAD environment, and the second approach was to utilize a convenient visualization environment.

1- CAD environments: [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997] such as AutoCAD and Microstation are open environments having their own programming language. Using the basic tools already available within the environment it is possible to build customized applications, which provides the application developer with good deal of flexibility, but because of the lack of simulation capabilities in the CAD versions available in the market. Also because this option was explored by the Brown & Root team, the decision was taken for using a visualization environment approach.

Investigating several visualization environments such as: Cimstation, Walkthru, Alias, Grasp, Review, Voyager, and Design Review (this effort is documented by Alciatore et al), revealed the selection of the Walkthru as the implementation environment for the
suggested system because of the following three (3) reasons [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]:

1- Walkthru was purposely developed for the construction industry therefore it was equipped with the basic functions to analyze construction situations

2- Open Source developers had added extra feature that could be used to program additional functions to Walkthru.

3- The availability of hardware platform in terms of a Silicon Graphics Iris workstation to run Walkthru.

4.2- What Can Walkthru Do and what Can’t?

Walkthru is developed and supported by Bechtel, and currently (in 1997 the publishing year of the article) is supported by Jacobus Technology. Walkthru is a real-time 3D visualization system [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

Walkthru simulates a walking through a 3D computer model, and can do the following:

1- Control user’s position

2- Provide controlled movement and positioning of objects within the 3D model

3- Define the natural degrees of freedom and master-slave relationships required to simulate the realistic motion of equipment (because of the second feature)

4- Record a sequence of moves and replaying the recorded moves when needed

5- Detect interference between objects using its on-line function
Therefore Walkthru had major features that are required to perform a heavy lift planning

**Walkthru can’t do:**

1- Manual customization of the definition files for each model.

2- No support for data integration where Walkthru can neither link a crane image to a load chart nor perform numerical computations based on the graphic representation.

3- Visualization and clearance check limitation that does not allow testing the clearance from other objects.

To remedy Walkthru’s limitations, there are two suggestions [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]:

1- To automatically manipulate the definition files based on the users request, an external shell to be developed

2- To link the graphic image to data, monitor the data, and make long loads picked at one end move realistically, new routines are to be added to Walkthru.

Open source developers added a procedure to Walkthru called *user motion function* which is left empty for customization. Then the data integration, monitoring, and load behavior routines were developed as a part of this procedure and then compiled and combined with Walkthru to produce a new executable called *CraneWalk*. [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

**4.3- HELPS System Architecture**
Figure 3 below shows the architecture of the suggested HELPS system where the C programming language is utilized to develop the HELPS system. Using a Silicon Graphics Iris workstation a prototype of the HELPS system was developed and installed on it. The user interacts with the system through a user shell interface which controls:

1- The creation of the final model file

2- The creation of the definition files

3- The access to the modified version of Walkthru which is called CraneWalk

User’s interaction with the shell allows selecting the necessary options and then run the simulation using CraneWalk. CAD applications are used to create the model files such as:

1- The site layout and geometry

2- Crane and Load geometry and characteristics
These models are stored in the CAD format (IGDS in this case). Once a site, crane, and load are specified, a final Walkthru model file is created by merging the geometries from each of the files using one of Walkthru’s tools called WALKIGDS.

An off-line creation of the site model file (object name file), the construction scheduling file, and the load definition files, is done, where:

- The object name file is a typical Walkthru file from which names are assigned to each layer of objects.
- The file that relates each layer in the site model file to the time when it is installed, is the construction schedule file
- The load definition file assigns a default weight for each load in the load model file.

The system takes care of automatically creating the other Walkthru definition files such as the hierarchy definition file and user-defined object motion menu files.

The following subsections provide detailed discussions of the shell and the programming routines that are added to Walkthru.

4.3.1 Walkthru’s User Interface

The user shell is the user’s interface utility with Walkthru. It was developed to perform the following tasks:

1. Standardize crane components
2. Design and implement programs for:
   - Providing a user interface
   - Creating model files
Creating other definition/control files using Walkthru format

4.3.1.1 Standardize crane components

It means using constant numbers to reference the common components of different crane models, which remain constant relative to other components; therefore it is an important step in shell development.

Motion menus and object hierarchies are generated in a standard Walkthru format, for any crane.

4.3.1.2 Programs’ Design and implementation

The user shell program was developed using a structured approach in three phases:

1- User-interface routines
2- Model-creation routines
3- Definition/control files creation routines.

A typical loop is used to implement the user interface in order to prompt the user with a basic structured menu-driven interface. This interface is used to:

- Select/edit a site, crane, load(s)
- Prepare lifting schedule to be used for a planning session
- Define the default weight of the load to be lifted during a planning session

How does merging of the site, load, and crane graphics files happen?

These files are merged together using WALKIGDS tool which produces the final model file. In this operation a number of layers (objects) in the site model are established by reading the site object name file and then uses WALKIGDS to merge the crane objects.
and load objects to layers higher than the maximum defined in the site file [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997].

4.3.2 Customizing and Improving Walkthru

Customization and enhancements in Walkthru are applied to the system by adding extra functionality through the development of programs added to the user motion function [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997] in Walkthru.

What is the user motion function?

1- It is a function that is called frequently when the standard user motion menu feature is selected within the Walkthru environment

2- It was created as a template to be customized by users (programmers) to enhance their Walkthru behavior.

This means that to enhance Walkthru, a programmer (Walkthru user) writes the required code as a component of user motion function and compile it to build a new version of Walkthru that contain the added functionality.

In HELPS system the user motion function was used to add the following three (3) basic functions:

4.3.2.1 Data integration [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

This module ensures that the graphic image on the screen is correctly mapped to the non-graphic information such as crane capacities and load weights.
4.3.2.2 Data monitoring [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

Actually Walkthru is not able to monitor and display the non-graphic information, therefore data-monitoring functionality is necessary to provide an interactive planning of the lift. In HELPS system, monitoring was provided using a windows form that displays all the lifting parameters, as shown in Figure-4 below, data-monitoring consists of the following nine (9) divisions:

1- Division one displays the make and the model of the crane.

2- The information about the location of the crane, crane’s orientation in global coordinates, and the orientation of the cab relative to the crane base, are all displayed in division two.

3- The third division displays characteristics of the load that will be attached or currently attached to the crane.

4- The fourth division displays the current boom length.

5- The fifth division in the window shows the length of the main hook line and current lift radius.

6- In the sixth division, the following information are displayed:
   a. The capacity of the crane in its current state
   b. The weight of the load being lifted at the boom
   c. The percent capacity used.
7- Divisions 7, 8, and 9 are dedicated to display the characteristics of a jib (crane’s arm) boom.

<table>
<thead>
<tr>
<th>Crane Model</th>
<th>Location</th>
<th>Crane Angle</th>
<th>Cab Angle</th>
<th>Active Load</th>
<th>Distance To Load</th>
<th>Angle To Load</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>BOOM</td>
<td>JIB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>Lift Radius</td>
<td>Hook Length</td>
<td>Capacity</td>
<td>Load Weight</td>
<td>% Capacity</td>
</tr>
<tr>
<td></td>
<td>+</td>
<td>4</td>
<td>-</td>
<td>7</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig-4 Layout of Monitoring Window [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

4.3.2.3 Load motion control [Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

Walkthru treats loads which are picked from their edges or any point that is not at its center of gravity (CoG) as if they were picked from its CoG which is not realistic as shown in figure-5 below. Therefore a load motion control module is necessary to be added to Walkthru to ensure a real motion of a long load that has been picked and lifted at one end only.
In motion control the load is rotated by the required angle to keep the far end of the load on the original plane until the load reaches its vertical position. An angle $\theta$ as shown in Fig-5 above represents the angle between position (b) and position (c). The control function calculates the angle using the following formula, in each time the user motion function is called:

$$\theta = \arcsin \left( \frac{\Delta H}{L} \right)$$  

[Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

Assuming that the height $H_1$ is the initial length of the line, and while the hook is moving up using the Walkthru’s motion menu, the custom program within the user motion function triggers the load motion control function which keeps track of the current line length $H_2$ and calculates the real angle of the load for the current hook.
position. This angle is passed as an argument to the WALK_ARGS data structure, resulting in a real display of the load’s position.

5- DISCUSSIONS AND CONCLUSION

It is not uncommon in construction sites to utilize heavy lifts especially for large construction projects where planning is required to perform the heavy lifts safely and efficiently. The lack of heavy lift planning tools confines the planners’ productivity and opens the doors for potential errors in the development of lifting plans. Utilizing the computer power and its appropriate tools will enhance the productivity and reliability of the planning activity. A key element in planning heavy lifts is visualization and simulation, which promotes the planner’s creativity and enables him/her to generate alternative solutions and efficiently communicates all parties of the heavy lifting activity.

Efficient computer tools should be available to support visualization which in turn benefits the planner by providing easy tool to discover the feasibility of the alternatives. Walkthru was selected as a suitable environment for developing the HELPS system for planning heavy lifts because it provides:

1- Visualization features
2- Basic analysis tools
3- Possibility of future tools.
However, it was not easy to customize the Walkthru’s version in order to produce HELP system. Therefore the version of HELP system that has been presented here has the following limitations:

1. **Altering Site Geometry**: since CAD functions are not incorporated inside Walkthru, the site geometry which can only be changed in CAD environment which is not convenient to conduct analysis dealing with a site with dynamic geometry changes. Integrating CAD feature within Walkthru or using visualization software with geometry creation features might be the solution.

2. **Depth issue in 3D display**: although 3D view is a good visualization feature, but practically it is not good for analysis because the real depth is not clear. This requires using multiple views to reach accurate estimate of the depth or obtains assistance from the coordinate outputs to position the hook/load over a specific point, which makes it a tiresome process to develop a detailed plan. Stereoscopic screens or virtual reality modeling might be the right solution.

3. **Restricted mouse usage**: [Varaghese K.; Dharwadkar P.; Wolfhope J.; O’Connor J. T., 1997] using the mouse during simulation to control the Degrees Of Freedom of the crane is not user friendly. A joystick-based interface would provide more user-friendly and realistic control of the simulation.

Furthermore there are extra suggestions to enhance HELP system by providing an optimization feature of the lifting path. Instead of leaving it to the user who has to determine the best lift path by simulating various lift paths and then selecting a suitable path. This automation can be implemented in two ways.
1. The system would automatically generate the possible good lifting paths, and the planner would select the optimal path based on his/her experience.

2. The system would automatically generate the possible good lifting paths (as in 1 above) and also would select the best path based on programmed selection criteria.

