

Comparative Study of Swarm Intelligence Behavior to Solve Optimization Problems

Dr. Hasanen S. Abdullah*

Received on: 26/5/2011

Accepted on: 8/9/2011

Abstract

The optimization problems usually need specific techniques to solve, therefore many approaches and methods were proposed to solve such problems, but there are many difficulties (limitations) still faced the problem solvers such as how to reach the solution (or solutions) with high performance and efficiency or with more accuracy results or with suitable behavior.

Thus the artificial intelligence tools are considered the best tools that can be used to solve the optimization problems, because the AI tools must decide two important aims: the problem reduction and the guarantee of solutions which lead to less the effect of the difficulties (limitations) and give more suitable criteria in performance, efficiency, and behavior. The swarm intelligent techniques are considered the most modern AI techniques which contains many approaches that are used to solve optimization problems with high performance and efficiency and suitable behavior

In this paper a specific study is made to the behavior of the swarm intelligence techniques and evaluates its performance to solve various problems, then there is a presentation to a scientific comparative section in which many approaches is presented that used different swarm intelligence techniques such as Ant Colony Optimization (ACO), Bees Algorithm (BA), and Particle Swarm Optimization (PSO) to solve various optimization problems and then make a comparison among them in term of behavior and performance. Finally we reach to scientific discussion and conclusions that distinguish among the presented approaches to prove that the swarm intelligence techniques success in solving practical, important, and applicable problems with high performance, efficiency, and special behavior.

دراسة مقارنة لسلوك الذكاء بالتحشد لحل مشاكل تحقيق الأمثلية

الخلاصة

إن مشاكل تحقيق الأمثلية عادة ماتحتاج الى تقنيات خاصة لحلها، لذلك هناك عدة طرق قد اقترحت لحل هكذا مشاكل، لكن تبقى هناك عدة صعوبات او معوقات تواجه الأشخاص الذين يفهمون بحل هذه المشاكل من هذه المعوقات هي كيفية الوصول الى الحل (الحلول) بأداء وكفاءة عاليين او بنتائج اكثر دقة او بسلوك مناسب.

لذا تعتبر أدوات الذكاء الاصطناعي من افضل الأدوات التي من الممكن استخدامها لحل مشاكل تحقيق الأمثلية بسبب ان أدوات الذكاء الاصطناعي عند استخدامها يجب ان تحقق هدفين مهمين هما: تقليص حجم المشكلة وضمان الوصول للحل والتي تؤدي الى تقليل تأثير المعوقات وتعطي معايير اكثر دقة في الأداء والكفاءة والسلوك. أن تقنيات الذكاء بالتحشد تعتبر من احدث تقنيات الذكاء الاصطناعي والتي تتضمن عدة طرق والتي من الممكن استخدامها لحل مشاكل تحقيق الأمثلية بأداء وكفاءة عاليين وسلوك مناسب.

في هذا البحث تم إنجاز دراسة خاصة لسلوك تقنيات الذكاء بالتحشد لحل مشاكل متنوعة و تقييم أدائها، ومن ثم هناك تقديم لمقارنة علمية من خلال عرض عدة اساليب التي تم من خلالها تم

إستخدام تقنيات مختلفة من تقنيات الذكاء بالتحشد مثل تقنية تحقيق أمثلية مستعمرة النمل، خوارزمية النحل، وتقنية تحقيق أمثلية حشد الجزيئة لحل مشاكل تحقيق امثلية متنوعة ومن ثم عمل مقارنة بينها من خلال السلوك والاداء لكل منها. في نهاية البحث لقد توصلنا الى مناقشة ونتائج علمية الاساس والتي من خلالها تم التمييز بين الأساليب المقدمة وذلك لأثبات إن تقنيات الذكاء بالتحشد قد نجحت في حل مشاكل عملية، مهمة، وقابلة للتطبيق بأداء وكفاءة عاليين المستوى وسلوك متميز.

Introduction

The evolutionary computation provides approaches for doing global and local search simultaneously. The main evolutionary paradigms are: evolution strategies, evolutionary programming, swarm intelligence, genetic algorithms, and genetic programming. They conduct probabilistic population-based search which is a powerful tool for broad exploration and local exploitation of the model space. The population-based strategy is an advantage over other global search algorithms such as simulated annealing and tabu search, which works with only one hypothesis at a time, and over algorithms for local search that perform only narrow examination of the search space. Their stochastic character is an advantage over the heuristic AI and machine learning algorithms that also search with one hypothesis. [1]

All biological systems result from an evolutionary process. The sophistication, robustness, and adaptability of biological systems represent a powerful motivation for replicating the mechanisms of natural evolution in the attempt to generate software and hardware systems with characteristics comparable to those of biological systems.

Even in the context of problem solving, artificial evolution is sometimes criticized by engineers

Because it contains elements of randomness and lacks formal proofs of convergence proper of other, model-based, optimization techniques. Indeed, artificial evolution is better employed in situations where conventional optimization techniques cannot find a satisfactory solution. [2]

Particle Swarm Optimization

Particle Swarm Optimization (PSO) was firstly introduced by Kennedy and Eberhart in 1995. It is a simple evolutionary algorithm which differs from other evolutionary algorithms in which it is motivated the simulation of social behavior. PSO has shown good performance in finding good solutions to optimization problems, and turned out to be another powerful tool besides other evolutionary algorithms such as genetic algorithms. Like other evolutionary algorithms, PSO is also a population-based search algorithm and starts with an initial population of randomly generated solutions called particles. Each particle in PSO has a position and a velocity. PSO remembers both the best position found by all particles and the best positions found by each particle in the search process. For a search problem in an n-dimensional space, a potential solution is represented by a particle that adjusts its position and velocity according to Eqs. (1) and (2):

$$V_i^{(t+1)} = W * V_i^{(t)} + C_1 * \text{rand}_1() * (pbest_i - X_i^{(t)}) + C_2 * \text{rand}_2() * (gbest - X_i^{(t)}) \quad \dots(1)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)} \quad \dots(2)$$

