TOTAL COMPLETION TIME MINIMIZATION IN A DRILLING SEQUENCE PROBLEM CONSIDERING TOOL WEAR: AN ANT ALGORITHMS APPROACH

A Thesis by

Annamalai Murugappan

Bachelor of Engineering, Annamalai University, 2000

Submitted to the College of Engineering
and the Faculty of the Graduate School of
Wichita State University in partial fulfillment of
the requirements for the degree of
Master of Science

December 2005

TOTAL COMPLETION TIME MINIMIZATION IN A DRILLING SEQUENCE PROBLEM CONSIDERING TOOL WEAR: AN ANT ALGORITHMS APPROACH

We have examined the final copy of this Thesis for form and content and recommend that it be accepted in partial fulfillment of requirements for the degree of Master of Science, with a major in Industrial Engineering.

____

Dr. Seyed Hossein Cheraghi, Committee Chair

We have read this Thesis and recommend its acceptance:

____

Dr. Krishna Krishnan, Committee Member

____

Dr. Behnam Bahr, Committee Member

DEDICATION

To my beloved parents

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my advisor Dr. Seyed Hossein Cheraghi for his patient guidance, advice, precious time and constant support throughout my graduate career at Wichita State University, without which my efforts would not have been complete and fruitful.

I would like to thank Dr. Krishna Krishnan and Dr. Behnam Bahr for their precious time, helpful comments and suggestions.

I would like to thank Krishnan Krishnaiyer for his valuable inputs for this thesis also I would to thank all my friends, who encouraged and supported me during my graduate studies.

Finally, I would like to thank my family, for their moral support and love, as well as their encouragement and example, which motivated me to further my education.

ABSTRACT

Drilling is one of the most common machining operations. An Aircraft skin consists of hundreds of different-sized holes distributed over a large area. Automated drilling/riveting machines are used today to perform the drilling/riveting process on large aircraft skins. These machines are capital intensive and their maximum utilization is vital to their economic viability. An issue that affects the utilization of these machines is the drilling sequence because usually there is 'n' number of holes that has to be visited. Determination of drilling sequence is similar to a Traveling Salesman Problem (TSP) and exhibits characteristics of an NP-hard problem. Two types of setups complicate this process further. Depending on the size of the holes, different-sized drills are selected, each size requiring a different setup for changing the tool. Also, as holes are drilled, the drill bits wear out and need to be replaced at the end of their tool life. This thesis presents an Ant-algorithm meta-heuristic to solve this sequencing problem. Results indicate that the procedure is effective in arriving at good solutions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE **PAGE**

CHAPTER I

INTRODUCTION

1.1 Introduction to the problem

Drilling is defined as “An implement with cutting edges or a pointed end for boring holes in hard materials, usually by a rotating abrasion or repeated blows” (Dictionary, 2005). It has been estimated that 95% of all machined parts have holes in them. In general, drilling operation on any part is very critical, particularly in many aircraft manufacturing process.

Aircraft skin consists of hundreds of different-sized holes distributed over a large area. Automated drilling/riveting machines are used today to perform the drilling/riveting process on large aircraft skins. These machines are capital intensive and their maximum utilization is vital to their economic viability. Complexity of drilling depends on many factors includes the number of holes, the machine, the material, and the method. Drilling operation time is an important factor that consists of cutting time and travel time. Cutting time is the time required for the drill bit to cut through the panels and retract fully at a hole location (Linn, 1998). Also in aircraft industries other important factors that affect the cutting time are the required hole quality. Drill travel time, on the other hand, is basically the time taken for the drill bit to move from one hole location to another. Owing to large number of holes and rivetings involved in aircraft parts, drill cutting time, travel time and sequencing become important process parameters. An effective sequencing minimizes production and setup time. In the lean manufacturing terms the travel time is “Muda”, and is one of the classic forms of “Seven waste”. Since it is a non-value added process, minimizing drill travel time can increase production capacity. Further more;

adopting an optimal drilling sequence could increase the throughput. Sequencing of the drilling operation is very challenging and difficult because usually there is ‘n’ number of holes that has to be visited. Determination of drilling sequence is similar to the Traveling Salesman Problem (TSP) problem, since in a TSP problem the salesman has to visit all the cities situated at various places. It is well known that TSP is a classical example of a NP-Hard problem (Linn, 1998). Adding two types of setups complicates this process further. Depending on the size of the holes, different-sized drills are selected, each size requiring a different setup for changing the tool. Also, as holes are drilled, the drill bits wear out and need to be replaced at the end of their tool life.

Generally heuristics are used for solving complex applications where classical optimization techniques cannot be applied. Meta-heuristics are used as a powerful tool for arriving at optimal or near-optimal solutions to any NP hard problem. According to Rardin and Uzsoy (2001) “heuristic optimization algorithms seek good feasible solutions to optimization problems in circumstances where the complexity of the problem or the limited time available for its solution does not allow exact solution”. There are several meta heuristics like Ant Colony Optimization, Genetic Algorithms, GRASP (Greedy Randomized Adaptive Search Procedures), Memetic Algorithms, Reactive Search, Scatter Search, Simulated Annealing, Tabu Search. In this thesis, an ant algorithm is used to provide a practical solution to a drilling sequencing problem.

Scientists have been fascinated by the orderly behavior of social insects such as ants and bees for centuries. Insects solve problems in a very flexible and robust way. Their daily task of searching for food, construction of nests, divisions of labor between various individuals is very effective. They adapt to abrupt changes in environment and continue functioning even when individuals fail to achieve their tasks. The simple and adaptive collective behavior of these social insects can be of significant importance in solving engineering problems, where their behavior can be designed as algorithms of optimization, routing and distributed control, from which powerful and notoriously difficult optimization problems have been solved.

The intelligence of insects have found its application in companies where there has been experiments with new practices and developing software that incorporates valuable lessons from these creatures. The traditional approach to sequencing has been broken by software such as “Ants”, which lays pheromone trials to optimum locations and movements within production facilities.

1.2 Research Focus and Objective

Sequencing problems attract a lot of attention in the computer science world because they are encountered in day-to-day life and are complex. A real life problem of arriving at an optimal drilling sequence for a set of holes to be drilled at specific Cartesian coordinates is considered. Adding the constraints of setup changes due to tool wear and changes in specified hole sizes makes this a complex NP-hard problem. Ant-algorithms meta-heuristics have been proved as a successful tool to solve such problems. This thesis considers setup changes due to both tool size changes and tool wear. A web-based ant algorithm meta-heuristic is proposed to solve this sequencing problem. It is anticipated that the proposed technology will lead to reduced drilling cycle-times, increased efficiency and lower cost in drilling operations. Nature seems to have an inherent capability for computation.

The objective is, “Given a set of ‘n’ holes in a part with different hole sizes distributed across the Cartesian plane, minimize the total completion time by considering the set up changes due to tool wear and change in drill sizes.”

1.3 Thesis Organization

The work has been organized in to five chapters. The first chapter introduces the problem and explains its significance. Chapter Two provides an overview and prior research done in Ant colony optimization, the area of total completion time minimization in a drilling sequence. Methodology used in this research is presented in the third chapter, with the implementation; the results are discussed in the fourth chapter. The final chapter presents the summary and conclusions of the research.

CHAPTER II

LITERATURE REVIEW

2.1 Background

Drilling is one of the basic and most common machining operations (Chandrasekhar et al., 1997). Drilling is essentially a hole producing process and constitutes major portion of overall machining operation. Drilling is a highly prioritized task especially when it is performed as an end operation in fabricating a part. Poor drilling can result in complete elimination of entire part itself and result in higher production and waste cost (Strenkowski et al, 2004). In a multi-product environment, with different setup times and processing times for individual product types, the demand for drilling economically and to produce quality holes still remains a difficult task to achieve. Any drilling process is greatly influenced by parameters such as ‘Setup Time’ (time required for the operator to alter one tool type to another. For example, drills with varying diameters), ‘Processing Time’ (time taken by a specific machine to execute a single drilling operation; it directly impacts tool wear) and ‘Tool Life’ (length of time until after which the drill would not be able to produce quality holes to specification).

Literatures on machine scheduling have not extensively addressed the issue of tool change. Even if it is considered, previous researches ignore the change induced by tool wear and rather assume that part mix alone drives the need for tool change. But this is not the case with real life situations as tool changes due to tool wear are ten times as frequent as those due to part mix (Gray et al., 1993). Developing an optimal drilling sequence that takes into consideration all the above influential factors is the complicacy involved in drilling process. Under these complex parameters many heuristics such as

Ants Algorithms have been deployed to arrive at optimal solution. A brief introduction of Ants Algorithms is discussed in the next part of this chapter.

2.2 Ant Algorithms

Generic heuristic methods are also called as ‘Meta heuristic’ or general local search methods have taken a rapid growth in past decades. They have undoubtedly captured the focus of most researchers in variety of fields. The success of these methods can be attributed to factors like their ease of implementation, their ability to consider specific constraints that fit practical applications and the superior quality of solutions they generate. Among the most flourishing techniques, some widely used methods are Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search (TS) and Ant Systems (AS) (Eric et al, 2000).

As the name implies, ant algorithm imitates the foraging behavior (behavior of wandering in search for food) of natural ants to search high quality solutions for hard combinatorial optimization problems. Besides the fact that ants live in colonies, they have got some remarkable characteristics. Even their very survival depends on the collective behavior of the colony as a whole. They benefit together as a colony rather than as a single individual of the society. Though individual ants have no special abilities, their indirect mode of communication (using chemical substances called pheromones) enables the entire ant colony to execute complex tasks, such as setting up the shortest route paths from their nests to feeding sources (Silva et al, 2001).

Small groups of ants in search of food head off from its nest by choosing a random path to get to the food source. The ants that pick up the shortest path release a strong trial of pheromone on their path and make it possible for other members of ant colony to trace the shortest path. The mechanism behind the strong trial of pheromone is that the ants that took the shortest path to food source return to the nest at the earliest. They travel along the shortest path twice (to go from the nest to the source and to return to the nest), thus, depositing more amount of pheromone on the shortest path than on longer paths. This theory helps the ant colony to opt the shortest pathway, in the midst of alternative pathways, from their nest to a food source. As a replicate of the above process, ant algorithms use artificial ants (mobile agents) to generate solutions for NP hard problems and these artificial ants direct their search based on the information collected during past simulations. The available information is constantly modified and updated through the environment to obtain enhanced solutions and to avoid early convergence of bad solutions Dorigo & Gambardella (1999); Hozefa & Bonabeau (1998).

