“An Improved ACO based on Estimation of Distribution”

Thesis submitted towards the partial fulfillment of the requirements for the award of the degree of

Master of Technology (VLSI Design & CAD)

Submitted by

Jitendra Kumar

Reg. No: 60761006

Under the supervision

Ankush Kansal

Assistant Professor, ECED

Department of Electronics & Communication Engineering

THAPAR UNIVERSITY, PATIALA

PUNJAB- 147001, INDIA

2010

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled: **"An Improved ACO based on Estimation of Distribution"** in partial fulfilment of the requirements for the award of the Degree of Master of Technology in VLSI AND CAD Engineering (Department of Electronics and Communication Engineering) is an authentic record of my own carried out under the supervision of Mr. Ankush Kansal, Assistant Professor, ECED.

The matter embodied in this thesis has not been submitted in any other University/Institute for the award of any degree.

Place: Patiala
Date: 21/07/10

Jitendra Kumar

Reg. No: 60761006

This is certified that the above statement made by the candidate is correct and true to best of my Knowledge.

21/7/10
Ankush Kansal
Assistant Professor, ECED
Thapar University Patiala
Punjab, INDIA – 147004

Dr. A. K. Chatterjee
Professor and Head
ECED
Thapar University, Patiala
Punjab, INDIA – 147004

22.7.10
Dr. R. K. Sharma
Dean of Academic Affairs
Thapar University, Patiala
Punjab, INDIA – 147004

ACKNOWLEDGMENTS

I would like to acknowledge with pleasure and gratitude the interest, guidance and mentorship of Ankush Kansal, my guide and advisor, towards the conception, execution and realization of this design Thesis. His invaluable suggestions and his direction and enthusiasm were a constant source of inspiration for me.

I would like to thank *Mrs. Alpana Agarwal* P. G. Coordinator, for facilitating me with the technology resources. Next, I wish to express my gratitude to *Professor (Dr.) A. K.Chatterjee*, Head, Department of Electronics and Communication Engineering for their excellent guidance and encouragement right from the beginning of this course. I am especially indebted to them for critical reviewing of my compilation to bring it in the submitted presentable form. I am also grateful for their trust in the choices I made to complete this thesis work.

This Thesis would not have been successful without *Mr Rohit* (IIT Delhi), who always motivated to ameliorate the project. I would also like to acknowledge the assistance and untiring spirit of my batch mate, in making this unique learning experience an enjoyable one. The final credits for this endeavor must be attributed to my family whose love, support and values have been my guiding light throughout these years.

My acknowledgements would not be complete without expressing my gratitude towards Almighty God. I feel very fortunate to come to know Him during all these years of my life and have continually been blessed by His endless love ever since. He is the true shepherd of my life.

Jitendra Kumar

ABSTRACT

In this Thesis a routing algorithm named Ant Routing System (ARS) is presented. ARS is based on a general-purpose met heuristic named "*An Improved ACO based on Estimation of Distribution*" which is a framework for building ant-inspired algorithms. ARS is applied as the routing algorithm in a point-to-point network Estimation of Distribution. ED-ACO yields significant improvements in performance over ACS.

Ant Colony System algorithm is one of the best algorithms of Ant Colony optimization however, the weaknesses of premature convergence, slow speed and low efficiency greatly restrict its application. An improved method to deal with the two main problems involved in ACO, slow speed of convergence and poor ability to search better solution at the end of the search procedure.

In order to improve the performance of the algorithms, a new Ant Colony Optimization algorithm based on Estimation of distribution (ED-ACO) is presented. ED-ACO is significantly improving the performance of slow speed & premature convergence. The MATLAB results show that ED-ACO is an effective and efficient way to solve combinatorial optimization problems.

ARS based on ED-ACO behaves differently depending on the relative priority of positive feedback, negative feedback and local heuristics, and that it is possible to adjust the parameters to achieve distribution of traffic over several paths when the network is heavily loaded, resulting in a higher throughput and lower loss.

In this Thesis, I have implement ED-ACO on MATLAB and show the improvement in the performance such that convergence rate become faster compared to other ACO based algorithm and also increases the error performance.

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

LIST OF SYMBOLS

α	Pheromone decay parameter
β	Relative importance of pheromone
τ	Pheromone on edge
η	Desirability of edge
δ	Distance between r and s with edge
L_{gb}	Globally best tour
$p(x_i)$	Probability of generating at bit position *i*.
$J_k(r)$	List of cities still to be visited
V	Set of edges
A	Set of pairs of edges

NOMENCLATURE

AS	Ant System
ARS	Ant Routing System
ACO	Ant Colony Optimization
TSP	Traveling Salesman Problems
ED-ACO	Ant Colony Optimization Based on Estimation of Distribution
ACS	Ant Colony System
MMAS	Max Min Ant System
ACOA	Ant Colony Optimization Algorithm
PTR	Pheromone Trail Replacement
RCACO	Rule based systems using continuous Ant-Colony Optimization
GSP	Good solution pool
TQACO	Quantum mechanism for Traveling salesman problem
QEA	Quantum-inspired evolutionary algorithm
MSACO	Multi-Direction Searching Ant Colony Optimization
MACO	Multiple Ant-Colony Optimization
NC-ACO	Noise Clustering Algorithm Using ant Colony Optimization
RCRSS	RoboCupRescue simulation system
TCO	The Time-Cost Optimization
SACO	Self-Adaptive Ant Colony Optimization
PSO	Particle Swarm Optimization
EACO	Enhanced Ant Colony Optimization
DACO	Dynamic Ant Colony Optimization

CPMM	Candidate Path Management Module
RPUS	Relativity Pheromone Updating Strategy
ERHM	Effective Repairing Heuristic Module
UC	Unit Commitment
AM-ACO	Ant Colony Optimization Algorithm To Aggregated Multicast
BCACO	Bi-Directional Convergence Ant Colony Optimization
SA	Simulated Annealing
SBACO	Simulated Bi-Directional Convergence Ant Colony Optimization
WNCACO	Well-Distributed on the initiation, the nearest neighbor Node choosing rules and with crossover operator

CHAPTER 1

INTRODUCTION

The goal of this Thesis is to investigate properties of a certain type of swarm intelligence commonly referred to as *ant algorithms*. To study the use of an ant algorithm operating upon a dynamic problem domain, that is, a domain that changes as a function over time.

With the growing importance of telecommunications and the Internet, more complex networked systems are being designed and developed. As networked systems become more complex so are the underlying software and the control rules. The challenges of dealing with the vast complexity of networking tasks such as load balancing, routing and congestion control accentuate the need for more sophisticated (and perhaps more intelligent) tools to solve these problem Specifically, To discuss and use an ant-inspired graph-based general-purpose algorithm met heuristic named *Ant Colony Optimization based on estimated distribution* as the basis of an implementation of a routing algorithm have the capability of finding short paths in graphs, and show an inherent adaptability that could be utilized to solve dynamic problems such as routing in a network [1].

Swarm intelligence is an area of research that over the last decade has experienced a boom in interest. Inspired by the seemingly intelligent behavior of swarms of primitive animals, swarm intelligence has proven to be a promising field of research in many different areas. A swarm of ants in the search for food shows the remarkable capability of finding shortest paths between a found food source and the anthill. Even though any single ant could be said to possess the capability of finding a short path from the anthill to a nearby food source, the probability of this occurring is very small, since an ant is not a very smart animal. The amazing thing is that when many ants cooperate on finding food, using pheromone trails as a simple indirect form of communication, the swarm of ants seems to be able to find a shortest path effectively. Another feature is their ability to adapt to a changing environment. If an obstacle is placed on the path from the food source back to the anthill, ants are capable of finding the shortest path around the obstacle and possibly find food sources closer to the anthill [2].

The emergent intelligence that a swarm of ants seems to possess is enticing from the perspective of computer science, because of the possibility to simulate and potentially exploit this behavior to solve algorithmic problems.

By observing and modeling the processes that occur in natural ants when working in their natural domain, it is possible to use this as inspiration to create a 'colony' of artificial ants, working in an environment represented by a graph and potentially experience a similar emergent intelligence. In fact, researchers in the field of biology have developed good theories on how ants communicate as a group and attempts to adapt these theories to a simulated domain in a computer have verified that it is possible to obtain emergent intelligent behavior from colonies of artificial ants [3],[4].

The Ant Colony System (ACS) is one of the best algorithms of ant colony optimization (ACO). It is developed by M.Dorigo et al. in the 1990s, inspired by the observation of real colonies of ants. Compared to the existing heuristics, ACS possesses the characteristics of positive feedback and distributed computing. Recently, ACS has been proposed to solve different types of combinatorial optimization problems especially to solve the NP-hard combinatorial optimization problems, large-scale complicated combinatorial optimization models, distributed control and clustering analysis problems. However, premature convergence and low efficiency is its deficiency and sometimes it is inefficient.In order to avoid these weaknesses [4].

Thomas Stuztle presented MAX-MIN Ant System (MMAS), M. Manfrin proposed Parallel ant colony optimization, and other improved ant colony optimization algorithms are presented. It is clear that great achievements have been made in improving the algorithm. But the premature and inefficient problems are still ready to be solved. Therefore, this Thesis tries to present ED-ACO to avoid the premature convergence and inefficient problems. The Estimation of Distribution Algorithms (EDAs) is a class of evolutionary algorithms that use probabilistic models to estimate a distribution over the search space, and then generate new solutions by sampling from these models. The probabilistic models are built by the global statistical information from the visited promising solutions. A distribution estimate can capture a building-block structure of a problem very accurately and ensure a very effective mixing and re-production of building blocks. This results in a linear or sub-quadratic performance of EDAs on combinatorial optimization problems. The ED-ACO is a new Ant Colony Optimization algorithm based on Estimation of Distribution. It combines the ideas of ACS and EDAs. The main idea is that it uses the probabilistic model based on estimating the distribution of promising edges to adjust the state transition rule and the global updating rule. This paper is organized as follows. To make the paper self-contained we first introduce the TSP, ACS and EDAs & introduce ED-ACO and present some computational results compared with ACS and MMAS on traveling salesman problems (TSPs). Improve the ED-ACO significantly by adding local search and then give some computational results [5].

A major problem in routing today is congestion. Routers risk sending packages through bottlenecks, thereby contributing to congestion, and increasing the probability of package death. If only routers had significantly different routing tables, a natural distribution could be expected, but unfortunately routers normally share information regarding which paths are the shortest. A router also tends to route all packages to a given destination through the same wire.

If routers could somehow individualize their choices, and also utilize a stochastic distribution of packages via different wires, then we believe that some of the bottleneck phenomena could be avoided. This will of course open up for a completely new discussion regarding unwanted side-effects and increased traffic on the network as a whole, but that is an entirely new problem, which we do not have the resources to discuss here. Suffice to say that in this project report we will concentrate on developing some basic route-finding and distributive capabilities for some limited settings. In the following we will give a description of what we believe is a productive way of routing packages in a network with limited capacity. Limited capacity in a network means that sending a stream of packages between routers (that are not neighbors) requires more than one distinct path in order to ensure package arrival because of the limited capacity of a single path. This necessitates distribution of package transfers over several paths. To believe that the ideal way to organize this distribution would be to sort the paths after lengths, shortest first, and then start utilizing the ordering by selecting the shortest paths first and then use them to their full capacity before utilizing the next in order [5].

In a dynamic domain such as a network, the ordering could change when traffic is directed over a specific path. Consequentially it is necessary to continuously reevaluate and adapt to the changes in latency over the different paths. The strategy we have tried to implement in Ant Routing System is thus to utilize the shortest paths to their full capacity, compensating for the fact that using them can cause them to cease to be the shortest because of the increased latency created by increasing queue lengths [6].

This can be done by converging to a number of good solutions, while avoiding stagnation by the use of exploration. Network routing is one of the possible uses for the abilities of artificial ant colonies, which potentially could utilize their adaptive behavior.

The quality of a routing algorithm could be measured by how fast the algorithm is able to transfer packages through the network while minimizing package loss. At an abstract level, a network can be viewed as a graph, with routers represented by vertices and wires represented by edges. If an ant-inspired algorithm is capable of finding a short path in a graph, can it also be that their adaptable nature allows them to find good solutions in a graph when costs on edges are time-varying stochastic variables1?

Propose that one of the problems with existing routing algorithms in use is their deterministic and static nature, given the highly dynamic nature of a network. In principle, a router calculates a single shortest path to other routers in the network on the basis of latency values that are updated with intervals.

It is important to note that edge cost varies determinedly as a function of package load and a number of predetermined properties (queue length etc.) in the system. The stochastic factor is introduced, as it is not presumed that sending frequency and sources is known a priori. sent deterministically over the shortest paths to their destination. In the intervals between updates, the routing algorithms thus route packages based on static information [7].

This approach present the problem that a network topology might have two distinct paths with almost equal length between a pair of fast routers, with a number of slower routers on the intermediate paths. Distribution over both paths could ensure the maximum throughput with a minimum of increased latency.

Another problem is that routing a package has a feedback mechanism: Routing a package to a router increments the latency to the receiver because the receiver must process the sent package, thereby making the latency values on which the shortest path was calculated invalid. In the case where two paths exist with almost equal length the latency increase from using the lowest latency path would eventually make the other path the fastest. One could argue that the information used by existing routing algorithms is recalculated at regular intervals, ensuring a relatively updated shortest path, but this could be viewed as fixing a problem when it arises instead of trying to avoid it.

ED-ACO Algorithm that continuously updates the information used for routing would constitute a solution better suited to this type of problem. The information would be based on the gathered experience with latency variations experienced by packages routed through the network.

The optimal solution would be a routing algorithm that distributes packages over the necessary number of different paths with minimal length seen over a period of time. To speculate that this is exactly what ant inspired algorithms can do. By continuously searching the domain in which they work, they show the ability of converging towards 'good' shortest paths, without losing the ability to adapt to changes in the domain. And ACO uses Negative feedback is often used to avoid premature convergence on a suboptimal solution (stagnation). If premature convergence happens, the ants are "satisfied" with the result found so far and have stopped looking for new and maybe better solutions.

To show that a successfully designed ant-algorithm will be able to mainly use the shortest path when the network is capable of handling the packages, but utilize distribution over several short routes when the network is congested [7].

Let us continue by specifying the problems to examine in this Thesis. Can an ant-inspired routing algorithm be used to obtain a optimize path in a point to point network, which result in improved throughput and reduced loss and minimize error in a point-to-point network?

The behavior of ant-inspired algorithms can be controlled by prioritizing different heuristics used to construct solutions. To investigate their effects in the context of routing we will answer the following question. How does an ant-inspired routing algorithm react to the prioritizing of different heuristics in the context of routing in a point-to-point network? To answer these questions we will run simulations examining the different effects.

The purpose of this Thesis is for us to implement, optimize and analyze ant algorithms. Our study-related goal is also to demonstrate understanding of the ACO met heuristic, a design template for ant-inspired algorithms, and to make an investigation of the properties of our ant algorithms, both from an analytical and from an empirical point of view.

1.1: THEORY OF ACO

Since the problems that are discussed in this project are graph problems it will initially try to establish some nomenclature to facilitate the further discussions. Now then continue to discuss ant algorithms and the ACO met heuristic that is a template for building ant inspired algorithms & then discuss EDACO & its state transition rule [1].

1.2: ANT COLONY SYSTEM FOR TSPS

A Traveling Salesman Problem can be represented by a complete weighted directed graph G = (V, A, δ) where V = {1, 2, . . . , n} is a set of cities, A = { (r, s) : r, s $\in$ V} is the edge set, and δ (r, s) = δ (s, r) is the distance between r and s with edge (r, s) $\in$ A. The aim is to find a minimal cost closed tour that visits each city exactly once. Let k be an ant whose task is to make a tour: visit all the cities and return to the starting one. Associated to k there is the list $J_k(r)$ of cities still to be visited, where r is the current city. An ant k situated in city r moves to city s using the following rule, called state transition rule:

$$S = \begin{cases} \arg\max_{u \in J_k(r)} \{\tau(r, u).\eta(r, u)^{\beta} \\ \text{if } q \le q_0 \text{ (exploitation)} \\ S \text{ otherwise (biased exploration)} \end{cases}$$

where τ (r, u) stands for the pheromone on edge (r, u), η (r, u) = 1/ δ (r, u) is the desirability of edge (r, u), β is a parameter which determines the relative importance of pheromone versus distance, *q* is a value chosen randomly with uniform probability in [0, 1], and q0 (0<q0<1) is a parameter. [1].

S is a random variable selected according to the random proportional rule given below

$$p_k(r, s) = \frac{[\tau(r, u)]^{\alpha} [\eta(r, u)]^{\beta}}{\sum [\tau(r, u)]^{\alpha}, [n(r, u]^{\beta}} \quad \text{if } S \in J_k(r), u \in J_k(r)$$

= 0 otherwise.

This state transition rule will favor transitions towards nodes connected by short edges with a high amount of trail. Formula shows how a transition can either exploit accumulated knowledge about the problem or explore new edges. While constructing a tour, ants visit edges and change their trail by applying the local updating rule.

$$\tau(r,s) \leftarrow (1-\rho).\tau(r,s) + \rho\tau\ 0$$

Where $0<\rho<1$ is the coefficient representing pheromone evaporation, and $\tau 0=(n*Lnn)^{-1}$ here Lnn is the tour length produced by the nearest neighbor heuristic and n is the number of cities. Once all ants have arrived at the destination, edges (r, s) belonging to the shortest tour change their trail by applying the following global updating rule

$$\tau(r,s) \leftarrow (1-\alpha).\tau(r,s) + \alpha\Delta\tau(r,s)$$

Where

$\Delta\tau(r, s)= (L_{gb})^{-1}$ if (r, s) Є globally best tour

= 0 otherwise.

Where $0<\alpha< 1$ is the pheromone decay parameter, and L_{gb} is the length of the globally best tour. Global trail updating provides a higher amount of trail to shorter tours. In a sense, this is similar to a reinforcement learning scheme in which better solutions get a higher reinforcement. The ACS algorithm can be described as follows ants are initially positioned on cities randomly. Each ant builds a tour by repeatedly applying the state transition rule. While constructing its tour, an ant also modifies the amount of pheromone on the visited edges by applying the local updating rule. Once all ants have terminated their tour, the amount of pheromone on edges is modified again by applying the global updating rule. [1], [2], [5], [8], [9].

1.3: ESTIMATED DISTRIBUTION BASED ANT SYSTEM

Informally, the EDAs work as follows initially better solutions are selected from a randomly generated population of solutions. Then, the true probability distribution of the selected set of solutions is estimated and new solutions are generated according to this estimate. The new solutions are then added into the original population, replacing some of the old ones. The process is repeated until the termination criterion is met. The original EDAs are for univariate problems, including PBIL (Population based incremental), UMDA (Univariate marginal distribution algorithm) etc.

PBIL is described below for its basic probability model will be used in ED-ACO. The PBIL probability model is a probability vector $p(x) = (p(x_1), p(x_2),\ldots, p(x_n))$, where $p(x_i)$ is the probability of generating a 1 in bit position *i*. The PBIL algorithm works as follow: In every generation, randomly generate M solutions, and then evaluate the probability vector and select N highest solutions to update p(x), where N<M. The probability update rule is

$$P_{l+1}(x) \backslash= (1-\alpha)\, P_i(x) + \alpha.\, 1/N \sum_{k=1}^{n} xi^k$$

Where $p_l(x)$ is the probability vector of l^{th} generation, $x_l^1, x_l^2, \ldots, x_l^N$ represent the N selected solution vectors, x_l^i is the i^{th} position in the solution vector which the probability vector is moved towards, and α is the learning rate [5].

1.4: ANT COLONY OPTIMIZATION BASED ON ESTIMATED DISTRIBUTION

The ED-ACS differs from the previous ACS in three main aspects:

i) The state transition rule also accumulates the probability knowledge about the problem.

ii) While ants choosing the next city, the probability factor is also evaluated in the random proportional rule. In the following we explain how the rules are changed [5].

iii) The pheromone addition in global updating rule is also consider the probability factor.

(1) STATE TRANSITION RULE: In the ED-ACO the state transition rule is as follows: an ant positioned on node chooses the city to move to by applying the rule given by

$$S = \begin{cases} \arg\max_{u \,\square\, J_k(r)} \{p(r,u)\tau(r,u).\eta(r,u)^{\beta}\} & \text{if } q \le q_0 \text{ (exploitation)} \\ S & \text{otherwise (biased exploration)} \end{cases}$$

Where p(r, u) is the probability vector represents the times of the previous ant passed the edge (r, u) in a tour.

(2) RANDOM-PROPORTIONAL RULE: In the ED-ACO the random-proportional rule also considers the probability distribution, and the rule is changed to

$$p_k(r,s) = \frac{[\tau(r,u)]^{\alpha}\,[\eta(r,u)]^{\beta}\,[p(r,u)]}{\sum [\tau(r,u)]^{\alpha},[n(r,u]^{\beta}\,[p(r,u)]} \quad \text{If } S \in J_k(r),\ u \in J_k(r)$$

= 0 otherwise

Where

$$p(r,u) \leftarrow p(r,u) + \Delta p(r,u)$$

Here

$$\Delta p(r,u) = \begin{cases} 1 & \text{if } p(r,u) \text{ tour done by ant } k \\ & \text{Otherwise} \end{cases}$$

The modified state transition rule favors transitions toward nodes connected by short edges and with a large amount of pheromone and probability. The parameter q0 determines the relative importance of exploitation versus exploration. If q<q0 then the best edge, according to previous state transition rule, is chosen (exploitation), otherwise an edge is chosen according to ED-ACO (biased exploration) [4].

POSITIVE FEEDBACK

When good solutions to problems are reinforced more (more pheromone is deposited on the path, that constitutes the solution) than solutions of lesser quality, this is referred to as positive feedback in the context of ant algorithms. In ant algorithms, artificial pheromone trails are deposited by ants in amounts proportionate to the quality of the solution. The evaluation of a solution is performed by a quality function *f* If the solution found is a particular good one positive feedback sometimes results in an autocatalytic effect (a "snow ball" effect), by which more and more ants are recruited to follow the path that constitutes that solution. This can lead to the system converging to a single solution.

The autocatalytic effect is not always desirable, since it may happen when the ants have found a locally optimal solution. The mass recruitment that occurs can leave the ants unable to discover new solutions. This situation is described as a stagnation of the system. It is the job of other mechanisms to ensure that the ants keep looking for better solutions [6].