[Varaghese K.; Dharwadkar P.; Wolfhope J.; O'Connor J. T., 1997]

Future work in this product might focus on the following areas:

1. Enhancing HELPS to overcome its limitations discussed in the previous section

2. Extending the system to perform planning for multiple crane lifts

3. Investigating the possibility of adding features to the system in order to archive lift cases to generate case-based ideas for planning lifts on future projects

4. Exploring implementing the system in a virtual reality based environment with networked planners interacting within the environment.
CHAPTER II

Cooperative Crane Lifts

Introduction

This chapter discusses the use of cooperative cranes for lifting in construction sites, where the cost effectiveness of heavy lift operations can be improved. For cooperative manipulators, path planning is very different from that in of a single manipulator. When the load to be lifted is very heavy or too large for a single manipulator to carry it, options such as cooperative use of multiple medium capacity manipulators, or using specially assembled equipment such as jacking systems might be a good choice.

However, the widespread use of cooperative crane lifts is growing although the development complexity of reaching a reliable system for planning heavy lifts, due to the advances of computing power and the availability of computer-aided tools that help building efficient and reliable planning systems.

As previous studies focused on planning lifting path, this chapter presents a vital task of the lift planning process, focusing on the same target and taking one more step further. This chapter discusses the issues associated with two cooperative cranes and whether their movement is synchronous or asynchronous and how can a cooperative manipulator system be built presenting the work done to develop a computer aided path planner for two cooperative lifting cranes
1- Issues to be addressed in Cooperative Cranes

In this discussion there are several problem-specific aspects or scenarios need to be formulated in order to develop a path planner for cooperative crane lifts. Such problem-specific aspects like:

1. DOF in two cooperative manipulators
2. Nature of movement (i.e. synchronous vs. asynchronous)
3. Slope of the load line (hoisting rope)
4. Criteria to identify a best lift path need to be considered.

In addition, to simplify the problem formulation which minimizes the complexity of path planning algorithms, appropriate assumptions need to be made based on the constraints on computational time.

1.1 Degrees Of Freedom (DOFs) in Cooperative Cranes

In general, a mobile lifting crane has eight (8) degrees of freedom 8-DOF, which includes:

a) A single DOF for the crane operations (swinging, hoisting, luffing, telescoping)
b) Three (3) DOFs to account for moving and rotating the crane base
c) When considering the load’s rotation around the vertical axis additions DOF is added for lifting the load
However a real world lattice boom crane Fig.1 below has the following six (6) DOF categories:

1- Location and orientation of the base in the site plan, i.e., \( C_x, C_y \) and \( R_b \) respectively

2- Swinging of the cab \([\Phi] \)

3- Luffing of the boom \([\theta] \)

4- Telescoping action of the boom \([L_b]\)

5- Hoisting \([h_L]\)

6- Slope of the load line about the \( x \) and \( y \) axes \([h_{\theta x}, h_{\theta y}]\).

To represent a unique configuration of the crane, a configuration set is expressed as follow: \([C_x, C_y, R_b, \Phi, \theta, b_L, h_L, h_{\theta x}, h_{\theta y}]\).

The discussions presented in this chapter is limited to fixed base lattice boom cranes (meaning that \( C_x, C_y, R_b \) are omitted). In addition, both base and cab are modeled as a single entity (meaning that \( b_L \) is not considered).

Also configuration variables \( h_{\theta x}, h_{\theta y} \), which represent the slope of the load line are not modeled as DOF because they are kept within the permissible limits in order to minimize the additional load transferred to each crane [Shapiro et al. 1991]. Hence, the configuration set of a single crane is simplified into \([\Phi, \theta, h_L]\).

Since this chapter focuses on path planning for cooperative crane lifts, therefore, the corresponding configuration set using two cranes is expressed as \([[[\Phi, \theta, h_L]C1] ], \{ [\Phi, \theta], [h_L]C2]\).
$\Theta, h_L [C2]$, where $C1$ and $C2$ represent the two cooperative cranes crane 1 and crane 2, respectively.

This configuration is referred to as crane $2 \times 3$ manipulator, because it consists of two cranes, each having three DOF.

**Fig-1 Degrees of Freedom for Lattice Boom Crane** [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]

### 1.2 Movement of Two Cooperative Cranes

The movement of cooperative cranes can be either synchronous or asynchronous, during the lift. When identical movement of cooperative cranes between pick and place locations happens, this movement would be synchronous. The following example is a synchronous movement for a hoisting operation from two to five units: Two successive configurations $\{[90,45,2]C1, [90,45,2]C2\}$ and $\{[90,45,5]C1 ,[90,45,5]C2\}$. 
When the cooperative cranes do not move identically, asynchronous movement is happening. The following example of two successive configuration sets \{[140,55,2]C1, [30,40,2]C2\} and \{[150,55,2]C1, [30,45,2]C2\}. Where, the first crane performs a swing operation from 140° to 150°; whereas, the second crane performs a luffing operation from 40° to 45°.

This chapter discusses both synchronous and asynchronous movements for planning the lifting path.

**1.3 Cooperation between Cranes (Load Line’s Angle)**

In multiple cooperative cranes, it is common to have a slope of the load line, due to cooperation constraints. In such cases, additional load is transferred to each crane because the load lines are not vertical [Shapiro et al. 1991]. Therefore extra case has to be considered so that the slope of load and the resultant additional load must be kept within permissible limits. However slope within limits is permitted since the load-line sway is not modeled as an active DOF. This slope is limited by considering the Euclidean distance between the boom tips of crane manipulators 1 and 2 to vary between object length \pm n (assuming that the object is lifted at the end).

Also a height difference between Hook 1 and Hook 2 results in a slope in the object plane. This slope results in extra loads acting on the manipulator system. Therefore the inclination of the object is limited to 20° with respect to the horizontal plane.
1.4 Collision among Cooperative Crane Manipulator System and Environment
Also due to cooperation constraints, there are four possible combinations of interference in the cooperative manipulator system that is also considered in this study:

1. Manipulator to object
2. Manipulator to obstacles in the environment
3. Object to obstacles; and
4. Manipulator to manipulator.

1.5 What is the Best Lifting Path?

An ideal lifting path is the path that minimizes the work done by the crane therefore it should have a minimum number of swing, luff, and hoist operations [Shapiro et al. 1991]. Further many other factors influence the feasibility and optimality of the path such as:

1. Motion priority among swing, luff, and hoist operations based on the ease of lift execution by the crane operator;
2. Capacity limits of each crane;
3. Minimum clearances needed between cranes, load, and site obstacles; and
4. Safety where avoiding lifts over dangerous operating areas is paramount.

Capacity limits and clearances are the real factors that determine the feasibility and of a lift path, while the other factors are influencing the optimality of the path. In this study, path planning is based on the work done by the manipulator and the clearances required between the cranes, obstacles, and object but there is no check for crane capacity limits in this work.
However, check for crane capacity utilization limits can be added to the path-planning algorithm without the need for major modifications.

1.6 Modeling Conditions

To model the cooperative crane lift problem, the following assumptions and conditions are considered:

- Identical cooperative cranes
- Same horizontal level for both the cooperative cranes
- Symmetrical equally distributed load to each crane
- Static obstacles in the construction site i.e. no change in the state of any obstacle during the lift.

Further geometrical representation of the cooperative cranes has been applied to certain simplifications:

- A solid of rectangular shape integrates both the crane’s base and cab
- A solid of rectangular shape represents both boom and obstacles in the construction site
- A line represents the geometry of the object lifted by cooperative cranes.

Fig-2 shows the simplified geometry of cooperative cranes based on the previous assumptions.
After modeling the simplified geometry of cooperative cranes, we get five key attributes of the study problem:

1- **Crane**:
   
a. Two identical cooperative cranes,

b. 3-DOF (three degrees of freedom for each crane):
   
   i. swinging
   
   ii. luffing
   
   iii. hoisting

   c. Both synchronous and asynchronous motion is considered; and

   d. Limits of slope of load line can be specified.

*Fig-2 Simplified geometry of two cooperative cranes* [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]
2- **Load**: symmetric and is distributed equally to each crane and lifted at the ends.

3- **Site**: obstructions to cranes and load are static.

4- **Feasibility criteria**: collision free path.

5- **Optimality criteria**: minimize work done.

The cooperative crane analysis is still complex even after these simplifications, since the assumptions helps to emphasize on the degrees of freedom and space/path representation issues.

Although these assumptions limit the practical application of the discussed system, but they do not affect the investigation of the proof-of-concept of the solution approach.

**2- SYSTEM DESIGN**

**2.1 Design Strategy**

In general, path planning for manipulators having more than 4-DOF (four degrees of freedom) is considered complex [Hwang and Ahuja 1992]. Development of a system with total of 6-DOF in the cooperative cranes (3-DOF for each crane) would be feasible, where three benchmark manipulator systems of increasing complexity could be formulated in order to deal with the complexity of the problem in a systematic manner.

Fig-3 below shows these formulated benchmark manipulator systems, which are planar 1 x 2 manipulator, planar 2 x 2 manipulator, and planar 2 x 3 manipulator.
Generally a planar $N \times M$ manipulator represents a system that consists of $N$ manipulators, each with $M$ DOF. Each solution approach was initially evaluated on the benchmark systems, which enabled the evaluation of the scalability of different approaches and enhance the selected approaches to solve the more complex dual crane system.
2.2 Design Tasks

Fulfillment of the following three (3) main tasks represents the design of a path planning system for cooperative crane lift:

1. **Modeling search space**: where the environment where the cooperative crane lifting including the crane(s) and all potential obstacles are represented so that a software
tool implements it and hands it over to the path planning simulation process, to do the 
path planning and obtain the optimal lifting path

2. **Selecting path searching technique**: where the automation feature is added to the 
path planning process (Automated Path Planning “Chapter III”)

3. **Selecting path searching options**: where the automation feature is added to the path 
planning process (Automated Path Planning “Chapter III”)

### 2.2.1 Modeling Search Space

Search space can be represented either in two ways:

1. **Real space**: is the Cartesian space with 3 axes X, Y and Z

2. **C-Space**: has the following characters
   a) Is not a real space, plotted between different degrees of freedom
   b) Manipulator’s Degrees of freedom (DOF) are represented by an axis for each 
      DOF.
   c) Each point in the graph represents a position of the manipulator (crane) in the real 
      space
   d) An obstacle point is designated in the C-space graph to represent an obstructed 
      manipulator in a given position i.e. if a manipulator is obstructed in a given 
      position, then the C-space point representing that position is allocated for an 
      obstacle space. On the other hand a free space is allocated in the C-space graph if 
      the manipulator is not obstructed for a specific position of its DOFs
Generally C-Space is preferred over the real space to represent the search space because of the following reasons:

1. C-space produces no redundancy in representing a configuration since each point refers to a unique configuration of the manipulator (which is expressed in terms of configuration variables).