2.3 Types of Ant Algorithms

An ACO algorithm is comprised of a number of cycles (iterations) of solution construction. Construction of complete solutions is carried out by a specified number of ants (the number of ants is a parameter) during each iteration, using the heuristic information at hand and the collected experiences of previous groups of ants. A digital analogue of trial pheromone deployed on the constituent elements of a solution represents the collected experiences of ants. The quality of solution obtained at the end of each iteration determines the amount of pheromone deposited. Typically, the initial phase of solution construction contains small deposits of pheromone whereas the final phase contains larger amounts of pheromone. Each ant performs a probabilistic pick from the set of available elements at each step of iteration and the element selection is based on a combined measure of pheromone deposits on that element and the heuristic value of that element's utility. Higher the amount of pheromone and heuristic value, an element is more likely to be selected.

Ant Colony Optimization (ACO) meta-heuristic was initially applied to obtain improved solutions for Traveling Salesman Problems (TSP). There exist a lot of coincidence between natural ants identifying its shortest path to food and artificial ants identifying the minimum cost paths for a Traveling Salesman. Dorigo, Bonabeau and Theralaz (2000) modeled TSP as a function of (C, L, W) subject to a set of constraints Ω where Ω is a function of (C, L, θ). Hence for the TSP, C is the set of cities, L is the set of links between cities, W is the weight of the links and Ω is the set of constraints imposed on the elements of C and L at a given time θ (Montgomery, 2002).

Gagne et. al. (2002) used TSP as a basis for modeling the single machine scheduling problem where the objective is to minimize the total time taken to process all jobs (the make span). The ACO algorithm yielded first-rate results equivalent to (in some cases better solutions) solutions obtained by other heuristics for the same problem. Furthermore, the computational time taken by ACO algorithm was appreciably lower.

In addition to the above applications, Ant algorithms can also be applied to work out Quadratic Assignment Problems, Vehicle Routing Problems, Graph Coloring Problems and Shortest Common Super Sequence Problems. Ant algorithms can be considered perfect examples of swarm intelligent systems. They also prove the fact that applying basic principles of particular natural phenomena to diverse optimization tasks can yield good quality solutions (Silva et al, 2001). Even if ACO algorithms cover up wide areas of application, the mainstream of problems handled by ACO are static and well defined combinatorial problems, that is, problems for which all the required information is on hand and the available information remains unchanged during problem solution. So it becomes exceedingly important to test the strength of ACO algorithms after systematically applying them to ill-structured problems, or to highly dynamic domains. Research efforts on this area are already on (Dorigo & Stutzle, 2000).

2.4 Total Completion Time

Total Completion Time of a given sequence of operation is the total sum of the processing times for individual operation with the sum of time taken to complete the previous operations. Processing time being a dominant factor in determining whether a station is bottleneck or not, can be influenced by parameters like cutting speed and feed rate which in turn directly impacts tool wear.

Chandru et al modeled the scheduling process of large scale integrated circuits in burn-in ovens considering the ovens as batch processing machines. The heuristic algorithms they developed provided superior values of minimized total completion times (equivalent to average job completion time) for both single and parallel machine problems (Chandru et al, 1993).

Allahverdi and Aldowaisan in their work "New heuristics to minimize total completion time in m-machine flow shops" compared the relative performance of existing heuristics and improved their error performance with least CPU time by proposing slight modifications to those algorithms through pair wise exchange rule. Besides this, they also built up several new heuristics which outperformed the modified existing heuristics in terms of both error value and CPU time (Allahverdi & Aldowaisan, 2002).

Amos Fiat and Gerhard Woeginger dealt with online scheduling of n jobs on a single machine with the objective to minimize the total job completion time. They formulated the problem in such a way that a job has to be scheduled immediately and irrevocably upon its arrival and no information about the job is known prior to its arrival. The quality of online algorithm was measured using a competitive ratio (Fiat & Woeginger, 1999).

The treatment of single machine scheduling problem where processing is interrupted due to tool wear & tool changes as well and the purpose as is common with all scheduling problems is to minimize the total job completion time (Akturk et al, 2003).

The problem taken for consideration by Akturk is structured using the following assumptions and is formulated as a mixed integer program.

- Presence of single CNC machine and n independent jobs
- Processing times of jobs are constant and known
- Only one type of tool with known constant life and unlimited availability is used
- Time for tool change is constant and known
- Tool change is carried out only at the end of a process and not during the process

The objective function for this problem is designed in such a way that it bears some useful insights into scheduling issues. Total completion time of a job has been partitioned into two components – one showing the total completion time without tool changes and the second showing the increase in job completion times as a result of tool changes. Since the problem formulated with all its constraints prove to be NP-hard, it cannot be solved beyond small instances and the need for heuristic algorithms arise. Various heuristic algorithms have been proposed to solve the problem and the relative performance of proposed heuristics is studied using a simple case study (Akturk et al, 2003).

Various heuristic algorithms are used to solve the NP- hard scheduling problem such as (1) Shortest Processing Time (SPT) Rule, (2) First Fit Decreasing (FFD) Rule, (3) Modified First Fit Decreasing (MFFD) Rule, (4) Expected Gain Index (EGI) Heuristic, (5) Knapsack (Knap) Heuristic, (5) Two Bin (2 Bin) Heuristic, and (6) Genetic Algorithm with Problem Space Search (GAPS) (Akturk et al, 2003).

It was observed from case study that SPT & FFD dispatch rules were outperformed by other proposed heuristics and the optimal solution computed by mixed integer program turned out to be different from all algorithms. Though 2 Bin and GAPS heuristics gave better results than others did, GAPS remained clearly the best (Akturk et al, 2003).

Akturk's work too has a few limitations and those can be regarded as future research directions. Some of the limitations of Akturk's study are

- Multiple tool types and different tooling requirements (depends on part mix) for jobs are not allowed.
- Absence of different scheduling criteria such as weighted completion time and weighted tardiness
- Fixed tool life and processing times for jobs (one can fit some distributions to these variables).

Krishnaiyer & Cheraghi (2003) have adopted Ant Algorithms for solving problems in real life manufacturing systems. The problems include (1) Drilling sequence optimization problem, (2) Single-machine scheduling optimization considering tool wear and (3) Single-machine scheduling considering total job changeover cost. The effectiveness of proposed approach is evaluated by performing relative analysis with solutions obtained using other existing Meta heuristics in the literature such as genetic algorithms. The authors have come up with a noticeable conclusion that Ant Algorithms is preferred over other Meta heuristics as it provides high level of scheduling flexibility and swift problem solving capabilities. The web-based implementation of proposed algorithm is considered advantageous as it eliminates the need for installing individual copies of algorithms on specific stand-alone computers. Rather it is bound by client-server architecture flexibility with a single installation of proposed algorithm in a server.

Single machine scheduling problem is a classic problem and has been the focus of numerous researches so far. Tool management is a key element that is closely associated with single machine scheduling and it comprises of tool scheduling, tool wear & tool replacement. Earlier researches in this area attribute the need for tool replacement only to part mix and ignore the emphasis on tool wear. But the proposed algorithm takes into consideration the tool wear and ensures that tool wear triggers the need for tool replacement. So, the problem under focus is modeled as, given a set of "n" independent jobs that are to be processed on a single machine with known individual processing times for each job, the objective is to minimize the total completion time of jobs. Also, the formulation takes the assumption that only one type of tool is used and unlimited number of tools of the specified type is present. After making required changes to WACSA (Web-based Ant Colony System Algorithm) methodology such that it lodges the tool wear and tool change constraints present in the problem, the algorithm was simulated for maximum permissible iterations to locate the global best job sequence. And the results are then compared with the best solution obtained using GAPS (other Meta heuristics) for the same problem. It is observed that the proposed algorithm resulted in a solution equal to the optimal solution obtained by GAPS but with a different sequence of jobs. Besides that, it also offers a number of alternate schedules with close optimal solutions. In case the shop floor constraints don't permit the execution of best schedule, the other alternatives can be investigated for implementation and this phase of work in particular adds high level of flexibility to WACSA (Krishnaiyer & Cheraghi, 2003).

Optimizing the drilling sequence is vital for diminishing the fluctuations in total time to complete all drilling. One of the most influential factors of total drilling time is the drill travel time, which can be defined as the total time taken for the drill to move from one location to another. As it is evident that drill travel time is directly related to the sequence the tool follows, the objective of this problem is to optimize the drilling sequence for a set of jobs that minimizes total production time. Three Meta heuristics with different evaluation criteria (1) Total travel time criteria, (2) Number of setups criteria and (3) Total distance traveled criteria have been proposed to solve this problem and option 1 yields the best sequence with least total production time. Hence, it is concluded that optimizing both the setup and travel time significantly impact the total drilling time. The solution obtained is close to the optimal one found using other heuristics and the proposed algorithm proves to be efficient as it took only a couple of seconds to arrive at the best solution (Krishnaiyer & Cheraghi, 2003).

CHAPTER III

METHODOLOGY AND IMPLEMENTATION

3.1 Problem Definition

The problem under study can be formally defined as follows:

Given a set of n holes with varying sizes distributed across a cartesian plane, the objective is to arrive at a drilling sequence to minimize total completion time considering the set up changes due to tool wear and change in tool size. A web-based ant algorithm meta-heuristic is developed to solve this scheduling problem. This chapter discusses the proposed Ant Colony Drilling Sequence (ACDS) suitable for drilling sequence optimization.

3.2 Ant Colony Drilling Sequence (ACDS)

Figure 1 shows a detailed flow chart of the ACDS methodology. The structure of the ACDS has five phases (1) Input, (2) Initialization, (3) Ant Generation, (4) Ant Walk, and (5) Output.

3.2.1 Input Phase

In the input phase user enters important parameters that guide the ants to make consecutive movements such as relative importance of trial (α), relative importance of visibility (β), trial Persistence (Rho), relative importance of exploitation (qo) and other input parameters like processing time (priori) of all holes, tool change time, hole sizes, tool life and cartesian coordinates of hole positions. The number of ants required for simulation is set to equal to the number of hole positions.

3.2.2 Initialization Phase

In the initialization phase, the distance between the nodes are calculated and stored in distance matrix. The distances between any pairs of nodes are generated by the distance formula. The distance D between any two points X_1, Y_1, and X_2, Y_2 is given by the following equation:

$$D = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \tag{1}$$

Based on the distance matrix the initial pheromone value (τ_0) is initialized and calculated based on the following equation:

$$\tau_0 = \frac{1}{(\sum_{i=1}^{n-1} D_{i,\ i+1}) \times n} \tag{2}$$

Where:

(τ_0) - Initial Pheromone Value

n - number of nodes

$\sum_{i=1}^{n-1} D_{i,\ i+1}$ - Sum of all the distances

A large positive value is assigned as the best initial tour length. This is done so that the result of the first iteration would become the current best and is established as a baseline for further comparison.