Where: X_i and V_i are the position and velocity of particle i , $pbest_i$ and $gbest$ are previous best particle for the i th particle and the global best particle found by all particles so far respectively, and w is an inertia factor, and $\text{rand}_1()$ and $\text{rand}_2()$ are two random numbers independently generated within the range of $[0,1]$, and c_1 and c_2 are two learning factors which control the influence of the social and cognitive components. One problem found in the standard PSO is that it could easily fall into local optima in many optimization problems. Some research has been done to tackle this problem. The standard PSO was inspired by the social and cognitive behavior of swarm. According to equation (1), particles are largely influenced by its previous best particles and the global best particle. Once the best particle has no change in a local optimum, all the rest particles will quickly converge to the position of the best particle. So enhancing the mutated probability of particles will lead to some changes of the best particle. These changes could help the best particle escape from local optima.[3] In this paper, a comparison has been conducted among several swarm intelligence approaches to solve some optimization problems by using various swarm approaches that

depend on the parallel processing environment to do their jobs such as ant colony optimization, bees algorithm and particle swarm intelligence.

Ant Colony Optimization [4]

Ant colony optimization (ACO) was inspired by the behavior of ant colonies searching for the shortest route to a food source. Although ants are almost blind and thus a single ant has limited capabilities, ants in colonies exhibit *foraging behavior* to find the shortest distance between their nest and the food.

When an ant encounters an intersection (e.g. an obstacle) that has two possible routes it locates the shortest possible route via *pheromone* laid by previous ants, as ants following a path will deposit some pheromone on that path. Ants detect the concentration of pheromone on each path and tend to choose, by probability, the path with the higher intensity of pheromone.

The inspiration derived from the foraging behavior of real ants, after the undertaking of extensive experimentation, has been transformed into a strategy that can be used to solve complex optimization problems. The ant agents used in the ACO metaheuristic are generally known as *artificial ants* in contrast to their natural counterparts, artificial ants are given the following additional abilities to solve more complex real-world optimization problems:

1. **Visibility:** artificial ants are given visibility when they encounter an intersection. With this given

artificial intelligence, ants are able to judge the distances of different paths at the intersection so that shorter paths are more favorable.

2. Memory: real ants are assumed have no memory and make decisions based only on the pheromone intensities of decision paths. In contrast, memory is given to artificial ants for storing records of previously visited paths.

3. Higher pheromone evaporation rate: pheromone evaporation reduces the intensity of all pheromone trails by an amount directly proportional to the intensity. Consequently, it can be seen as a means of encouraging exploration of unvisited paths by reducing the overall gap between pheromone trail intensities. Pheromone evaporation also takes place during the foraging process of real ants, but at a much slower rate. In contrast, higher evaporation rates are suggested for artificial ants, especially when solving more complex problems.

4. Daemon actions: daemon actions are those actions that cannot be performed by an individual ant, for example, additional pheromone being laid on the shortest route found so far, and is optional in the ACO metaheuristic.

Bees Algorithm [5]

A colony of honey bees can be seen as a diffuse creature which can extend itself over long distances in multiple directions in order to exploit a large number of food sources at the same time. In principle, flower patches with plentiful amounts of nectar or pollen

that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees. The foraging process begins in a colony by scout bees being sent to search for promising flower patches. Scout bees search randomly from one patch to another. During the harvesting season, a colony continues its exploration keeping a percentage of the population as scout bees. When they return to the hive, those scout bees that found a patch which is rated above a certain threshold (measured as a combination of some constituents, such as sugar content) deposit their nectar or pollen and go to the "dance floor" to perform a dance known as the "waggle dance". This dance is essential for colony communication, and contains three vital pieces of information regarding a flower patch: the direction in which it will be found, its distance from the hive and its quality rating. This information helps the bees to find the flower patches precisely, without using guides or maps. Each individual's knowledge of the outside environment is gleaned solely from the waggle dance. This dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it. After waggle dancing on the dance floor, the dancer bee (i.e. the scout bee) goes back to the flower patch with follower bees that were waiting inside the hive. The number of follower bees assigned to a

patch depends on the overall quality of the patch. This allows the colony to gather food quickly and efficiently. While harvesting from a patch, the bees monitor its food level. This is necessary to decide upon the next waggle dance when they return to the hive. If the patch is still good enough as a food source, then it will be advertised in the waggle dance and more bees will be recruited to that source.

As mentioned above, the Bees Algorithm is an optimization algorithm inspired by the natural foraging behavior of honey bees. The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of sites selected out of n visited sites (m), number of elite sites out of m selected sites (e), number of bees recruited for the best e sites (n_{ep}), number of bees recruited for the other ($m-e$) selected sites (n_{sp}), initial size of patches (n_{gh}) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of the sites visited by the scout bees are evaluated in step 2. The valuation of fitness would depend on the optimization problem, but in general 'fitness' is taken as the value of the objective function being optimized.

In step 4, bees that have the highest fitnesses are designated as "selected bees" and sites visited by them are chosen for neighborhood search. Then, in steps 5 and 6, the algorithm conducts searches in the

neighborhood of the selected sites, assigning more bees to search near to the best e sites. The bees can be chosen directly according to the fitnesses associated with the sites they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighborhood of the best e sites which represent more promising solutions are made more detailed by recruiting more bees to follow selected bees than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm. However, in step 6, for each patch only the bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This constraint is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population – representatives from each selected patch and other scout bees assigned to conduct random searches. The Bees Algorithm was adopted in this work as it had proved to have a more robust performance than other intelligent optimization methods for a range of complex problems.