2. Constraints on the manipulator movement can be expressed specifically.

3. C-space can be used to evaluate path for different pick locations since path planning only requires performing a search between the pick and place locations.

4. The real space approach requires solving the inverse kinematics problem where determining the DOF (Degree Of Freedom) of the manipulator at certain location, becomes a sub-problem of the lifting path-planning task [R. Ruddy H.; Varghese K., 2002]. While in C-Space there is no need to solve inverse kinematics problems because the path is represented in terms of configuration variables.

A planar manipulator with two angular degrees of freedom is presented in Fig. 4-(a) above. Since this manipulator is planar, its Cartesian space is limited to two dimensions.

Figs. 4-(a & b) show two configurations [135, 45] and [45, 135] of the manipulator in real space position \( P(x, y) \), where Fig. 4-c shows these configurations represented as points in C-Space. If there is interference or the values of DOF are out of limits, a C-Space configuration might become infeasible.
Since there are six degrees of freedom (6-DOF), the C-Space for the cooperative crane problem would be of six dimensions. To obtain a feasible cooperative crane configuration:

- The load line slope should be within the allowed limit
- No intervention among cranes, object, and obstacles
- The DOF ranges should be within the allowed limits

### 2.2.2 Searching Techniques for Lift Path

Advanced searching techniques such as those used in AI (Artificial Intelligence) can also be used in this engineering context. Some of these searching techniques for planning the lift path are:

1. **Exhaustive search (unguided exploration)**: iteratively produces potential solutions to a given problem, without any guidance it explores the search space systematically, checks to see if the problem is solved, and continues until a correct solution is generated, at which point the solution is returned. *Depth first* and *Breadth first* searches fall under this searching category, which is inefficient because at each step it does not check the goal node’s closeness

2. **Heuristic search (guided exploration)**: based on certain rules, it tries at each step to find the most promising direction. Hill climbing and A* are two common heuristic search techniques

3. **Probabilistic search** *(Johnson and Picton 1995)*: such as genetic algorithm (GA), simulated annealing, and ant colony optimization (ACO). They use randomness in addition to certain guidelines. Their ability to escape from local minima is one of their most advantages
2.2.3 Path Searching Options

Path searching options are also discussed in Chapter III “Automated Path Planning”. A search can be performed either as a tree search in open space or as a graph search in feasible space, for the selected search space representation and search technique. Since both feasible and infeasible configurations could be found in the open search space, validity checks can be done during the search for only those configurations that are hit during the search [Sivakumar Pl.; Varaghese K.; Ramesh Babu N.; 2003].

Searching in a feasible space involves validate feasible configurations by comprehensively testing all configurations of cooperative cranes, then build the connectivity relationship among the feasible configurations, and the path is then obtained by searching the graph containing feasible configurations.

2.3 Implementation Tasks

To build a software system for cooperative lift path planning where users (planners) can use the system would typically consist of three (3) main modules:

1- **Input module**: allows user to interact with the system and collects necessary inputs in order to perform the path simulation

2- **Path search module**: implements the selected searching algorithm and performs searching for the optimal lifting path, where an automation feature is added to the path planning process.
3- **Path Display module**: an output device that displays/prints the search results, where the user gets the lifting path information.

Since the concern of this chapter is to discuss path planning for cooperative crane lifting and present different parameters related to it, details about the automation of the path planning are left for next chapter. In chapter III, we will discuss in more details the automated path planning for crane lifting and cover a literature review for two (2) different automated path planning systems for cooperative cranes:

1- The first system is about Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]

2- The second system is about Collision Free Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005]

Furthermore Chapter III provides a quick review on the path planning used in the system HELPS presented in Chapter I.

### 3- CONCLUSION AND DISCUSSIONS

Cost-effectiveness of heavy lifts can be improved by the use of cooperative cranes. However, due to the complexity of developing a reliable lift plan makes it difficult for its widespread. The efficiency and reliability of the lift plan can be improved by the availability of computer aided planning systems. The discussion in this chapter covers
different parameters related to path planning in cooperative cranes that should be considered in the design and development of such a system.

Automating path-planning will be much appreciated by lift planners since they will be able to reach a reliable lift plan. Other areas, such as manufacturing are interested in the use of cooperative manipulators to do the job quicker in less cost.

Critical design factors for cooperative cranes are discussed in this chapter such as modeling the lifting search space, and modeling the degrees of freedom in two cooperative cranes. The discussions suggested that C-Space representation of the lifting environment is superior to the real space model because C-Space solves the inverse kinematics problem, C-Space results in no redundancy in representing a configuration, and C-Space allows for specific representation of the manipulator’s constraints.

Also it was suggested that a system with total of 6-DOF in the two (2) cooperative cranes (3-DOF for each crane) would be feasible for design and implementation. In order to deal with the complexity of the problem in a systematic manner three (3) benchmark manipulator systems of increasing complexity could be formulated and considered in testing the system.
CHAPTER III

Automated Path Planning for Mobile Cranes

1- Introduction

Chapter I, discussed the heavy lift planning and showed how it can be manually performed using conventional planning tools such as scaled models and drawings where data from different resources such as project schedules, crane capacity tables, and site layouts, has to be integrated. It was clear that in order to produce a detailed lift plan for crane lifts; the manual process has to be iteratively performed until reaching a reasonable degree of confidence. This iterative process is very time consuming that might take weeks or months to complete. A system called HELPS has been explored where it was found that the system allows planners to represent the lifting environment using an off-line 3D AutoCAD file, then the system uses Walkthru program to perform the planning simulation using the 3D file and tries to find the best crane location that allows for the optimal lifting path.

In chapter II, an attempt has been made to explore different issues that might come across during lifting path planning for cooperative cranes. Issues such as representing the lifting work environment and modeling the Degrees Of Freedom (DOF) in two (2) cooperative cranes are the most important issues that have been addressed. Since the objective of Chapter II was to provide generic discussions on cooperative crane lifts with their associated issues without going through specifics, this chapter III is going to provide the specific discussions on two (2) different cooperative crane lifts.
This chapter III investigates the concepts and techniques used to automate the crane path planning in 3D space. The chapter discusses automating the path planning for mobile cranes, where the system automatically performs searching for potential paths and then finds the optimal path(s) on behalf of the path planner. In this case path planners do not need the same high level of experience that was necessary to manually simulate the path planning, find the crane location and find the optimal path.

This chapter contains six (6) sections as follow:

1- Introduction (current section)

2- Overview of the concepts for path planning used in robotic manipulators, and then it discusses the concepts of configuration space applied to crane lifting.

3- Present a design and implementation of an automated crane lift planning system that implements the concepts in the previous section into a CAD environment using the heuristic search approach to find and optimize the lifting path

4- The fourth section discusses the implementation of path planning in cooperative crane lifts using heuristic search approach as well

5- The fifth section discusses the implementation of path planning in cooperative crane lifts using genetic search approach

6- Finally conclusion and discussions based on the material introduced in the chapter

2- Path Planning in Robotic and Crane Manipulators
To perform robotic path-planning, two tasks are involved:

### 2.1 Space Representation

Determines and represents the free space and obstacle space within the workspace, which could be represented by either real space or transformed space:

- **Real space representation**: 2-D and 3-D representation of manipulators and obstacles in the workspace. As mentioned in Chapter II, one of the major limitations of the real space approach is the inverse kinematics issue where determining the DOF (Degree Of Freedom) of the manipulator at certain location, becomes a sub-problem of the lifting path-planning task.

- **Transformed space representation**: simplifies the space representation and avoids the inverse kinematics problem, by transforming the real space into a constraint represented space.

In both representations the free space and obstacles have to be identified with either one of the following three techniques:

a) **Road map methods**: Visibility graph and Voronoi diagrams are techniques that can be used to construct roadmaps. The free space is identified by un-intersected lines connecting the start and the end nodes. Vertices of obstacles are identified as the free space, where obstacle space is the remaining part of the space.

b) **Exact cell decomposition**: a collection of non-overlapping regions called cells, represent the free space. The shape of the region and obstacles determines the shape of cells.
c) **Approximate cell decomposition**: where cells of simple uniform shape are identified as free or obstructed, and the resulting representation is practically considered correct (although it is not mathematically perfect).

### 2.2 Finding Path

This function tries to find the optimal path between the start (pick) and end (place) nodes (locations) using any graph searching algorithm. As mentioned in chapter II, AI searching techniques such as heuristic search (A*, ), probabilistic search such as (simulated annealing, genetic algorithms “GA”, and ant colony optimization “ACO” which will be introduced in Chapter IV), could be utilized.

### 2.3 Representing Workspace using Configuration Space (C-space)

The system uses C-space representation of the workspace environment, which has the same characteristics discussed in chapter II under section **2.2 Design Tasks**

Fig-1 below illustrates how C-space approach works for a 2-link planner manipulator, which has two (2) angular DOF: \( \theta_1 \) and \( \theta_2 \), each with a range of zero to 180 degrees. The obstacle in real space is presented by the purple shaded area in Fig-1. The pick and place locations along with the manipulator’s successful configurations: point 1, and point 8. Also the other possible configuration paths are shown 1 to 7. [R. Ruddy H.; Varghese K., 2002]
Fig-2 below shows the equivalent 2D space for the manipulator and obstacle, where each dimension (axis) of the 2D represents one of the two degrees of freedom of the manipulator [R. Ruddy H.; Varghese K., 2002]. The path between the pick and place locations in Fig-2 below passes through the corresponding configurations 1 – 7 shown in Fig-1 above.
Once the C-space is built, searching the space for an optimal path between a given pick and place configurations takes place, where any standard graph-search algorithm can be used to perform the search, including cost-functions and heuristics specific to a particular domain.

3- Data Flow Diagram (DFD) For Generic Automated Path Planning System

The following diagram in Fig-3 below is a Data Flow Diagram (DFD) for a system that generates the path planning for crane lift:
4. Automated Lift Planning System for Single Lifting Crane

This section explores an automated lift planning system for mobile cranes introduced [R. Ruddy H.; Varghese K., 2002]
This system deals with a Single crane only and uses two levels of heuristic search. The first search goes through the entire search space to find potential paths, and the second search goes through the constrained search space which is defined based on the first search results in order to find the optimal lifting path. Lifting space has been modeled as C-Space using AutoLisp and only three (3) Degrees Of Freedom are considered for the lifting crane.