3.2.3 Ant Generation

Ant generation consists of creating a starting hole matrix, a visited hole matrix, and a tour matrix for each ant. Only one ant is placed randomly at each hole at the beginning of the simulation. A number that lies between 0 and 1 is randomly generated. By multiplying the generated random number with the number of holes, an initial hole is randomly chosen for each ant. The value obtained is rounded up to the nearest integer. This hole is inserted into the initial starting hole matrix and the tour matrix for each ant. The visited hole matrix is then updated with a value of 1.

3.2.4 Ant Walk

The next hole that ant selects is based on the distance from the present hole to the next, and the amount of trail (pheromone) intensity that is between one hole to that of the other hole. A random number, "q" (exploration) is generated between 0 and 1, which is compared with "q_0" (exploitation). The movement of ants from one hole to another is controlled by the following equation pseudo-random-proportional action choice rule or state transition rule "L" (equation 3) and the probability transition rule "S" (equation 4).

$$L = \begin{cases} \underset{j \notin Tabu_k(i) \neq 1}{\arg\max} \left\{ (\tau_{ij})^{\alpha} \, x (\eta_{ij})^{\beta} \right\} & if \quad q \leq q_0 \\ S & Otherwise \end{cases} \tag{3}$$

$$S = P_{ij}^{k}(t) \begin{cases} \dfrac{(\tau_{ij})^{\alpha} \, x (\eta_{ij})^{\beta}}{\sum_{j \notin Tabu_k(i) \neq 1} (\tau_{ij})^{\alpha} \, x (\eta_{ij})^{\beta}} & if \quad S \in Tabu_k(i) \neq 1 \\ 0 & Otherwise \end{cases} \tag{4}$$

Where

α is relative importance of trial and β is relative importance of visibility

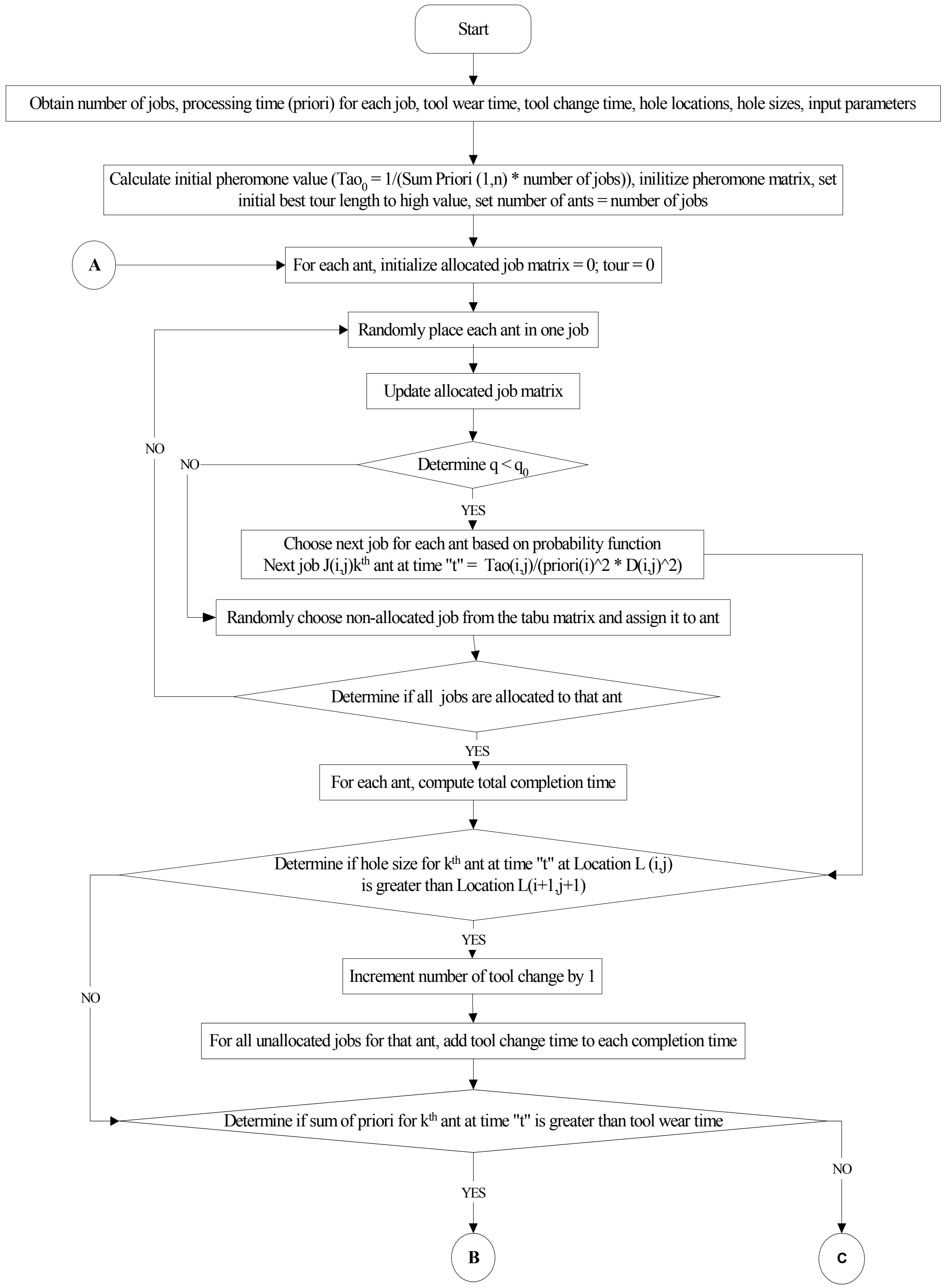
The tour matrix is updated with a new hole number after the ant moves to the next node. At the new hole's location, the hole that has already been visited will be shown by "1". The pheromone trail intensity on each edge is updated after each ant has successfully completed one movement, using the "local pheromone update". The local pheromone is updated until all ants visit all holes once. This sequence constitutes a valid tour. The length of tour for each ant is computed by using the tour and distance matrix. All of these values are arranged in ascending order, with the lowest value stored as $L_{BestIteration}$. The next step is the global pheromone updating process. At this stage, each ant would have successfully completed one valid tour, the "ant walk".

The Ant walk stage accommodates the tool change due to tool wear and change in tool size. After holes are assigned to each ant the total completion time for each ant is computed. The hole size at any new location is compared with the previous hole. If the hole size is different the number of tool change is increased by 1 and the tool change time is added to all unallocated holes. If it is of the same size the tool remains unchanged, the sum of the processing time is calculated to check whether the tool life of that particular tool has exceeded. If the processing times exceed the tool life, the tool is changed and the tool change time is added to the total completion time for the remaining unallocated holes.

The allocation of holes continues until all holes are allocated. The total completion time and sequence is stored as the best sequence. After each cycle the total completion time for the current iteration is compared with the best sequence and the lowest total completion time is taken as the new best value. The only stopping criterion in the program is the maximum iteration value. The cycle continues until the maximum iteration value is reached.

3.2.5 Output

Output stage prints out the best tour matrix for the entire iteration. Distance matrix is used to calculate the length between any two pairs of holes and added together to give the tour's length. The entire process computation time is also displayed. Also ACDS does not need previous best solution for its algorithm, and in most cases, previous best-optimized solution will be unknown.


Start
Obtain number of jobs, processing time (priori) for each job, tool wear time, tool change time, hole locations, hole sizes, input parameters
Calculate initial pheromone value (Tao0 = 1/(Sum Priori (1,n) * number of jobs)), inilitize pheromone matrix, set initial best tour length to high value, set number of ants = number of jobs
A
For each ant, initialize allocated job matrix = 0; tour = 0
Randomly place each ant in one job
Update allocated job matrix
NO
NO
Determine q < q0
YES
Choose next job for each ant based on probability function
Next job J(i,j)kth ant at time "t" = Tao(i,j)/(priori(i)^2 * D(i,j)^2)
Randomly choose non-allocated job from the tabu matrix and assign it to ant
Determine if all jobs are allocated to that ant
YES
For each ant, compute total completion time
Determine if hole size for kth ant at time "t" at Location L (i,j) is greater than Location L(i+1,j+1)
YES
NO
Increment number of tool change by 1
For all unallocated jobs for that ant, add tool change time to each completion time
Determine if sum of priori for kth ant at time "t" is greater than tool wear time
NO
YES
B
C