NEGATIVE FEEDBACK

Negative feedback is often used to avoid premature convergence on a suboptimal solution (stagnation). If premature convergence happens, the ants are "satisfied" with the result found so far and have stopped looking for new and maybe better solutions.

An example of negative feedback in ant algorithms is artificial pheromone evaporation. Artificial pheromone evaporation is usually implemented as a reduction in pheromone, which reduces the amount of pheromone to a percentage of the original amount. This implies that the amount of pheromone removed is proportionate to the (original) amount of pheromone. This means that good paths need to be continually reinforced by ants in order to ensure that they retain their pheromone value since they evaporate faster [6].

(3) GLOBAL UPDATING RULE: In ED-ACO only the globally best ant is allowed to deposit pheromone. This choice, together with the use of the pseudo-random-proportional rule, is intended to make the search more directed: ants search in a neighborhood of the best tour found up to the current iteration of the algorithm. [4] Global updating is performed after all ants have completed their tours. The pheromone level is updated by applying the global updating rule of same to ACS; however the $\Delta\tau(r, s)$ is changed to:

$$\Delta\tau(r, s) = p(r, s)/\eta.L_{gb} \quad \text{If } (r, s) \in \text{ global best tour.}$$
$$= 0 \text{ otherwise}$$

CHAPTER 2

LITERATURE SURVEY

To proposed a memory-based ant colony algorithm for efficiently solving the bipartite sub-graph problem. In the proposed algorithm, artificial ant has memory of solution found previously, and can use it to construct a new solution. The proposed algorithm is evaluated by performing a large number of simulations. The simulation results showed that the proposed algorithm works remarkably well and is superior to its competitors to solve the bipartite sub-graph problem [10].

An Improved Ant Colony Optimization Algorithm Based on Dynamic Control of Solution Construction and Mergence of Local Search Solutions gives better performance then conventional ACO An improved ant colony optimization algorithm is proposed in this paper. Comparing with the conventional ant colony optimization algorithm, the proposed method has two highlights. First, a newly strategy based on the dynamic control of solution construction is adopted. The purpose of this strategy is to ensure ants to exploit the solutions at the beginning of searching procedure with large probability while at the end of the searching procedure the solutions provided by each ant are obtained by searching around the best-so-far solution. Second, to obtain a more reasonable solution, a mergence mechanism, based on the local search result of each ant, is employed. The experiments demonstrate that the proposed method has better performance than the conventional ACO algorithm [11].

The Improved Ant Colony Optimization Algorithm focuses on giving an improved method to deal with the two main problems involved in ACO, slow speed of convergence and poor ability to search better solution at the end of the search procedure, the improved algorithm chiefly includes two techniques: a more effective solution construction method based on dynamic control of parameter and a novel mergence mechanism of local search solution which will utilize the local search solution of each ant. The improved method will accelerate the convergence speed and give the algorithm with stronger ability to find better solutions in the total search procedure. q0 roughly speaking, the general scheme of the improved method is designed as the conventional ACO algorithm, but in the step of solution construction, apply new solution construction method based on dynamic control of q0. And after the step of local search we apply mergence of local search solutions. The details of the newly techniques will be given in the following subsections [12].

Ant Colony Optimization (ACO) algorithm is a new meta-heuristic for hard combinational optimization problem. An improved version of the ACO algorithm based on dynamic control of solution construction and mergence of local search solutions is proposed. Experiments have demonstrated that the proposed algorithm is more effective than the conventional ACO in the terms of convergence speed and the ability to finding better solutions [13].

Improving Noise Clustering Algorithm Using ant Colony Optimization, Noise clustering, as a robust clustering method, performs partitioning of data sets reducing errors caused by outliers. In many applications outliers contain important information and their correct identification are crucial. The original ant system algorithm is simplified leading to a generalized ant colony optimization algorithm that can be used to solve a wide variety of discrete optimization problems. It is shown how objective based clustering models function such as noise clustering can be optimized using particular extensions of this simplified ant optimization algorithm. Experiments with artificial dataset show that ant clustering (NC-ACO) produces better results [14].

An intuitive and easily realizable robust clustering scheme based on NC has been presented. FCM-ACO algorithm has been extended on the noise clustering algorithm. It can minimize objective function, but for finding cluster centers has not achieved enough improvement. For feature work we can extend FCM-ACO algorithm on the mega clustering algorithm. Also we can apply NC-ACO on the real and high dimensional data sets. Also The FCM-ACO algorithm presented here can be easily modified to be applied to other kinds of fuzzy optimization problems such as, for example, the fuzzy component assignment problem, fuzzy decision making problems, and other relaxations of discrete optimization problems [15].

“Making concessions in order to gain advantages” Improved Ant Colony Optimization for improving Job Scheduling Problems, the author *liusuqin, shuojun, menglingfen, lixingsheng* says Ant Colony Optimization (ACO) converges on the optimal path with pheromones cumulating and updating, adopting the mechanism of distributed parallel search. Ant colony system is well self-adoptive and dynamic with making full use of current feedback, which is similar to the dynamic performance of the grid and is proved to be an effective algorithm to solve scheduling problems. But the existing ant colony algorithm cannot solve the scheduling problems liking misusing good performance resources for minor purposes. This paper presents a “making concessions in order to gain advantages” Algorithm an improved algorithm based on Ant Colony Optimization (ACO) algorithm for job scheduling problems. Experimental results show that improved ACO approach can solve the problem and outperform ACO [16].

Author *ZHANG Fei Jun et. al.* [17] *Wei* proposed the ant system is a new meta-heuristic mainly for hard combinatorial optimization problems. It has been unexpectedly successful and known as Ant Colony Optimization (ACO) in recent years. Nowadays, a series of improvements have been made to the ACO, most of which focus on the exploitation of gather information to guide the search of ant colony towards better solution space but neglect the exploration of new tours. In order to enlarge the ants’ searching space and diversify the searching solutions, Meeting ACO is proposed here. The main strategy used in this new algorithm is to combine pairs of searching ants together to expand the diversification of the search. To make up the influence caused by limited number of meeting ants, a threshold constant is applied to make the algorithm function normally. As proved by the simulation experiments, the Meeting ACO is ranked among the best ACO for tackling the TSP problems.

A Meeting ACO was proposed here. The main strategy used in this new algorithm is to combine pairs of searching ants together. By doing so, the searching solutions can be diversified and the premature stagnation could be avoided at the same time. In order to counteract the limited number of meeting ants in the iteration, the update law is selected due to the comparison results between the number of meeting ants and the threshold constant. As shown by the simulation experiments, the Meeting ACO is proved to be one of the best ACO while tackling the TSP [18].

Ant Colony Optimization based on Estimation of Distribution for the Traveling Salesman Problem *Xu Chang, Xu Jun, Chang Huiyou* says,Ant Colony System algorithm is one of the best algorithms of ant colony optimization. However, the weaknesses of premature convergence and low efficiency greatly restrict its application. In order to improve the performance of the algorithm, a new Ant Colony Optimization algorithm based on Estimation of Distribution (ED-ACO) is presented. ED-ACO uses probabilistic model based on estimating the distribution of promising edges to adjust the state transition rule and the global updating rule. Furthermore, ED-ACO is significantly improved by extending with a local search procedure. We apply ED-ACO to traveling salesman problems and compare it to the previous finding. The results show that ED-ACO is an effective and efficient way to solve combinatorial optimization problems. The ED-ACS differs from the previous ACS in three main aspects:

(i) The state transition rule also accumulates the probability knowledge about the problem,
(ii) The pheromone addition in global updating rule is also consider the probability factor,
(iii) While ants choosing the next city, the probability factor is also evaluated in the random-proportional rule.

ED-ACO an improved Ant Colony System based on Estimation of Distribution. ED-ACO yields significant improvements in performance over ACS and performs at least at the same level of performance as MMAS. The differences between ED-ACO and ACS are mainly in the way the probability factor is considered while choosing the next city and updating global information. Furthermore, we extended ED-ACO by adding a local search phase and demonstrated that thereby the solution quality of ED-ACO can be significantly improved. The experiments also show that ED-ACO with local search outperforms ACS with additional local search [5].

An innovative ant colony algorithm called memory-based ant colony algorithm is proposed to solve the bipartite sub-graph problem. In the proposed algorithm, artificial ant has memory of solution found previously, and can use it to construct a new solution. Besides, in the proposed algorithm two kinds of pheromone and two kinds of heuristic information are also adopted to reinforce the search ability. The proposed algorithm is tested on a large number of instances and compared with other algorithms. The experimental results show that the proposed algorithm is superior to its competitors.

“Making concessions in order to gain advantages” improved Ant Colony Optimization for improving Job Scheduling Problems is present. Preliminary test shows that this approach has better performance than ACO, especially solving for “misuse good performance resources for minor purpose” problems. Our paper only take CPU speed into consideration, any other resources should be considered, for example, memory capacity, hard disk capacity [19].

Solving Continuous Optimization Using Ant Colony algorithm, one shortcoming of ant colony optimization is that it cannot be applied on continuous optimization problems directly. In this paper we propose a new approach for solving continuous optimization problems using ant colony algorithm. While the method maintains the framework of the classical ant colony algorithm, it replaces the discrete frequency in the ant selecting probability by a continuous probability distribution formula using the continuous integral instead of discrete summation. We also use the direction towards the optimum in each dimension as the heuristic information guiding the ants’ searching. Experimental results on benchmarks show that our algorithm not only has faster convergence speed than other similar methods, but also effectively improves the accuracy of solution and enhances its robustness [20].

Improved Ant Colony Optimization with Particle Swarm Optimization Operator Solving Continuous Optimization problems, Ant Colony Optimization (ACO) has the disadvantages such as easily relapsing into local optima and. Aimed at improving this problem existed in ACO; several new betterments are proposed and evaluated. In particular, pheromone mutation and Particle Swarm Optimization operator were inducted. Then an improved Ant Colony Optimization with Particle Swarm Optimization operator was put forward. It was tested by a set of benchmark continuous function optimization problems. And the results of the examples show that it cannot easily run into the local optimum and can converge at the global optimum.

Improved Ant Colony Optimization gives better result comparatively ACO the biggest error is 1% and N Cycle is 1000 iterations, the optimal rate is 100%. The best solution from 25 running is 0.999999, the worst is 0.999116. The average optimal solution of 25 times is 0.999591[21].

Improving Retrieval Effectiveness using Ant Colony Optimization, Software reuse is only effective if it is easier to locate and appropriately modify a reusable component than to write it from scratch. It is the use of existing software knowledge or artifacts also known as software components to build new software. There are two main problems in software reuse. First, classifying software modules in a component library is a major problem in software reuse. Second, identifying appropriate software components in a library or software component retrieval is an important task in software reuse: after all, components must be found before they can be reused. Many researchers have proposed various techniques to search and retrieve components. Proposed technique helps re-user to identify and retrieve software component. In its first step it matches keywords, their synonyms and their interrelationships. And then makes use of ant colony optimization, a probabilistic approach to generate rule for matching the component against the re-user query.

Ant Colony Optimization algorithm for retrieval of software components, The method shows very good values of precision and recall. As the method directs towards only the relevant components; recall of the method is always seen as 1. Higher value of precision can be achieved by in case of exact queries belonging to single or restricted domain. Precision degrades gracefully in case of indistinct queries, because this system extracts multiple domains belonging to these queries [22].

Elitist-Mutated Ant System versus Max-Min Ant System: Application to Pipe Network Optimization Problems, The Ant Colony Optimization Algorithm (ACOA) is a new class of stochastic search algorithm proposed for the solution of combinatorial optimization problems. Different versions of ACOA are developed and used with varying degrees of success. The Max-Min Ant System (MMAS) is recently proposed as a remedy for the premature convergence problem often encountered with ACOAs using elitist strategies. The basic concept behind MMAS is to provide a logical balance between exploitation and exploration. The method, however, introduces some additional parameters to the original algorithm, which should be tuned for the best performance of the method adding to the computational requirement of the algorithm. An alternative method to MMAS is proposed in this paper and applied to pipe network optimization problem. The method uses a simple but effective mechanism, namely Pheromone Trail Replacement (PTR), to make sure that the global best solution path has always the maximum trail intensity. This mechanism introduces enough exploitation into the method and more importantly enables one to exactly predict the number of global best solutions at each iteration of the algorithm without requiring calculation of the cost of the solutions created. The sub-colony of repeated global best solutions of the iterations is then mutated, such that a predefined number of solutions survive the mutation process. Two different mutation mechanisms, namely deterministic and stochastic mutation processes, are introduced and used. The first one uses a one bit mutation with a probability of one on some members of the sub-colony, while the second one uses a uniform mutation on the whole sub-colony. The probability of mutation in the second mutation process is adjusted at each iteration, so that the required number of global best solutions survives the mutation. The method is shown to produce results comparable to the MMAS algorithm, while requiring less free parameter tuning. The application of the method to a benchmark example in the pipe network optimization discipline is presented and the results are compared.

A new ACO algorithm was presented as an alternative to the Max-Min Ant System. The method exploits automatically balanced exploitative and explorative features. The exploitation of the method is provided by a simple but effective free-parameter procedure in which the global-best solution pheromone intensity is replaced by the current maximum pheromone trail, each time the global-best solution is updated. This procedure was shown to introduce enough exploitation into the method ensuring the convergence of the search to the global-best solution, irrespective of the value of the evaporation factor. The method offers the advantage of exactly predicting the number of global-best solutions of the iteration without requiring calculation of the cost of the trial solutions. Two mutation mechanisms, one deterministic and the other stochastic, were then used on the predicted global-best solutions to introduce a balancing exploration into the algorithm. The deterministic approach uses a one-bit mutation on a number of global-best solutions while in the stochastic one, all the global-best solutions undergo a uniform mutation process with an automatically calculated probability. Both of the mutation procedures were devised such that a predefined number of global-best solutions survive the mutation. The proposed algorithm was tested against a benchmark example in the water distribution network optimization literature and the results compared with that of MMAS. The results show that the proposed algorithm produces solutions comparable to that of MMAS, while introducing less free parameters to be tuned [23].

Designing Fuzzy-Rule-Based Systems Using Continuous Ant-Colony Optimization, The design of fuzzy rule based systems using continuous ant-colony optimization (RCACO).RCACO determines the number of fuzzy rules and optimizes all the free parameters in each fuzzy rule. It uses an online rule generation method to determine the number of rules and identify suitable initial parameters for the rules and then optimizes all the free parameters using continuous ant-colony optimization (ACO). In contrast to traditional ACO, which optimizes in the discrete domain, the RCACO optimizes parameters in the continuous domain and can achieve greater learning accuracy. In RCACO, the path of an ant is regarded as a combination of antecedent and consequent parameters from all the rules. A new path-selection method based on pheromone levels is proposed for initial-solution construction. The solution is modified by sampling from a Gaussian probability density function and is then refined using the group best solution. Simulations on fuzzy control of three nonlinear plants are conducted to verify RCACO performance. Comparisons with other swarm intelligence and genetic algorithms demonstrate the advantages of RCACO.

RCACO for FS optimization in a continuous space, the rule-generation approach in RCACO helps to automatically generate rules and determine their proper initial parameters. The framework of RCACO for parameter optimization is based on a graphic representation of the optimization problem and consists of nodes, edges, and ant paths. Therefore, RCACO can be regarded as a direct extension of ACO to solve continuous-optimization problems. A new path-selection strategy based on pheromone levels is proposed. The innovative idea of incorporating group best ant into new solution generation improves learning performance. Performance comparisons with different SI and GAs verify the effectiveness and efficiency of RCACO. The hybridization of other optimization algorithms and RCACO may further improve optimization performance and will be studied in the future [7].

GSP-ANT: An Efficient Ant Colony Optimization Algorithm with Multiple Good Solutions for Pheromone Update, Ant colony optimization (ACO) is a met heuristic for various optimization problems, especially the hard combinatorial optimization problems. However, existing ACO algorithms suffer from search stagnation and exorbitantly long computation time. To alleviate these shortcomings, an improved ACO algorithm, called GSP-ANT, is presented in this paper. It maintains a good solution pool (GSP) and alternately uses the optimal solution and suboptimal solutions in the pool to update pheromone. This enables ants to transfer among different solution regions and accordingly explore larger solution space. On the other hand, once a solution in the GSP is selected, it is continuously used for pheromone update in a certain number of iterations with the aim of exploiting the neighborhood of this solution intensively. By this means, both the intensification and diversification of the search are considered. The performance of GSP-ANT is examined experimentally on typical traveling salesman problems. Computational results indicate that GSP-ANT is a promising approach [24].

Description of GSP-ANT: The basic principle hiding in ACO is that search in the neighborhoods of good solutions might find even better solutions. In fact, during the run process of ACO, many solutions which are slightly worse than *S*gd will be produced. Their neighborhoods might contain some high-quality solutions. Therefore, if we can use these good solutions properly, it is hopeful to further enhance the performance of ACO. GSPANT, the algorithm proposed in this paper, follows basic algorithmic schemes of MMAS for static combinatorial optimization problems, whereas it adopts a new pheromone update rule. It does not employ the single solution *S*gd (or *S*id) for pheromone update, but depends on a set of good solutions found so far.

A new ACO algorithm called GSP-ANT as an attempt to explore the potential of enhancing the performance of ACO. Its main feature is to employ the solutions contained in a good solution pool for pheromone update. Our preliminary experimental results show that GSP-ANT offers promising performance. Although we take the TSP as an example while describing GSP-ANT, no specific information of TSP is used in designing algorithmic strategies. So GSP-ANT belongs to the general ACO framework and can be applied to other problems. In addition, a possible parallel strategy for GSP-ANT is simply analyzed in this paper; however we did not implement it in the experiment for hardware limitation. This avenue of investigation merits further research .

Quantum Computing-based Ant Colony Optimization Algorithm for TSP, *Xingwai Miao,Sheng Liu* says A novel self-adaptive Ant Colony Optimization algorithm based on Quantum mechanism for Traveling salesman problem(TQACO) is proposed. Firstly, initializing the population of the ant colony with superposition of Q-bit, Secondly, using self-adaptive operator, namely in prophase we use higher probability to explore more search space and to collect useful global information; otherwise in anaphase we use higher probability to accelerate convergence. This mechanism offers the ability to escape from local optima and can self-regulate the production of diverse antibodies. Because of the quantum superposition and rotation it can maintain quite nicely the population diversity than the classical evolutionary algorithm, because of the self-adaptive operator it can obtain more optimal solution and the solution quality is improved significantly.

Ant Colony Optimization based on Quantum mechanism for TSP (TQACO). Q-bit and quantum rotation gate adopted in quantum-inspired evolutionary algorithm (QEA) are introduced into TQACO to represent and update the pheromone respectively. Due to its linear superposition of Q-bit, usually it can obtain optimal solution faster and the solution quality is improved significantly with self-adaptive operator. Integration of the quantum computing in the Ant Colony Optimization procedure can yield significant improvements in both the convergence [25].

Multi-direction Searching Ant Colony Optimization for Traveling Salesman Problems, Traveling salesman problem (TSP) is one of the most famous NP-hard problems, which has wide application background. Ant colony optimization (ACO) is a nature-inspired algorithm and taken as one of the high performance computing methods for TSP. Classical ACO algorithm like ant colony system (ACS) cannot solve TSP very well. The present paper proposes an ACO algorithm with multi-direction searching capacity to improve the performance in solving TSP. Three weight parameter settings are designed to form a new transition rule, which has multi-direction searching functions in selecting the edges of the TSP tour. The experimental results of solving different kinds of TSP problems indicate the proposed algorithm performs better than the famous ACO algorithm ACS.

An improved ant colony optimization (MSACO) for solving traveling salesman problems. The improvement is mainly on a new transition rule which can lead artificial ants search in local, global and comprehensive directions. In all of the test instances including small, bigger and large size TSP, MSACO performs much better than ACS [26].

On the Dynamic Ant Colony Algorithm Optimization Based on Multipheromones, *Jun-liang CHEN* says An algorithm DACO (Dynamic Ant Colony Optimization Algorithm Based on Multipheromones) is put forward to apply to the dynamics of web services state and QoS in service composition optimization. In order to denote users' needs more accurately, this algorithm sets multiple pheromones. The DACO is also improved based on experiment in order to make it better and faster converge to optimization value. Simulation experiment in this Report shows that the DACO is more effective than Ant Colony Algorithm and a Genetic Algorithm applied to services composition.

There is only one kind of pheromone in ACO, which cannot deal with the question of multiple attributes in web services composition; in ACO, the path and path weight is stable, and cannot fit for dynamic web services composition. Additionally, ACO chiefly makes use of positive feedback to enhance relative optimization. In this method, when evolving to a certain iterative, the increasing enhancement of the pheromone on local optimization paths makes the ants gather on fewer paths, and prematurity and stagnancy will appear. In this case, the optimization value is only local. Aiming at addressing the above disadvantages, the DACO is put forward to fit for the dynamic services composition optimization and to promote the algorithm's effectiveness [27].

Web services state is random and instable and the QoS attributes of the service will change sometimes, so in the process of services composition optimization some cases should be considered, such as uselessness of the services, increase of new services, and QoS change. These questions can be solved by deleting an arch, adding an arch or changing the weight of the architecture in the services composition graph. Adding an arch must meet such requirements as not forming loop. When services are useless, the corresponding arch need to be deleted, which is a more complicated case, for the deleting of one arch may lead to the emerging of suspension points in the graph.

ACO applied in the services composition optimization system, and put forward an improved DACO which can address such problems in the optimization process such as dynamics instability, multiple QoS attribute limitations of web services. The effectiveness of the algorithm is verified in a “tourism service recommendation system”, with the result that the DACO has better performance than typical Ant Colony Algorithm and the Genetic Algorithms applied to services composition. Hopefully, and make a research that applies ACO to semantic web to make the improved algorithm individualized.

Multi-Colony Parallel Ant Colony Optimization on SMP and Multi-Core Computers, The propose Multi-Colony Parallel Ant Colony Optimization is an effective implementation of the Ant Colony Optimization metaheuristic on actual shared-memory parallel computers. To deal with the management of multiple colonies which use a global shared memory to exchange information, The report considerable speedups on a SMP node of multi-core processors while witnessing solution quality equal or greater than the original sequential implementation.