Swarm Intelligence Approaches to Solve Problems

In this section there is a presentation to some approaches that used the swarm intelligence techniques to solve various problems.

Swarm Intelligence for Routing in Mobile Ad Hoc Networks (Sirmahn) [6]

Mobile Ad Hoc Networks are communication networks built up of a collection of mobile devices which can communicate through wireless connections. Routing is the task of directing data packets from a source node to a given destination. This task is particularly hard in Mobile Ad Hoc Networks: due to the mobility of the network elements and the lack of central control, routing algorithms should be robust and adaptive and work in a decentralized and self organizing way. In this paper, we describe an algorithm which draws inspiration from Swarm Intelligence to obtain these characteristics. More specifically, we borrow ideas from ant colonies and from the Ant Colony Optimization (ACO) framework. In an extensive set of simulation tests, we compare our routing algorithm with a state-of-the-art algorithm, and show that it gets better performance over a wide range of different scenarios and for a number of different evaluation measures. In particular, we show that it scales better with the number of nodes in the network.

This approach has described AntHocNet, a routing algorithm for MANETs which was inspired by

ideas from Swarm Intelligence, and more specifically by the framework of ACO. The algorithm combines reactive and proactive behavior to deal with the specific challenges of MANETs in an efficient way. Routing information is learned through Monte Carlo sampling of paths using repeatedly and concurrently generated ant agents, as is common in ACO routing algorithms. This learning process is supported by a secondary process, called pheromone diffusion. Pheromone diffusion provides an alternative way to learn pheromone information, using an information bootstrapping mechanism rather than Monte Carlo sampling. It is used to help update pheromone on existing paths and to provide guidance to ants in search of new paths. We have evaluated the algorithm in an extensive set of simulation tests. The tests were carried out in a commercial simulator and comparisons were each time made with AODV (Ad-hoc On-demand Distance Vector), a de facto standard in the community. AntHocNet was shown to outperform AODV over the wide range of tested scenarios in terms of delivery ratio, average end-to-end delay and average jitter, while generating a comparable amount of control overhead. An important observation was that the advantage of AntHocNet over AODV grew for larger networks, especially in terms of overhead, suggesting that AntHocNet is more scalable than AODV.

Application of the Bees Algorithm to the Training of Learning Vector Quantization Networks for Control Chart Pattern Recognition (ABTLVNCPR) [7]

Control charts are employed in manufacturing industry for statistical process control (SPC). It is possible to detect incipient problems and prevent a process from going out of control by identifying the type of patterns displayed by the control charts. Various

techniques have been applied to this control chart pattern recognition task. This approach presents the use of Learning Vector Quantization (LVQ) networks for recognizing patterns in control charts. The LVQ networks were trained, not by applying standard training algorithms, but by employing a new optimization algorithm developed by the authors. The algorithm, called the Bees Algorithm, is inspired by the food foraging behavior of honey bees. The approach first describes the Bees Algorithm and explains how the algorithm is employed to train LVQ networks. It then discusses the recognition of control chart patterns by LVQ networks optimized using the Bees Algorithm.

The algorithm requires a number of parameters to be set, namely: number of scout bees (n), number of elite bees (e), number of patches selected out of n visited points (m), number of bees recruited for patches visited by "elite bees" (nep), number of bees recruited for the other ($m-e$) selected patches (nsp), size of patches (ngh)

and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitnesses of points visited by the scout bees are evaluated in step 2.

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) H/Forming new population.
4. Select elite bees and elite sites for neighborhood search.
5. Select other sites for neighborhood search.
6. Recruit bees around selected sites and evaluate fitnesses.
7. Select the fittest bee from each site.
8. Assign remaining bees to search randomly and evaluate their fitnesses.
9. End While.

In step 4, bees that have the highest fitnesses are chosen as "elite bees". Then, in steps 5 - 7, the algorithm conducts searches in the neighborhood of the elite bees and of the other selected bees. The latter can be chosen directly according to the fitnesses associated with the points they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighborhood of the elite bees which represent more promising solutions are made more detailed by recruiting more bees to follow elite bees than other selected bees. Also within scouting, differential recruitment is one of the key operations of the Bees Algorithm. Both scouting and differential recruitment are used in nature. However, in step 7, for each

site only one bee with the highest fitness will be selected to form the next bee population. In nature, there is no such a restriction. This restriction is introduced here to reduce the number of points to be explored. In step 8, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population - representatives from each selected patch and other scout bees assigned to conduct random searches.

This approach has described the use of the Bees Algorithm to train the LVQ neural network for control chart pattern recognition. Despite the high dimensionality of the problem - each bee represented 2160 parameters that had to be determined - the algorithm still succeeded to train more accurate classifiers than that produced by the standard LVQ training algorithm. Future work will be directed at investigating classifiers based on other neural network paradigms and comparing their performances.

3.3 A Swarm Intelligence Approach to Distributed Mobile Surveillance (SIDMS) [8]

In the post-9/11 world, new and improved surveillance and information-gathering technologies have become a high-priority problem to be solved. Surveillance systems are often needed in areas too hostile or dangerous for a direct human presence. The field of robotics is

being looked to for an autonomous mobile surveillance system. One major problem is the control and coordination of multiple cooperating robots. Researchers have looked to the distributed control strategies found in nature in the form of social insects as an inspiration for new control schemes. Swarm intelligence research centers around the interactions of such systems and how they can be applied to contemporary problems. In this thesis, a surveillance system of mobile autonomous robots based on the principles of swarm intelligence is presented.

The specific goal of this thesis work is to investigate the tolerance of swarm intelligent systems to errors in communication. The objectives of this research are:

- To demonstrate a communication weakness in many of the current swarm intelligent systems and propose a viable solution to safeguard against the problem and improve performance.
- To develop a surveillance system of multiple autonomous robots using a swarm intelligence approach as a test bed for the experiments.
- To create reusable code structures, hardware, and research methodologies.