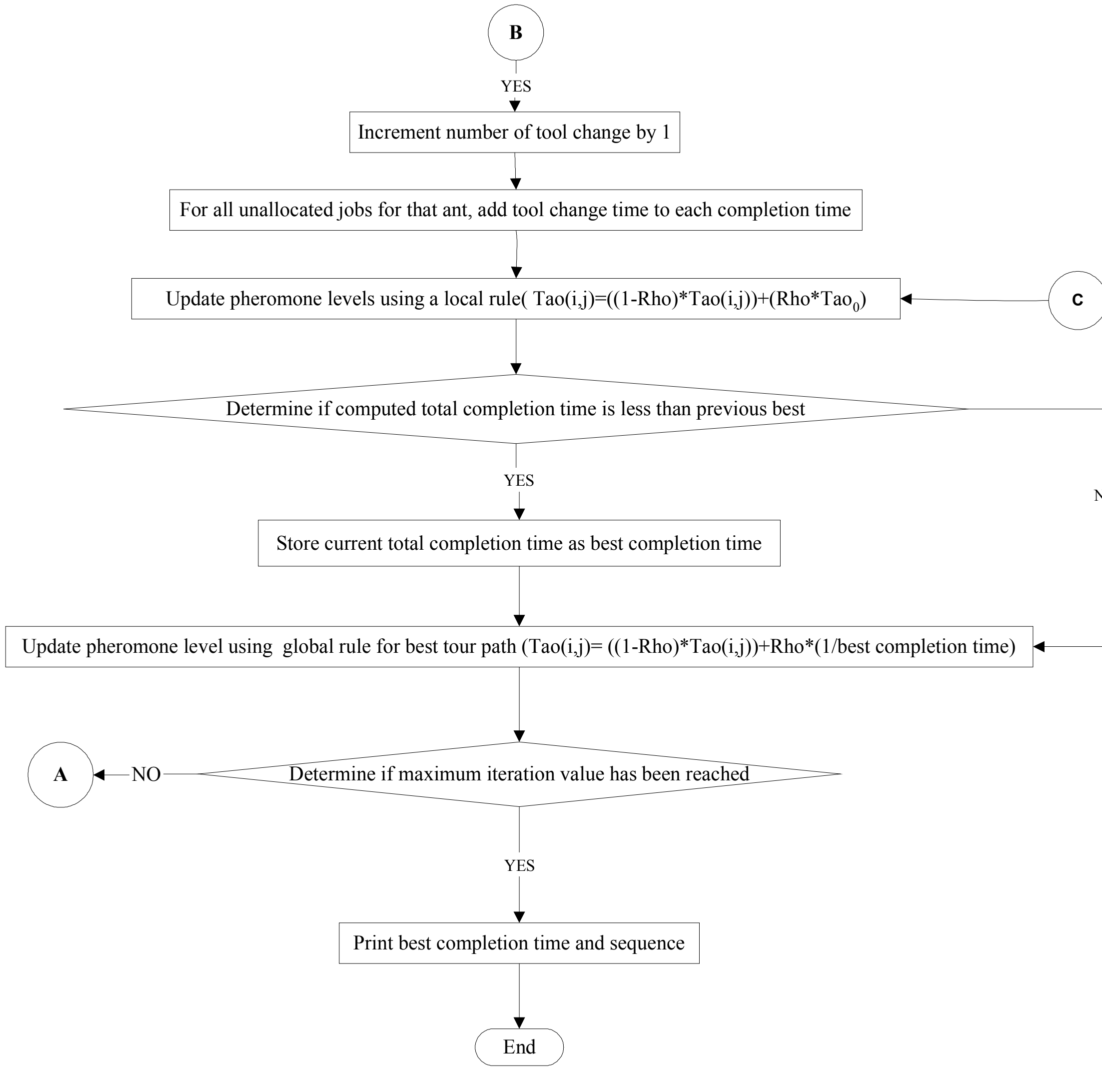
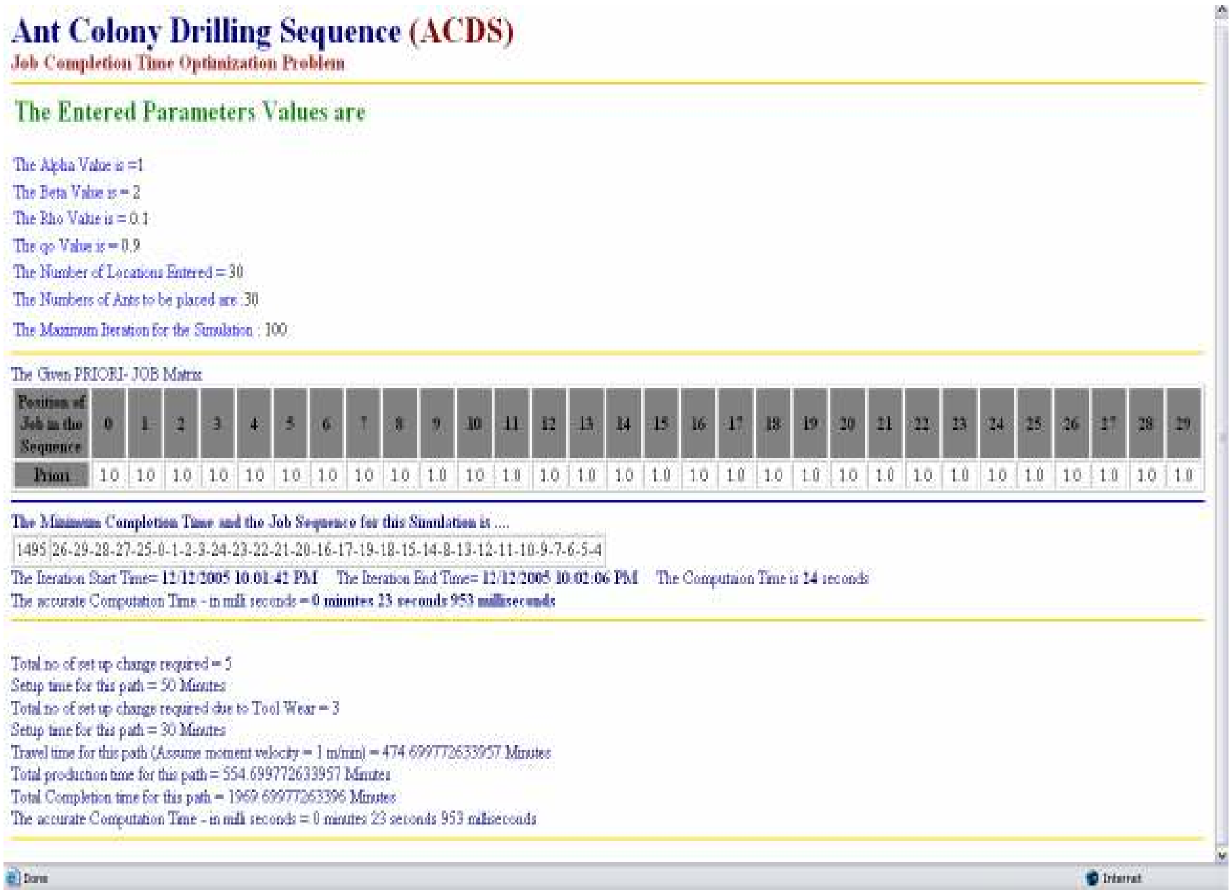


Figure 1. Ant Colony Drilling Sequence methodology

3.3 Implementation

ACDS methodology was programmed as a client-server model. All the parameters are hard coded in to the program where the user can enter the values and the algorithm is run on the web server and results are presented back to the client browser. Microsoft active server page with Visual basic script was used to program the ACDS methodology. Microsoft visual Interdev was used as the interface to write the programs. Microsoft Windows XP professional, with Pentium 4 processor 2.4 GHz, 512MB RAM was used as the server.

Figure 2 shows the screen shot of the output of web-based ACDS algorithm. The first section of the output shows the user entered input parameters including the process time for each hole; the middle section displays the best sequence obtained for the simulation that has the minimum total completion time along with the start and end time of the iteration. The final half summarizes the total number of setup changes required due to different hole sizes and changes due to tool wear along with the time taken for both the setups. This half also displays the travel time for the best sequence.



Ant Colony Drilling Sequence (ACDS)

Job Completion Time Optimization Problem

The Entered Parameters Values are

The Alpha Value is =1
The Beta Value is = 2
The Rho Value is = 0.1
The qo Value is = 0.9
The Number of Locations Entered = 30
The Numbers of Ants to be placed are :30
The Maximum Iteration for the Simulation : 100

The Given PRIORI- JOB Matrix

Position of Job in the Sequence	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Priori	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

The Minimum Completion Time and the Job Sequence for this Simulation is ....

1495	26-29-28-27-25-0-1-2-3-24-23-22-21-20-16-17-19-18-15-14-8-13-12-11-10-9-7-6-5-4

The Iteration Start Time= **12/12/2005 10:01:42 PM** The Iteration End Time= **12/12/2005 10:02:06 PM** The Computaion Time is **24** seconds
The accurate Computation Time - in milli seconds = **0 minutes 23 seconds 953 milliseconds**

Total no of set up change required = 5
Setup time for this path = 50 Minutes
Total no of set up change required due to Tool Wear = 3
Setup time for this path = 30 Minutes
Travel time for this path (Assume moment velocity = 1 m/min) = 474.699772633957 Minutes
Total production time for this path = 554.699772633957 Minutes
Total Completion time for this path = 1969.69977263396 Minutes
The accurate Computation Time - in milli seconds = 0 minutes 23 seconds 953 milliseconds

Done Internet

Figure 2. Screenshot of the result of Ant Colony Drilling Sequence.

CHAPTER IV

RESULTS AND DISCUSSION

4.1 Purpose

The purpose of this chapter is to validate the developed Ant Colony Drilling Sequence (ACDS) algorithm and present the results. ACDS algorithm was validate using the hole location of an Oliver 30 problem.

The developed ACDS model is validated by comparing the results with those of another validated model developed by (Krishnaiyer and Cheraghi, 2003) and by observing the variation in total completion time by varying tool life and setup time and keeping other parameters constant.

Table 1 shows the processing time and size of the holes used as the input data for the drilling sequence optimization problem. X and Y coordinates of the Oliver30 problem was used as the hole locations.

Table 1. Input data for drilling sequence problem (Oliver 30)

Hole No.	Processing Time	Size of the Hole	X Position	Y Position
0	1	1	18	54
1	1	1	22	60
2	1	1	25	62
3	1	1	7	64
4	1	1	2	99
5	1	1	41	94
6	1	1	37	84
7	1	2	54	67
8	1	2	54	62
9	1	2	58	69
10	1	2	71	71
11	1	2	74	78
12	1	2	87	76
13	1	2	83	69
14	1	3	64	60
15	1	3	68	58
16	1	3	71	44
17	1	3	83	46
18	1	3	91	38
19	1	3	82	7
20	1	3	62	32
21	1	4	58	35
22	1	4	45	21
23	1	4	41	26
24	1	4	44	35
25	1	4	25	38
26	1	4	24	42
27	1	4	18	40
28	1	4	13	40
29	1	4	4	50

4.2 Comparison of ACDS and WACSA

Krishnaiyer (2003) proposed a Web based ant colony system algorithm (WACSA) to solve drilling sequence optimization problem considering setup, occurring due to change in tool sizes. It is expected that the total production time calculated using ACDS algorithm would be more than WACSA because of the additional setup time caused due to tool wear.

The input data used in WACSA was the same Oliver30 problem for the hole locations (X and Y coordinates). The sizes of the holes were categorized into 14 different sizes, 1 through 14 along with processing time of 1 time unit and the setup time of 4.5 time units. The following equations (5) and (6) shows how the total production time is calculated by both the algorithms,

WACSA: Total Production Time (TPT) = Travel time + Setup Time (Change in tool size) - (5)

ACDS: Total Production Time (TPT) = Travel time + Setup Time (Tool wear and Change in tool size) - (6)

Figure 3 shows the results obtained from WACSA and ACDS for the same input parameters with ACDS incorporating tool life of 2 time units. It is clear from the graph that total production time for ACDS is more and the difference can be attributed to the time taken for the additional number of setup occurred due to tool wear.