The ACO metaheuristic offers an interesting potential for parallelization, but also some challenges due to its solution construction phase and the data structures it has to maintain. Since it is a relatively recent resolution approach, a limited amount of literature can be found on its parallelization. *Bullnheimer* have proposed two parallelization strategies for the Ant System on a message passing, distributed-memory architecture. The first one is a low-level, synchronous strategy that aims to accelerate computations by distributing ants to processors in a master-slave fashion. At each cycle, the master broadcasts the pheromone structure to slaves, which then compute their tours in parallel and send them back to the master. The time needed for these global communications and synchronizations implies a considerable overhead. The second strategy aims to reduce it by letting the algorithm perform a given number of cycles without exchanging information. The authors conclude that this partially asynchronous strategy is preferable due to the considerable reduction of the communication overhead. The works of *Randall and Lewis Craus and Rudeanu* . *Stutzle* et. al. [28] are based on a similar parallelization scheme .

Another parallelization strategy, also based on a messagepassing, distributed memory architecture, is presented by *St¨utzle* parallel execution of multiple independent copies of the same algorithm. *Middendorf* extended this scheme by introducing four information exchange strategies between multiple ant colonies. Exchange of globally best solution, circular exchange of locally best solutions, circular exchange of migrants and circular exchange of locally best solutions plus migrants. It is shown that it can be advantageous for ant colonies to avoid exchanging too much information and exchanging too often. Giving up on the idea of exchanging whole pheromone information, they based their strategy on the exchange of a single solution at each exchange step. *Chu ,Manfrin* have also proposed different information exchange strategies for the multiple ant colony scheme. Many parameters are studied like the topology of the links between processors, the frequency of information exchange and the nature of the exchanged information. These strategies are implemented using MPI on distributed memory architectures. As it can be seen, a growing interest to parallel ACO has been manifested in the last few years. Also, much of the research so far has been directed towards two main parallelization strategies. Parallel ants and multiple ant colonies and also note that apart from our previous works where implement a parallel ACO with Open MP to solve an industrial scheduling problem propose a shared memory implementation for the TSP and compare the performance of shared and distributed memory implementations all parallel implementations were made according to the message passing paradigm on distributed-memory architectures [29].

Since our previous works were based on the parallel ants parallelization scheme, no actual multi-colony implementations based on shared-memory architectures can be found in the literature. With the growing progress and availability of SMP and multi-core architectures, we believe that the shared-memory paradigm is worth more investigation for parallel ACO. And also note that papers devoted to the TSP limits their attention to relatively small problem instances. Since one of the main benefits of parallel computing to metaheuristics is the possibility of tackling larger problems, attention should be directed towards instances that contain more than a few hundred cities. To partially fill this gap and as a report of our current research on shared-memory parallel ACO, we propose a parallel implementation based on a multi-colony strategy and report experiments on a shared-memory parallel computer.

A parallel multicolony implementation of ACO on shared memory computers. Two strategies were used. Multiple independent colonies and multiple cooperating colonies. We showed that both strategies, which are the most popular approaches in the ACO literature, can be efficiently implemented on SMP and multi-core computers containing up to 8 processors. In fact, the proposed implementations managed to provide comparable or better solution quality with considerable speedups. With the increasing availability of SMP and multi-core architectures, shared-memory parallel metaheuristics are a promising way of providing efficient and robust optimization tools for a wide range of researchers and practitioners. Still, as it is the case in the field of parallel ACO and parallel metaheuristics in general, much can still be done as this paper brings its share of questions and research avenues. For example, a better understanding of the implicit system costs and global memory contention could help us improve speedups and scalability on high numbers of processors. Also, even though we used a rather simple information exchange strategy for our cooperative strategy, we realize that much more thought can be put into it. And plan both to implement efficient schemes found in the literature and to exploit the shared memory concept to devise new and better ones [30].

Multiple Ant-Colony Optimizations for Network Routing, An ANT is a mobile agent that is capable of solving various kinds of routing and congestion problems in computer networking by continuously modifying routing tables in respond to congestion. In a distributed problem solving paradigm, a society of ANTs (each contributing some information) collaborate to solve a larger problem. In recent years, Ant-based algorithms were used to solve classical routing problems such as: Traveling Salesman Problem, Vehicle Routing Problem, Quadratic Assignment Problem, connection-oriented /connectionless routing, sequential ordering, graph coloring and shortest common super sequence. This paper introduces the general idea of Ant-based algorithms with a focus on Ant Colony Optimization (ACO), and their features, strengths, weaknesses and applications in network routing. The contribution of this paper is the proposal of a multiple ant-colony optimization (MACO) approach for network routing.

With the growing importance of telecommunications and the Internet, more complex networked systems are being designed and developed. As networked systems become more complex so are the underlying software and the control rules. The challenges of dealing with the vast complexity of networking tasks such as load balancing, routing and congestion control accentuate the need for more sophisticated (and perhaps more intelligent) tools to solve these problems.

ACO algorithms and drew analogies to solving problems in network routings. However, it is noted that most ACO algorithm suffers from the problem of stagnation. While discussed and compare the various approaches of mitigating stagnation, the contribution of this paper is proposing a new approach for mitigating stagnation. By adopting multiple colonies of ants to search for optimal paths enhances adaptiveness and reduces the chance of stagnation. In a dynamic network where links and nodes can change constantly, MACO increases the probability of allowing new or better paths to be explored. In network routing, the MACO approach can be realized by maintaining different pheromone tables. One of the possible applications of MACO is solving the load-balancing problem in circuit-switched network. Work on applying MACO for load balancing in circuit-switched network has already been carried out and experimental results are reported in an upcoming paper by both authors. Nevertheless, this research does not claim that MACO will totally eradicate stagnation. This research offers a new direction for Ant-based Optimization and it is hope that this work can shed new light in ant-based research [31].

A Rescue Robot Path Planning Based on Ant Colony Optimization Algorithm is proposed by *Xiaoyong Zhang*. A robot path planning algorithm based on improved ant colony optimization. The ant colony algorithm is used for a global path planning in robot rescue. A target attracting function is introduced to guide the searching process which can improve the search quality of ant colony algorithm in the complex and dynamic environment. The affectivity of proposed algorithm is verified in a standard test bed, RoboCupRescue simulation system.

Robot path planning is an important part of the robotics. Its main task is to find a shortest path without barrier or with a minimum price from the designated original node to destination node in the environment which has obstacles. Path planning is an important ability of intelligent robots, and has great significance for accomplishing the robot's task. RoboCup Rescue simulation system is a rising project of RoboCup competition. It simulates the urban disasters scene and the activity of rescue robot in real life after earthquake. Rescue robot must make a dynamic path planning depending on current environment information. A successful path planning makes it possible for robot to arrive at the destination and rescue quickly. In addition, the quality of path planning has a direct impact on the effectiveness of rescue .

Ant colony algorithm is a kind of colony intelligent algorithm and it has well application prospect. The aim at the complexity of the environment that rescue agents facing in, and introduced the path planning for agent based on improved ant colony optimization. This approach, by using the ant colony algorithm for global path planning, adopts the objective attraction function to guide ants' search, which improves the search quality of the ant colony algorithm in complex and dynamic environment. By applying it in the RCRSS, the performance of this algorithm is proved in terms of validity and practicability.

An Ant Colony Optimization Algorithm Based on the Nearest Neighbor Node, Choosing Rules and the Crossover Operator The ant colony optimization algorithm (ACO) has a powerful capacity to find out solutions to combinatorial optimization problems, but it still has two defects, namely, it is slow in convergence speed and is prone to falling in the local optimal solution. Against the deficiencies of this algorithm, in this study to proposed an ACO based on basic ACO algorithm based on well-distributed on the initiation, the nearest neighbor node choosing rules and with crossover operator. In the initiation of the algorithm, the convergence speed of the ACO is increased by distributing the ant colony evenly in all the cities and adopting the nearest neighbor choosing node rule and making crossover computation among better individual ants at the end of each round of cycle when each ant chooses the next city. The experiment results indicate that the ACO proposed in this study is valid.

In order to improve the performance of the ACO in these two aspects, many scholars have conducted extensive research and come up with many improvement methods. proposed an MMAS algorithm (max-min ant system) whose basic idea is to limit the pheromone density within the range of [min, max] so as to overcome the stagnation problem and meanwhile enhance the pheromone density along the path an individual ant has traveled in each iteration, thus increasing the convergence speed of the ACO. Reference proposed an approach to improve the ACO by way of changing the pheromone density dynamically. Specifically, on the one hand, this approach depends on the dynamic change volatilization coefficient to alter the local pheromone density; on the other hand, when the overall pheromone density is being updated, the pheromone density along the shortest path will be strengthened, while the pheromone density along the longer paths will be weakened. Reference proposed an algorithm on the basis of the ant evolution rules. The reference proposed a meeting algorithm whose basic idea is to use two ants to complete the search of the same path so as to quicken the search speed and overcome the limitation of the ACO's deficiency of being easy to fall in the local optimal solution, Mutation strategy to quicken the local search. Increasing the convergence speed and obtaining the global optimal solution. The two deficiencies remain to the bottleneck constraining [32].

The ACO from being widely applied in large-scale optimization problems. For this end, on the basis of the solution to the Traveling Salesman Problem (TSP), we studied an ACO algorithm based on well-distributed on the initiation, the nearest neighbor node choosing rules and with crossover operator, and hereafter WNCACO for short. The experiment results show that the algorithm proposed in this study can substantially increase the convergence speed of the ACO.

WNCACO proposed in this study can rapidly converge to the best result. In its initiation, this algorithm distributes all the ants well into various cities so as to obtain the global optimal solution and then overcome the limitation of being easy to fall into the local optimal solution. Every time, when each ant is finding the next city, the nearest neighbor node choosing rule will be adopted and crossover operation will be made among the better results in each cycle so as to attain the goal of realizing the fast convergence of the algorithm. Our experiments have shown that the method employed by the present study is effective to increase the convergence speed of the algorithm.

A new approach of reducing redundancy condition is put forward based on an information consistency relationship of equivalent classification. The best coverage of data coordinated about the decision-table of an information decision system and the significance of attribute Ri to the system is chosen as a rule of heuristic information of the attribute j by way of ant colony optimization. An algorithm of attribute reduction based on ant colony optimization is suggested to solve N-P hard problem in derivation of a minimal set of attribute reduction. The merit of the optimization of attribute reduction on ant colony algorithm is validated by the proposed application here.

Applying Self-adaptive Ant Colony Optimization for Construction Time-Cost Optimization, The time-cost optimization (TCO) problem is a multi objective problem, which attempts to strike a balance between resource allocation costs and project schedule duration. In this paper, a self-adaptive ant colony optimization (SACO) with changing parameters based on information entropy has been employed to model time-cost optimization problem, which overcome the intrinsic weakness of premature of the basic ant colony optimization (ACO) by adjusting parameters according to mean information entropy of the ant system. A computer simulation with Matlab7.0 based on a prototype example has been carried out on the basis of SACO for TCO problem. The test results show that the SACO for TCO model can generate a more optimal cost under the same duration and achieve a better Pareto front than other models. Therefore, the SACO can be regarded as a useful approach for solving construction project TCO problems.

A self-adaptive ACO has been introduced to model TCO problem so as to optimize the project objectives of duration and total cost. We use information entropy to measure the uncertainty of the ant system. The bigger entropy is, the more uncertainty ants choose the path; the smaller entropy is, the more certainty ants choose the next cities. The parameters α and β have been controlled by the mean information entropy S , and when the $S < 0.01$ the algorithm stops. In the early stage of the algorithm, α is small and β is big, which makes the ants search solutions in global feasible space and overcome the premature and trapping local best solutions [33].

Ant Colony Optimization for the Traveling Salesman Problem Based on Ants with Memory is improved ACO based algorithm proposed by *Bifan Li1, Lipo Wang1* **.**A new model of Ant Colony Optimization (ACO) to solve the traveling salesman problem (TSP) by introducing ants with memory into the Ant Colony System (ACS). In the new ant system, the ants can remember and make use of the best-so-far solution, so that the algorithm is able to converge into at least a near-optimum solution quickly. To tested the algorithm in 3 representational TSP instances and compared the results with the original ACS algorithm. According to the result we make amelioration to the new ant model and test it again. The simulations show that the amended ants with memory improve the converge speed and can find better solutions compared to the original ants.

Mant is a very simple but interesting novel approach to the ACO system. It has been shown to compare favorably with ACS algorithm, and Its amelioration model probabilistic Mant got inspiring performance in ACS. However, competition on the TSP is very tough, and a utilization of best-so-far tour information which converges these solutions to a near-optimum seems to be a useful strategy [34].

Ant Colony Optimization for Configuration, An inherent difficulty in enumerative search algorithms for optimization is the combinatorial explosion that occurs when increasing the size of the input. Among incomplete algorithms that address this issue, Ant Colony Optimization (ACO) uses a combination of random and heuristic methods plus reinforcement learning, which proved efficient on a wide range of configuration problems. This paper presents results in applying an ACO-based algorithm to configuration, which to the best of our knowledge was never investigated before. We describe how the nature of unbounded configuration problems impacts the ACO approach due to the presence of set-variables with open domains. We propose an ACO framework able to deal with those issues through an original pheromone model and algorithm. We also present the use of Particle Swarm Optimization (PSO) to converge towards good parameter sets. Finally, we provide early experimental results, both for random problem instances *and the"racks*" optimization problem.

A configuration task consists in building (a simulation of) a complex product from components picked from a catalog of types. In the general case, neither the number nor the actual types of the required components are known beforehand. Components are subject to relations, and their types are subject to inheritance. Constraints generically define all the valid products. A configurator expects as input a fragment of a target object structure, and expands it to a solution of the configuration problem, if any, adding all necessary elements during search. This is usually achieved with an enumerative procedure. This first-order logic problem is semi-decidable in the general case. The reader can refer to for a deeper introduction to configuration [35].

The presented a framework and early experimental results in using ant colony optimization to address finite model search for configuration problems. To the best of our knowledge, using stochastic techniques was yet unexplored in this field. Furthermore, the presented work deals with original first-order issues in ACO algorithms which may be reused outside of configuration’s scope. The results are encouraging considering the difficulty of unbounded configuration problems. Ongoing research involve using value and variable heuristics, still absent from these results, implementing known ACO algorithm improvements, as well as mixing ACO with local search.

An Approach of Optimal Path Generation using Ant Colony Optimization, *Dorigo, M. & Stuetzle* proposed and this idea can be used for software testing. Software Testing is one of the indispensable parts of the software development lifecycle and structural testing is one of the most widely used testing paradigms to test various software. Structural testing relies on code path identification, which in turn leads to identification of effective paths. Aim of the current paper is to present a simple and novel algorithm with the help of an ant colony optimization, for the optimal path identification by using the basic property and behavior of the ants. This novel approach uses certain set of rules to find out all the effective/optimal paths via ant colony optimization (ACO) principle. The method concentrates on generation of paths, equal to the cyclomatic complexity. This algorithm guarantees full path coverage. A novel algorithm for the path prioritization in a simple and effective manner, it has been identified that one of the software engineering areas with a more suitable and realistic use of artificial intelligence techniques is software testing and those techniques are known as a met heuristic approach. The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be used to find optimal paths through graphs. The Algorithm is inspired by the behavior of ants in finding paths from the colony to food. C. Peng Lam and others use this ACO to generate test sequences for state-based software testing.

A model for path prioritization technique by using ant colony optimization, The result got by applying proposed method is very encouraging. And taken the directed graph approach to model the system and shown the different paths of the model during the execution. After successful execution of algorithm, it automatically selects a best path sequence which covers the maximum coverage at least once. Soft computing techniques can evaluate path based testing which may ultimately help software industry to a greater extent. A number of extensions and applications of this model may be possible by using the different met heuristic techniques.

Relativity Pheromone Updating Strategy in Ant Colony Optimization for Constrained Unit Commitment Problem, *Songsak Chusanapiputt* proposedA development of the Enhanced Ant Colony Optimization (EACO) based on a novel approach of the Relativity Pheromone Updating Strategy (RPUS) for solving constrained unit commitment problem which cooperates with the Candidate Path Management Module (CPMM) embedded the Effective Repairing Heuristic Module (ERHM) in reducing search space and recovering a feasible optimality region so that a high quality solution can be acquired in a very early iterative. The adoption of RPUS not only enhances the search convergence of EACO, but also provides relatively pheromone information that is suitably exploited for a good guidance of search process. The EACO algorithms have been performed on a test system up to 100 generating units with a scheduling time horizon of 24 hours. The numerical results show better economical saving in the total operating cost when compared to the previous literature results. Furthermore, the proposed EACO topology can remarkably speed up the computation time of ant colony optimization algorithms, which is favorable for a large-scale UC problem implementation [36].

A novel algorithm of ant colony optimization based on the relativity pheromone updating strategy has been presented to the constrained unit commitment problem. It is essential to enhance a standard ACO performance based UC problem with the addition of effective algorithms for providing more suitability of pheromone updating paradigms in order to obtain a satisfactory UC solution within a reasonable CPU time. By avoiding the infeasible of UC solutions and recovering high quality candidate path neighbors, in this paper, the effective repairing heuristic module incorporated with the candidate path management module is introduced as an effective tool for UC problem solution so that the execution time of the proposed

EACO is significantly reduced. Study results have shown that these new decision schemes can effectively improve the solution optimality of the method at a sight cost of computational time increment. The performance of the proposed EACO algorithm is compared to those of the existing literature methods in dealing with the constraints of UC problem and reaching better UC solutions have been validated in this paper. Additionally, due to the proposed EACO affords the benefits of good reasonability to operate based on the populations evolving and the pheromone trail intensity evaluations according to their search performance relatively, noticeably improved. When compared to the DACO. These advantages indicate that the proposed EACO has satisfactory performance for applying to unit commitment problems [37].

An Ant Colony Optimization Algorithm to Aggregated Multicast Using the Idea of Bin Packing, Large-scale deployment of multicast applications is limited by the number of states that are set in routers for multicast groups. As a new approach to multicast state reduction, aggregated multicast forces multiple multicast groups sharing a common distribution tree. An ant colony optimization algorithm to aggregated multicast is proposed. Inspired by bin packing problem, relative fullness is used as an important component to define fitness function. To improve the algorithm's convergence time, heuristic information is introduced according to changes of aggregated trees' bandwidth waste rate. After each iteration a new pheromone update rule is proposed. Simulation results show that this algorithm performs well in scenarios with bigger bandwidth waste rate or larger network scale. Compared with greedy algorithm by running for the same amount of time and in the same network topology, the algorithm has better optimization performance.

Ant colony optimization algorithm AM-ACO is proposed. Inspired by bin packing problem, relative fullness is used to define fitness function. To improve the algorithm's convergence time, heuristic information is introduced according to the changes of aggregated trees' bandwidth waste rate in aggregation process. after each iteration there are always several solutions with the same minimum number of aggregated trees and the relative fullness of each aggregated tree is different. To utilize the knowledge comprehensively, a new pheromone update rule is proposed. Unlike traditional algorithms, candidate tree set needs not to be computed in AM-ACO algorithm, and the executive time for each iteration is not sensitive to bandwidth waste rate any more. In this way, the AM-ACO algorithm performs also well in scenarios with bigger bandwidth waste rate or larger network scale. Compared with greedy algorithm, the AM-ACO algorithm has better optimization performance.

A Novel Bi-directional Convergence Ant Colony Optimization with SA for Job-Shop Scheduling, *Yan-hong Wang and Peng-zhu Pan* proposed The bi-directional convergence ant colony optimization can effectively solve job-shop scheduling, but it often fall into local optimization result. While the simulated annealing has some characteristics can make the convergence process jump out of the local optimum. In this paper an application of the simulated annealing to be integrated into the bi-directional convergence ant colony optimization is proposed to tackle the complex job-shop scheduling problem. First, we propose an improved ant colony optimization structure for this problem by extending the convergence process with a simulated annealing procedure for the job-shop scheduling problem. Then we develop a new solution generating rules, which uses random choose, judge, exchange three steps for constructing the simulated annealing to fit the special job-shop scheduling problems and improve the constructed solutions. Therefore, when the ant colony optimization fall into a local optimum, simulated annealing is used to help the intermediate result jumping out of local optimum, and the global optimum is improved in turn. To compare this algorithm to an exist bi-directional convergence ant colony optimization by Wang, Cao and Dai to job-shop scheduling. Simulation results show that our algorithm works well when applied to the complex job-shop scheduling instances.

A bi-directional convergence ant colony optimization (BCACO) has been presented, which is based on an elite strategy to improve the convergence rate. However, the results were still quite far from the performance of the most satisfactory algorithm, which would easily fall into a local optimum when iterating. Integrated the simulated annealing to ant colony optimization, and proposed a SA based bi-directional convergence ant colony optimization is proposed try to tackle the job shop scheduling problems. It employ the simulated annealing to help to improve the capable of global search, and thus to enhance the efficient of bi-directional convergence ant colony optimization. The performance tests were carried out by using the classical job shop instance and the result of the algorithm were compared with the other competing algorithms. The results show that SBACO is especially suited to the application in job-shop scheduling instances; it can make the better quality of solution than other algorithm, and improve the speed of convergence. However, to the large-scale problem, the quality of solution has some reduce. Hence, increase the quality of solution of large-scale problem would be of interest.

Ant Colony Optimization for Multi-objective Optimization Problems, A generic algorithm based on Ant Colony Optimization to solve multi-objective optimization problems. The proposed algorithm is parameterized by the number of ant colonies and the number of pheromone trails. This algorithm is parameterized by the number of ant colonies and the number of pheromone trails. And tested four variants of this algorithm on the multiobjective knapsack problem (MOKP). m- ACO4(1,m), where m is the number of criteria to optimize, returns globally the best results [38].

A New Pheromone Control Algorithm of Ant Colony Optimization, in this algorithm author proposed. Ant Colony Optimization algorithm is a new meta-heuristic that combines distributed computation, auto-catalysis (positive feedback) and constructive greedy heuristic in finding optimal solutions for combinational optimization problem Compared to previous meta-heuristics, such as genetic algorithms and simulated annealing algorithms, ACO algorithm is still one of best methods to solve TSP. The traveling salesman problem (TSP) is one of the most important problems in combinational optimization. In relation to ACO approach of TSP, the main search mechanism is the pheromone control. While building a tour, each ant modifies the amounts of pheromone on the edge. Once all ants have constructed the tours, once more the amounts of the pheromone on all edges are modified according to update rule of the pheromone. Therefore, the pheromone control algorithm is the most important in ACO. And propose a new pheromone control algorithm. The proposed a new pheromone control algorithm. The proposed algorithm prevented the premature convergence. The experiments verified that the proposed algorithm is effective for improvement of search performance and for reduction of processing steps.