It was shown in both simulation and physical hardware that current system utilizing swarm intelligent control schemes have a major weakness involving false communication. If a malicious force infiltrates the communication of a robot malfunctions, system that

blindly follows external communication could enter a terminal deadlock state. To remedy this situation, a communication threshold scheme was implemented and tested. This threshold prevented the robots from responding to external communication that did not meet certain levels of criteria. Also, if following a signal did not lead to anything of interest, the robot entered an isolation state for a period of time where it did not respond to any external communication. Finally, a surveillance system using swarm intelligence control scheme was developed as a proof of concept and a test bed for the previously mentioned experiments.

Result Calculations [8]

To gauge the effectiveness of the system, the positional certainty of the target's location is measured. To do this, the target's position is recorded at 1 second intervals during operation. Each time the target is spotted, its perceived location is recorded along with the time of the spotting. At every time step, the distance between the perceived and actual position is calculated using the following formula:

$$Distance = \sqrt{((X_{actual} - X_{perceived})^2 + (Y_{actual} - Y_{perceived})^2)}$$

Because the angle between the perceived location and actual location is still unknown, this distance represents the radius of a circle of location certainty. The area of the certainty region is calculated with the following formula:

$$Certainty Area = \Pi * Distance^2$$

To obtain the final measure of effectiveness, the certainty measure is compared to the total area as follows:

$$Percentage\ of\ area\ known\ to\ contain\ target = \frac{Certainty\ Area}{Total\ Area} * 100$$

This measure allows for comparisons across multiple area configurations as the area size is accounted for.

Simulation Test Procedure [8]

The following procedures were used to make the simulate results more accurate and easily repeatable.

- Time- Because the running time of StarLogo is dependent on the processor making the computations and the current load on that processor, real time was not a suitable way to measure test duration. Each test allows the target to make 450 movements; the same amount of work is done regardless of the system configuration running the tests. This movement limit corresponds to the 300 second time limit in the physical tests because at their current speeds an ER1 can travel approximately 1.5 times its length in a second.
- Collisions- To model the disadvantages of robot-to-robot collisions, delays are added to the robot movement. When two or more robots occupy the same patch, they wait for .1 seconds and then move past one another.
- Area - A 12 patch by 12 patch area was used to mimic the physical hardware. Each table used as a barrier

in the hardware testing was approximately four robot lengths wide. The simulation scale is one patch to one robot length.

Backtracking Ant Algorithm (BTAA) [9]

In this work, the backtracked ant system is a viable new approach to stochastic combinatorial optimization. The main characteristics of the search algorithm; are positive feedback, distributed computations, and the use of backtracked heuristic search, and it helps to find acceptable solutions in the early stages of the search process.

This algorithm is applied on famous problem, the network routing, and report simulation results. Parameter selection and the early setups of the algorithm are discussed.

The backtracked ant system (AS) effectiveness can be applied to other types and techniques of AS that are [AS, Elitist EAS, Rank-Based AS, Max-Min AS, and ACO], then a comparison is made between them depending on (time execution minimum cost, optimization solution). Finally, it is proved that backtracked ACO technique has the best good performance and retrieves appropriate results its memory.

In this work (approach) facts can be reached from the application (network routing) results, the distribution time is increased with increasing the number of ants, therefore, the system will decrease the time by decreasing the number of ants by using backtrack ant systems and blocking some ants, the operations of the update pheromone are computed in the first,

second, third,..., iterations, but the update of pheromone will stop in the last iterations in the system work; therefore the execution time will decrease. Also by applying the algorithm to the AntNet (network routing), and from the experimental application results of the ant systems it is discovered that the ACO method is the best one among the others, gives good results and performance, like minimum cost, minimum execution time, minimum tour time, and nearest iteration. In the AS methods the deposited pheromone and the updated pheromone are applied to all paths between the nodes in the graph, while in the ACO method these two types of pheromones are applied only to the paths of the best tour solution.

Comparison of Methods

In this section there is a comparison study among the three previous methods to use variety approaches of the swarm intelligence to achieve a practical various application in which selected approach such as bees algorithm (BA), ant colony optimization (ACO), and the particle swarm optimization (PSO).

Three methods are chosen to be a cases study in this paper to be compared in their performance, behavior or accuracy through implement a proposed approach of the swarm intelligence to solve different problems and then we discuss these approaches to give a practical studying for each problem that had been solved, the first method (case 1 SIRMAHN) "*Swarm Intelligence for*

Routing in Mobile Ad Hoc Network", the second method (case 2 ABTLVNCPR) "*Application of the Bees Algorithm to the Training of Learning Vector Quantization Networks for Control Chart Pattern Recognition*", the third method (case 3 SIDMS) "*A Swarm Intelligence Approach to Distributed Mobile Surveillance*".

Case 1 Sirmahn

We investigate AntHocNet's performance for varying level of mobility and node density, for increasing network sizes, and for different data traffic patterns, we also show results reporting the performance of the algorithm at a smaller time scale followed by the evolution of the end-to-end delay over a course of simulation session of the test run. For different network sizes, we increase the number of nodes (up to 800) and the area size together, keeping the node density constant. [6] The Ant Colony Optimization algorithm is used in this method so to make the mobile Ad Hoc Network adaptive routing with various states parameters. Number of simulation results is used to evaluate the method by monitoring its algorithm behavior with some properties of the network scenario by varying parameters (pause time & number of nodes) in each state of the system behavior, the results of the AntHocNet method are compared with the results of the AODV method in term of delay and delivery ratio, jitter and routing overhead, and delay, delivery and overhead for increase node density.