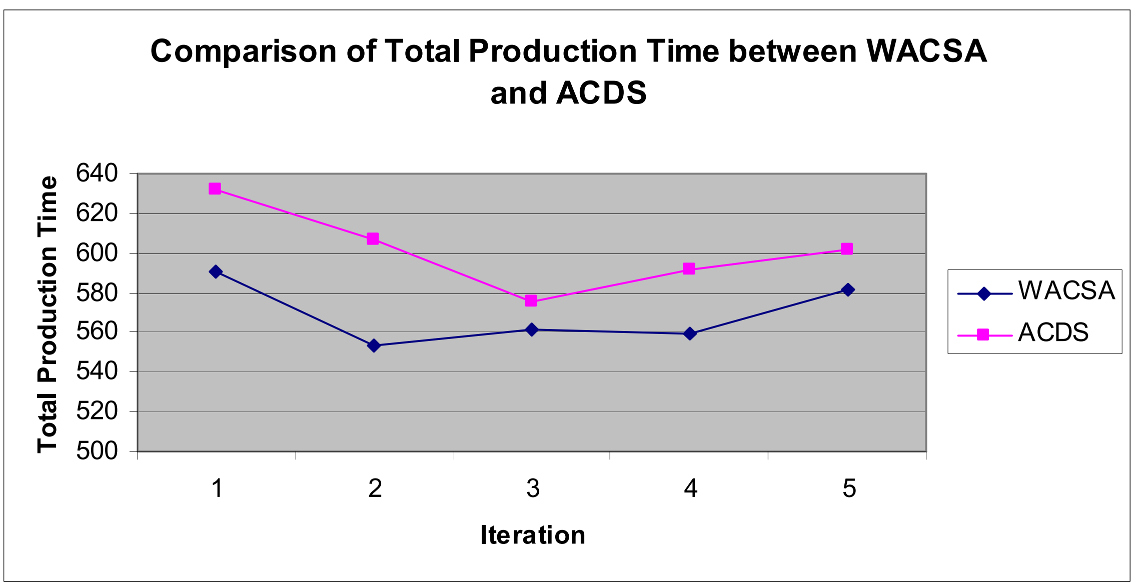


Figure 3. Comparison of total production time between WACSA and ACDS

4.3 Variation of Input Parameters

4.3.1 Effect of Change in Tool Life

To further validate the model, tool life values were changed and the effect was analyzed. The input data shown in Table 1 is used for this purpose. Two cases were considered. In the first case the tool life was set to 4 time units while in the second case some of the tool lives were set to 5 time units. It is expected that the total completion time for the later to be less because of the increase in tool life. Table 2 compares the average total completion time for the two cases

Table 2. Effect of change in tool life on total completion time

Tool Life	Constant Tool life (4 time units)	Different Tool Life (4 and 5 time units)
Average Total Completion Time	1929.3	1917.94

It is clear that the average total completion time of drilling sequence with tools that has different tool life (4 and 5 time unit) is less than that of the total completion time with tools that has constant tool life (4 time unit). Irrespective of the tool life the other factor that contributes to the total completion time is the travel time, therefore there can be occasions when the total completion time can be more for sequence with slightly more tool life.

4.3.2 Variation of Tool Life

The developed model is also validated by varying the input parameters such as tool life and observing the variation in the total completion time. It is expected that total completion time decreases as the life of the tool increases.

Table 3 shows the results of simulation for different tool life with tool change time maintained at a constant time of 10 time units. The tool life was varied from 1 to 5 time units.

Table 3. Drilling sequence optimization results with constant tool change

Tool Life	Total completion time
1	5302.4
2	3141.2
3	2366.7
4	1991.0
5	1892.6

It could be inferred that higher the tool life the total completion time decreases. The best Total Completion Time occurs when the tool has a life of 5 time units. The solution gets better as we progress with the tool that has better tool life.

4.3.3 Variation of Tool Change Time

The developed model is also validated by varying setup time (time taken to change the tool) due to tool wear and change in tool size and observing the variation in the total completion time. It is expected that total completion time increases as the setup time increases.

Table 4 shows the results of simulation for different setup times with the tool life maintained constant at 4 time units. The setup time was varied from 1 to 5 time units.

Table 4. Drilling sequence optimization results with constant tool wear

Setup Time	Total completion time
1	992.93
2	1173.69
3	1225.91
4	1327.40
5	1422.14

It can be inferred that when the setup time increases, so does the total completion time. So it can be concluded that the results obtained from developed algorithm is logical.

The above procedures show that the developed algorithm performs as expected.

4.4 Computational Analysis

In order to evaluate the computational capability of the proposed algorithm, problems of different sizes were solved. For this purpose the input data set obtained from Boeing (Table 7) was used. It is expected that the computation time would increase as the number of holes (size of the problem) increases. Table 5 shows the computation time for various numbers of holes. The number hole sizes were varied from 30 to 50 with an increment of 5 holes. Due to the system capability (RAM capacity) only maximum of 50 hole locations was considered, over which the system generates an error message.

Table 5. Computation time for various hole sizes

No. of Holes	Computation Time
30	24
35	37
40	55
45	77
50	104

It can be inferred from the table that the computation time increases for larger number of holes. Figure 4 shows the graphical representation of Table 9.

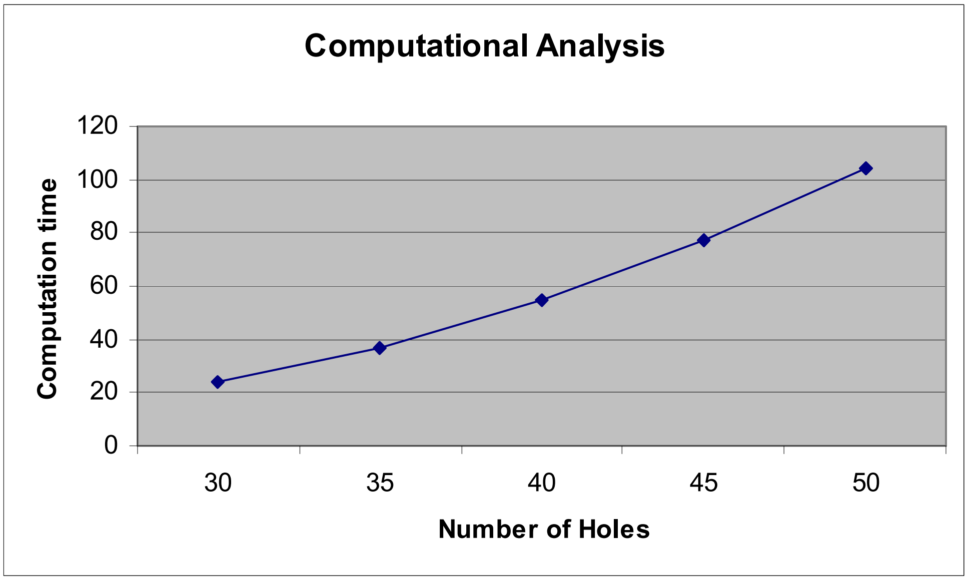


Figure 4. Evaluation of computation time for problems with different hole sizes

4.5 Application to Real World Problem

ACDS algorithm was applied to two different real world data sets. Hole location of an Oliver30 problem and actual hole locations data set on an aircraft part from Boeing was used and the results are presented below.

4.5.1 Drilling Sequence Optimization Problem (Oliver 30)

Considering the same Oliver 30 input data (Table 1) used for validating ACDS algorithm the developed program was run for 100 iterations and Table 6 lists the top 5 sequences obtained with the total completion for each of the sequence. The assumptions made are as follows: the sizes of the holes were categorized into 4 different sizes, 1 through 4 along with processing time of 1 time unit. The simulations were run with tool life of 4 minutes and tool change time of 10 minutes. Due the programming constraints the 30 holes were numbered from 0 to 29. The total completion time for a single sequence is computed manually using Excel and compared with the results obtained from the program.

The algorithm parameters were as follows: relative importance of trial (α) = 1, relative importance of visibility (β) = 2, trial persistence (Rho) = 0.1, relative importance of exploitation (qo) = 0.9. Table 6 displays the total completion time for all the five sequences.

Table 6. Top five solutions obtained from ACDS algorithm (Oliver 30)

	Sequence 1	Sequence 2	Sequence 3	Sequence 4	Sequence 5
	25	26	25	29	29
	26	25	26	28	28
	29	27	27	27	27
	28	28	29	26	26
	27	29	28	25	25
	0	3	0	0	0
	1	0	1	1	1
	2	1	2	2	2
	3	2	3	3	3
	24	24	24	24	24
	23	23	23	23	23
	22	22	22	22	22
	21	21	21	21	21
	20	20	20	20	20
	16	16	16	19	16
	19	17	19	18	17
	18	18	18	17	19
	17	15	17	16	18
	15	19	15	15	15
	14	14	14	14	14
	8	8	8	8	8
	7	13	7	7	13
	9	12	9	9	12
	10	11	10	10	11
	13	10	11	11	10
	12	9	13	13	9
	11	7	12	12	7
	5	6	5	5	6
	6	5	6	6	5
	4	4	4	4	4
Total Completion Time	1924.5	1925.4	1895.2	1923.3	1978

4.5.2 Drilling Sequence Optimization Problem (Boeing)

ACDS algorithm was applied to another real world problem, actual hole locations data set on an aircraft part from Boeing was used and the results are discussed below.

Appendix A shows the actual data set received from Boeing. Table 7 shows processing time and size of the hole used as the input data for the drilling sequence optimization problem. The original data set received contained 283 different hole locations with three dimensional coordinates (X, Y and Z). Due to the programming constraints and system capability only 50 hole locations were taken and the Z coordinate was eliminated. The assumptions made on processing time and hole sizes were similar to that of the Oliver 30 problem.

Table 8 lists the top 5 sequences obtained with the total completion time for each of the sequence, which indicates that ACDS can be used to arrive at alternative sequences. This is a comprehensive advantage over other heuristics, providing much needed flexibility to sequencing problems.

Figure 5 displays the tour path of one of the sequence obtained from ACDS algorithm and the optimal tour path that can be generated. Boeing data points are used for this purpose. Also the distance traveled is calculated for both the cases. The optimal tour path has traveled 51 units where as the ACDS generated tour path has traveled 61 units. The difference in tour length can be attributed towards change in hole sizes and tool wear. Also from the comparison graph it is clear that ACDS algorithm follows a logical pattern. It can be concluded that ACDS algorithm generates a near optimal tour path for a given set of holes.