On the Implementation of Ant Colony Optimization Scheme for Improved Channel allocation in Wireless Communication, *P.M. Papazpglou* says channel allocation in wireless communication systems is one of the fundamental issues. The corresponding allocation schemes cannot be static due to the dynamically changing traffic conditions and network performance. Thus, more sophisticated strategies adapted to current network conditions must be investigated and applied. Recently, various approaches have been proposed for channel allocation based on intelligent techniques such as multi-agent technology and genetic algorithms. These approaches constitute heuristic solutions to resource management problem. The ant colony optimization approach has been proposed for solving the channel allocation problem in wireless communication systems. In this paper, a comprehensive heuristic approach for solving the channel allocation problem based on intelligent techniques such as multi-agents and ant colony optimization is proposed. Moreover, important implementation issues such as thread execution sequence are also presented. Finally, the simulation results show the performance improvement of the proposed ant colony optimization algorithm as well as the multi-agent modeling approach.

The efficiency of channel allocation schemes is a major issue in wireless communication networks. Channel allocation schemes cannot be static due to the dynamically changing traffic conditions and network performance. Thus, a more sophisticated model must be designed. The Multi Agent concept gives the opportunity to adapt the network services to current user needs and network behavior. The sequence of channel allocation and call servicing based on defined priorities affect the network performance. Applying the multi agent concept combined with several herein proposed novel cooperative/competitive negotiation schemes to channel allocation, simulation model performance can be significantly improved.

On the other hand, in a large scale wireless network, even small differences in the channel allocation procedure may largely affect network performance. Thus, more efficient channel allocation schemes based on intelligent techniques (especially ant colony optimization) must be further investigated. Although multi threading technology constitutes a valuable tool for alternative implementation of simulation models which support concurrent events, however, on the other hand, the existence of serious drawbacks (e.g. deadlocks, synchronization, etc) based on specific technology and platform constraints must be taken into consideration .

Ant Colony Optimization Algorithm for Topology Design of Distributed Local Area Networks, Ant colony optimization (ACO) is a powerful optimization technique that has been applied to solve a number of complex optimization problems. One such optimization problem is network topology design of distributed local area networks (DLANs). The problem requires simultaneous optimization of a number of objectives, such as monetary cost, average network delay, hop count between communicating nodes, and reliability under a set of constraints. This paper presents a multi-objective ant colony optimization algorithm to efficiently solve the DLAN topology design problem. The multi-objective aspect of the problem is handled by incorporating fuzzy logic in the ACO algorithm. The performance of fuzzy ACO is evaluated through comparison with a fuzzy simulated annealing algorithm. Empirical results suggest that the fuzzy ACO produces results of equal quality when compared with a fuzzy simulated annealing algorithm.

An approach for topology design of distributed local area networks based on a multi-objective ant colony optimization algorithm. The multi-objective aspects were addressed by incorporating fuzzy logic in the algorithm. The performance of the fuzzy ACO was evaluated with respect to different algorithm parameters. A comparison with fuzzy simulated annealing algorithm showed that the fuzzy ACO produced solutions of the same quality as that of fuzzy simulated annealing. [39].

CHAPTER 3

GAPS IN STUDY PROBLEM DEFINITION & METHODOLOGY

Ant Colony Optimization (ACO) is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems. The first algorithm which can be classified within this framework was presented in 1991 and, since then, many diverse variants of the basic principle have been reported The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with a posteriori information about the structure of previously obtained good solutions. Metaheuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic either a constructive heuristic starting from a null solution and adding elements to build a good complete one, or a local search heuristic starting from a complete solution and iteratively modifying some of its elements in order to achieve a better one. The metaheuristic part permits the low level heuristic to obtain solutions better than those it could have achieved alone, even if iterated. Usually, the controlling mechanism is achieved either by constraining or by randomizing the set of local neighbor solutions to consider in local search [11], [40].

ACO is a class of algorithms, whose first member, called Ant System, was initially proposed by Colorni, Dorigo and Maniezzo The main underlying idea, loosely inspired by the behavior of real ants, is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective behavior emerging from the interaction of the different search threads has proved effective in solving combinatorial optimization (CO) problems.

3.1: THE ANT COLONY OPTIMIZATION METAHEURISTIC

The Ant Colony Optimization (ACO) metaheuristic is a recently proposed discrete optimization metaheuristic for solving NP-hard problems. In the following it will present a formal definition of the problem representation on which ACO algorithms work. And will describe formally how solutions are constructed in the representation and present the ACO metaheuristic as pseudo-code.

The central properties of ACO are based upon the self-organized collective foraging behavior of ants. The key property collective in the foraging behavior of ants is their ability to find shortest paths between the location of their anthill and the location of food sources. When the ants move on a path between their anthill and the location of a food source they lay a pheromone trail. Other ants can then follow these generated paths of pheromone trails. This means that ants tend to converge on the same path, as illustrated in the example above.

In ACO, solutions are constructed repetitively by adding solution components to partial solutions stochastically. Solutions are constructed by taking into account (i) heuristic information when adding solution components (if available), and (ii) (artificial) pheromone trails which change dynamically based on the experience of the ants. Stigmergy handles the propagation of experience between ants [1], [5].

A number of algorithms following the ACO metaheuristic have been presented in recent years. The problems solved can be divided into two domains – static and dynamic problems. The basic properties of the algorithm do not change because of this, and the problems are in both cases represented by graph

The ACO metaheuristic has been used as a template for algorithms that has achieved world-class performance in both domains. Nevertheless, because of the ants' inherent ability to adapt to changes in the environment, it find that ACO algorithms are especially well suited to solving for example routing problems in networks, which constitute a dynamic problem domain.

Ant colony optimization is a new metaphor for solving combinatorial optimization problems and has been applied broadly in recent years. It has been applied to many combinatorial optimization problems, e.g. traveling salesman problems, network routing problems, graph coloring problems, quadratic assignment problems and so on.

In general, one of the most important aspect in study of heuristic algorithms is the balance between intensification and diversification. Too much emphasis on the former can make agents converge to a local optimum and to much emphasis on the latter can cause an unstable state, though these two factor are essential as we need to accelerate convergence and the latter to find better solutions [9].

3.2: MAIN VARIANTS OF ACO - MAX-MIN ANT SYSTEM

MAX-MIN Ant System (*MM*AS) is an improvement over the original Ant System idea. *M*MAS was proposed by StÄutzle and Hoos and introduces the following two changes:

1. Only the best ant can update the pheromone trails,
2. The minimum and maximum values of the pheromone are limited.

Equation takes hence the following new form:

$$\tau(r,s) \leftarrow (1-\rho).\tau(r,s) + \Delta\,\tau best$$

Where

$\Delta\tau$ *best* is the pheromone update value defined by:

$$\Delta\tau\ best = Q.(L_{best})^{-1} \text{ if } (r, s) \text{ ▯ globally best tour}$$
$$= 0 \text{ otherwise.}$$

L_{best} is the length of the tour of the best ant. This may be (subject to the algorithm designer decision) either the best tour found in the current iteration-iteration-best, Lib-or the best solution found since the start of the algorithm-best-so-far, Lbs-or a combination of both. Concerning the limits on the minimal and maximal pheromone values allowed, respectively τ min and τ max, StÄutzle and Hoos suggest that they should be chosen experimentally based on the problem at hand. The maximum value τmax may be calculated analytically provided that the optimum ant tour length is known. In the case of the TSP, τmax is given by:

$$\tau\ \max = \frac{1}{\rho\ .L}$$

Where L is the length of the optimal tour, The minimum pheromone value *τ* min should be chosen with caution as it has a rather strong influence on the algorithm performance. They present an analytical approach to finding this value based on the probability *p*best that an ant constructs the best tour found so far. [1] This is done as follows. First, it is assumed that at each construction step an ant has a constant number *k* of options available. Therefore, the probability that an ant make the right decision. (i.e., the decision that belongs to the sequence of decisions leading to the construction of the best tour found so far) at each of *n* steps is given by.

$$pdec = \sqrt[n-1]{pbest}$$

The analytical formula they suggest for finding *τ min* is:

$$\tau \min = \frac{\tau \max .(1 - pdec)}{k . pdec}$$

The process of pheromone update in *MM*AS is concluded by verifying that all pheromone values are within the imposed limits:

$$\tau (r, s) = \tau \max \text{ if } \tau (r, s) \geq \tau \max$$

$$\tau (r, s) = \tau \min \text{ if } \tau (r, s) \leq \tau \min$$

MAX-MIN Ant System provided a significant improvement over the basic Ant System performance. While the first implementations focused on the TSP it has been later applied to many other combinatorial optimization problems [1].

3.3: ANT COLONY SYSTEM

Another improvement over the original Ant System was Ant Colony System (ACS) introduced by Gambardella and Dorigo. The most interesting contribution of ACS is the introduction of a *local pheromone update* in addition to the pheromone update performed at the end of the construction process (called here *offline* pheromone update).

3.4: PARALLEL ANT COLONY ALGORITHMS

To describe and compare different approaches for parallel ACO algorithms that are suitable for networks of workstations or parallel computers. So far parallel ACO algorithms have been designed mainly for homogenous parallel systems where all processors or workstations are similar or they follow the master-slave paradigm where the central master distributes the work to the other processors. In the last case the master can cope with differences in speed between the workstations/processors by sending the slower processor less work. Unfortunately, not much work has been done so far on non centralized approaches to ACO for heterogeneous parallel systems, although this is very relevant for practical applications. Therefore we concentrate on this aspect in the experimental part of this section where we investigate a non centralized ACO on a network of workstations which offer different computational power for the ACO due to other load. Hardware parallelization and parallelization for processor arrays which consists of many simple processing elements.

In order to evaluate the quality of a parallel ACO algorithm it should be compared with am multistart ACO. This means that several runs of a sequential ACO algorithm are executed on different processors independently. Some additional profit may be gained for the multistart ACO when the sequential ACO algorithms are started with different parameter values. Parallel ACO algorithms can be classified with respect to several criteria that are described in the following. Two of them most basic criteria are:

- Is the algorithm a parallelization of standard ACO or a specially designed parallel ACO algorithm? The aim of a parallelization of standard ACO algorithm is to increase the run time without changing the optimization behavior of the algorithm. In contrast the specially designed parallel ACO algorithms try to change the standard ACO algorithm so that the parallel version works more efficiently. One approach is to do information exchange between the processors not at every iteration. This can also have a positive effect on the optimization behavior because the colonies of the processors can specialize to different regions of the search space.
- Does the algorithm use a centralized approach or a decentralized approach? Typically, in a centralized approach there is one processor that collects the solutions or the pheromone information from all other processors. Then it does the pheromone update and computes the new pheromone matrix which is then send to the other processors. This process on the central processor is often called the master process and other processes are the slave processes. In a decentralized approach every processor has to compute the pheromone update by itself using information that it has received from other processors [8].

3.5: RANK BASED VERSION OF THE ANT SYSTEM

The ant system is a new meta-heuristic for hard combinatorial optimization problems. It is a population-based approach that uses exploitation of positive feedback as well as greedy search. It was first proposed for tackling the well known Traveling Salesman Problem (TSP), but has been also successfully applied to problems such as quadratic assignment, job-shop scheduling, vehicle routing and graph colouring. Introduce a new rank based version of the ant system and present results of a computational study, where we compare the ant system with simulated annealing and a genetic algorithm on several TSP instances. It turns out that our rank based ant system can compete with the other methods in terms of average behavior, and shows even better worst case behavior.

The quality of the solutions produced by the ant system could be improved using so-called elitist ants. The idea of the elitist strategy in the context of the ant system is to give extra emphasis to the best path found so far after every iteration. The concept of ranking can be applied and extended to the ant system as follows: after all m ants have generated a tour, the ants are sorted by tour length ($L1 < L2 < : : : < Lm$), and the contribution of an ant to the trail level update is weighted according to the rank R of the ant. In addition to that, only the w best ants are considered.

Many combinatorial optimization problems still require heuristic search approaches, even though today's computers are very powerful. To solve these problems local search procedures such as simulated annealing or tabu search are widely used. The local search itself is based on the concept of neighborhood, and neighborhood solutions. Defining these neighborhoods is a very difficult task for certain problems.

Here lies one major advantage of the ant system instead of altering existing solutions, in each iteration new solutions are generated. Therefore the ant system does not depend on neighborhoods. Compared to genetic algorithms, another population based method; its advantage is that the finding of appropriate crossover operators - still an unsolved problem – is unnecessary. At the same time, positive feedback within the population, one major advantage of population based methods, can still be exploited in the ant system approach.

The computational study in which we compared the algorithm with simulated annealing and a genetic algorithm lead to the following conclusions. In general the ant system can compete with the other two meta-heuristics. For large problems it seems to outperform the other methods regarding average and especially worst case behavior. Within the ant system algorithms, our new rank based version outperformed the others in any respect [1], [3].

3.6: IMPROVED ANT COLONY OPTIMIZATION

Improved ant colony optimization algorithm for traveling salesman problem,which adopts a new probability selection mechanism by using Held-Karp lower bound to determine the trade-off between the influence of the heuristic information and the pheromone trail. The experiments showed that it can stably generate better solution for the traveling salesman problem than rank-based ant system and max-min ant colony optimization algorithm [24].

HK-ACO PROBABILITY SELECTION MECHANISM

From ACO Probability equation, α and β are two parameters which determine the relative influence of the pheromone trail and the heuristic information. The role of the parameters _ and _ is analyzed in the following. Supposed α = 0, the equation become

$$p_k(r,s) = \frac{[\eta(r,u)]^{\beta}}{\sum [n(r,u]^{\beta}}$$

From the ACO Probability equation, the selection probability increases with the increase of . Furthermore, here η(r, u) = 1/δ(r, u) is an a priori available heuristic value. Therefore, If α = 0, the closest cities are more likely to be selected. This corresponds to a classical stochastic greedy algorithm. On the other hand, suppose β = 0, the equation become

$$p_k(r,s) = \frac{[\tau(r,u)]^{\alpha}}{\sum [\tau(r,u]^{\alpha}}$$

if β = 0, only pheromone amplification is at work. This method will lead to the rapid convergence of a stagnation situation with the corresponding generation of tours which, in general, are strongly suboptimal. Hence a trade-off between the influence of the heuristic information and the pheromone trail exists. In order to achieve the tradeoff between heuristic information and the pheromone, this paper introduces a new approach to adjust the parameters α and β according the equations

$$\alpha = C - di\, j/(Lbsp + \delta(r, s) + Lrsp)$$

$$\beta = B + di\ j/(Lbsp + \delta(r, s) + Lrsp)$$

Where parameter C and B are constant, and set to be 1, 5 correspondingly that are suggested to be advantageous. Lbsp is the length of subtour visited by ant. Lrsp is the length of subtour not visited by ant except city r and s. δi j/(Lbsp + δ(r, s)+ Lrsp) express the proportion of the selected edge to the optimal length.

The Held-Karp lower bound provides a very good problem-specific estimation of optimal tour length for the traveling salesman problem. This measure, which is relatively quick and easy to compute, has enormous practical value when evaluating the quality of near optimal solutions for large problems where the true optima are not known.

This study investigates an improved ant colony optimization to the traveling salesman problem. For this purpose, Held-Karp lower bound is adopted to improve the solution quality. The results showed that the average deviation from optimal solution can be decreased using the proposed algorithm. Additionally, a new probability selection mechanism is utilized to accelerate the algorithm convergence. The feasibility and effectiveness of proposed algorithm are verified through the Traveling Salesman Problem, which is a classical NP-hard combinatorial optimization problem. The experiments demonstrate that the proposed algorithm is an effective algorithm for the traveling salesman problem [24].

3.7: ESTIMATED DISTRIBUTION BASED ANT SYSTEM

The EDAs work as follows initially better solutions are selected from a randomly generated population of solutions. Then, the true probability distribution of the selected set of solutions is estimated and new solutions are generated according to this estimate. The new solutions are then added into the original population, replacing some of the old ones. The process is repeated until the termination criterion is met. The original EDAs are for univariate problems, including PBIL (Population based incremental Learning) , UMDA (Univariate marginal distribution algorithm) etc.

PBIL is described below for its basic probability model will be used in ED-ACO. The PBIL probability model is a probability vector $p(x) = (p(x_1), p(x_2),\ldots, p(x_n))$, where $p(x_i)$ is the probability of generating a 1 in bit position *i*. The PBIL algorithm works as follow: In every generation, randomly generate M solutions [23].

The ED-ACS differs from the previous ACS in three main aspects, The state transition rule also accumulates the probability knowledge about the problem, The pheromone addition in global updating rule is also consider the probability factor, While ants choosing the next city, the probability factor is also evaluated in the random-proportional rule. In the following we explain how the rules are changed [4].

By adding local search procedure optimization heuristics start from a given tour and attempt to reduce its length by exchanging edges chosen according to some heuristic rule until a local optimum is found. The most used and well known tour improvement heuristics by local search in which two, three edges are exchanged respectively. Most of the improved ACO algorithms focus on the local search methods and the pheromone updating rule. However, all of the existed ACO algorithms only use the local search results of each ant to find the best-so-far solution. How to make full use of the known information about the local search results has not been studied before. In the conventional ACO algorithm, each ant will search its neighborhood to find a better solution in each iterative procedure. Unfortunately, only the result provided by one optimal ant is used to form the best-so-far solution and other results are given up. But in fact each ant may find a local optimal solution. In order to make full use of the local search results, and propose a method to merge effective local optimal solutions in the results of all ants to the best-so-far solution. It will accelerate the convergence sharply especially for large combination optimization problem [40]. .

3.8: PROBLEM DEFINITION

In this thesis, to study the use of an ant algorithm operating upon a dynamic problem domain, that is, a domain that changes as a function over time. "*An Improved ACO based on Estimation of Distribution*" as the basis of an implementation of a routing algorithm have the capability of finding short paths in graphs; This Thesis tries to present ED-ACO to avoid the premature convergence and inefficient problems. The Estimation of Distribution Algorithms (EDAs) is a class of evolutionary algorithms that use probabilistic models to estimate a distribution over the search space, and then generate new solutions by sampling from these models. And show an inherent adaptability that could be utilized to solve dynamic problems such as routing in a network.

3.9: OBJECTIVE

On the basis of the previous work of ant colony optimization, in this thesis try to improve the performance of ARS basis algorithm ED-ACO by adding probability factor of previous transition in its state transition rule and the advantages of ED-ACO over another Ant Colony Optimization based algorithms

- To avoid the premature convergence in ACO by using of estimation of distribution
- To improve the Convergence rate become faster as compared to other ACO based algorithm
- To improve the error performance by minimizing error

METHODOLOGY

Up to now, most of the improved ACO algorithms focus on the local search methods and the pheromone updating rule. However, all of the existed ACO algorithms only use the the probability factor of previous iteration to find better optimize path. In this thesis try to make full use of the known information based on estimation of distribution. In the conventional ACO algorithm, each ant will search its neighborhood to find a better solution in each iterative procedure. Unfortunately, only the result provided by one optimal ant is used to form the best-so-far solution and other results are given up. So it reduces the convergence of the solution but by using estimation of distribution it will accelerate the convergence sharply especially for lower iteration values in combination optimization problem.

CHAPTER 4

SIMULATION RESULTS & DISCUSSION

4.1: ANT COLONY OPTIMIZATION BASED ON ESTIMATION OF DISTRIBUTION

ACS possesses the characteristics of positive feedback and distributed computing. Recently, ACS has been proposed to solve different types of combinatorial optimization problems especially to solve the NP-hard combinatorial optimization problems, large-scale complicated combinatorial optimization models, distributed control and clustering analysis problems.

However, premature convergence and low efficiency is its deficiency and sometimes it is inefficient. In order to avoid these weaknesses the performance of ACO can be enhanced using ED-ACO. ED-ACO to avoid the premature convergence and inefficient problems. The Estimation of Distribution Algorithms (EDAs) is a class of evolutionary algorithms that use probabilistic models to estimate a distribution over the search space, and then generate new solutions by sampling from these models.

The probabilistic models are built by the global statistical information from the visited promising solutions. The ED-ACO is a new Ant Colony Optimization algorithm based on Estimation of Distribution. It combines the ideas of ACS and EDAs. The main idea is that it uses the probabilistic model based on estimating the distribution of promising edges to adjust the state transition rule and the global updating rule [23].

In this Section, give some computational results. That shows better convergence, better through put and comparatively less error ACO and MAX-MIN Ant algorithm. The EDAs work as follows initially better solutions are selected from a randomly generated population of solutions. Then, the true probability distribution of the selected set of solutions is estimated and new solutions are generated according to this estimate. The new solutions are then added into the original population, replacing some of the old ones. The process is repeated until the termination criterion is met.

Most of the improved ACO algorithms focus on the local search methods and the pheromone updating rule. However, all of the existed ACO algorithms only use the local search results of each ant to find the best-so-far solution. How to make full use of the known information about the local search results has not been studied before. In the conventional ACO algorithm, each ant will search its neighborhood to find a better solution in each iterative procedure.

4.2: DIFFERENCE BETWEEN ACO &ED-ACO

The ED-ACS differs from the previous ACS in four main aspects:

- The state transition rule also accumulates the probability knowledge about the problem,
- The pheromone addition in global updating rule is also consider the probability factor,
- While ants choosing the next city, the probability factor is also evaluated in the random proportional rule.
- PBIL (Population based incremental Learning) model is used to calculate probability vector after each iteration.

4.3: PBIL PROBABILITY MODEL

The PBIL probability model is a probability vector p(x)=(p(x1), p(x2),…, p(xn)), where p(xi) is the probability of generating a 1 in bit position *i*. The PBIL algorithm works as follow: In every generation, randomly generate M solutions, and then evaluate the probability vector and select N highest solutions to update p(x), where N≤M. The probability update rule is

$$P_{l+1}(x) \backslash= (1-\alpha)\, P_i(x) + \alpha.\, 1/N \sum_{k=1}^{n} xi^k$$

Where $p_l(x)$ is the probability vector of l^{th} generation, x_l^1, $x_l^{\,2}$, …, x_l^N represent the N selected solution vectors, x_l^i is the i^{th} position in the solution vector which the probability vector is moved towards, and α is the learning rate [23].