The method with various states in which the net job with different level of mobility that gives the average delay and delivery ratio for different levels of mobility by comparing the average end-to-end packet and the packet delivery ratio with the pause time (in second) between the AntHocNet and the AODV, from this, we can say that AntHocNet is better than the AODV in averaging packet delay with high ratio by reducing this average in more amount of packet delay measured with pause time in second; as well as the packet delivery ratio is reduced in an acceptable amount when using the AntHocNet that the AODV also measured with pause time in second, see figure (3.1.1).

Also the AntHocNet various states with different level of mobility in term of average delay jitter and routing overhead network comparing with the AODV method for the same purpose of net job, from this, the average jitter delay net with the using of the ACO with the routing Ad Hoc Mobile network is more preferred than used the AODV method in which the averaging amount is reduced with good ratio; in contrast the routing overhead is higher with the AntHocNet than the AODV this gives an idea that the ACO makes the network environment has a high routing overhead this property because the adaptation in the net through the using of the ant colony optimization algorithm to adapt the net behavior also measured with

pause time in second see Figure (3.1.2).

Again the AntHocNet has three states of the network job that are: the averaging end-to-end packet delay, packet delivery ratio, and routing overhead which measured with number of nodes in each test, through looking at the net job details we can conclude that AntHocNet reduce the amount of the average end-to-end packet delay time with very big ratio than the AODV method, while the packet delivery ratio is increased in using the AntHocNet over the use of AODV this because the guidance to ACO for searching a new path in the network routing, in the routing overhead the curve in AODV is grown gradually with increasing the number of nodes but in the AntHocNet the curve is not regular that is in the beginning intervals is more but in the ending intervals is less than in the AODV, see figure (3.1.3). Figures (3.1.4) is shown the same states of the network job but for increasing network sizes. Figure (3.1.5) represents the evolution of the end-to-end delay over the course of a test run.

(Case 2 Abtlvncpr)

In terms of the Bees Algorithm, each bee represents an LVQ network with a particular set of reference vectors. The aim of the algorithm is to find the bee with the set of reference vectors producing the smallest value of the error function. The LVQ network training procedure using the Bees Algorithm thus comprises the following steps: [7]

1. Generate an initial population of bees.
2. Apply the training data set to determine the value of the error function associated with each bee.
3. Based on the error value obtained in step 2, create a new population of bees comprising the best bees in the selected neighborhoods and randomly placed scout Bees.
4. Stop if the value of the error function has fallen below a predetermined threshold.
5. Else, return to step 2.

In this method the bees algorithm is used with an adaptation tool to train the LVQ Neural network for pattern recognition application (control chart) so to make the accuracy of the training procedure more preferred but with the increasing of the number of iterations, thus the table (1) bellow shows the values of the parameters which adopted for the bees algorithm. Table (2) bellow presents the classification results obtained for runs of the bees algorithm.

The effects of number of iteration of the LVQ neural network on the classification training accuracy is clear, from this effect, we can say that the training accuracy is enhanced with the increasing of the iterations number but with high increasing this means if we want to obtain a very high accuracy training for the pattern recognition we must increase the neural network iterations with large number reach to many thousands of iterations, see figure (3.2.1).

Also with the looking on table (3) there is a comparison results of

different pattern recognizers by using the standard LVQ neural networks one time and LVQ with bees algorithm another time in term of learning accuracy and test accuracy, it can be clearly observed that the learning (training) accuracy in the LVQ with bees algorithm is better than in the standard LVQ neural network as well as the test accuracy.

(Case 3 SIDMS)

In this method the assumption swarm intelligence approach is used to determine the robot movements so to reach to the target's position as a suitable movement calculation among many tests were used to the same purpose such as communication disruption test, independent test, cooperative test, and threshold test. The proposed method use two type of comparing results: the large scale results and the physical imitation results, in each type there are two measured parameters to evaluate each method from others.

These two parameters: [8]

- *Positional Certainty*
Positional certainty is the percentage of the area, which the target's position is narrowed down to. Lower percentages represent a more accurate placement of the target.
- *Number of Sightings*
The number of sightings reports the number of chances a robot had to take a measurement of the target's position.

In the *Large Scale Results* simulations the system work depend on the number of robots were used to do the movement which is limited with used hardware and the results to

be depend with available of robotics resources these results will be more active with increasing the robotics resources.

In the test system with large scale results; the large scale positional certainty represents a simulation results by comparing the positional error with number of robots reach to less than 300, clearly had observed that the positional certainty in the communication disruption test is higher than the other tests, while the cooperative test gives positional certainty more accurate than the other tests, see figure (3.3.1), while the large scale sightings is obtained by comparing the number of target sightings with the number of robots also reach to less than 300, as it can be reported the communication disruption tests behave the same way as in the first parameter by producing results separated from the other tests, but in all tests the number of sightings make peak when the number of robots increase without cause a big increasing in the number of sightings, see figure (3.3.2).

While in the *Physical Imitation Results* simulations are designed to model for the behavior with the movement, note that the same tests were used in the large scale results were used here also but the number of robots is limited here from one to three robots to de the tests.

Respectively in the test system with physical imitation results; the simulation positional certainty compared with number of robots limit to three as a maximum, we can report

that the cooperative, threshold, and communication disruption tests produce results more accuracy than the independent robots, see figure (3.3.3), in the other side, the test system has results of simulation number of sightings also compared with at most three robots, as it pointed the communication disruption has slightly stable increasing in the number of sightings but has a lower end, while the cooperative test produces a maximum number of sightings as all, see figure (3.3.4).

(Case 4 BTAA)

In this method the researcher aims to enhance the ant algorithm by embedding the backtracking heuristic technique in the ant algorithm environment so to make the algorithm more efficient in performance and behavior to solve the network routing problem in different situation of nodes number which is associative with the number of ants in each iteration, the proposed algorithm is applied in solving the network routing problem in the term of path cost and run time to execute this path from the source to the destination nodes then compare the result that obtains from the old methods with the results that obtained from updating the old methods based on the using of the backtrack technique in heuristic search.