4.1 Development of Configuration Space (C-space) for Cranes

C-space discussed earlier in the previous section in this chapter handles only 2DOF, but in general (as mentioned in Chapter II) there are a maximum of eight (8) degrees of freedom for a loaded crane. This 8-DOF includes:

d) A single DOF for the crane operations (swinging, hoisting, luffing, telescoping)
e) Three (3) DOFs to account for moving and rotating the crane base
f) When considering the load’s rotation around the vertical axis additions DOF is added for lifting the load

Considering all the eight (8) DOFs (including crane travel and load swing) makes it very complex computation of the lifting path. The scope of this system is limited to only the three (3) most common DOFs used in lift operations which are: swinging, hoisting and luffing. However the implemented C-space can easily be extended to handle higher DOFs.
The shape of the C-space for 3-DOF (swinging, luffing, and hoisting) crane with no external obstructions, is presented in Fig-4 below, where its circular shape results from the crane’s swinging motion. Because the space is discrete and the limitation of hoisting the load varies from different luffing angles, the stepping of the C-space arises [R. Ruddy H.; Varghese K., 2002].

![Fig-4 C-space diagram for a 3-DOF crane](image-url)

Fig-4 C-space diagram for a 3-DOF crane [R. Ruddy H.; Varghese K., 2002]
4.2 Path Searching Criteria

As mentioned earlier, once the C-space graph is ready the costs of traversing the space are calculated based on the factors that influence the path selection based on the path planners’ point of view. This system design identifies the following factors that contribute to the cost of traversal for a given path:

1- Direct costs: the cost of performing a unit displacement of an operation in relative performance swinging, hoisting, and luffing

2- Indirect costs: includes:

   - Change in direction costs: the cost of changing the direction of motion by stopping one motion and starting another.

   - Proximity to obstacles: means the cost of abusing a specified clearance from an obstacle

To reduce the search time a heuristic is required, where a heuristic is employed to direct the search in the right direction by evaluating the validity of the next node based on the actual distance to the place location. Once reaching a reasonable solution, it can be refined by performing a more detailed search in the proximity of the established path [R. Ruddy H.; Varghese K., 2002]
4.3- System Implementation

The system discussed in this study consists of the following six (6) modules:

4.3.1 Input Module

This data entry module is implemented as a set of AutoLisp (is a small dynamically scoped LISP “list processing language”) functions that prompt the user to enter the following input requirements:

1- **Shape and dimension of the load to be lifted**: a CAD drawing of the shape is generated as a result.
2- **Crane configuration file**: a pre-prepared CAD file contains the crane drawing for a given boom length organized in layers
3- **File of the load capacity chart**: text file that specifies the crane capacity for different boom lengths and lifting radii
4- **Minimum clearance and weight of the load**: are numeric inputs that are written to a file and used later by the system

Based on the given inputs the following two values are computed to be used by the next module for workspace generation:

1- The maximum working radius of the crane (which depends on the capacity and clearance of the load to the ground)
2- The minimum working radius of the crane (which depends on the clearance of the load to the boom).

4.3.2 Workspace Creation Module
This module uses AutoLisp to generate a 3D view of the crane’s workspace using the maximum and minimum working radii of the crane computed in the input module. After entering the CAD drawing file of the site, specifying the location of the crane, and using the crane location as the center of the workspace geometry the module merges the geometry of the 3D workspace with the site geometry and creates the workspace, where a graphical subtraction using the “Subtract” function of AutoCAD results in eliminating all objects outside the workspace. The output file of this module is used as an input to the next module (C-space generation).

### 4.3.3 C-Space Creation Module

This module is also implemented using AutoLisp to generate the C-space, where a subtracted site geometry (generated in the previous module) is initially merged with the crane and load geometry. An AutoLisp function controls the different DOFs in a nested control structure to cover all possible configurations of swing, luff and hoist [R. Ruddy H.; Varghese K., 2002]

In each configuration, its obstructed status is checked by existing Auto-cad interface detection routines, and records the results to an output C-space file. This C-space file is used as an input for the creation of the free-space graph in the next module (initial path determination).

### 4.3.4 Generating Initial Path Module
Depth-first search algorithm is the main algorithm used in this module, which is modified with a forward looking distance heuristic similar to the directed hill-climbing. Unobstructed C-space nodes are used for the search-space of this system.

The free-space graph is created by this module using the C-space file (output of the previous module), then pick and place configuration nodes are specified. The search starts its job to find the potential lifting path between the pick and place configurations. This path is written to a file as a sequence of nodes traversing between the pick and place nodes.

4.3.5 Path Optimization Module

This module refines the path generated in the previous module by adding more searching criteria. Path optimization starts with generating a constrained search space around the generated path and scans through the full path length. A free-space graph is constructed which is limited to the unobstructed configurations that falls within the constrained space. Based on the optimization criteria, a detailed search runs against the constrained free space in order to reach an optimized path. Also a heuristic depth-first search algorithm is used based on the estimated lowest cost to reach the destination.

When the search reaches a certain improvement in the path, it sets this path as its target to continue its improvement and reach a better path. This iterative process continues and terminates only when either:

1- Reach the end of the constrained-search space
2- Achieve a specified number of cost improvements
3- Achieve a specified cost
4- Elapsing a specified time

This module produces an output file specifying the sequence of nodes that have been traversed between the pick and place nodes, and represent the optimized path

5. Automated Lift Planning Systems for Two Cooperative Lifting Cranes

This section discusses two (2) implementations of path planning systems for two cooperative cranes:

1. The first system performs Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search [Sivakumar Pl.; Varaghese K.; Ramesh Babu N.; 2003]
2. The second system is a Collision Free Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005]

5.1 Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search

The system is based on Heuristic search, where hill climbing search and A* search are used in open C-space. Also a tree based search in open C-Space is considered as a searching option since pre-computation of feasible C-Space can be computationally expensive for higher dimensions of C-Space.
5.1.1 Implementation of Path Planning Algorithms

The system implements two (2) path searching algorithms:

1- Tree based hill climbing search in open C-Space

2- Tree based A* search in open C-Space.

The system also implements the same three (3) main modules mentioned earlier as follow:

5.1.2 Input Module

The elements of the input data are [Sivakumar Pl.; Varaghese K.; Ramesh Babu N.: 2003]:

1. Site geometry

2. Dimensions of each crane

3. Length of the load

4. Number of obstacles

5. Location and dimensions of each obstacle

6. Pick and place configurations

7. Step size for each DOF

8. Number and the DOF that vary concurrently between two successive steps

9. Tolerance used to limit the slope of the load line.

These data are written to an ASCII text file, and it is read by the system using a C++ programming code.

5.1.3 Path Searching Module
To find the lifting path and then optimize it, hill climbing and A* search were used. While the local search hill climbing keeps only the information about the current node and its neighbors at any time during the search, the A* is a global search that maintains the configuration date of all nodes that are explored and the linking relationship between them until reaching the place configuration. A* search uses Linked lists Data Structure to maintain node chaining during the A* search [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]

5.1.3.1 Hill Climbing Search
The Hill Climbing Search algorithm is explained in Appendix A for the logic for a hill climbing used in this work.

5.1.3.2 A* Search
The A* Search algorithm is explained in Appendix A for the logic for a hill climbing used in this work.

5.1.3.3 Hill Climbing and A* Search Details
Generation of neighbors, feasibility check, and computation of heuristic search function, are the three major processes common in both techniques. While the implementation of the first two steps are the same in both hill climbing and A* searches, the third step is not the same.

5.1.3.4 Creating Neighbors
A fixed number of neighbors are generated for each configuration, and for each configuration, the total number of neighbors generated depends on the possible combinations of DOF variations for cooperative cranes during its movement from one step to another. Please refer to [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003] for more details.

5.1.3.5 Feasibility Check

The feasibility of cooperative crane configuration depends on the following three factors:

- Cooperation among the cranes
- Interference
- The limits for DOF

5.1.3.6 Cooperation Checks

The perfect coordination between two cranes happens when the distance between their boom tips is always maintained to the length of the object carried by the two cranes, which unfortunately is not always practically possible [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]. This means that a difference between the object length and the distance between the boom tips happen and as a result a slope in the load line is formed which causes an additional load to be transferred to the cranes. Therefore the slope of the load line and the additional load transferred due to this slope should be kept within permissible limits, to ensure acceptable cooperation.

5.1.3.7 Interference Check

Four (4) possible combinations of interference in the cooperative crane system can be found, which include [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]:

71
1. Crane component to load
2. Crane component to obstacles in the construction site
3. Load to obstacles
4. Crane boom to the cooperative crane boom.

5.1.3.8 Check for Limits of DOF

A procedure to monitor the limits of the DOF (swing, luffing, and hoisting) has been developed and integrated to the system to ensure that each DOF is always maintained within its permissible limit.

5.1.3.9 How Does The Heuristic Function Work?

A heuristic function is used by the path planner to find the most hopeful direction in order to move from one node to the other. For reaching the best lift path, this system considers only the work done by the manipulator. According to the theories of physics, the work done is estimated to be the Euclidean distance traveled by the object in real space corresponding to the movement of cooperative cranes between pick and place configurations [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003]. The heuristic function is calculated differently between the Hill Climbing search and the A* search, as follow:

1- For Hill climbing search is equal to the Euclidean distance between the current configuration and the place (end point) configuration.
2- For the A* search, is the sum of the actual distance from pick (start point) to current 
configuration and the Euclidean distance between current configuration to place (end 
point) configuration.

5.1.4 Path Display Module

Lifting path means the route or the step-by-step movement of cooperative cranes. The 
path for the crane 2 x 3 manipulator is expressed as follows:

\([C_1, C_2]_{\text{PICK}}, [C_1, C_2]_1, [C_1, C_2]_2, [C_1, C_2]_3, [C_1, C_2]_{N-1}, [C_1, C_2]_N, [C_1, C_2]_{\text{PLACE}}\)

Where \(N\) represents the number of steps representing the path and \([C_1, C_2]\) represents a 
unique configuration of cooperative cranes, which is \([[\Phi, \theta, h_L]_{C_1}], [[\Phi, \theta, h_L]_{C_2}]\).
AutoCAD is utilized to simulate and display the generated path in 3D. Customized toolbars were developed to control each DOF of cooperative cranes using Object ARX (Autodesk 1999), as a part of this platform, where a screen snapshot of this platform is shown in Fig. 5 above. Furthermore, this platform can provide a visual representation on checking cooperation and interference between cranes [Sivakumar Pl.; Varaghese K.; Ramesh Babu N.: 2003].

5.1.5 RESULTS AND DISCUSSION

Three test scenarios are used to illustrate the performance of path-planning approaches. Both hill climbing and A* are applied in each scenario, and the results are compared. The first scenario is based on synchronous movement; whereas, the second and third are based on asynchronous movement.

For complete details on the results and associate discussions, please refer to: Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search

5.2 Path Planning of Cooperative Crane Manipulators Using Genetic Algorithm

5.2.1 Introduction

This system implements the concepts and configurations for cooperative cranes as discussed in the previous chapter (Chapter II), in terms of problem formulation, path planning concepts (such as C-space modeling of lifting site, and crane’s DOF), cooperation between two cranes, and collision among cranes. Also while the system
discussed in the previous section uses two graph searching methods: Hill Climbing and A* Searching, this system applies more sophisticated searching algorithms to find the optimal lifting path, which is the Genetic Algorithm (GA).