4.4: FLOW DIAGRAM DESCRIPTION FOR ED-ACO

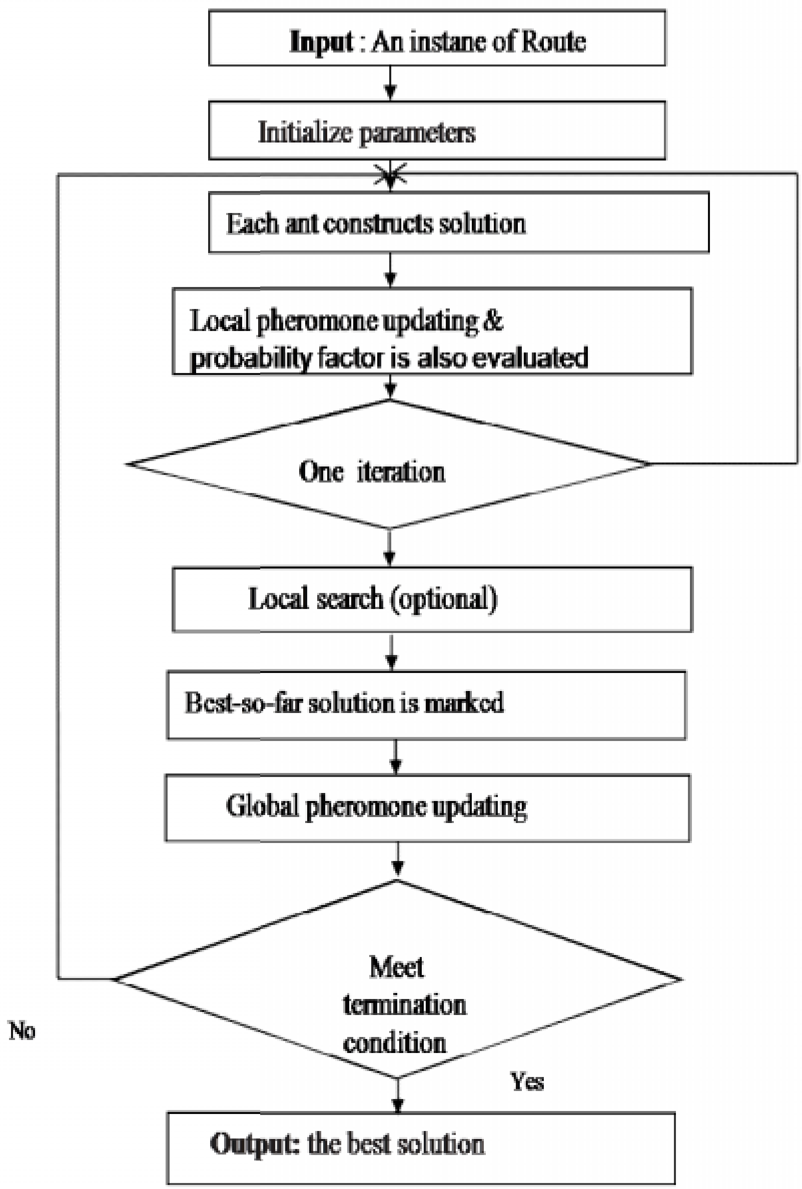

Figure 1: Flow Diagram of ED-ACO

Initialize the probability vectors & other parameters should be initialized in ED-ACO

Loop at this level each loop is called an iteration each ant is positioned on a starting city. And each ant to find the best-so-far solution and applies a local pheromone update rule.

Loop at this level each loop is called a step each ant applies updated state transition rule to incrementally build a solution and best far solution is marked.

Finally updated global updating rule is applied until the termination criteria is met and find the best optimize output.

4.5: EXPERIMENTAL RESULTS BASED ON TSPLIB

The results give an indication of the performance with respect to solution quality of ED-ACO, the results of ED-ACO compared to ACO for some TSPs. All the test problems can be found in TSPLIB

Table 1: Comparison of ED-ACO with ACS [5]

Problem	**Optimum**	**ACO**			**ED-ACO**		
		Best	**Average**	**%Error**	**Best**	**Average**	**%Error**
Eil 51	426	426	428.1	0.00%	426	427.4	0.00%
KrA100	21282	21282	21420.0	0.00%	21282	21296.9	0.00%
D 198	15780	15888	16054.0	0.68%	15780	15857.4	0.00%
Lin 318	42029	42407	42937.6	0.90%	42181	42250.8	0.28%
Pcb 442	50778	51268	51690.0	0.96%	51026	51236.9	0.39%

An easier comparison to other existing improved variants of Ant Colony Optimization (ACO) also give the results obtained with ED-ACO for these problems, see Table 1. Results on ACO and ED-ACO are from and respectively. The number associated with each instance is the number of edges, In all experiments of the following sections the numeric parameters are set to the following values: **β=2, q0=0.9, α=ρ=0.1**, the number of ants used is **m=20**, and was run for 1000 iterations, It is obviously to see the performance of ED-ACO improves considerably over ACO It is obviously to see the performance of ED-ACO improves considerably over ACO [41].

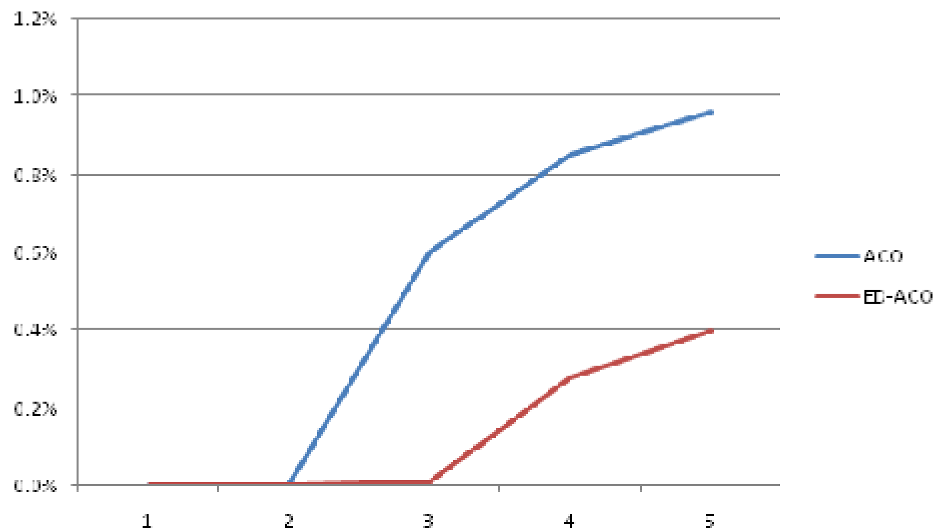

Figure 2: The Error Percentages of the ACO, and ED-ACO [5].

The error percentage graph shows ED-ACO gives better accuracy to find optimize path comparatively to ACO. Where1, 2, 3, 4 & 5 show the 51, 100, 198, 318 & 442 cities respectively.

TABLE 2: Comparison of ED-ACO and ACS with Local Search [5]

Problem	Optimum	ACS +LS			ED-ACO +LS		
		Best	Average	%Error	Best	Average	%Error
Eil 51	426	426	426	0.00%	426	426	0.00%
KrA100	21282	21282	21282	0.00%	21282	21282	0.00%
D 198	15780	15780	15780	0.00%	15780	15781	0.00%
Lin 318	42029	42029	42086	0.00%	42029	42065.6	0.00%
Pcb 442	50778	50795	50914	0.03%	50778	50902.4	0.01%

Local search an improved method to deal with the two main problems involved in ACO, slow speed of convergence and poor ability to search better solution at the end of the search procedure & its improve Error performance of ED-ACO considerably over conventional Ant System based algorithms.

4.6: SIMULATION RESULT BASED ON MATLAB

In all experiments of the following sections the numeric parameters are set to the following values: **β=5, q0=1.0, α=1, ρ=0.65**, the number of ants used is **m=20**, and was run for 100 to 1000 iterations, and discuss the results comes from the MATLAB.

RESULT FOR 100 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 100 iteration the results of ACO & ED-ACO are

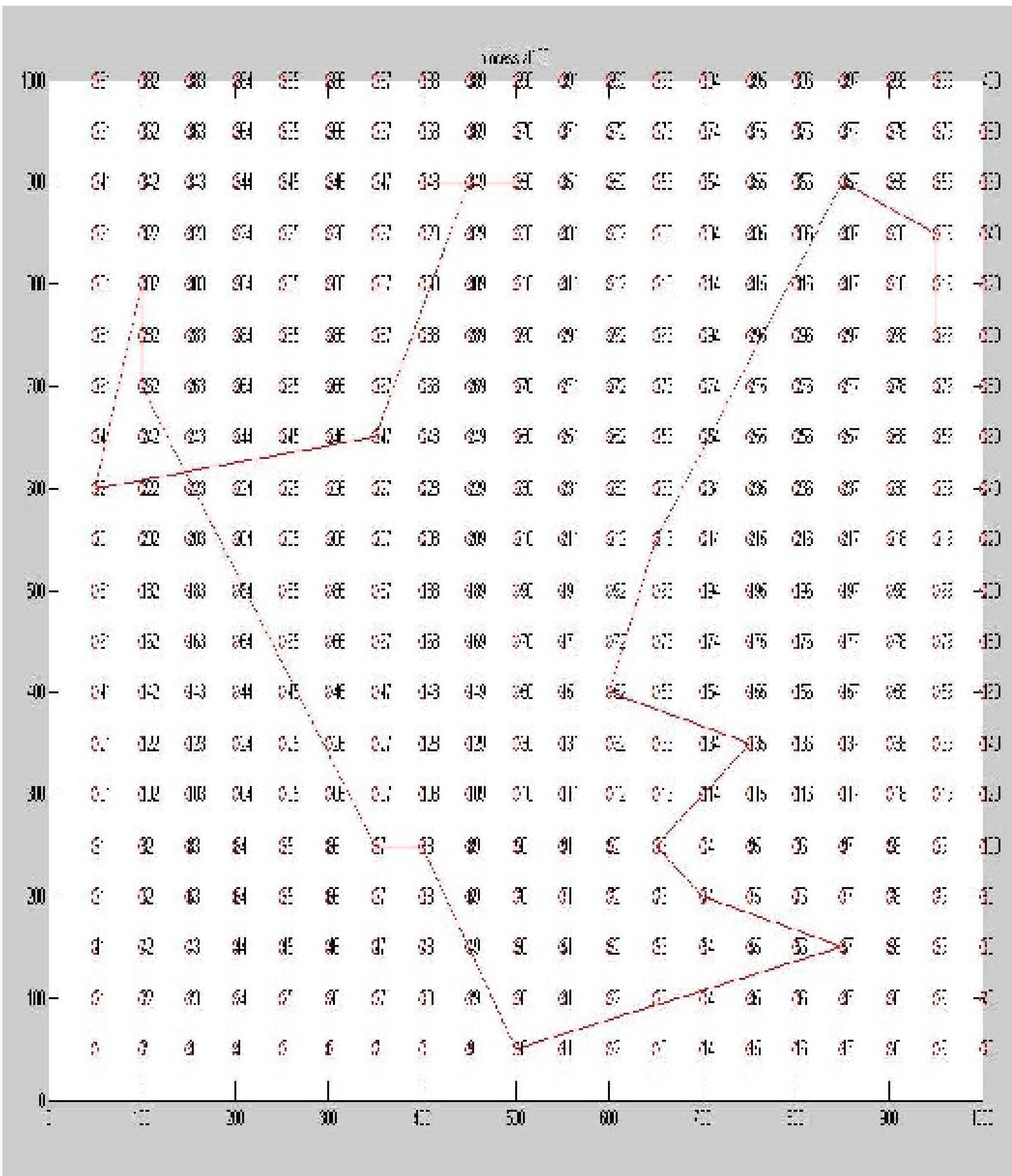

Figure 3: MATLAB Simulation Result of ACO for 100 Iteration

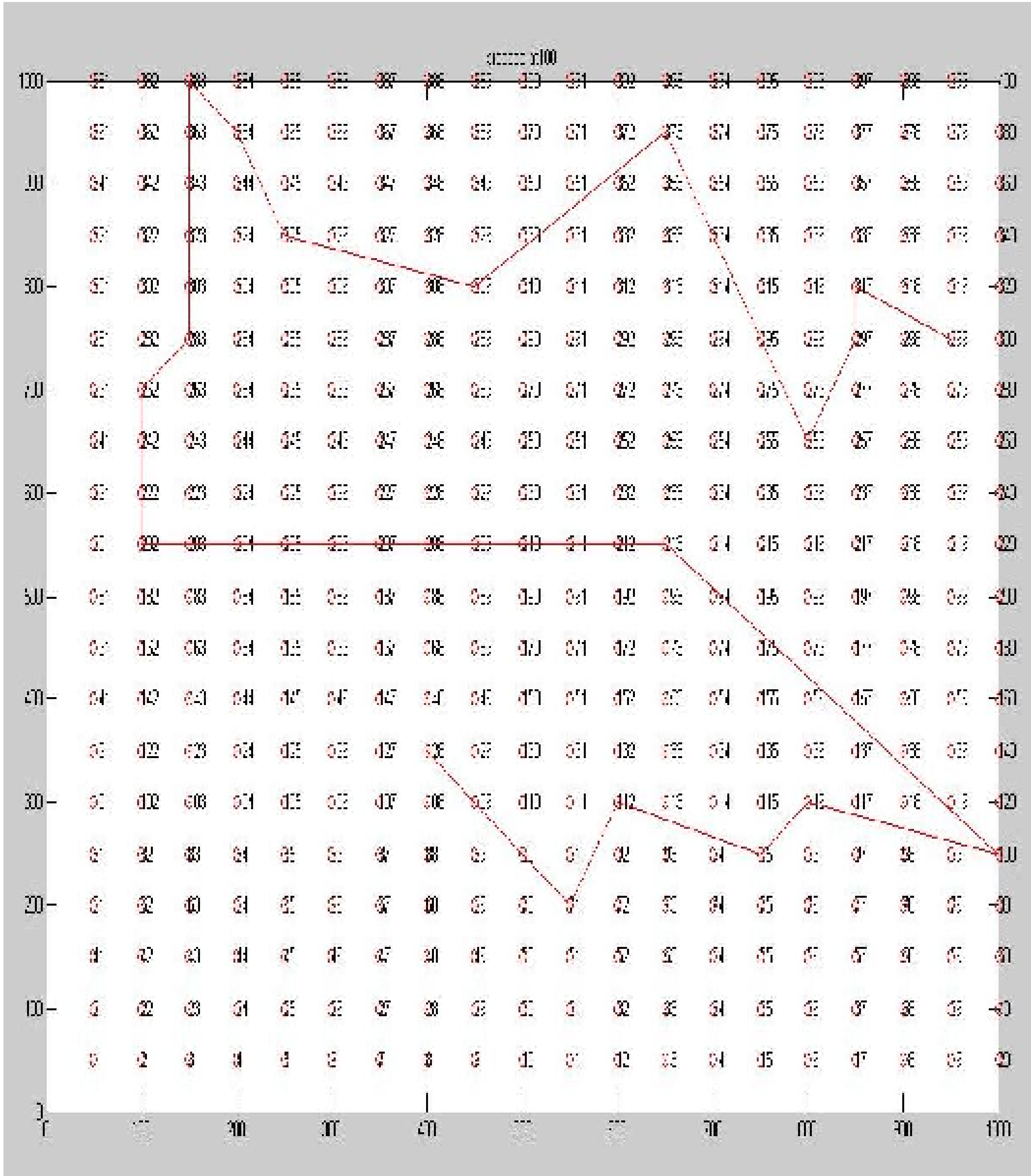

Figure 4: MATLAB Simulation Result of ED-ACO for 100 Iteration

For 100 iterations in ACO optimize path there is loop from the node 221-302-282-262 that would reduce the convergence of the path. Path would be better optimizing if path passes from node 31 to 221 & from node 262 to 247 & there is branches in the graph that also reduce optimization. On the other hand ED-ACO graph has no loop or branch so gives better optimize path compared to ACO.

RESULT FOR 200 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 200 iteration the results of ACO & ED-ACO are

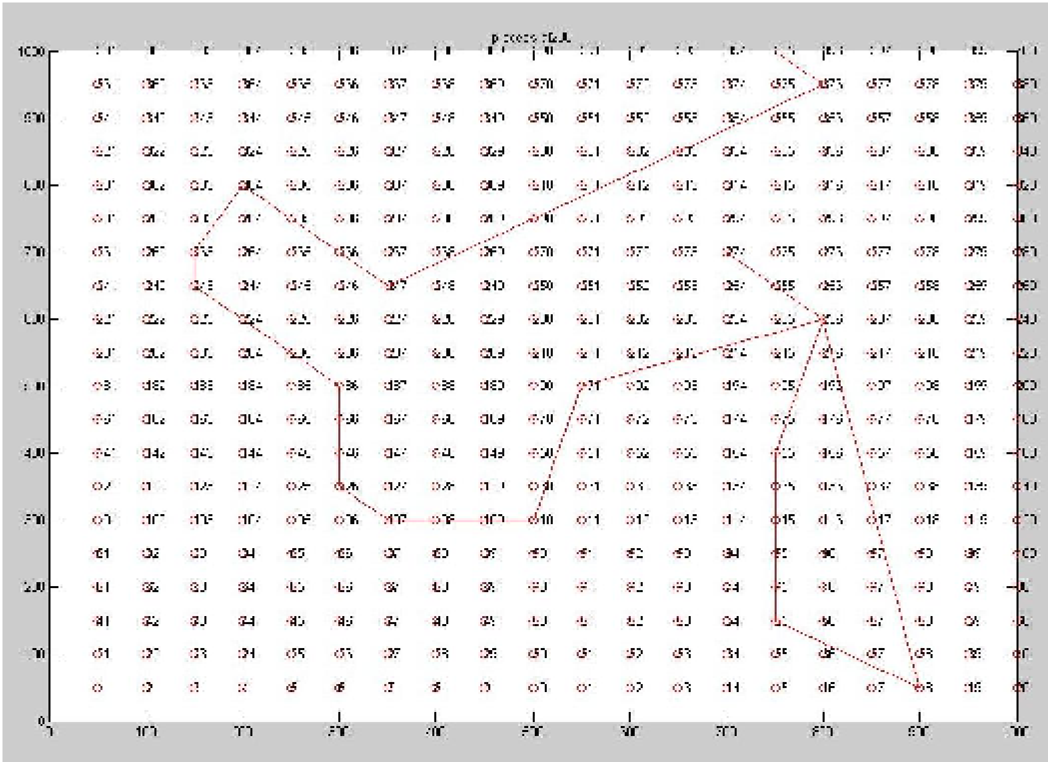

Figure 5: MATLAB Simulation Result of ACO for 200 Iteration

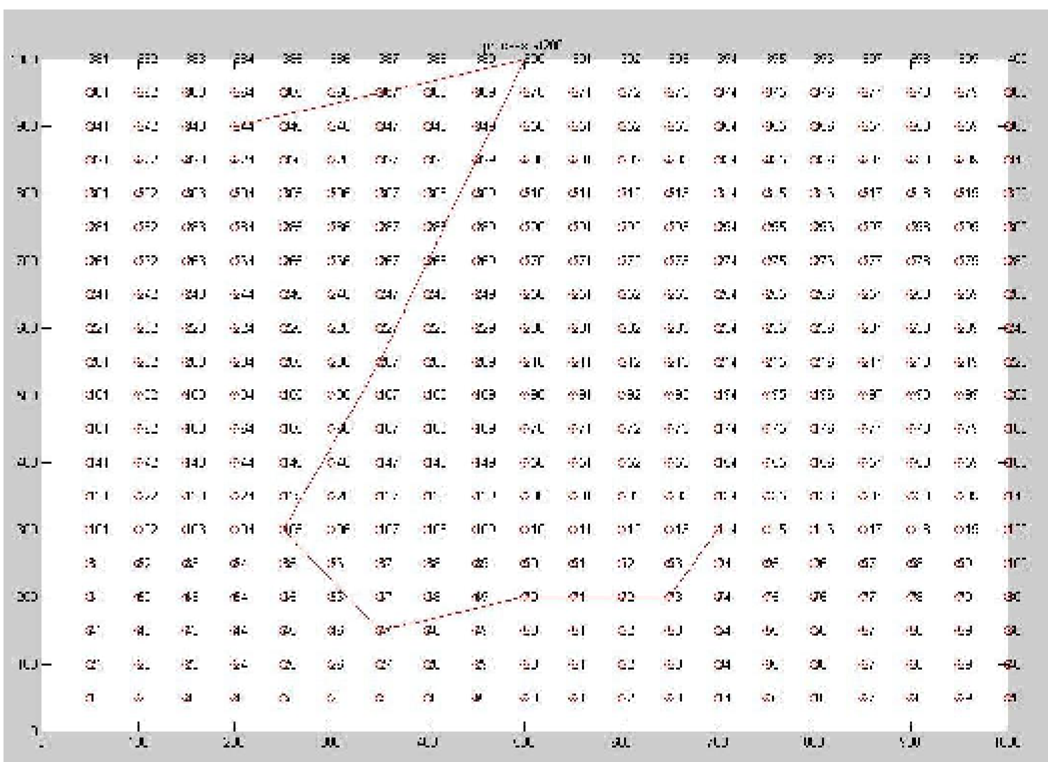

Figure 6: MATLAB Simulation Result of ED-ACO for 200 Iteration

For 200 iterations in ACO graph there is a loop between nodes 8, 55, 236 that would reduce the convergence of the path, the path would be more optimize if it passes from 91 to 55. On the other hand in ED-ACO path there is no loop is formed so ED-ACO finds more optimize path compared to ACO.