Experimental Results of Network Routing

1. In this practical application five methods of AS [Ant System (AS), Elitist AS, Rank-Based (AS), Max-Min AS, ACO] are used, then a comparison is made between the old

algorithm of AS and the updated Backtracked AS algorithm.

2. Best results are obtained in the updated algorithm for all the five methods in comparison with the old algorithm.

3. Among these five methods it is noted that the ACO method has the best results among them, which gives a minimum cost path and shortest length tour in minimum execution time.

4. Updated Ant Colony Optimization (UACO) is the best method especially in the range of $[0.7n \rightarrow 0.85n]$, $n =$ No.of ants], exactly in this range the (UACO) is the best. [9]

Five standard methods (ACO, AS, ELITIST, RANK, and MAXMIN) of Ant algorithm were used in solving the routing in network with five update methods (UACO, UAS, UELITIST, URANK, and UMAXMIN), the results of update methods are clearly better than the standard methods in term of the shortest path with different number of nodes, starts from 10 nodes (with various ants number 5, 8, 10), 20 nodes (with various ants number 5, 10, 14, 18, 20), to 30 nodes (with various ants number 5, 10, 15, 18, 24, 30), see figure (3.4.1).

Consequently the update methods give results, totally, better than the old methods in term of system run time or time of execute the ant algorithm to solve the network routing problem with different number of nodes associate with various number of ants as it the same in the term of the shortest path test, see figure (3.4.2).

Discussion and Conclusion

As it mentioned in details, the four previous swarm intelligence approaches present different methods to solve same problems in one time and various problems in other time, that is the case1 (SIRMAHN) and case4 (BTAA) are similar in the application execution in term of network routing but differ in the specific functionality of the network routing type, further more case1 results of average delay, delivery ratio, average jitter and overhead are evaluated by comparing these network tasks with the AODV results because it is considered the standard Ad-Hoc tasks manager environment, in the other side case4 results of path cost (shortest path) and execution run time are considered to evaluate the enhanced approach therefore the results of the BTAA are compared with standard ant system, five methods of ant system (AS, Elitist AS, Rank-based AS, Max-Min AS, and ACO) are enhanced to form new five methods to be compared with the old ones in the system performance of the network routing. Now we have the following facts:

- Case1 is applied to the Ad-Hoc net while case4 is applied to normal network.
- Case1 results were evaluated through AODV approach while case4 results were evaluated through the standard AS.
- It is clear that the different level of mobility, increasing node density, and increasing network size are basic parameters in case1, while

the number of nodes, number of iteration, and number of ants are the basic parameters in case4.

Case2 (ABTLVNCPR) and Case3 (SIDMS) are two different approaches to solve problems, case2 used the bees algorithm in training the LVQ Neural Network to perform the pattern recognition application to increase the training accuracy but in the other side the bees algorithm causes increasing the number of training iteration, while Case3 (SIDMS) present an individual approach to determine the robot movement to reach the target with best position; four methods (Independent, Cooperative, Communication Disruption, and Threshold) were used in this approach to determine the best robot movements to its goal position in two system environment the large scale and the physical imitation. Now we have the following facts:

- Case2 is applied to the neural network as an adaptive tool, but case3 is applied to the robot movement to its position as an optimization tool.
- Case2 results were evaluated with standard LVQ in term of learning accuracy and test accuracy, while case3 results were evaluated between four suggested methods.
- Case2 is accomplished in a single system environment, but case3 is accomplished in two system environment in term of positional certainty and number of sightings in the both large scale and physical imitation.

- It is clear that the number of used robots is affected on the positional error and the number of target sightings thus in the large scale the number of robot is reach to more than 200 robots in contrast in the physical imitation the number of robots is 3 as max.

Through the observations from the four previous approaches to solve verity types of problems (applications) such as network job, neural network learning (training) and robot move to target position by using three types of swarm intelligence (ant system, particle swarm, and bees algorithm), the following items represent our conclusions obtained during this comparative studying:

- The swarm intelligence technique can be used in many fields of the system life cycle such as machine learning (neural network) and different practical problems especially optimization problems such as network routing and robot movement.

- Swarm intelligence success in achieving different network jobs with different network types.

- Through using the bees algorithm as an adaptive tool to train the neural network an advantage is obtained by increasing the learning accuracy and test accuracy but in contrast this cause increasing the number of neural network training iterations thus led to increasing the system run time as a disadvantage.

- As it know the robotics world is practically difficult to run, swarm intelligence present an efficient and

simple way to accomplish the robot movement and different jobs.

- When the system environment is changed the swarm intelligence algorithm still work efficiently but slightly less or more than the previous environment according to the parameters changing, such as in the robot movement through large scale or physical imitation.

References

- [1]Nikolay Y. Nikolaev and Hitoshi Iba, "*Adaptive Learning of Polynomial Networks*", *Genetic programming, back propagation, and Bayesian Methods*, Springer, 2006.
- [2]Dario Floreano and Claudio Mattiussi, "*Bio Inspired artificial Intelligence*", *Theories, Methods, and Techniques*, the MIT Press, 2008.
- [3]Hui Wang, Yong Liu, Sanyou Zeng, and Changhe Li, "*Opposition _Based Particle Swarm Algorithm with Cauchy Mutation*", 2006.
- [4]Wai Kuan Foong, "*Ant Colony Optimization for Power Plant Maintenance scheduling*", Ph.D Thesis, University of Adelaide, April 2007.
- [5]D. T. Pham, Anthony J. Soroka, Afshin Ghanbarzadel, Ebube Kir Koc, Sameh Orti, and Michael Packianather, "*Optimizing Neural Network for Identification of wood Defects Using the Bees Algorithm*", IEEE International Conference on Industrial Informatics 1_4244_9701_0106, 2006.