5.2.2 Search-Technique Concepts

Because of the increasing use of GA-based search and optimization techniques in robot motion planning, this system utilizes GA to optimize the angular moves of the cooperative manipulators in moving the load/object from pick location to place location.

A fitness function is formulated that considers two (2) main constraints:

1- Guaranteeing coordination in cooperatively handling objects [Kohout P. 2003]

2- Preventing collision of the manipulator system with itself and with the environment [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005].

The fitness evaluation is suggested to be performed in the following two phases which reduces the computational time, as shown in the program flowchart at Fig-6 (a, b) below:

- Coordination checking phase
- Coordination and collision checking phase
Inputs:
- Pick/Place locations
- Manipulator, Object and Obstacle dimensions
- DOF limits in manipulator system
- Crossover probability and mutation
- Maximum Generation
- Convergence Criteria
- Population size
- # of intermediate configuration position between pick and place points

GA Run – Initial Population Generation

Evaluation of the fitness function

Coordination Phase Fitness Evaluation

Reproduction (or) selection

Two point Cross Over

Parameter Based Mutation

Coordination Phase Fitness Evaluation

Termination of Coordination Phase

No

Yes

Coordination and Collision Phase Fitness Evaluation

Fig-6-a GA Search Program Flow (Coordination Phase) [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005]
So with the dual fitness evaluation, the program ensures that only coordinated moves are evaluated for collision.

To avoid jumps caused by standard crossover operator to path strings, two techniques were developed for each stage:

1- A standard two-point crossover followed by a parameter-based mutation to smooth the string, in the first phase

2- A parameter-based crossover was developed to minimize jumps in the paths, in the second phase
5.2.3 Implementation Strategy

This system models and implements a 2 x 3 cooperative manipulator system using GA, where an input module is generated to enter input information regarding:

1. Pick and place configuration of the manipulator system
2. Number of obstacles
3. Location and dimension of each crane

The GA technique is used in this system to search the optimal path and consists of the following steps, (as shown in Fig-1 (a, b) ).

1. **Initialization**: where chromosomes are randomly created. Diverse population is very important at this stage otherwise the results might not be optimal.
2. **Evaluation**: A fitness value is assigned to each chromosome where each chromosome is evaluated on how well the chromosome solves the problem at hand.
3. **Selection**: the fittest chromosomes are elected for spread into the future generation based on the fitness level they have.
4. **Recombination**: where recombination of individual and pairs of chromosomes takes place, modification and then put back into the population.

5.2.4 Generating Initial Path Strings

The initial path strings as shown in Fig. 7 below, are populated using the values of upper/lower limits listed in Table-1, where an example of a path string represents a unique position of the manipulators in the C-space along with fifteen (15) different
configurations generated randomly (within the range limits) and inserted between pick
and place locations. Each intermediate configuration consists of six joint angles (3 DOFs
“swing, luffing, and hoist” for each manipulator).

<table>
<thead>
<tr>
<th>No</th>
<th>Arm</th>
<th>Lower Limit</th>
<th>Upper Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Base-swing</td>
<td>0</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>Boom-luff</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>3</td>
<td>Hoist height</td>
<td>0</td>
<td>Depends on luff angle of boom and geometry of load</td>
</tr>
</tbody>
</table>

Table-1 Limiting Values of Movement for Different Arms of Crane Manipulator [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005]

The GA algorithm runs the initial population with its two phases of operation
(“Coordination checking phase” and the “Coordination and collision checking phase”)

For more details on the GA implementation please refer to [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005]

5.2.5 Program Termination

The convergence criterion terminates the GA, where the entire genetic process of
coordination and collision phase fitness calculation, reproduction, parameter-based
crossover and bitwise mutation is repeated until no improvement in the solution occurs for 10 consecutive generations. At the end of the coordination and collision phase, the fittest string that has minimum distance and a collision-free path is selected to be the desired optimal solution.

6. CONCLUSION

This chapter reveals the possibility of effective use of C-space (configuration space) concept, Artificial Intelligence (AI) techniques such as heuristic search, and Genetic Algorithms in planning and optimizing the crane lifting path. This comes in the cost of implementing efficient software and hardware systems for both C-space generation and searching techniques.

Also cost-effectiveness of heavy lifts can be improved by the use of cooperative cranes. However, the complexity of developing a reliable lift plan enforces the utilization of the computing power and its aided planning systems. The work presented in this chapter represents an initial step toward the development of such automated systems.

This chapter discussed three (3) different automated path planning systems:

1- **System I**: Automated path planning system for a single mobile crane lift [R. Ruddy H.; Varghese K., 2002]

2- **System II**: Automated path planning system for cooperative crane lift using Heuristic search [Sivakumar Pl.; Varaghese K.; Ramesh Babu N.; 2003]
3- **System III**: Automated path planning system for cooperative crane lift using Genetic Search Algorithm [A. D. Ali M.S.; Ramesh Babu N.; Varghese K., 2005]

As presented in **System I**, AutoLisp is utilized to generate a 3D view of the crane’s workspace, create the C-space considering three (3) Degrees Of Freedom for the lifting crane. A double search strategy can be implemented to reach an optimal result for the lifting path planning where the first search uses depth-first search modified with a forward looking distance heuristic and produced a general direction of a feasible lifting path, then the second search also uses a heuristic depth-first search to do the detailed search within a limited search tunnel in order to reach an optimal lifting path.

The experiments on **System I** showed that although using CAD environment produces overhead, but its utilization is necessary for implementing engineering tasks

Unfortunately the performance of CAD can not be enhanced with parallel processing; however implementing heuristic search with C language can take good advantage of the parallel processing.

Future work regarding **System I** should focus on:

1. AutoCAD’s Runtime Extension (ARX) module to be used instead of AutoLisp to enhance the implementation efficiency
2. Developing efficient heuristic logics to improve the search
3. Consider modeling higher degree of freedom levels
4. Explore alternate searching algorithms that produce more efficient results such as genetic search

**System II** C-space is used to model the working space and 3 DOF is considered for each crane. The system utilizes two heuristic search methods: a tree-based hill climbing search and a tree-based A* search in open C-Space. Assessment of the effectiveness of these search methods was done in three different scenarios that required the synchronous and asynchronous movements of cooperative cranes with single and multiple obstacles.

Hill climbing was found effective when space is not confined. A* search on the other hand provides slower performance but is able to generate near optimal paths in confined spaces. A* becomes much more slower if additional cranes are added to the system or if addition degrees of freedom such as telescoping are included, because of the increase of the C-Space dimension

Although **System II** proved its concept in automating path-planning for cooperative crane lifts, but it cannot be used directly for automated planning in practical cases because of the following limitations:

1. Simplified geometry of crane and load
2. The heuristic search is based on distance only
3. Omitted issues such as unsymmetrical loads and its distribution on two cooperative cranes.
Further work regarding System II includes:

1. Overcoming the limitations of the system mentioned above.
2. Investigating the improvement of the execution speed when performing a search in a finer resolution.
3. Produce higher dimensions of C-Space that consider telescoping and mobility of crane components.
4. Experiments with powerful hardware, improved heuristics based on clustering techniques, parallel processing, and alternate search methods such as genetic algorithms.

Genetic Algorithms that introduced in System III proved to produce more efficient results than the conventional A* search, both in the distance traveled and in the computation time for automated path planning of 2 x 3 (2 manipulators with 3 DOF each) cooperative manipulators. This approach performed search in the entire C-space, including both feasible C-space and obstacle C-space. The search identified the collision-free configurations for cooperative handling of the object between the pick and the place locations.

The GA search and optimization mechanism applied its technique in two-stages:

1- Coordination phase: where an optimized path with coordination is produced, using its coordination fitness measure.
2- **Coordination and Collision phase**: where a safe, feasible, and collision free path is produced in this stage. Parameter-based crossover is used in this second phase because of the coordinated strings produced in the first phase, and any two-point crossover might exchange the configuration parameters among the strings, which could affect coordination among the manipulators.

However System III presents the following challenges:

1. The execution process requires a manual coordination, which introduces practical difficulties. Further research work is underway for developing an automated execution using robotic work cell.

2. The dynamic load effects during motion. Here also work is in progress towards addressing this issue for cooperative manipulators using telescopic capabilities.
CHAPTER IV

Tools and Techniques for Automated Path Planning

Introduction

Based on discussions introduced in the previous chapters, this chapter goes through tools and techniques used for automated path planning in terms of the Information Systems side. Graphical modeling of construction site specially the lifting area such as CAD packages, software development tools that enable developers to build efficient algorithms for searching graphs and doing computations are some examples of such tools and techniques.

The chapter also introduces an Ant Colony Optimization (ACO) algorithm as an intelligent bionic optimization graph searching algorithm after it has proved its efficiency in path planning systems for robotic motions. A model for an automated path planning for single lifting cranes that utilizes ACO algorithm is suggested, where ACO is basically used to optimize the lifting path. Due to the limitation of this paper, the proposed model has not been implemented in real life and therefore no testing and/or results are discussed.

In the previous chapter (Chapter III), Heuristic and Genetic algorithms have been utilized to find the optimal lifting path for single and for two cooperative cranes planning. The
following sections provide an overview on how ACO is modeled and discuss the possibility of its utilization in the area of path optimization in crane lift planning.

1. Path Planning Tools and Techniques

The software tools for lifting path planning can be categorized into two categories:

1.1 Graphical Tools

Such as CAD package which is designed specifically for analysis, modeling, and design of engineering projects. It has an easy to user GUI (Graphical User Interface), allows users to model and analyze engineering projects in 2D and 3D. This tool has been used all over the systems discussed previously in this paper, to model construction sites and specifically lifting area.

1.2 Computational Tools

Chapter I introduced the HELPS systems that utilized Walkthru system to perform the path planning for heavy lifts. Chapter III discussed the automated lift planning for mobile cranes. This system used LISP software to model the C-Space and perform the planning simulations enforcing the manipulator’s DOFs. C programming language has been utilized in that system to perform the two level heuristic first-depth searches. The two other systems presented in Chapter III, have utilized Artificial Intelligence techniques such as Hill Climbing, A* and Genetic Algorithms to perform the search for path and the path optimization.
2. Ant Colony Optimization (ACO) Overview

2.1 What is ACO?

Ant Colony Optimization is a new class of natural algorithms stimulated by the scavenging behavior of natural ant colonies. ACO algorithm is a probabilistic technique used to find best paths through graphs. It has been brought up by Marco Dorigo in 1992 in his PhD thesis.