RESULT FOR 300 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB results of ACO & ED-ACO for 300 iteration is

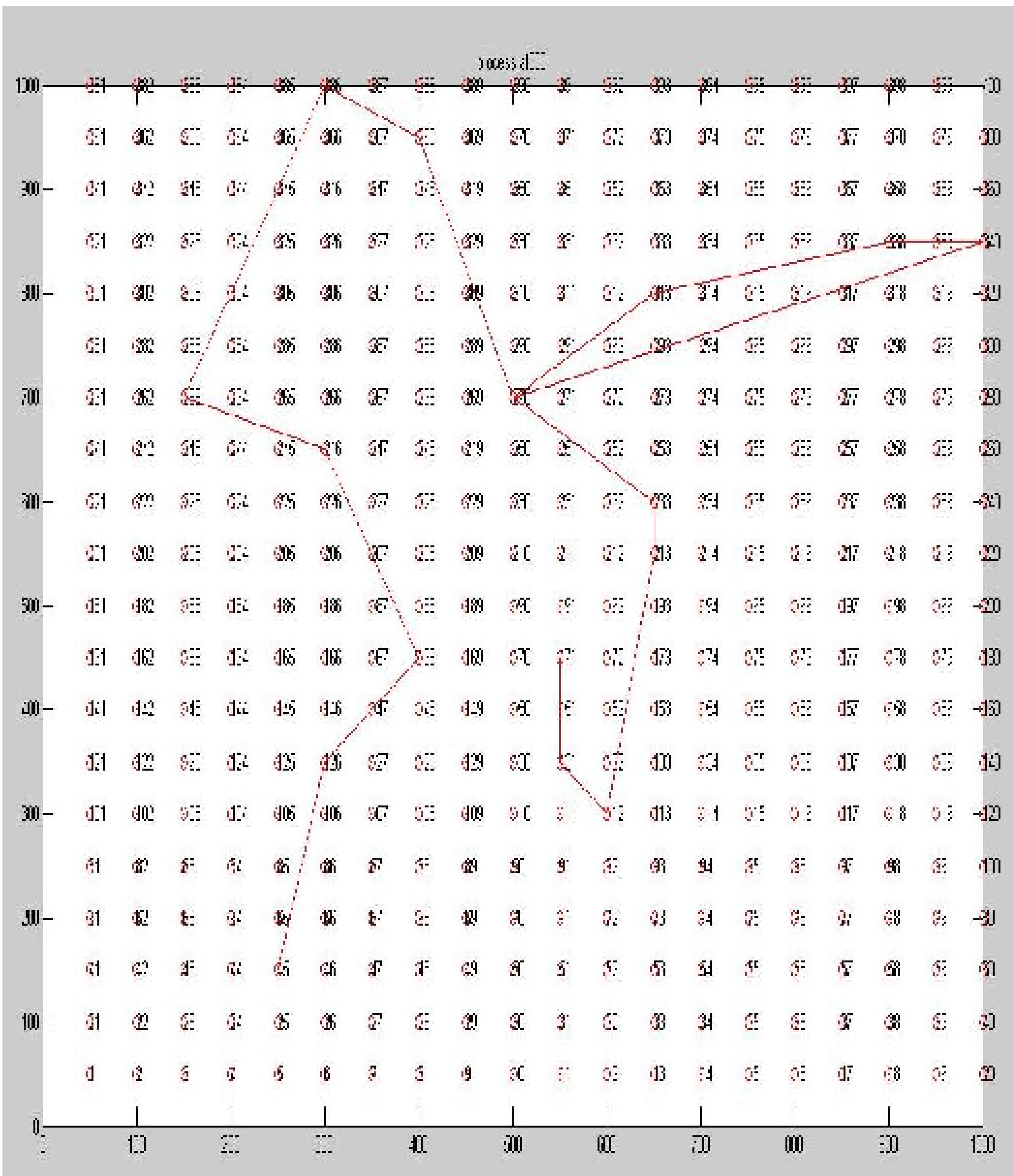

Figure 7: MATLAB Simulation result of ACO for 300 Iteration

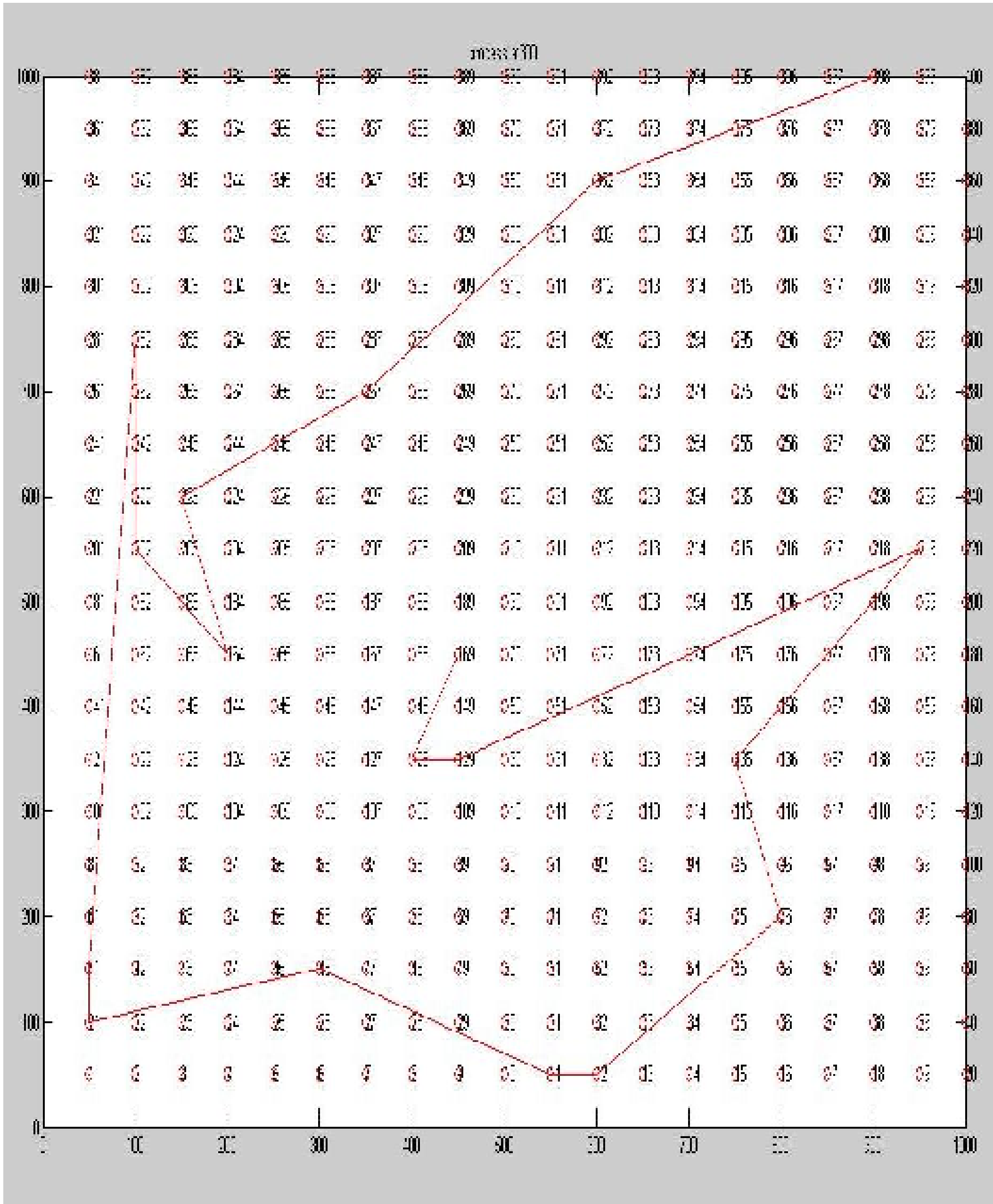

Figure 8: MATLAB Simulation Result of ED-ACO for 300 Iteration

For 300 iterations in ACO graph there is a loop from the node 270-313-338-340 that reduce the convergence path would be more optimize if it passes from node 368 to 313 and ED-ACO find more optimize path comparatively ACO having no loop or branch.

RESULT FOR 400 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 400 iteration the results of ACO & ED-ACO are

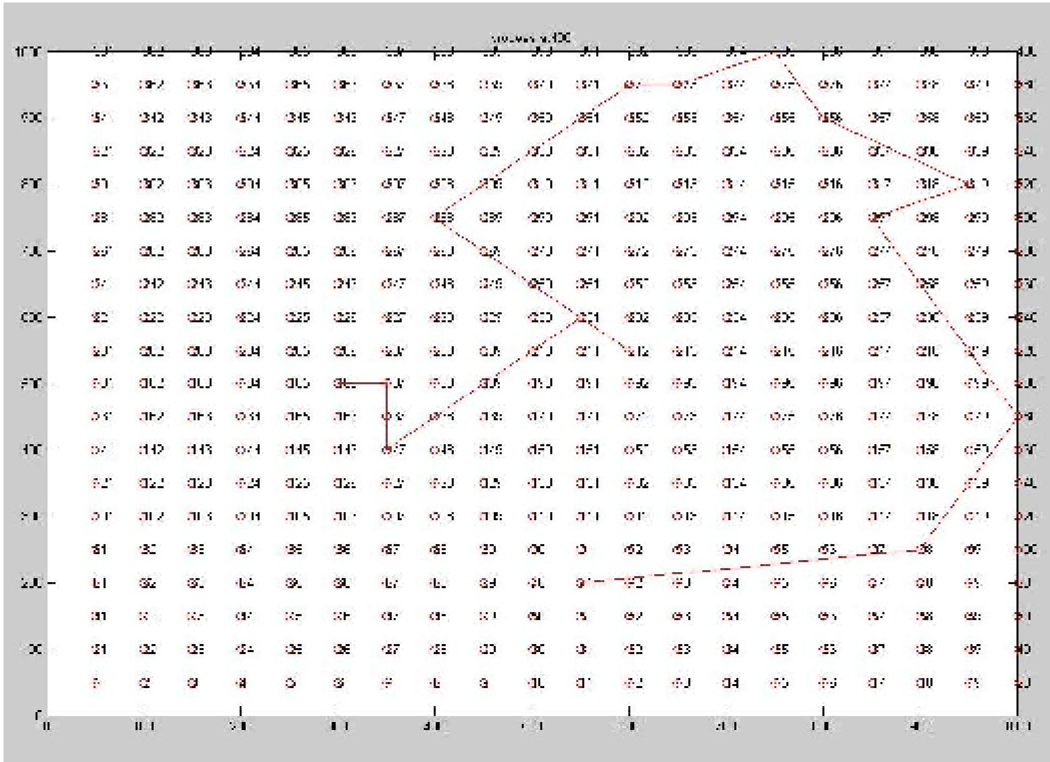

Figure 9: MATLAB Simulation Result of ACO for 400 Iteration

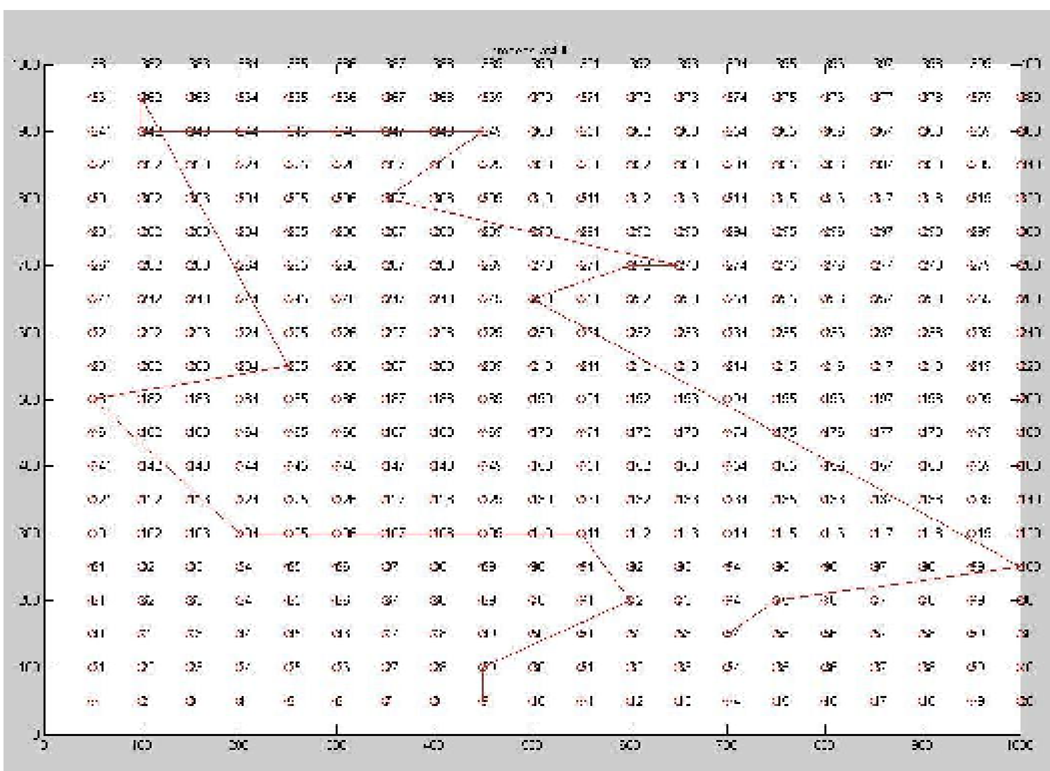

Figure 10: MATLAB Simulation Result of ED-ACO for 400 Iteration

For 400 iterations in ACO graph there is a branch in optimize path from node 212 to 231 so it is not well optimize path. If path passes from 210-212-231 the path would be more optimize. On the other hand there is small loop from node 342 to 362 reduce the convergence but it will gives more optimize path compared to ACO.

RESULT FOR 500 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 500 iteration the results of ACO & ED-ACO are

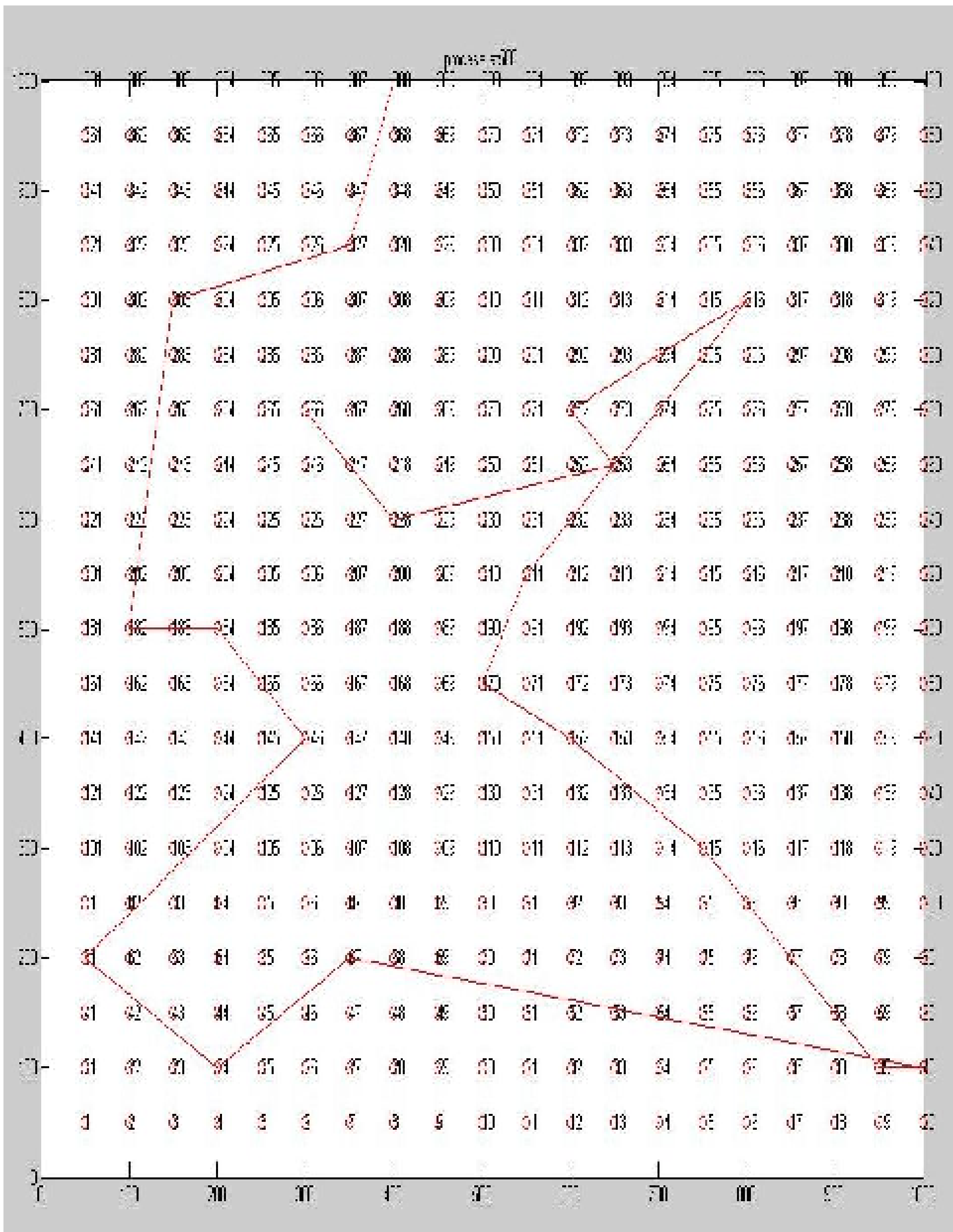

Figure 11: MATLAB Simulation Result of ACO for 500 Iteration

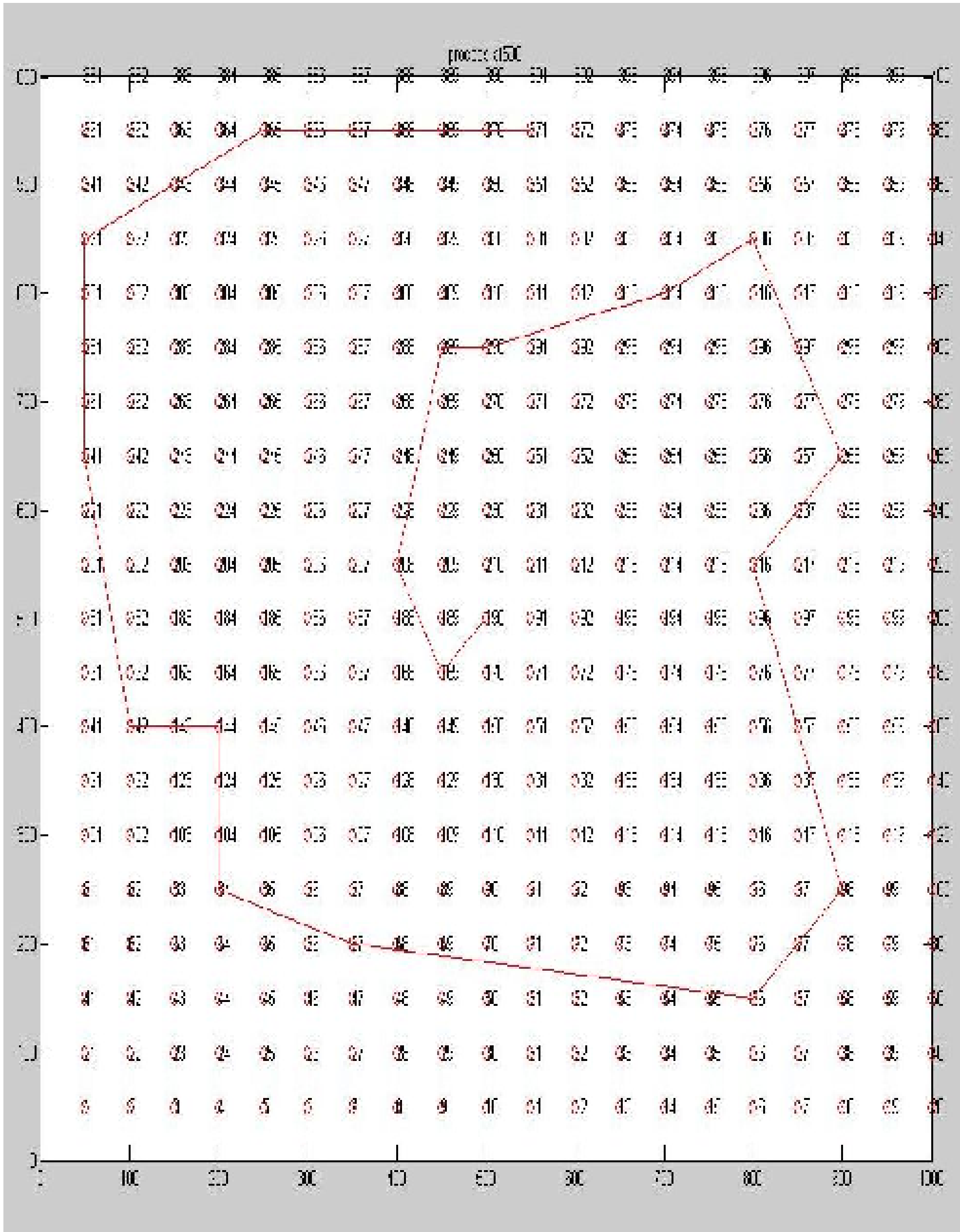

Figure 12: MATLAB Simulation Result of ED-ACO for 500 Iteration

For 500 iterations in ACO graph there is loop at the node 253-272-316 that reduce the convergence of the optimize path if path passes from 272 to 228 the path would be more optimize. On the other hand ED-ACO gives more optimize path compared to ACO.