- [6]Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella, "*Swarm Intelligence for Routing in Mobile Ad Hoc Networks*", 0-7803-8916-6/05 IEEE, 2005.
- [7]D. T. Pham, Sameh Orti, A. Ghanbarzadeh, and E. Koc, "*Application of the Bees Algorithm to the Training of Learning Vector Quantization Networks for Control Chart Pattern Recognition*", 0-7803-9521-2/06 IEEE, 2006.
- [8]Michael Brian Marshal, "*A Swarm Intelligence Approach to Distributed Mobile Surveillance*", Master Thesis, Virginia Polytechnic Institute and State University, 2005.
- [9]Ashty Mahdy Aaref, "*Backtracking Ant Algorithms*", Ph.D Thesis, University of Technology, Computer Sciences Department, 2009.

Table (1) the parameters of the bees algorithm [7]

Bees Algorithm parameters	Symbol	Value/ range
Population	n	200
Number of selected sites	m	20
Number of elite site	e	1
Patch size	ngh	0.01
Number bees around elite points	nep	20
Number of bees around other selected points	nsp	10

Table (2) LVQ classification results [7]

Number of runs	Train accuracy	Test accuracy
1	94.24%	93.39%
2	97.10%	96.05%
3	96.20%	95.51%
4	96.93%	96.56%
5	97.92%	95.10%
6	95.71%	94.36%
7	96.17%	95.63%
8	97.03%	96.44%
9	97.89%	95.65%
10	96.40%	96.03%
Max	97.92%	96.56%
Min	94.24%	93.39%
Mean	96.56%	95.47%

Table (3) result of different pattern recognizers [7]

Pattern recogniser	Learning accuracy	Test accuracy
LVQ (stand.)	95.18%	92.31%
LVQ (Bees)	96.56%	95.47%

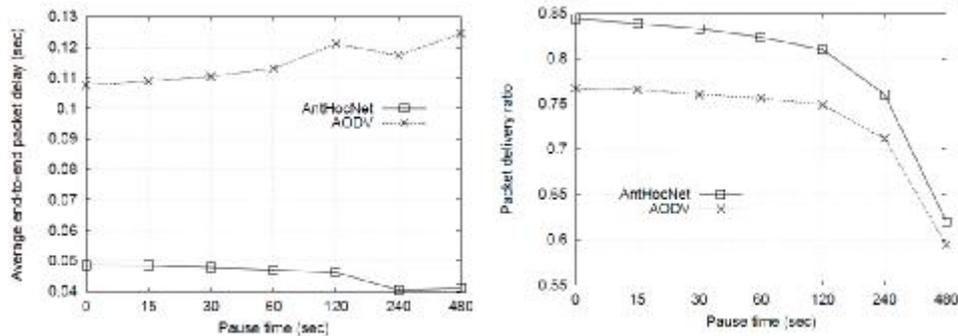


Figure (3.1.1) Average Delay and Delivery Ratio for Different Levels of Mobility [6]

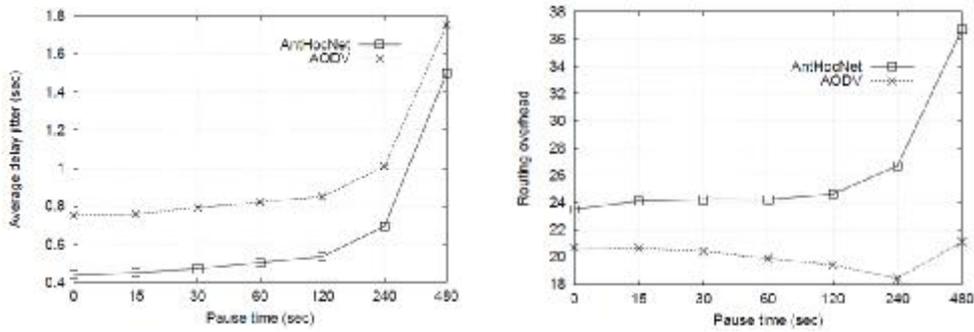


Figure (3.1.2) Average Jitter and Overhead for Different Levels of Mobility [6]

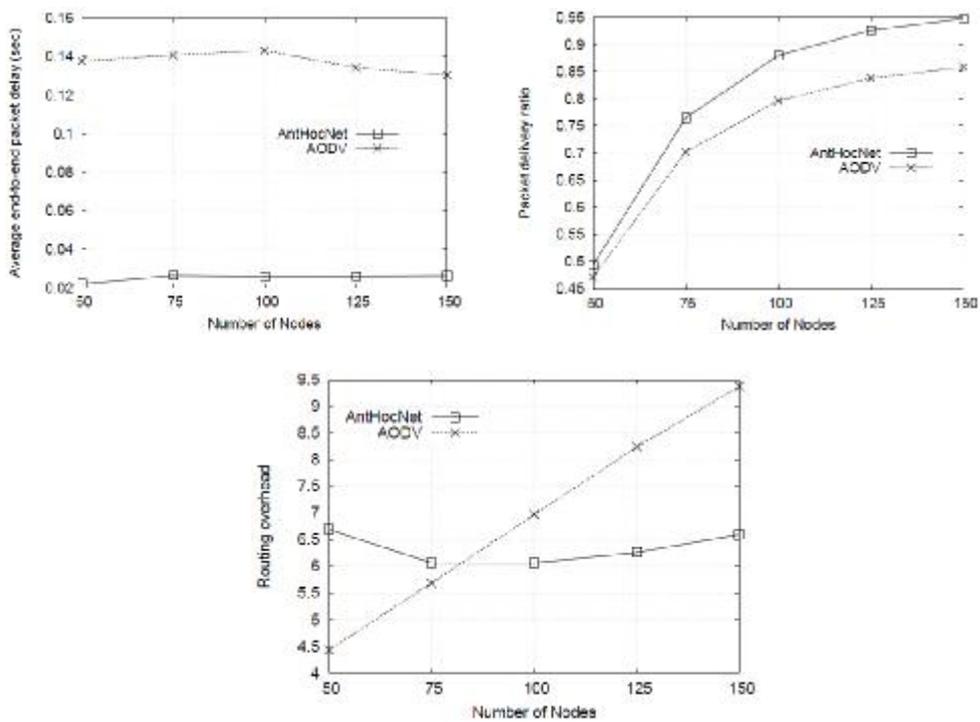


Figure (3.1.3) Average Delay, Delivery Ratio and Overhead for Increasing Node Density [6]