The ant is blind and weak creature; however by cooperating with each other, the colony of ants demonstrates complex behavior. When ants want to find the closest route to a food source or some other interesting landmark, they start walk randomly, and once food is found they return back to their camp (colony) where they lay down pheromone (a special chemical used by ants to communicate with each other) traces during their back trip. Other ants looking for food would likely follow this pheromone trail rather than randomly traveling. More pheromone would be added to the pheromone path by the other ants while their returning back trip if they found the food (for more information about ant’s communication please see Ant communication).

Pheromone trail starts to evaporate overtime, resulting in reduction of its attractive strength, and the more time it takes for ants to travel down the path and back again, the more time the pheromones have to evaporate. On the other hand a short path, takes less
time to go through, and the pheromone density remains high since pheromone is poured on the path in a rate that could be faster than its evaporation rate.

This means that pheromone evaporation has the advantage of determining whether a path is short or long. It also has the advantage of avoiding conflict with a locally optimal solution.

The purpose of the ant colony algorithm is to imitate the ant’s behavior using simulated ants that walk around the graph representing the problem to solve.

2.2 Example using ACO algorithm in Artificial Ants

In Fig. 1-a below, assume that the distances between points D and H, between B and H, and between B C + C D (where C is half the way between D and B) are equal to 1 unit distance. The example will try to discover what happens at regular discrete intervals of time: $t = 0, 1, 2, \text{etc.}$ Assuming that:

1. At each time unit, there are 30 new ants come to B from A, and 30 to D from E
2. Each ant walks at a speed of 1 unit distance per 1 time unit
3. While walking an ant lays down at time $t$ a pheromone trail of intensity $\tau = 1$
4. To make the example simpler, pheromone trail evaporates completely and instantaneously in the middle of the successive time interval ($t + 1, t + 2$).

In Fig. 1b below, at the beginning ($t = 0$) there is no trail yet, but there are 30 ants are in B and 30 in D. Ants’ choice about which way to go (from B/D to C or H) is totally
random, i.e. the probability to go from B/D to C or H is 50/50. Therefore expected 15 ants from each node will go toward H and 15 toward C (Fig. 2b).

In Fig. 1c below, at time $t = 1$ the 30 new ants that come to B from A find a trail of intensity 15 on the path that leads to H (laid by the 15 ants that went that way from B), but a trail of intensity 30 on the path to C, (equals to the sum of the trail laid by the 15 ants that went that way from B and by the 15 ants that reached B coming from D via C since the whole distance BC + CD = 1). In this case the probability of choosing a path is not 50/50 anymore. Therefore the expected number of ants going through C will be the double of those going through H (20 versus 10 respectively). The same probability applies for the new 30 ants in D which came from E.

Ultimately with continuing this process all of the ants will choose the shortest path (i.e. reaching A or E through C)

The principle of ACO is at a given point the path that has the highest probability to be chosen by ants is the path that was heavily chosen by preceding ants (i.e. with a highest trail level). In addition, high trail levels always associated with short paths.
3. Architecture of the Suggested Model

The suggested model introduces a single crane manipulator system that utilizes the Ant Colony Optimization (ACO) algorithm in order to efficiently reach the best crane lifting path, where collision prevention is taken care by other components called RAPID (Robust and Accurate Polygon Interface Detection) programs. The following sections discuss the system’s architecture:

3.1 Geometric Representation of the proposed Crane Lifting System

The proposed algorithm offers three (3) different ways for the geometric representation of the crane lifting site which are: C-Space (Configuration Space), Grid Representation, and Probabilistic Roadmap (PRM)
**I- C-Space**: it has been discussed in very detail in the previous chapters (Chapter II, III), and has been used in modeling working environments in both single crane lift and two cooperative crane lifts. As mentioned earlier, this approach is a 2-D representation of the working space. If C-Space representation is selected then the suggested system architecture can simply be modeled similar to any of the three (3) systems discussed in Chapter III, where the C-space models the lifting site and enforces the DOFs on the path planning simulation. A two way searching algorithms can be used: the first uses heuristic Hill Climbing search to find the feasible path(s), then the second uses ACO to optimize the feasible path which ends by the optimal lifting path.

**II- Grid Representation**: This approach generates a 3D tessellated model of the real solid objects of the working environment (such as obstacles, load, etc). Fortunately generating tessellated format is supported by most of the CAD packages including Pro/Engineer and AutoCAD, which makes the algorithm more popular that it can accept any solid model generated by any CAD packages. In addition, most of the collision detection programs (including RAPID) accept only the approximated triangular sides of the original model.

The tessellated file (*.stl) is an ASCII or binary format used in manufacturing, contains a list of triangular planes that approximates a computer generated solid model. It is the standard input for most rapid prototyping machines, where it defines an object’s surface by means of a set of adjacent triangles as shown in the Fig-2 below:
Each vertex of the triangle (number of triangles determines its accuracy) is presented by its X, Y and Z Cartesian coordinates in the .stl file, along with the coordinates of the normal vector to the triangle. Furthermore, each edge is shared only by two triangles.

The proposed ACO algorithm is based on recognizing the available paths in the given 3D tessellated model that is represented by the *.stl file. There are two suggested methods to detect collision in order to determine the availability of paths between pick and place locations:

1. Using collision detection library RAPID (Robust and Accurate Polygon Interface Detection)

2. Incorporating the functionality of RAPID library in the software design of the artificial ant used in the system. In this case each ant is equipped with its own sensor that allows it to move safely without collision.
The first suggestion is preferred since it reduces the design complexity of the artificial ant, where RAPID library components are called to check for collision in each movement of the ants.

1. RAPID is a C++ library developed at the Department of Computer Science, University of North Carolina for collision detection of large environments composed of unstructured models [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006].

**III- Probabilistic Roadmap (PRM)** [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]

This approach has the advantage of representing complicated high level Degrees Of Freedom (6-DOF or more) of cooperative 3-D robot motions, which is assumed to be similarly achieved in the case of crane motions.

Generating PRM takes two phases: *Pre-processing phase* and the *path planning phase*.

1. **Pre-processing phase**: builds a *C-space* (using deterministic local planner) where a set of collision-free configuration spaces are generated and with a simple path planning technique applied to pairs of neighboring nodes, the collision-free configuration spaces are interconnected to a network.

2. **Path planning phase**: where any two nodes A and B in the network are connected based on the connected two configurations. Then the network is searched for a sequence of edges connecting A and B, where an advanced technique can be used to improve the path finding process.
Each Degree Of Freedom (DOF) of the lifting crane is specified and implemented within its limits. Checks for collision with obstacles, loads and the crane itself are performed in the resulting configuration. A straight line in C-space represents a local planner (phase 1) that has been used, while the other planner (phase 2) is the one that can be used for the lifting cranes which can move restricted by its DOF's which works as follow:

a) Odd joints are simultaneously translated each time interval along a straight line in the workspace that connects its workspace configuration \( p \) to its workspace at configuration say \( q \).

b) The inverse kinematics of the lifting crane used in the path planning, is computed in order to adjust the position of the even joints.

c) To enhance the roadmap, a number of nodes between 1/3 and 1/2 of the initial nodes are generated. Select a node \( x \) and expand it using a probability distribution function. Expanding configuration \( x \) is performed by assigning a random value for each DOF in an interval centered on its value at \( x \) about 1/6 of range of the DOF.

### 3.2 Degrees of Freedom of the Lifting Crane

Based on the selected method of representing the geometry of the working environment, the implementation of the Degrees of freedom (DOF) of the lifting crane, differs. For example in the Grid representation, the lifting crane’s DOF could be incorporated in the design of the RAPID library (that is responsible for generating the available paths between pick and place locations), by means of fitness functions. In this case the paths
generated by RAPID library will be practically acceptable since they are determined based on the crane’s DOF that is applied by the fitness functions’ enforcement.

On the other hand if the PRM representation is selected, then the DOF is automatically implemented as discussed under the “Path Planning Phase” of PRM above.

To reduce the design complexity of the artificial ant, the DOF considered in this model is assumed to be similar to those discussed in the previous chapter (Chapter III), where only three (3) DOF are considered out of the eight (8) degrees of freedom (DOF) that a conventional crane manipulator can have. These 3 DOF are: [Swing, Luff, and Hoist], where a single manipulator would have the 3 DOF are represented as [Φ, θ, h].

3.3 Collision between Crane and Environment

As discussed earlier, collisions between crane and its environment (obstacles, loads, etc) are detected by RAPID library (in the first suggestion) or by the roadmap planners (in the second suggestion).

3.4 The Ant System

The flow chart diagram in Fig-3 below describes the process used in the Ant system:
As discussed earlier, The ACO algorithm is based on using pheromone trail laid down by individual ants. An ant tends to follow the path that its trail has higher intensity of pheromone. The pheromone trail evaporates over time, and if no more pheromone is laid by other ants, it looses its intensity. A path chosen by large number of ants would have high pheromone intensity which makes it more attractive for other ants to choose that path.

The pheromone trail intensity $\tau$ (for example) depends on two parameters:

- $\rho$ the evaporation rate of the pheromone trail.
- $\sigma$ amount of pheromone trail in unit length, laid on edge of ant $k$ at a period of time.

Fig-3 Structure of the Grid-Based Ant Colony Algorithm [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]
If C-Space representation of the crane lifting work space is selected then the suggested system architecture can simply be modeled similar to any of the three (3) systems discussed in Chapter III, where C-space models the lifting site and enforces the DOFs on the path planning simulation. Then a two way searching algorithms can be used: the first uses heuristic Hill Climbing search to find the feasible path(s), then the second uses ACO to optimize the feasible path and end by finding the optimal lifting path. The Data Flow Diagram (DFD) in Fig-4 below shows how the system works.

3.4.1 DFD for suggested lift planning system using ACO in C-space

The following Data Flow Diagram (DFD) in Fig-4 shows how the system works in this case:
However in the suggested system the grid representation of the crane lifting work space is the selected option because of its visual advantage over C-Space representation (i.e. 3-D vs 2-D). Then the system employs artificial ants to perform the path optimization task. These artificial ants are software programs designed to work as intelligent multi-agent system, where each agent (ant) works on behalf of the lifting crane. The ant (agent) in this case must inherit all the characteristics of the lifting crane in order to model the lifting crane and work in its behalf. Crane’s characteristics are passed to each artificial
ant (agent) as initialization parameters during instantiation of the ant object. These crane characteristics are:

1. Geometric shape of the crane, where the crane’s geometry is simplified as suggested in Chapter II, section 1.6 Modeling Conditions
2. Crane capacity (Max load that it can carry)
3. DOF of the crane, as discussed earlier in this chapter Degrees of Freedom of the Lifting Crane

An artificial ant performs a complete tour which in this system is defined as traveling from the pick location to the place location, by choosing the grid points according to a probabilistic state transition rule called random-proportional rule. The ant’s movement mimics the movement of the crane’s arm (boom) in regards to the crane’s position, and is subjected to the crane’s DOF which guides each movement of the crane’s boom. In random-proportional rule, an ant selects the following points that are closest to the place location with higher pheromone intensity [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006].