RESULT FOR 600 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 600 iteration the results of ACO & ED-ACO are

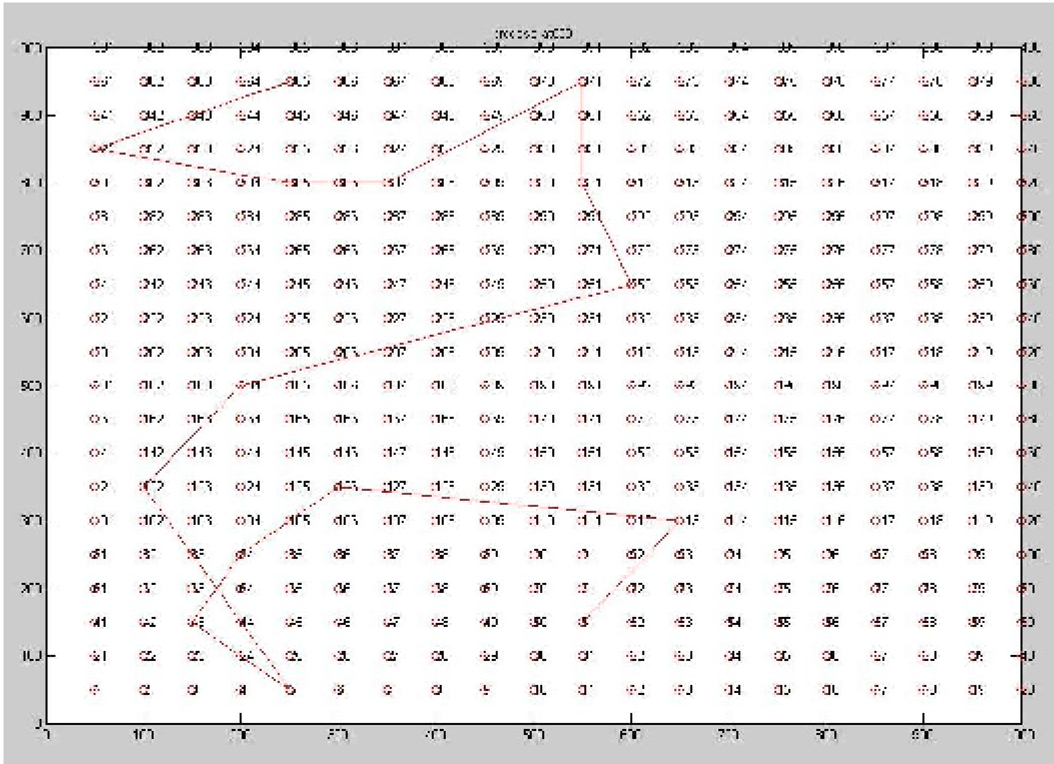

Figure 13 MATLAB Simulation result of ACO for 600 Iteration

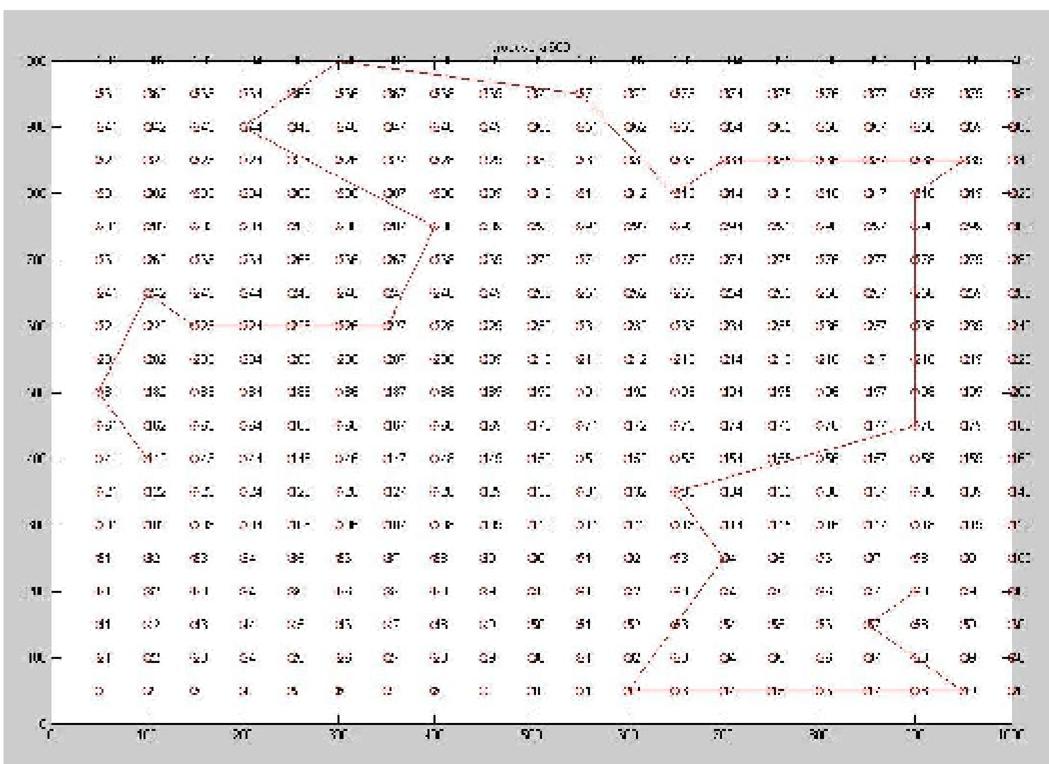

Figure 14: MATLAB Simulation result of ED-ACO for 600 Iteration

For 600 iterations in ACO graph there is loop from node 43-5-44-83 that would reduce optimization. For more optimize path it should be passes from node 83 to 43 & 44 to 84. On the other hand ED-ACO based path have no loop so gives more optimize path compared to conventional ACO algorithm

RESULT FOR 700 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 700 iteration the results of ACO & ED-ACO are

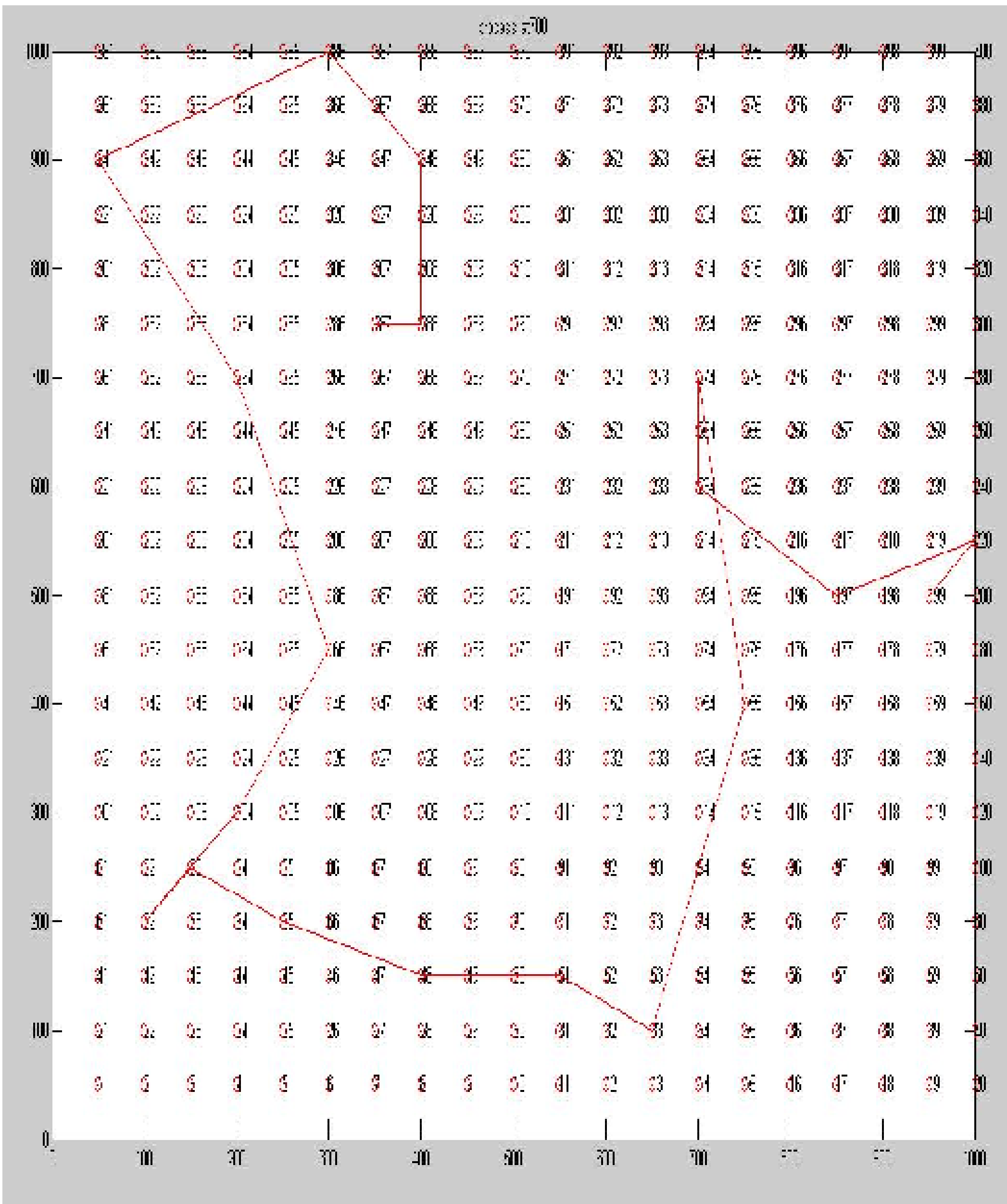

Figure 15: MATLAB Simulation result of ED-ACO for 700 Iteration

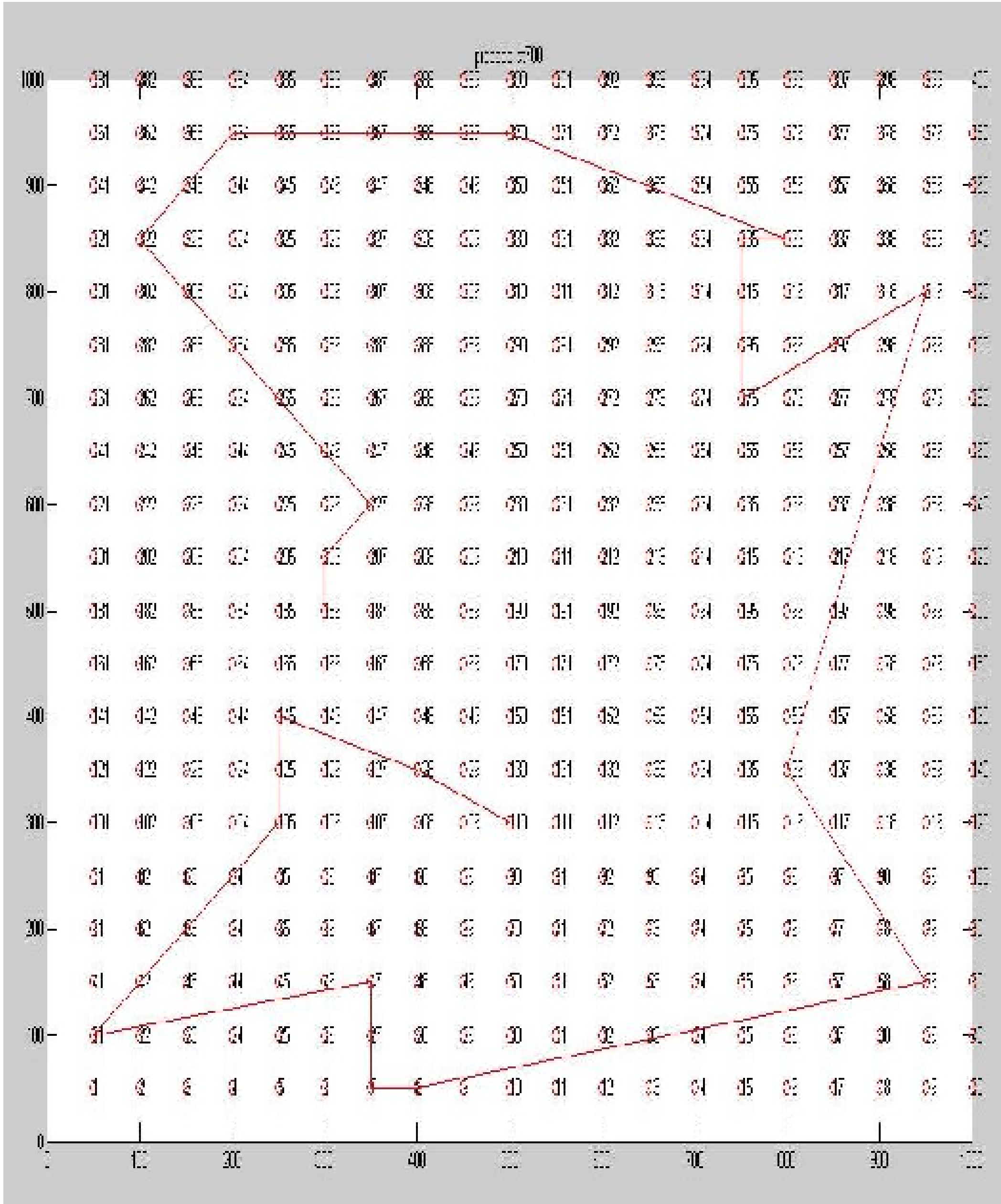

Figure 16: MATLAB Simulation result of ED-ACO for 700 Iteration

For 700 iterations in ACO path there is small loop from the node 234-254-274-155 and branch from node 62 to 83 that reduce the convergence of the path. Path would be more optimize if it passes from node 155 to 234 & 274 to 197. On the other hand ED-ACO find more optimize path comparatively to ACO based path.

RESULT FOR 800 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 800 iteration the results of ACO & ED-ACO are

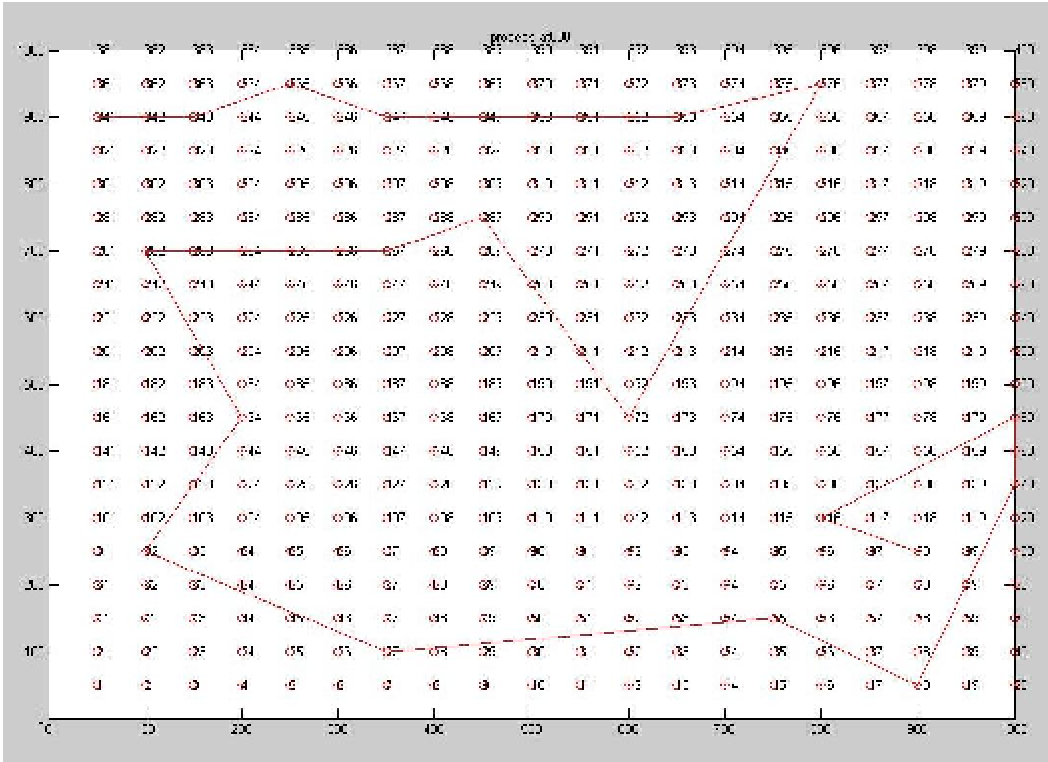

Figure 17: MATLAB Simulation result of ACO for 800 Iteration

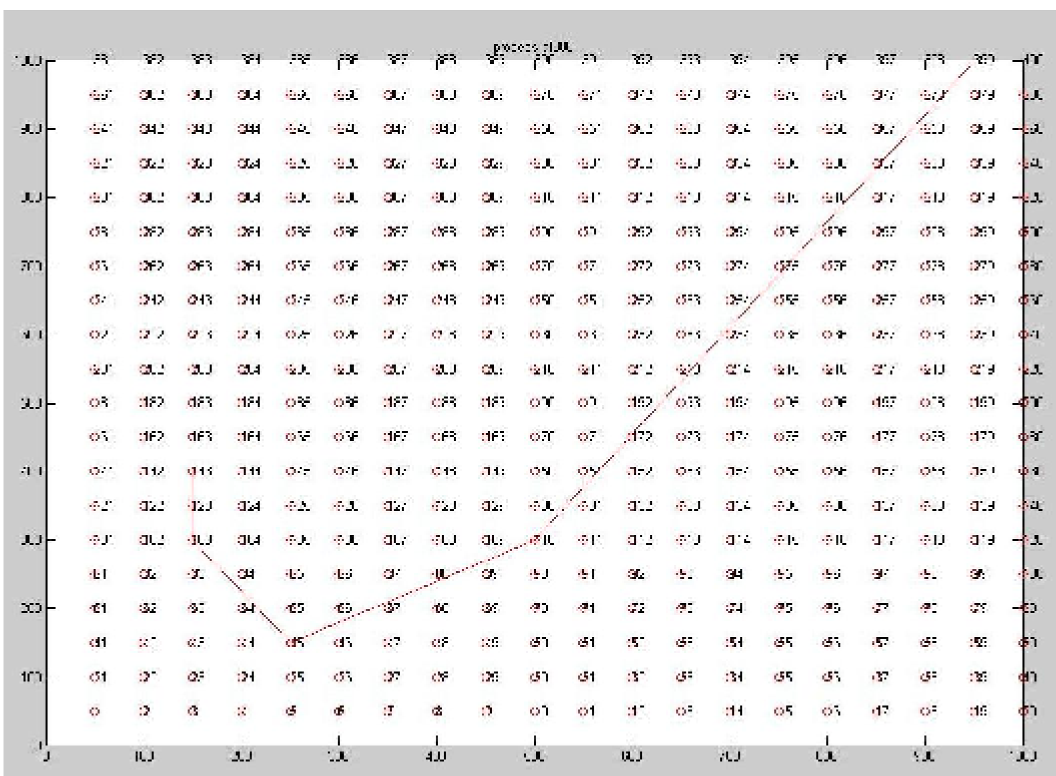

Figure 18: MATLAB Simulation result of ED-ACO for 800 Iteration

For 800 iterations in ACO graph there is no loop in the path better convergence compared to previous results but its optimization can be increased if path passes from 79 to 98 in place of 79 to 140. On the other hand ED-ACO gives optimize path compared to ACO.

RESULT FOR 900 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 900 iteration the results of ACO & ED-ACO are