Table 7. Input data for drilling sequence problem (Boeing)

Hole No.	Processing time	Size of the Hole	X Position	Y Position
0	1	1	196.1880	-100.3863
1	1	1	195.3389	-100.3851
2	1	1	195.3389	-101.3307
3	1	1	196.1880	-101.3328
4	1	1	196.1880	-102.2790
5	1	1	195.3389	-102.2761
6	1	2	195.3389	-103.2208
7	1	2	196.1880	-103.2246
8	1	2	196.1879	-104.1695
9	1	2	195.3389	-104.1648
10	1	2	195.3389	-105.1078
11	1	2	196.1879	-105.1133
12	1	2	195.3388	-108.2136
13	1	3	196.1878	-108.2186
14	1	3	196.1878	-109.0844
15	1	3	195.3388	-109.0805
16	1	3	195.3388	-109.9455
17	1	3	196.1877	-109.9482
18	1	3	196.1877	-110.8099
19	1	3	195.3387	-110.8082
20	1	4	195.3387	-111.6685
21	1	4	196.1876	-111.6691
22	1	4	196.1875	-112.5257
23	1	4	195.3387	-112.5261
24	1	4	195.3386	-113.3810
25	1	4	196.1875	-113.3796
26	1	4	196.1874	-114.2304
27	1	4	195.3386	-114.2327
28	1	4	194.0344	-113.7009
29	1	4	193.4439	-114.4780
30	1	1	192.8541	-115.2519
31	1	1	193.9085	-115.2522
32	1	1	195.3386	-116.0030
33	1	1	196.1874	-115.9822
34	1	1	196.1870	-117.8477
35	1	1	195.3384	-117.8459
36	1	2	195.3383	-118.7736
37	1	2	196.1869	-118.7797
38	1	2	196.1867	-119.7063
39	1	2	195.3383	-119.6958
40	1	2	195.3382	-120.6122
41	1	2	196.1866	-120.6272
42	1	2	196.1863	-122.4608
43	1	3	195.3380	-122.4407
44	1	3	195.3378	-124.4173
45	1	3	196.1858	-124.4413
46	1	3	196.1856	-125.4996
47	1	3	195.3377	-125.4723
48	1	3	195.3376	-126.5622
49	1	3	196.1853	-126.5906

Table 8. Top five solutions obtained from ACDS algorithm (Boeing)

	Sequence 1	Sequence 2	Sequence 3	Sequence 4	Sequence 5
	21	16	25	21	22
	22	17	24	22	23
	23	18	27	23	24
	24	19	26	24	27
	25	20	49	25	49
	26	23	48	26	48
	27	22	47	27	47
	28	25	46	28	46
	49	49	45	29	45
	48	48	44	31	44
	47	47	43	30	43
	46	46	42	32	42
	45	45	41	33	41
	44	44	40	34	40
	43	43	39	35	39
	42	42	38	36	38
	41	41	37	37	37
	40	40	36	49	36
	39	39	35	48	35
	38	38	34	47	34
	37	37	33	46	33
	36	36	32	45	32
	35	35	31	44	31
	34	34	30	43	30
	33	33	29	42	29
	32	32	28	41	28
	31	31	23	40	25
	30	30	22	39	26
	29	29	21	38	21
	20	28	20	20	20
	19	24	19	19	19
	18	27	18	18	18
	17	26	17	17	17
	16	21	16	16	16
	15	14	15	15	15
	14	15	14	14	14
	13	12	13	13	13
	12	13	12	12	12
	11	11	11	11	11
	10	10	10	10	10
	9	9	9	9	9
	8	8	8	8	8
	7	7	7	7	7
	6	6	6	6	6
	5	5	5	5	5
	4	4	4	4	4
	3	3	3	3	3
	2	2	2	2	2
	1	1	1	1	1
	0	0	0	0	0
Total Completion Time	4529.72	4490.71	4536.95	4520.34	4539.32

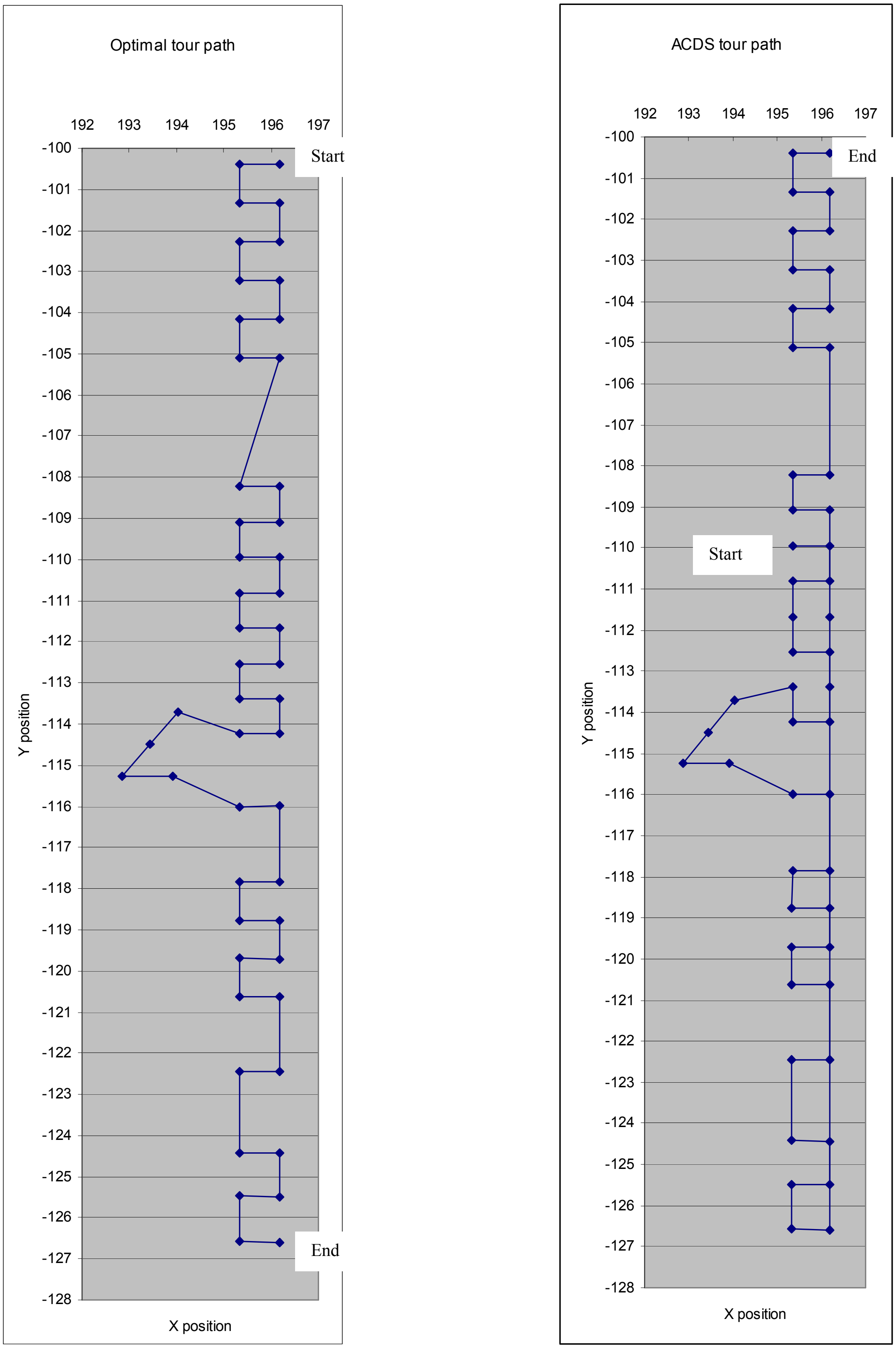


Figure 5. Comparison of ACDS generated and optimal tour path from Boeing data

CHAPTER V

CONCLUSION

5.1 Introduction

Necessity is the mother of all inventions. The advancement in the field of science and technology and development of convenient, easy to use software tools with everyday decreasing cost and increasing consistency had made ideas into reality of solving real world sequencing problems. Sequencing has become much easier with increase in speed of computers. By applying heuristic, improved and better solutions could be derived, even with lesser computational time than the conventional method. This thesis work manifests the efficiency of ant algorithm in solving one of the most complex real life problems--the drilling sequence problem. Determination of drilling sequence for a multi-product environment exhibits all the characteristics of an NP-Hard problem and is handled in the same way as a Traveling Salesman Problem (TSP).

5.2 Significance

The significance of this work lies mainly on the deficiency of earlier machine scheduling literatures, that is, ignoring the issue of tool change. Literatures up to date have modeled machine-scheduling problems considering that the tool change is either induced by part mix (krishnaiyer et al, 2003) or tool wear (Akturk et al, 2002). The proposed approach is innovative and unique in its own way as it overcomes the drawbacks so far in modeling a machine sequencing problem "as is" in real life (considers both the part mix and tool wear). Adding much value to our contribution, are the notable benefits that emerge out of proposed approach: minimized total completion (drilling) time which directly impacts labor hours required for the task, effective utilization of available resources, reduction in overall cycle time of the entire process, and thereby meeting the product deadline well in advance. In addition to previous works multiple tools types with variable processing time and different tool life problems can use this algorithm, but also may require problem specific heuristics. The novelty of this approach is by itself a drawback, as its performance relative to other algorithms cannot be evaluated.

5.3 Future Scope

This method could be applied to a real world problem and the input data can be obtained directly from the machine or work center. The effect of various parameters used in the algorithm such as Alpha, Beta could be studied to vary the ant routing ability to find the next hole. Further more the due date or effect of completion date and time of the process could be included with make the problem closer to the real life manufacturing situation but also increases the complexity. The algorithm can be applied to larger size problems by using higher end computers.

LIST OF REFERENCES

1. Akturk, M, S., Jay, B, G., and Evrim, D, G., 2003, Scheduling with Tool Changes to Minimize Total Completion Time: A Study of Heuristics and Their Performance. *Naval Research Logistics*, 50 (1), 15-30.

2. Anderson, C., and Bartholdi, J. J., 2000, Centralized Versus Decentralized Control in Manufacturing: Lessons from Social Insects. *Proceedings of Complexity and Complex Systems in Industry*, September, 2000, University of Warwick, 92-105.

3. Bland, J. A., 2001, Optimal Structural Design by Ant Colony Optimization. *Engineering Optimization*, 33, 425-443.

4. Bonabeau, E., Dorigo, M., and Theraulaz, G., 2000, Inspiration for Optimization from Social Insect Behavior. *Nature*, 406, 39-42.

5. Botee, M. H., and Bonabeau, E., 1998, Evolving Ant Colony Optimization, *Advance Complex System*, 1, 149-159.

6. Caroline, G., Wilson, L. P., and Marc, G., 2002, Comparing an ACO Algorithm with Other Heuristics for the Single Machine Scheduling Problem with Sequence Dependent Setup Times. *Journal of the Operational Research Society*, 53, 895-906.

7. Cicirello, V. A., and Smith, S., 2001, Ant Colony Control for Autonomous Decentralized Shop Floor Routing. *Proceedings of the 5th International Symposium on Autonomous Decentralized Systems, IEEE Computer Society*.

8. Colorni, A., Dorigo, M., and Maniezzo, V., 1992, An Investigation of Some Properties of an Ant Algorithm. *Proceedings of the parallel problem solving from Nature (PPSN 92)*, Brussels, Belgium, 509-520.