After completion of certain number of turns (\(N_{\text{turns}}\)) by all ants, a global pheromone updating rule (global updating rule, for short) is applied as depicted in the program flow chart in Fig-5 below. After the global updating, current set of ants removed and another set of ants start from the pick location to explore the place location. The process continues until the number of turns reach to the maximum number of turns (MAX_TURNS).
Note that, the parameter $N_{\text{turns}}$ has been set such that, most of the ants in the initial set were able to reach the place location [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006].

The pseudo-code of the ant colony algorithm used by the artificial ants is as follows [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]:

```
Initialize
  turn = 0
  turnsRemaining = $N_{\text{turns}}$ + 1
Loop
  Release a new set of ants from the starting point
  Loop
    turn = turn + 1
    turnsRemaining = turnsRemaining -1
    For each ant 'a' in the current set
      If ant 'a' does not reach target point Then
        Move to the next grid point using random propositional rule
      Else
        Ant 'a' stops exploring
      End If
    End For
  Until (turnsRemaining = 0)
  Apply the global pheromone update rule using ants that reached to the target point
  Update optimal path best so far
  Remove the current set of ants
  turnsRemaining = $N_{\text{turns}}$ + 1
Until (turn <= MAX_TURNS)
```
As mentioned earlier, the state transition rule used by the proposed ant system, called a random-proportional rule (probabilistic state transition rule), is given by equation (1)
below which gives the probability with which an ant \( k \) at node \( r \) chooses to move to the node \( s \) [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]:

\[
P_k(r, s) = \frac{[\tau(r, s)][\eta(r, s)]^\beta}{\sum_{u \in J_k(r)}[\tau(r, u)][\eta(r, u)]^\beta}, \text{ if } s \in J_k(r), \text{ otherwise } 0 \quad \text{(1)}
\]

Where \( \tau \) is the pheromone, \( \eta = \frac{1}{\delta} \) is the inverse of the distance \( (\delta) \) from point \( s \) to the target point, \( J_k(r) \) is the set of neighbor points of \( r \) that remain to be visited by ant \( k \) positioned on the point \( r \) (to make it easier), and \( \beta \) is a parameter which determines the relative importance of pheromone versus distance \( (\beta > 0) \).

As mentioned earlier, the global updating rule is implemented as follows:

Ants that were able to complete their tour within the number of allocated turns \( (N_{\text{turns}}) \), allow to update pheromone levels of their visited edges.

\[
\tau(r, s) \leftarrow (1 - \rho)\tau(r, s) + \sum_{k=1}^{m} \Delta \tau_k(r, s) \quad \text{………… (2)}
\]

Where:

\[
\Delta \tau_k(r, s) = \frac{1}{L_k}, \text{ if } (r, s) \in \text{ trip done by ant } k; \text{ otherwise } 0
\]

\( 0 < \rho < 1 \) is the pheromone reduction parameter due to evaporation, \( L_k \) is the length of the tour performed by ant \( k \), and \( m \) is the number of ants that were able to complete their tour within the predetermined turns \( N_{\text{turns}} \) [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006].
3.5 Data Flow Diagram (DFD) For the Ant System

Based on the detailed description, the following Data Flow Diagram (DFD) in Fig-6 below shows how the lift planning system works:

![DFD Diagram]

Fig-6 DFD for suggested Crane Lift Planning using Ant Colony Optimization Algorithm in Data Grid (Tessellated) Model

3.6 Principle of Grid Based ACO algorithm

To illustrate the principle of grid based ACO algorithm, the following simple example in Fig-7 below uses ACO in 2D grid [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]:

[Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]
Assuming that RAPID has generated two (2) routes starting from the pick location “K” leading to the place location “P” in addition to one additional route leading to the grid point C2 from the pick location K: R1 (K-A1-A2-A3-P), R2 (K-B1-P) and R3 (K-C1-C2). From the geometry of the location presented by the grid in Fig-7 below, and using math, we can find that \( R_1 = 2 \times R_2 \). Assume that initially ten (10) ants are at the pick location K and initial pheromone level for each edge is 100, \( N_{\text{turns}} = 20 \), \( \rho = 0.01 \), and \( \beta = 5 \).

These ants have to select one of the paths R1, R2, and R3 according to the random propositional rule represented by formula #1. If the distance between two neighboring grid points is 1 unit, then \( C_1-P = A_1-P = \sqrt{6^2 + 1^2} = \sqrt{37} = 6.083 \), and \( B_1-P = 5 \).
Probability of selecting the grid point A1 (or route R1) $p1$ (according to formula #1 above) [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]

\[
P_1 = \frac{100 \ast (1/6.083)^5}{100 \ast (1/6.083)^5 + 100 \ast (1/6.083)^5 + 100 \ast (1/5)^5} = 0.214374 \approx \frac{1}{5}
\]

\[
P_2 = \frac{100 \ast (1/5)^5}{100 \ast (1/6.083)^5 + 100 \ast (1/6.083)^5 + 100 \ast (1/5)^5} = 0.57125124 \approx \frac{3}{5}
\]

\[
P_3 = \frac{100 \ast (1/6.083)^5}{100 \ast (1/6.083)^5 + 100 \ast (1/6.083)^5 + 100 \ast (1/5)^5} = 0.214374 \approx \frac{1}{5}
\]
1. According to the probability values of P1, P2 and P3, the chances of selecting routes R1, R2, and R3 are 6, 2, and 2 ants respectively. Obviously route R2 (K-B1-P) takes the highest number of ants (6), since it has the highest probability.

2. When $N^{turn} = 20$ (i.e. after 20 turns), algorithm updates pheromone levels (according to formula #2 above) are as follows [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006]:

   a. Each edge on R2 route
      
      \begin{equation*}
      \text{edge on R2 route} = (1 - 0.01) \times 100 + 6 \times (1/6) = 100
      \end{equation*}

   b. Each edge on R1 route
      
      \begin{equation*}
      \text{edge on R1 route} = (1 - 0.01) \times 100 + 2 \times (1/12) = 99.17
      \end{equation*}

   c. Each edge on R3 route
      
      \begin{equation*}
      \text{edge on R3 route} = (1 - 0.01) \times 100 = 99
      \end{equation*}

   Since ants that followed the route R3 get lost because they are unable to find the place location “P”, the algorithm only evaporates the pheromone levels of the edges on that route [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006].

As we see in this example, a higher pheromone level for the shortest path (R2) than that in the path (R1) is set by the algorithm. In addition, the pheromone levels of lost path also decrease [Thantulage G.; Kalganova T.; Fernando W.A.C., 2006].

4. DISCUSSIONS AND CONCLUSION
In this chapter, tools and techniques used for path planning of crane lifts are discussed. An ant colony algorithm has been proposed for automatic 3D crane lift planning. The suggested system offered three (3) alternatives for representing the crane lifting work environment:

1. C-space (Configuration space) similar to the three (3) systems discussed in the previous chapter (III), where in this case the architecture of the proposed system would be much easy since it can be the same architecture of any system of the three (3) systems discussed in Chapter III, with only one difference. The difference is the optimization technique, where ACO replaces the heuristic (Depth-first search / A* in systems I / II) or the Genetic Algorithm (system III)

2. Grid-based representation: where a tessellated format of the solid model (includes all obstacles) is generated in a text file having .stl extension. The C++ library, RAPID (encapsulates the DOF of the cranes used in the lifting plan), is incorporated into the program for collision detections and finding the available lifting paths. The .stl file is passed to the algorithm as the RAPID can handle only the triangular shapes, where, the number of triangles used to approximate the obstacles dictates the accuracy of the collision detection.

3. Probabilistic Roadmap (PRM) representation: where a roadmap of paths in the free configuration space (C-space) is constructed in two phases: Pre-processing phase and the path planning phase. Each Degree Of Freedom (DOF) of the lifting crane is specified and implemented based on its range allowances, during the generation of a random configuration. Checks for collision with obstacles, loads and the crane itself are performed in the resulting configuration.
The ACO algorithm generates the optimal set of the lifting paths linking the pick and the place points.

Ant system has been proposed where the grid representation of the working space is assumed to be the selected option, where RAPID library finds all potential paths, then artificial ants are employed to perform the path optimization and provides planners with the optimal lifting path. The artificial ants’ system works similar to a multi-agent system where each ant performs a complete tour (i.e. traveling from the pick location to the place location) by choosing the grid points according to the random-proportional rule that selects the next node that has higher pheromone intensity.
CHAPTER V

CONCLUSION AND RECOMMENDATIONS

Productivity and reliability of heavy crane lifts planning will be improved by utilizing the computer power which allows crane lift planners to test feasibility and tolerances of different lifting paths in no time by means of simulating the critical parameters of the lift in a virtual environment. In this report, a literature review for four (4) path planning systems have been discussed in details in the first three (3) chapters. The first chapter discussed heavy lift planning in general and explored a heavy lift planning system (HELPS). Exploration of the other three (3) lift planning systems took place in Chapters III, after introducing the main concepts of automated lift planning systems in both single and two cooperative cranes. Chapter IV provided summary about software tools and techniques used in crane lift planning. Since Ant Colony Optimization (ACO), has not been utilized before in planning lift crane systems, chapter IV introduced a description of a suggested crane lift planning system that utilizes ACO in a crane lift planning system to obtain the optimal lifting path.

Based on the discussions of these different lift planning systems presented through this essay, it appears that the systems presented in the first three chapters (I, II, III) are limited in scope where the Degrees of Freedom (DOF) of the lifting crane(s) are limited to three (3-DOF). Although chapter III introduced cooperative lifting cranes systems with collision free, along with utilizing sophisticated AI searching techniques (such as heuristic and genetic algorithms), but there is still a major concern in the design of these
systems regarding the limitations of the utilized 2-D C-Space in order to model the working space along with the limited three degrees of freedom (3-DOF). This concern signals for the need of more advanced lift planning systems that can efficiently handle higher levels of cranes’ degrees of freedom.

Chapter IV attempts to address these concerns by exploring some advanced techniques in representing working space (Probabilistic Roadmap “PRM” and Grid representation), and optimizing crane lifting path (Ant Colony Optimization “ACO” algorithm). These techniques have been effectively employed in building systems for planning and automating robotic motion paths, but have not been utilized yet in planning crane lifting path. The suggested lift planning system discussed in chapter IV utilizes ACO for path optimization as an advanced AI tool. Despite of its scope limitation which was also limited to three degrees of freedom (3-DOF), but grid representation of the work space has been utilized to provide a more convenient working environment for the optimization ants to work with. Furthermore the system’s design can be easily enhanced to handle higher degrees of freedom (6-DOF or more), if Probabilistic Roadmap (PRM) representation of the working space is used.