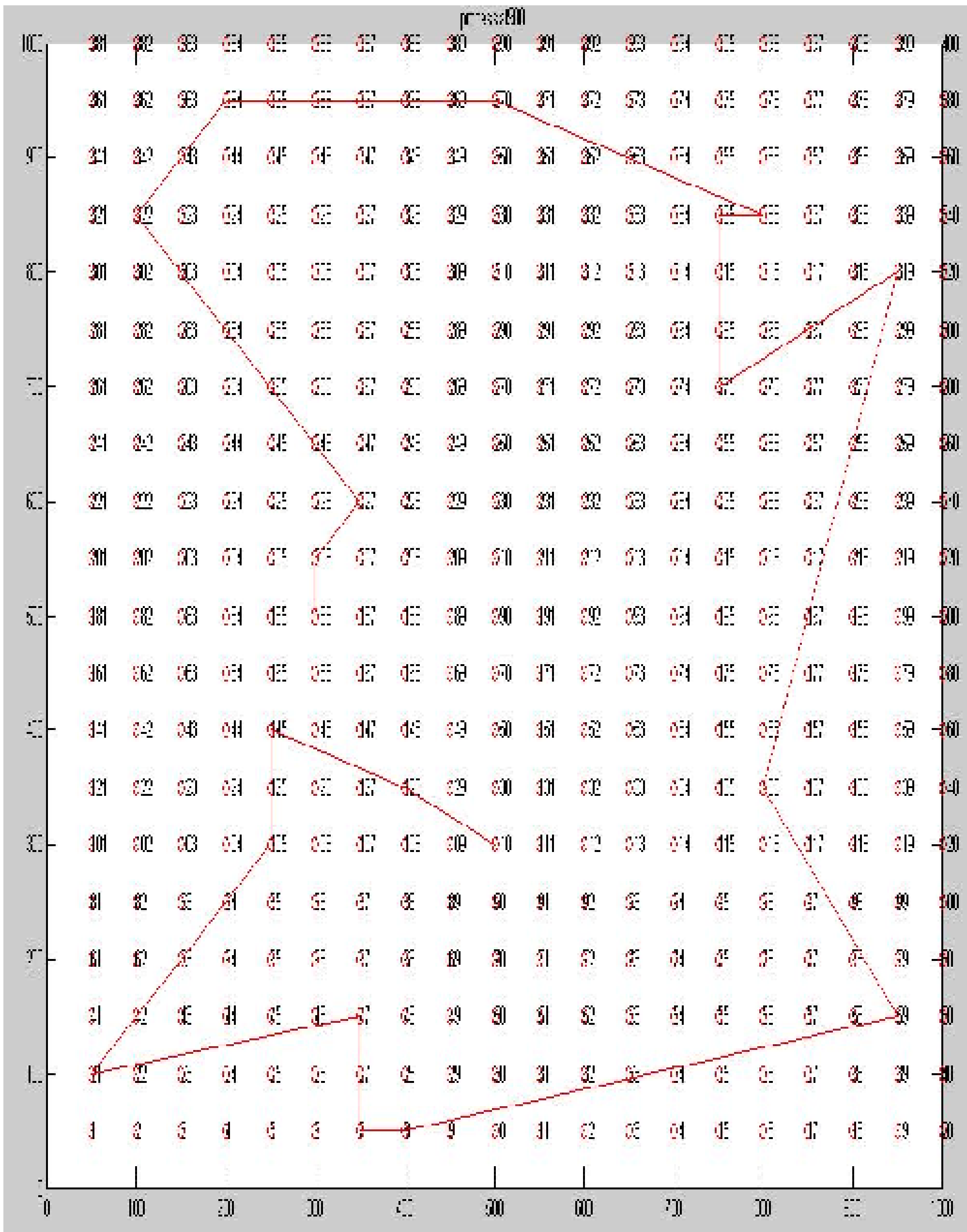

Figure 19: MATLAB Simulation result of ACO for 900 Iteration

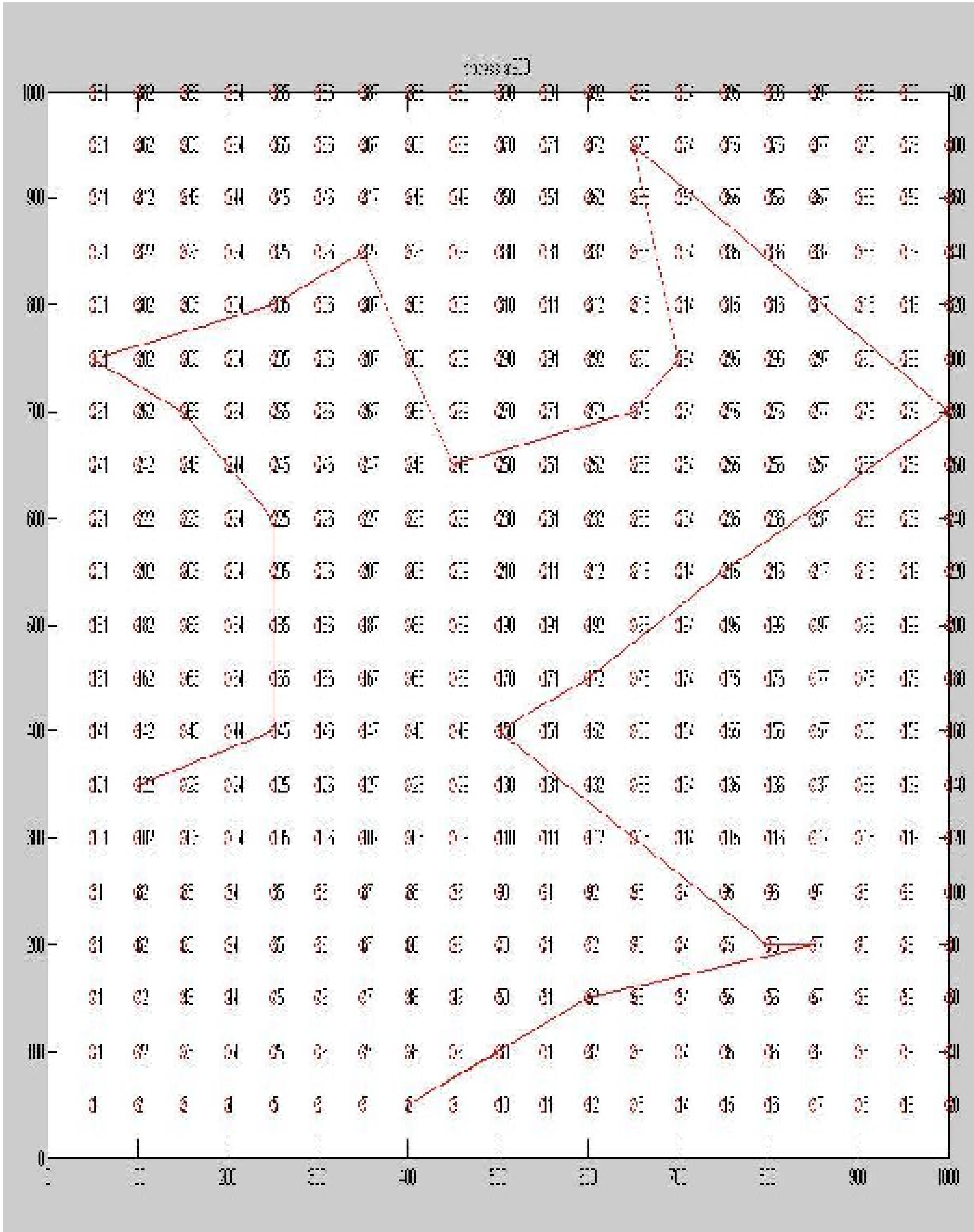

Figure 20: MATLAB Simulation result of ED-ACO for 900 Iteration

For 900 iterations, both ACO & ED-ACO having no loop or branch in it's optimize path both of algorithm gives almost same result with higher number of iteration.

RESULT FOR 1000 ITERATION: All the parameters are taken same for both ACO & ED-ACO, the MATLAB result for 1000 iteration the results of ACO & ED-ACO are

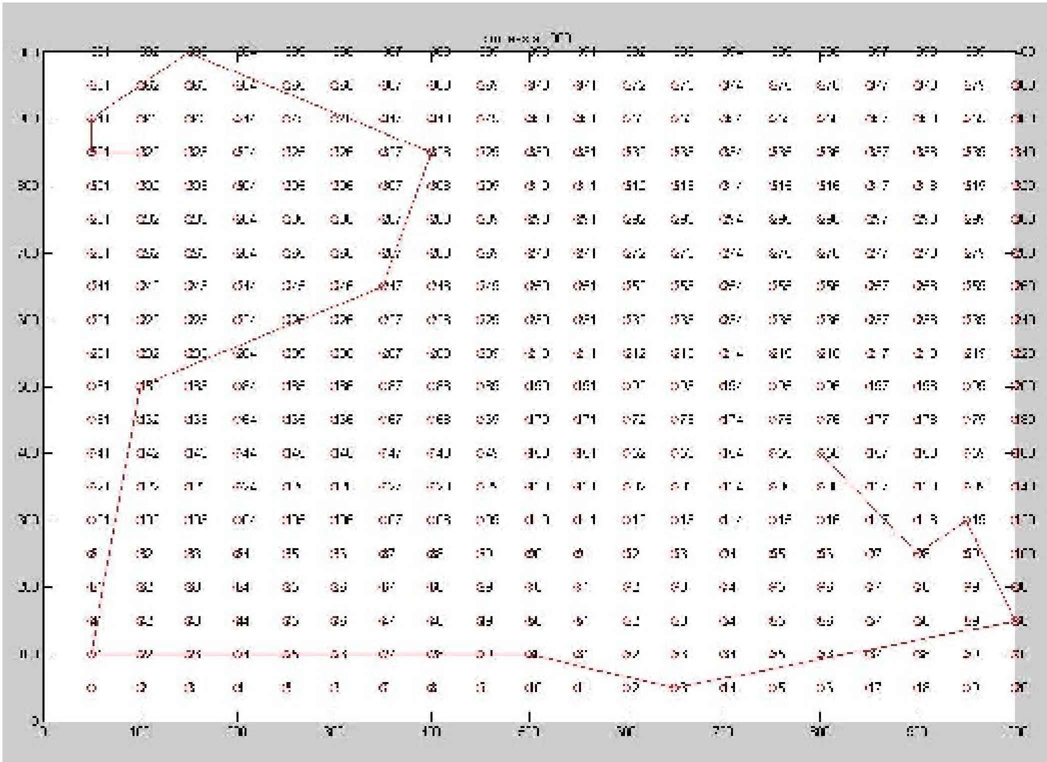

Figure 21: MATLAB Simulation result of ACO for 1000 Iteration

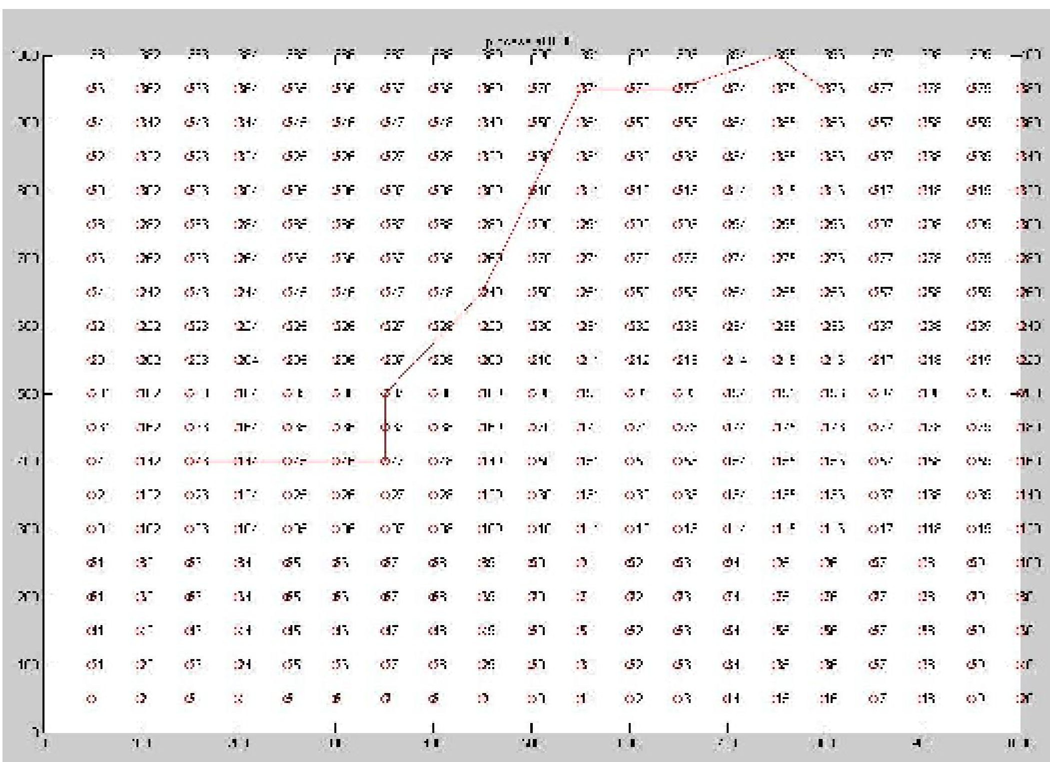

Figure 22: MATLAB Simulation result of ED-ACO for 1000 Iteration

By increasing the no of iteration performance of ACO increases. For 1000 iterations ACO gives better optimization compared to previous ACO results. But for more iteration ED-ACO gives more optimize path as result shown by simulation based MATLAB graphs.

4.7: SUMMARY

The Estimation of Distribution Algorithms (EDAs) is evolutionary algorithms that use probabilistic models to estimate a distribution over the search space, The Estimation of distribution works as initially better solutions are selected from a randomly generated population of solutions. Then, the true probability distribution of the selected set of solutions is estimated and new solutions are generated according to this estimate. The new solution uses by ants while choosing the node, the probability factor are also evaluated in the random proportional rule. And these new solution helps to improve the performance of algorithm. Finally points come from the results based on MATLAB. And find out the following points discuss below

- With lower number of iteration ED-ACO gives maximum improvement because of its state transition rule uses the probability to go next state (inversely proportional to the length go to next state) which works as memory to go next state.
- With higher number of iteration convergence of ED-ACO increases rapidly compared to ACO although at higher no of iteration convergence of ACO also increases and both of algorithms gives almost same result.

Ant Colony System based on Estimation of Distribution. ED-ACO yields significant improvements in performance over ACO and performs at least at the same level. And give better convergence & error performance then other optimization algorithms. And avoid premature convergence.

CHAPTER 5

CONCLUSIONS & FUTURE SCOPE

5.1: CONCLUSION

ED-ACO, an improved Ant Colony System based on Estimation of Distribution. ED-ACO yields significant improvements in performance over ACO and performs at least at the same level of performance as ACO. The differences between ED-ACO and ACO are mainly in the way the probability factor is considered while choosing the next city and updating global information. That increases the convergence.

Estimation of distribution this strategy is used to accelerate the convergence of the ACO algorithm. Estimation of distribution algorithms start from an initial solution and try to find a better solution in a reasonable neighborhood of the current solution. It has testified experimentally that TSPLIB Estimation of distribution will improve the performance of ACO, thereby the solution quality of ACO using Estimation of distribution significantly improved.

The MATLAB results also show that ED-ACO outperforms comparatively ACO. ED-ACO basis ant algorithm graph shows its error percentage is lower for same number of ants & nodes. On the other hand Ant colony Optimization ant system comparatively large error then ED-ACO. And MATLAB graphs shows that ED-ACO gives better convergence & optimizes solution then ACO based algorithms. With lower number of iteration ED-ACO gives maximum improvement because of its state transition rule uses the probability to go next state (inversely proportional to the length go to next state) which works as memory to go next state.

5.2: FUTURE SCOPE OF WORK

ED-ACO avoids premature convergence gives better optimization then other ACO based algorithm. This algorithm would be useful if implementing in simulated point to point network. Because this algorithm using both negative as well as positive feedback. Positive feedback useful to find optimize path while negative feedback avoid congestion.

REFERENCES

[1] Dorigo, M. & Stuetzle T. "The ant colony optimization metheuristic: Algorithms, applications and advances", Technical Report IRIDIA/2000-32, Belgium: University Libre de Bruxelles, 2000.

[2] Bonabeau, E., Dorigo M. & Theraulaz G. Swarm Intelligence: Form Natural to Artificial systems, Oxford University Press, 1999.

[3] Bonabeau, E., Dorigo M. & Theraulaz G. "Ant algorithms and stigmergy" pp. 851-871, 2000.

[4] Bonabeu E. & Theraulaz G. "Swarm smarts", Scientific American, pp. 72-79, March issue, 2000.

[5] Xu Chang, Xu Jun, Chang Huiyou "Anit Colony optimization Based on Estimation of Distribution for the Traveling Salesman Problem", Advanced Computer Control (ICACC), 2nd International Conference, 2010.

[6] Colorni, A., Dorigo M. & Maniezzo V. "Ant system- An Autocatalytic optimizing process", Technical Report IRIDIA/1991-016, Italy: Politecnico di Milano, 1991.

[7] Colorni, A., Dorigo M. & Maniezzo V. "The Ant System: Optimization by a colony of cooperating agents", IEEE Transactions on system, Man and Cybernetics-part B, Vol. 26, No.1, pp. 1-13, 1996.

[8] Cormen, T.H. Leiserson C.E., Rivest R.L. & Stein C. Introduction to Algorithms Second Edition, MIT Press/McGraw-Hill, 2001.

[9] Di Caro, G. & Dorigo M., "Ant colonies for Adaptive Routing in Packet-switched Communications Networks", Parallel Problem Solving from Nature, pp. 673-682, 1998.

[10] Di Caro, G., Dorigo M. & Gambardella L.M. "Ant Algorithms for Discrete Optimization", Artificial Life 5(2) pp. 137-172, 1999.

[11] Di Caro, G. Dorigo M. & Stuetzle T. "Guest editorial -Ant algorithms", pp. v-vii, 2000.

[12] Dorigo, M. & Gambardella L.M. "Ant Colonies for the Traveling Salesman Problem", Technical Report IRIDIA/1996-3, Belgium: University de Bruxelles, 1997.

[13] ZHAO Juan-ping, LIU Jin-gang, GAO Xian-wen and CHEN Ying-qiao "Research of Path Planning for Mobile Reboot based on Improved Ant

Colony Optimization Algorithm" Advanced Computer Control, 2nd International Conference. 2010.

[14] Masaya Yoshikawa' Mashario Fukui2, and Hidekazu Terai2 "A New Pheromome Control Algorithm of Ant Colony Optimization" Nagoya Smart Manufacturing Application, ICSMS International Conference, Dept. of Inf. Eng. Meijo Univ. 2008.

[15] Zahra Hajihashemi & Behrooz Minaei" Improving Noise Clustering Algorithm Using ant Colony Optimization" Sci. & Technol., Tehram Computer Science and Software Engineering, International Conference Comput. Eng. Dept, Iran Univ. 2008.

[16] liusuqin, shuojun, mengligfen, lixingsheng ""making concessions in order to gain advantages" Improved Ant colony Optimization for improving Job Scheduling Problems" Pet. (Huadong), Donying, China Intelligent system global conference, Dept. of Computer. Sci, China Univ, 2008.

[17] LI Hui-min WANG Zhuo-fun "Applying Self-adaptive Ant Colony Optimization for Construction Time-Cost optimization", 2009.

[18] ZHANG Fei Jun and GAO Wei "Meeting Ant Colony Optimization" Wuhan knowledge Acquisition and Modeling Workshop, KAM workshop, IEEE International Symposium, Post-Grad. Coll., Wuhan Polytech Univ. 2008.

[19] Ahmed Yousuf Saber, "Memory-Bounded Ant Colony Optimization with Dynamic Programming and A Local Search for Generator Planning" Nagoya Power Systems, IEEE Transactions, IEEE Transactions Toyota Technol. Inst. 2007.

[20] Ling Chen 12 Haiying Sun1, Shu Wang1 "Solving Continuous Optimization Using Ant Col ony Algorithm" Yangzhou, China Future Information Technology and Management Engineering, FITME, 09. Second International Conference Dept. of Compt. Sci, Yangzhou Univ, 2009.

[21] Yang XIAOI and Xuemei SONG2, Zheng YA03 "Improved Ant Colony Optimization with Particle Swarm Optimization Operator Solving Continuous Optimization Problems" Computational Intelligence and Software Engineering, CiSE, International Conference on Guihua & Shebei Dept., Hebei Polytech. Univ, Tangshan, China, 2003.

[22] Sandeep G. Khode and Rajesh Bhatia "Improving Rerieval Effectiveness using Ant colony Optimization" Advances in Computing, Control, &

Telecommunication Technologies, Act 09. International Conference on Dept. of Comput. Sci & Eng. Thapar University, Patiala, India, 2009.

[23] Zhigang Ren, Zuren Feng, Member IEEE and Zhaojun Zhang "GSP-ANT: An Efficient Ant Colony Optimization Algorithm with Multiple Good solutions for Pheromone Update" Intelligent Computing and Intelligent Systems, ICIS IEEE International conference on State Key Lab. For Manuf. Syst. Eng., Xian Jiatong Univ, Xi'an, China.

[24] Wing Cong, "Ant colony optimization algorithm with multiple good solution" Post-Grad. Coll, Wuhan Polytech. Univ, Wuhan Knowledge Acquisition and Modeling Workshop, KAM Workshop IEEE International Symposium, 2008.

[25] Xiaming You, Xingwai Miao, Sheng Liu "Quantum Computing-based Ant Colony Optimization Algorithm for TSP" Power Electronics and Intelligent Transportation System (PEITS), 2nd International Conference on Coll. of Electron. & Electr. Eng. Sci. Shanghai, China 2008.

[26] Zhaoquan Cai "Multi-direction Searching Ant Colony Optimization for Traveling Salesman Problems" Computational Intelligence and Security, CIS '08. International Conference on Network Center, Huizhou Univ. Huizhou, China, 2008.

[27] Ya-mei XIA, Jun-liang CHEN, Xiang-wu MENG" On the Dynamic Ant colony Algorithm Optimization Based on Multipheromones" TENCON '02. Proceedings. IEEE Region 10 conference on Computers, Communications, Control and Power Engineering Inst. of Mech. Eng, Dalian Univ. of Technol., China, 2008.

[28] C.H. Peng, "An application of parallel ACO algorithm in power network node code multi-scheme optimization", Computer Engineering and Technology (ICCET), 2nd International Conference on Department of Electrical & Electrical Engineering East China Jiaotong University, Nanchang 330013, Jiangxi Province, China, 2010.

[29] Yan-hong Wang and Peng-zhu Pan "A Novel Bi-directional Convergence Ant Colony Optimization with SA for Job-Shop Scheduling" Computational Intelligence and Software Engineering, CISE, International Conference on Syst. Engg. Ins, Shenyang Univ. of Technol., Shenyang, China, 2008.

[30] Pierree Delisle, Michael Krajecki and Marc Gravel "Multi-colony Parallel Ant Colony optimization on SMP and Multi-core computers" Nature & Biologically Inspired Computing, NABIC World Congress on Dept. de Math. et Inf, Univ. de Reims Champagne-Ardenne, Reims, France, 2009.

[31] Kwang Mong Sim and Weng Hong Sun "Multiple Ant-Colony Optimization for Network Routing" Cyber worlds, Proceedings. First International Symposium on Dept. of Inf. Eng., Chinese Univ. of Hong Kong, Shatin, China 2002.

[32] SHU Yunxing 1,2, GUO Junen 1, Ge Bo1 "An Ant Colony Optimization Algorithm Based on the Nearest Neighbor Node Choosing Rules and the crossover Operator" Computer Science and Software Engineering, International Conference on Lyoyang Inst. of Sci & Technol., Luoyang, 2008.

[33] Dong-Sheng Xu, Xiao-Yan Ai, Li-Ning Xing "An improved Ant Colony Optimization for Flexible Job shop Scheduling Problems", Computational Sciences and Optimization, CSO, International Join Conference on Dept. of Inf. Technol., Yulin Univ. Yulin, China, 2009.

[34] Bifan Li1, Lipo Wang1,2, and Wu Song3 "Ant colony Optimization for the traveling Salesman Problem Based on Ants with Memory" 34 Natural Computation, ICNC, Fourth International Conference on Coll. of Inf. Eng, Xiangtan Univ, Xiantan, 2008.

[35] Patrick Albert and Mathias Kleiner "Ant Colony Optimization for Configration", Tools with Artificial Intelligence, ICTAI, 20th IEEE International Conference ILOGS S.A., Gentilly, 2008.

[36] Songsak Chusanapiputt, Dulyatat Nualhong, Sujate Jantarang, and Sukumvit Phoomvuthisam "Relatively Pheromone Updating Stratgy in Ant Colony Optimization for Constrained Unit Commitment Problem" Power System Technology, International Conference on Dept. of Electr. Power Eng. Mahanakorn Univ. of Technology, Bangkok, 2006.

[37] Fangjin Zhu, Xiangxu Meng, Hua Wang, Member, IEEE, Shanwen Yi "An Ant Colony Optimization Algorithm To Aggregated Multicast Using The Idea of Bin Packing", Information, Computing and Telecommunication, IEEE Youth conference on Sch. of Comput, Sci. & Technol. Shandong Univ, Jinan, China, 2009.

[38] In'es Alaya "Ant Colony optimization for Multi-objective optimization Problems" Tools with Artificial Intelligence, 19th IEEE International Sympsium, IEEE Comput. Sci. Dept. Univ. of Pretoria, Tshwane, 2008.

[39] Salman A. Khan and Adnries P. Engelbrecht" Ant colony Optimization Algorithm for Topology Design of Distributed Local Area Networks" Swan intelligence Symposium, IEEE Comput. Sci. Dept. Univ. of Pretoria, Tshwane, 2008.

[40] http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB95/.495

[41] Jianxin Han School of Statistics and Mathematics, Shandong Finance University "An Improved Ant Colony Optimization Algorithm Based on Dynamic Control of Solution Construction and Mergence of Local Search Solutions" Natural Computation, Fourth International Conference on Sch. of Stat. & Math, Shandong Finance Univ., Jinan 2007.