9. Colorni A., Dorigo, M., and Maniezzo, V., 1992, Distributed Optimization by Ant Colonies". *Proceedings of the 1st European Conference on Artificial Life*, F. Varela and P. Bourgine (eds.), Paris, France, 134-142.

10. Dell'Amico, A. M., and. Trubian, A. M., 1986, Applying Tabu Search to the Job-Shop Scheduling Problem. *Annals of Operations Research*, 41, 231-252.

11. Digalakis, J., and Margaritis, K., 2000, A Survey for Meta-Heuristic Algorithms. *Internal Report 04 University of Macedonia- Parallel Distributed Processing Laboratory*, Thessaloniki, Greece.

12. Dorigo,M., Di Caro, G., and Gambardella, M, L.,1999, Ant Algorithms for Discrete Optimization, *Artificial Life*, 5(2), 137-172.

13. Dorigo, M., and Dicaro, G., 1999, The Ant Colony Optimization Meta-Heuristic. *New Ideas in Optimization*, Corne D., Dorigo, M., and Glover, F., (eds.), McGraw-Hill, England, 11-32.

14. Dorigo, M., and Gambardella, M. L., 1997, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66.

15. Dorigo, M., and Gambardella, M. L., 1997, Ant Colonies for the Traveling Salesman Problem. *Bio Systems*, 43, 73-81.

16. Dorigo, M., Gambardella, M. L., 1996, A Study of Some Properties of Ant-Q. *Internal Report TR/IRIDIA/1996-4,* University Libre de Bruxelles, Belgium.

17. Dorigo M., Maniezzo, V., and Colorni, A., 1991, Positive Feedback as a Search Strategy. *Technical Report No. 91-016,* Politecnico di Milano, Italy.

18. Dorigo, M., Maniezzo, V., and Colorni, A., 1996, Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 26(1), 29-41.

19. Dorigo, M., and Stuetzle, T., 2000, The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances. *Technical Report IRIDIA/2000-32, IRIDIA,* University Libre de Bruxelles, Belgium.

20. Feldmann, M., Dirk, B., 2003, Single-machine scheduling for minimizing earliness and tardiness penalties by meta-heuristic approaches. *Computers & Industrial Engineering*, 44(2), 307–323.

21. Gambardella, M. L., and Dorigo, M., 1996, Solving Symmetric and Asymmetric TSPs by Ant Colonies. *Proceedings of the IEEE Conference on Evolutionary Computation (ICEC96)*, IEEE Press, 622-627.

22. Gambardella, L. M., and Dorigo, M., 1995, Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. *Proceedings of the 12th International Conference on Machine Learning*, A. Prieditis and S. Russell, Morgan Kaufmann (eds.), Tahoe City, CA, 252–260.

23. Gambardella, L. M., Taillard, E., and Dorigo, M., 1999, Ant Colonies for the Quadratic Assignment Problem. *Journal of the Operational Research Society*, 50, 167-176.

24. Gravel M., Price, W., and Gagn, C., 2001, Scheduling Continuous Casting of Aluminum using a Multiple-Objective Ant Colony Optimization Meta-Heuristic. University of Laval, Quebec, Canada. (To be published)

25. Hao, J. K., 2001, *Meta-Heuristics for Combinatorial Optimization.* September 29th 2001.

26. Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, Michigan.

27. Huang, J. S., 2001, Enhancement of Hydroelectric Generation Scheduling using Ant Colony System Based Optimization Approaches, *IEEE Transactions on Energy Conversion*, 16(3), 296-301.

28. Koenig, S., 2001, Biologically Inspired Multi-Agent Systems: Ant Robotics, *College of Computing, Georgia Institute of Technology*. http://www.cc.gatech.edu/fac/Sven.Koenig/ants.html

29. Krishnaiyer, K., Cheraghi, S. H., 2002, Ant Algorithms: Review and Future Applications. *Proceedings of the Annual Institute of Industrial Engineers Research Conference* (IERC 2002), May, 2002, Orlando, Florida, USA.

30. Krishnaiyer, K., Cheraghi, S. H., 2003, Ant Algorithms: A Web-Based Implementation and Applications to Manufacturing Systems Problems

31. Krukowski, J., *Ant Algorithms*. http://usbusiness-review.com/0111/06.html

32. Laguna, M., 2001, Global Optimization and Meta-heuristics. *Encyclopedia of Life Support Systems*. September 29th 2001. http://bus.colorado.edu/Faculty/Laguna/

33. Lee. D., Kazadi., S., and Goodman., R., 2000, Swarm Engineering for TSP. *Proceedings of the ANTS'2000 From Ant Colonies to Artificial Ants: 2nd International Workshop on Ant Algorithms*, Brussels, Belgium

34. Madureira, M. A., 1988, Meta-Heuristics for Single-Machine Scheduling Problems. *Working Group on Intelligent Manufacturing Systems Workshop 4*, 1988, Nancy, France.

35. Maniezzo, V., and Carbonaro, A., 1999, Ant Colony Optimization: An Overview. *Proceedings of MIC'99 III Metaheuristics International Conference*, 1999, Brazil.

36. *National Institute for Standards and Technology*. (n.d). December 2002. http://www.nist.gov/dads/HTML/nphard.html

37. Oliver, I. M. , Smith D. J., and Holland, J. R. C., 1991, A Study of Permutation Crossover Operations on the Traveling Salesman Problem. *Proceedings of the 2nd International Conference on Genetic Algorithms*, 224-230.

38. Patch, K., 2000, Ants Solve Tough Problems. *Technology Research News*. http://www.trnmag.com/Stories/092700/Ant_Optimization_092700.html

39. Rardin, L. R., and Uzsoy, R., 2001, Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics*, 7, 261–304.

40. Roli, A., and Milano, M., 2001, Metaheuristics: A Multi Agent Perspective. *DEIS Technical Report No: DEIS-LIA-01-006*, Bologna, LIA Series No. 50.

41. Sautter, U., 2000, Software that Imitates the Behavior of Ants could Make Highway and Telecom Traffic More Efficient. *Insect Power - Time Europe*. http://www.time.com/time/europe/specials/ff/trip4/ants.html

42. Stützle, T., and. Dorigo, M., 2001, An Experimental Study of the Simple Ant Colony Optimization Algorithm. *Proceedings of the 2001 WSES International Conference on Evolutionary Computation (EC'01)*, WSES-Press International.

43. Stützle, T., and Hoos, H., 1997, Improvements on the Ant System: Introducing the MAX-MIN Ant System. *Proceedings of the* 3rd *International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA97*), University of East Anglia, Norwich, UK, Springer Verlag, 245-249.

44. Stützle, T., and Hoos, H., 1997, The MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. *Proceedings of the IEEE 4th International Conference on Evolutionary Computation*, Indianapolis, Indiana, USA, 308-313.

45. Stützle, T., and Hoos, H., 1997, The MAX-MIN Ant System and Local Search for Combinatorial Optimization Problems: Towards Adaptive Tools for Global Optimization. *Proceedings of the 2nd Metaheuristics International Conference (MIC-97)*, Sophia-Antipolis, France, 191-193.

46. Taillard, D. E., Gambardella, M. L., and Gendreau, M., et al., 2001, Adaptive Memory Programming: A Unified View of Meta-Heuristics. *European Journal of Operational Research*, 135, 1-16.

47. Talbi, E. G., 2002, A Taxonomy of Hybrid Meta-Heuristics, *Journal of Heuristics*, 8, 541-564.

48. Ulysses16 and Ulysses 22. *TSP Data Points*. http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsp/

49. Yadav, S., Xu, Y., and Xue, D., 2000, A multi-level heuristics search algorithm for production scheduling. International Journal of Production Research, 38, 2761-2785.

50. Zwaan, S. V., and Marques, C., 1999, Ant Colony Optimization for Job Shop Scheduling. *Proceedings of the 1999 Workshop on Genetic Algorithms and Artificial Life GAAL'99*, Lisbon, Portugal.

51. Strenkowski, J. S., Hsieh, C.C., and Shih, A.J., 2004, An Analytical Finite Element Technique for Predicting Thrust Force and Torque in Drilling. International Journal of Machine Tools and Manufacture, 44, 1413-1421.

52. Allahverdi, A.1., Aldowaisan, T., 2002, New heuristics to minimize total completion time in m-machine flowshops. International Journal of Production Economics, 77, 71-83.

53. Chandru, V., Lee, C. Y., and Uzsoy, R., 1993, Minimizing total completion time on batch processing machines. International Journal of Production Research, 31, 2097-2121.

APPENDIX

Appendix A: Table 9. Input data points from Boeing

Hole No.	X COORDINATE	Y COORDINATE	Z COORDINATE
0	196.1880	-100.3863	50.0811
1	195.3389	-100.3851	50.1219
2	195.3389	-101.3307	50.1352
3	196.1880	-101.3328	50.0944
4	196.1880	-102.2790	50.1225
5	195.3389	-102.2761	50.1634
6	195.3389	-103.2208	50.2064
7	196.1880	-103.2246	50.1654
8	196.1879	-104.1695	50.2232
9	195.3389	-104.1648	50.2643
10	195.3389	-105.1078	50.3372
11	196.1879	-105.1133	50.2959
12	195.3388	-108.2136	50.6853
13	196.1878	-108.2186	50.6428
14	196.1878	-109.0844	50.7695
15	195.3388	-109.0805	50.8126
16	195.3388	-109.9455	50.9530
17	196.1877	-109.9482	50.9091
18	196.1877	-110.8099	51.0617
19	195.3387	-110.8082	51.1064
20	195.3387	-111.6685	51.2729
21	196.1876	-111.6691	51.2273
22	196.1875	-112.5257	51.4059
23	195.3387	-112.5261	51.4525
24	195.3386	-113.3810	51.6452
25	196.1875	-113.3796	51.5974
26	196.1874	-114.2304	51.8020
27	195.3386	-114.2327	51.8509
28	194.0344	-113.7009	51.7959
29	193.4439	-114.4780	52.0245
30	192.8541	-115.2519	52.2657
31	193.9085	-115.2522	52.2010
32	195.3386	-116.0030	52.3232
33	196.1874	-115.9822	52.2669
34	196.1870	-117.8477	52.8284
35	195.3384	-117.8459	52.8809
36	195.3383	-118.7736	53.1878
37	196.1869	-118.7797	53.1354