Also the design and implementation of the ant’s motion within the ant system (discussed in the proposed planning system) can be enhanced to reflect the movement of two or more cooperative cranes under the DOF restriction of each cooperative crane in regards to the cranes’ positions. [Mohamad, M.M.; Taylor, N.K.; Dunnigan, M.W. 2006]. Therefore to reach a design of collision free cooperative crane lifting path planning
system that handles high levels degrees of freedom (possibly all the 8-DOFs), the modified proposed system might be the right option.

The following section provides conclusion and recommendations specific to each chapter:

In **Chapter I - Heavy Lift Planning in Construction**, eight (8) factors (criteria) were identified as main factors that determines whether it is possible to use heavy crane lifts or not, and were discussed in detail. It has been found that following these eight (8) criterion results in performing a manual path planning for heavy lifts. Description of a general path planning for heavy lifting system has been suggested. Also Walkthru development tool was selected for implementing the HEavy Lifting Planning System (HELPS). Because Walkthru is a self contained tool that has enough advantages qualifying it to perform planning path for heavy lifts. Walkthru allows: controlling user’s position (lifting position); providing controlled movement and positioning of loads and objects within the 3D model; defining the natural degrees of freedom and master-slave relationships required to simulate the realistic motion of the lifting crane; recording a sequence of crane moves and replaying the recorded moves when needed; and detecting interference between objects using its on-line function

However, some limitations in the discussed version of HELPS were found, overcoming these limitations will increase the productivity of the planning task. Recommendations regarding this system should focus on handling HELPS limitations discussed in Chapter
I; expanding the system’s capabilities to perform multiple crane lifts; adding the archiving feature so in order to maintain a history of lift cases; and exploring the potential of implementing the system in Virtual Reality environment.

**Chapter II - Cooperative Crane Lifts** discusses path planning for cooperative crane lifts, specifically when there are two cooperative cranes doing the lift. Concepts of cooperative crane lift have been discussed specially handling Degrees Of Freedom (DOF) of the two cooperative cranes, and modeling the lifting site. Discussions assumed three (3) DOF for each of the two cooperative cranes, and also recommended using C-space rather than real space to model the lifting site because of its advantages such as its ability to produce unique representation of each configuration; its ability to specify constraints in the manipulator’s movement; its ability to evaluate path for different pick locations; and most important it is free from the inverse kinematics problem.

In **Chapter III - Automated Path Planning of For Mobile Cranes**, automation of path planning process for crane lifts was discussed in both single and cooperative cranes. The effective use of C-space (configuration space) concept, Artificial Intelligence (AI) techniques (heuristic search, and Genetic Algorithms) in planning and optimizing the crane lifting path, would help building robust automated systems that are appreciated by path planners and user. This comes in the cost of implementing efficient software and hardware systems for both C-space generation and searching techniques.
Also cost-effectiveness of heavy lifts can be improved by the use of cooperative cranes. However, the complexity of developing a reliable lift plan enforces the utilization of the computing power and its aided planning systems. The work presented in this chapter represents an initial step toward the development of such automated systems.

1. Chapter III discussed three (3) different automated path planning systems: a) Automated path planning system for a single mobile crane lift; b) Automated path planning system for cooperative crane lift using Heuristic search; and c) Automated path planning system for cooperative crane lift using Genetic Search Algorithm. Recommendations in this chapter focus on the usage of AutoCAD’s Runtime Extension (ARX) module instead of AutoLisp to enhance the implementation efficiency of System (a); developing efficient heuristic logics to improve the search; consider modeling higher levels of Degree Of Freedom levels; explore alternate searching algorithms that produce more efficient results such as ACO and GA; resolving the limitations of the three (3) systems that have been discussed in the chapter; investigating the improvement of the execution speed when performing a search in a finer resolution; produce higher dimensions of C-Space that consider telescoping and mobility of crane components; experiments with powerful hardware, improved heuristics based on clustering techniques, parallel processing, and alternate search methods such as genetic algorithms; investigate the suitability of the GA approach for more complex cooperative manipulator applications where more degrees of freedom DOF of each cooperative manipulator are considered (such as 2 x 4 i.e. two manipulators each with 4 DOF).
1. **Chapter IV – Tools and Techniques For Automated Path Planning**, discusses tools and techniques used in path planning systems for crane lifts in general and specifically the system tools that are discussed through the report. Ant Colony Optimization (ACO) algorithm has been utilized in several path planning and research areas involving robots, and it proved its power and efficiency in this area. However it has not been used before in path planning for crane lifts. Therefore ACO was proposed in this chapter for automatic 3D crane lift planning. The suggested system offered three (3) alternative options for modeling the work environment of crane lifting:

a) C-Space representation similar to that used in the three (3) systems discussed in Chapter III. If this option is selected, then the suggested system can easily be designed similar to anyone of the three systems presented in Chapter III, where the second heuristic or genetic search would be replaced by ACO searching algorithm to perform optimization of the feasible lifting path that has been generated by the first heuristic search.

b) Grid-based representation: where a tessellated format of the solid model (includes all obstacles) is generated in a text file (with .stl extension). The C++ library, RAPID (Robust and Accurate Polygon Interface Detection) which encapsulates the DOF of the cranes used in the lifting plan, is incorporated into the program to perform collision detection and searching for the feasible lifting paths.

c) Probabilistic Roadmap (PRM) representation: where a roadmap of paths in the free configuration space (C-space) is constructed in two phases: Pre-processing phase and the path planning phase. Each Degree Of Freedom (DOF) of the lifting
crane is specified and implemented based on its value range, during the
generation of a random configuration. Checks for collision with obstacles, loads
and the crane itself are performed in the resulting configuration.

The ACO algorithm generates the optimal set of the lifting paths that are linking the pick
and the place points

An ant system has been proposed based on the grid modeling of the working space,
which employs artificial ants to perform the path optimization and provides planners with
the optimal lifting path. The artificial ants’ system works similar to a multi-agent
intelligent system where each ant performs a complete tour (i.e. traveling from the pick
location to the place location) by choosing the grid points according to the random-
proportional rule that selects the next node that has higher pheromone intensity.

General recommendations regarding the design of planning path systems for lifting
cranes might include investigating optimization tools that utilize sophisticated AI
techniques such as ACO algorithms for co-operative manipulators, and deal with higher
degrees of freedom (i.e. 2 x 4; 2 x 5; 2 x 6; 2 x 7; and 2 x 8). Also utilizing the power of
computer’s multiprocessing, and taking advantage of the low cost of RAM chips, disk
storage, and computer hardware in order to build powerful computing systems that
provide high performance simulations for complex lifting scenarios involving more levels
of crane’s Degrees Of Freedom.
APPENDIX A

Hill Climbing

The logic for a hill climbing search used in Chapter III as presented in [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003] is as follows:

1. Set the current crane configuration to the pick configuration
2. Generate neighbors for the current configuration
3. Perform feasibility check for the neighbors generated
4. Identify the feasible neighbors
5. Compute the heuristic search function for the feasible neighbors
6. Choose the best neighbor based on the heuristic function value
7. Set the current configuration to the neighbor chosen
8. If the current configuration is equal to the place configuration, stop, and go to step 9

   or else go to step 2
9. Print the path

A* Search

The logic for A* search used in Chapter III as presented in [Sivakumar Pl.; Varaghese K.; Ramesh Babu N. 2003] is as follows:

1. Declare open list and closed list to store the nodes that are expanded and ready for expansion
2. Compute the heuristic function value of the pick configuration and add it to the open list
3. Identify the best configuration in the open list
4. Remove that configuration from the open list and add it to the closed list
5. If the best configuration is equal to the place configuration, go to step 9
6. Generate neighbors for the best configuration and perform a feasibility check for all
7. Compute the heuristic function value for each feasible neighbor and add it to the open list
8. Point these neighbors toward the best configuration to retain the parent-child relationship, and go to step 3
9. Trace the path from place to pick configuration
10. Print the path
Keywords

Algorithm
Ant Colony Optimization (ACO) Algorithm
A* Search Algorithm
Configuration Space (C-space)
Construction
Cooperation
Crane; Equipment
Degrees Of Freedom
Depth-First Search
Heuristic Search
Hill Climbing Search Algorithm
Inverse Kinematics
Lift; Lift Path
Manipulator
Model; Modeling
Planning
Roadmap Model
Robot; Robotic
Site
Search
Tessellated Model
REFERENCES

1- Development of Computerized HEavy Lift Planning System for Crane Lifts (HELPs)
by “Koshy Varghese, Parmanand Dharwadkar, John Wolfhope & James T. O’Connor
1997”

2- A Computer-AIDED System For Planning and 3D Visualization of Multiple Heavy
Lift Operations by “Satyanarayana Reddy D, Koshy Varghese, Srinivasan N 2007”

3- Automated Path Planning of Cooperative Crane Lifts Using Heuristic Search by “PL.
Sivakumar; Koshy Varghese; and N. Ramesh Babu 2003”

4- Collision Free Path Planning of Cooperative Crane Manipulators Using Genetic
Algorithm by “M. S. Ajmal Deen Ali; N. Ramesh Babu; and Koshy Varghese, 2005”

5- Automated Path Planning for Mobile Crane Lifts by “H. Raghunatha Reddy & Koshy
Varghese 2002”

6- CodeProject: Genetic and Ant Colony Optimization Algorithms, by Peter Kohout.
2003

7- Robot Path Integration in Manufacturing Processes: Genetic ...

8- A Grid-based Ant Colony Algorithm for Automatic 3D Hose Routing, by Kohout P.
2006

9- Articulated Robot Motion Planning Using Ant Colony Optimizations by Mohamad,
M.M.; Taylor, N.K.; Dunnigan, M.W. 2006

10- Genetic algorithm with ant colony optimization (GA-ACO) for ...

11- Ant colony optimization - Wikipedia, the free encyclopedia

12- Automated Planning And Visualization Of Mobile Crane Operations Based On
Building And Schedule Information by “Kevin Tantisevi and Burcu Akinci”
13- **Experiments in Real-Time Spatial Data Acquisition for Obstacle Detection** Jochen Teizer, Katherine A. Liapi, Carlos H. Caldas, and Carl T.


16- **Automation in Construction - Journal of Information Technology in Construction**

17- **Computer-Aided Civil and Infrastructure - Engineering Journal of Construction Engineering and Management**


19- **Ant-Communications**

20- **Marco Dorigo: Ph.D.Thesis Optimization, Learning and Natural Algorithms**