38	196.1867	-119.7063	53.4584
39	195.3383	-119.6958	53.5107
40	195.3382	-120.6122	53.8495
41	196.1866	-120.6272	53.7975
42	196.1863	-122.4608	54.5277
43	195.3380	-122.4407	54.5805
44	195.3378	-124.4173	55.4563
45	196.1858	-124.4413	55.4019
46	196.1856	-125.4996	55.9069
47	195.3377	-125.4723	55.9617
48	195.3376	-126.5622	56.5126
49	196.1853	-126.5906	56.4563
50	196.1850	-127.6506	57.0189
51	195.3374	-127.6196	57.0760
52	195.3373	-128.6648	57.6618
53	196.1847	-128.6985	57.6039
54	195.1530	-135.1454	61.9250
55	194.0686	-135.0795	62.0072
56	193.5020	-135.8195	62.6654
57	194.4971	-135.8806	62.5865
58	195.4923	-135.9408	62.5086
59	195.0385	-136.6890	63.1768
60	193.8787	-136.6171	63.2707
61	193.8783	-137.2740	63.8277
62	195.0381	-137.3474	63.7318
63	195.0377	-137.9509	64.2556
64	193.8779	-137.8723	64.3499
65	193.8776	-138.4634	64.8802
66	195.0373	-138.5473	64.7877
67	195.0369	-139.1816	65.3699
68	193.8772	-139.0964	65.4644
69	193.8768	-139.6619	66.0012
70	195.0366	-139.7510	65.9074
71	195.0362	-140.3131	66.4524
72	193.8764	-140.2202	66.5454
73	193.8761	-140.7713	67.0970
74	195.0359	-140.8681	67.0048
75	193.4650	-117.5500	52.9063
76	193.4650	-118.4000	53.1845
77	193.7417	-119.2895	53.4730
78	192.9932	-119.2866	53.5233
79	193.7418	-120.3937	53.8775
80	192.9933	-120.3831	53.9266
81	193.7419	-122.3931	54.6784
82	192.9937	-122.3748	54.7272
83	193.7421	-124.3609	55.5563
84	192.9941	-124.3391	55.6066
85	193.7422	-125.4208	56.0676

86	192.9943	-125.3957	56.1180
87	193.7424	-126.5059	56.6200
88	193.7425	-127.5597	57.1856
89	192.9948	-127.5312	57.2380
90	193.7426	-128.6013	57.7737
91	192.9951	-128.5707	57.8270
92	192.5600	-117.5500	52.9658
93	192.5600	-118.4000	53.2454
94	191.6300	-117.5500	53.0281
95	191.6300	-118.4000	53.3093
96	190.8194	-117.5500	53.0835
97	190.8193	-118.4000	53.3659
98	189.9688	-117.5500	53.1426
99	189.9687	-118.4000	53.4265
100	189.1183	-117.5500	53.2028
101	189.1182	-118.4000	53.4881
102	188.2679	-117.5500	53.2641
103	188.2677	-118.4000	53.5508
104	187.4176	-117.5500	53.3265
105	187.4173	-118.4000	53.6146
106	186.5674	-117.5500	53.3901
107	186.5665	-118.4000	53.6797
108	183.4814	-117.0000	53.4484
109	183.4814	-117.9100	53.7536
110	182.6020	-117.0000	53.5190
111	182.6020	-117.9100	53.8257
112	181.7226	-117.0000	53.5908
113	181.7226	-117.9100	53.8992
114	180.8433	-117.0000	53.6641
115	180.8433	-117.9100	53.9740
116	179.9641	-117.0000	53.7387
117	179.9641	-117.9100	54.0503
118	160.0547	-120.4047	57.2980
119	160.0624	-122.1381	58.1548
120	160.0563	-126.8683	60.9848
121	160.0653	-128.4263	62.0883
122	160.8929	-129.7803	62.9526
123	160.8899	-130.5122	63.5185
124	160.8866	-131.2348	64.0972
125	161.0696	-131.5756	64.3436
126	161.0769	-132.2515	64.9260
127	161.0953	-133.3130	65.9016
128	162.0513	-133.4027	65.7859
129	163.0076	-133.4913	65.6715
130	161.8233	-134.7099	67.1157
131	161.1180	-134.2752	66.8340
132	160.9342	-134.5671	67.1693
133	160.0491	-133.2122	66.0284

134	160.0459	-131.8575	64.7855
135	160.0387	-131.1578	64.1970
136	160.0387	-130.4361	63.6174
137	160.0387	-129.7049	63.0505
138	158.9383	-129.6064	63.1784
139	158.9383	-130.3364	63.7470
140	158.9383	-131.0569	64.3285
141	158.9380	-131.7542	64.9195
142	158.7421	-132.4335	65.5752
143	158.7422	-133.1015	66.2055
144	158.9369	-133.7365	66.7827
145	158.1027	-129.5307	63.2768
146	158.1003	-130.2594	63.8469
147	158.0977	-130.9785	64.4299
148	158.0288	-131.6683	65.0309
149	157.6507	-132.3296	65.7123
150	157.6403	-132.9937	66.3446
151	157.8510	-133.6289	66.9211
152	158.0297	-134.2783	67.5394
153	156.5842	-132.2260	65.8474
154	156.5850	-132.8891	66.4796
155	156.7654	-133.5199	67.0613
156	155.6358	-132.1332	65.9695
157	155.6320	-133.4196	67.2247
158	154.7270	-132.0425	66.0872
159	154.7238	-133.3260	67.3450
160	153.8179	-131.9507	66.2064
161	153.8337	-132.6152	66.8449
162	153.8155	-133.2314	67.4667
163	152.9092	-131.8570	66.3259
164	152.9254	-132.5212	66.9668
165	152.9180	-133.1369	67.5883
166	151.9712	-131.7576	66.4493
167	151.9649	-132.4052	67.0817
168	151.9637	-133.0353	67.7189
169	158.9383	-116.8494	55.9536
170	158.9383	-117.7594	56.3050
171	158.9383	-118.6694	56.6789
172	158.9383	-119.5410	57.0586
173	157.2873	-116.9820	56.2280
174	157.2873	-117.8920	56.5865
175	156.4098	-116.9820	56.3506
176	156.4098	-117.8920	56.7111
177	155.5326	-117.8920	56.8379
178	154.6557	-116.9820	56.6025
179	154.6557	-117.8920	56.9670
180	153.7792	-116.9820	56.7319
181	153.7792	-117.8920	57.0984

182	152.0057	-117.0399	57.0238
183	152.0057	-117.9244	57.3852
184	151.0650	-116.9387	57.1305
185	151.0650	-117.8487	57.5022
186	145.0348	-116.6754	58.0334
187	145.0348	-117.4760	58.3655
188	144.0314	-116.9258	58.3155
189	144.0325	-117.9220	58.7417
190	143.1116	-116.9387	58.4893
191	143.1116	-117.8490	58.8804
192	137.0485	-116.6793	59.5698
193	137.0485	-117.4739	59.9162
194	136.1400	-116.8915	59.8511
195	136.1399	-117.8950	60.3016
196	135.1068	-116.9387	60.0929
197	135.1068	-117.8487	60.5040
198	133.6800	-117.1350	60.4928
199	133.6800	-118.0450	60.9138
200	132.7700	-117.1350	60.6973
201	132.7700	-118.0450	61.1208
202	131.8600	-117.1350	60.9052
203	131.8600	-118.0450	61.3312
204	130.9500	-117.1350	61.1168
205	130.0400	-117.1350	61.3319
206	130.0400	-118.0450	61.7630
207	125.5280	-116.7911	62.2934
208	125.5443	-117.7195	62.7305
209	124.1242	-114.1193	61.5519
210	124.1360	-115.2199	61.9750
211	124.1679	-116.2863	62.4196
212	124.1777	-116.9737	62.7307
213	123.3811	-114.1209	61.7453
214	123.4082	-115.2210	62.1660
215	123.4428	-116.2875	62.6118
216	123.4528	-116.9744	62.9240
217	122.5083	-115.2208	62.4048
218	121.4542	-115.2197	62.6890
219	120.3732	-114.1193	62.5511
220	120.3681	-115.2180	62.9871
221	120.3419	-116.3897	63.5059
222	112.1134	-115.2065	65.4385
223	85.7850	-104.3304	72.1384
224	86.7463	-104.3301	71.7500
225	87.2617	-105.0965	71.6413
226	86.3009	-105.0967	72.0308
227	85.3402	-105.0968	72.4206
228	84.3796	-105.0970	72.8103
229	83.4189	-105.0972	73.2001

230	82.4588	-105.0973	73.5914
231	81.5006	-105.0975	73.9872
232	80.5433	-105.0977	74.3852
233	79.5425	-105.1492	74.8111
234	78.6286	-105.1264	75.1899
235	79.3196	-106.1352	75.0830
236	80.5444	-106.1355	74.5678
237	81.5015	-106.1357	74.1670
238	82.4595	-106.1359	73.7686
239	83.4194	-106.1361	73.3746
240	84.3799	-106.1363	72.9823
241	85.3405	-106.1365	72.5900
242	86.3011	-106.1367	72.1978
243	87.2618	-106.1369	71.8058
244	86.8316	-106.8970	72.1251
245	85.8717	-106.8965	72.5190
246	84.9119	-106.8959	72.9132
247	86.3570	-115.5394	75.6651
248	87.1809	-116.4234	75.8377
249	87.1724	-117.7224	76.7535
250	85.2749	-115.6390	76.2350
251	86.0710	-116.4252	76.3692
252	86.3181	-117.7203	77.1714
253	85.4642	-117.7180	77.5901
254	84.9622	-116.4271	76.9031
255	84.1947	-115.7387	76.8096
256	83.1169	-115.8378	77.3886
257	83.8546	-116.4290	77.4393
258	84.6110	-117.7162	78.0101
259	83.7584	-117.7147	78.4314
260	82.7481	-116.4315	77.9776
261	82.0427	-115.9351	77.9745
262	82.9066	-117.7136	78.8545
263	81.6445	-116.4347	78.5220
264	82.1009	-117.7131	79.2597
265	82.0942	-118.4232	79.8496
266	80.9725	-116.0317	78.5678
267	80.5440	-116.4389	79.0726
268	81.3003	-117.7165	79.6707
269	81.3066	-118.4512	80.2860
270	80.3919	-117.7115	80.1352
271	79.3655	-116.3293	79.5839
272	78.8024	-116.6073	80.0771
273	79.2748	-117.7091	80.7167
274	80.3917	-118.4439	80.7652
275	79.2952	-118.4348	81.3479
276	78.3846	-118.1765	81.6035
277	78.5835	-119.2527	82.5630

278	79.2908	-119.4255	82.3408
279	79.2924	-120.0330	83.0102
280	78.5844	-119.8605	83.2322
281	78.5447	-120.4360	83.9347
282	79.2466	-120.6066	83.7149