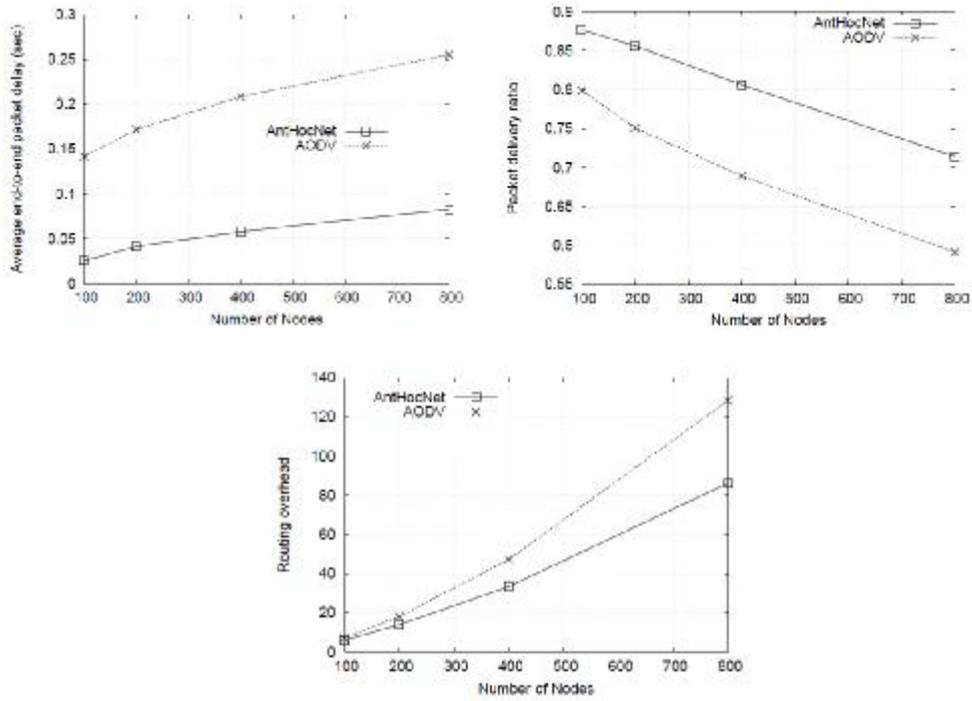


Figure (3.1.4) Average Delay, Delivery Ratio and Overhead for Increasing Network Sizes [6]

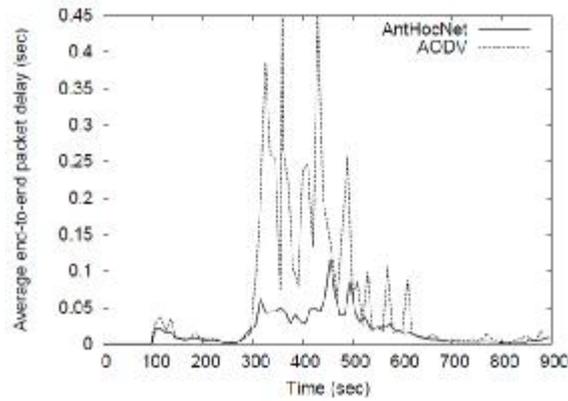


Figure (3.1.5) Evolution of the end-to-end Delay over the Course of a Test Run [6]

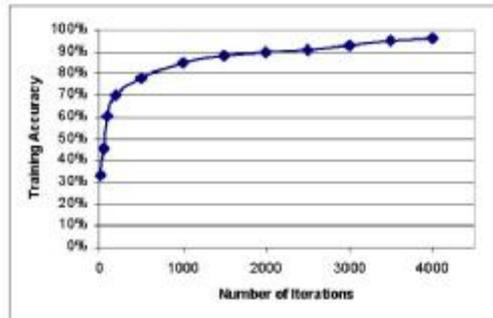


Figure (3.2.1) Effect of Number of Iterations on the Classification Accuracy [7]

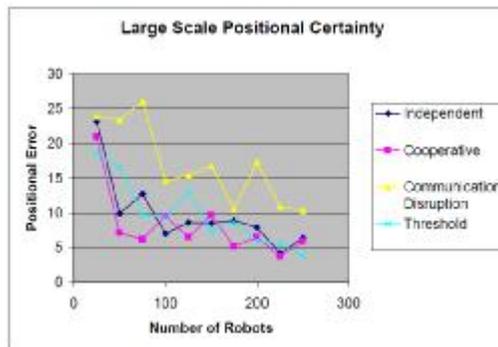


Figure (3.3.1) large scale results of positional certainty [8]

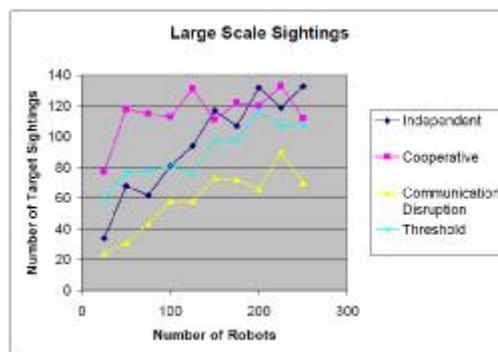


Figure (3.3.2) large scale results of number of sightings [8]

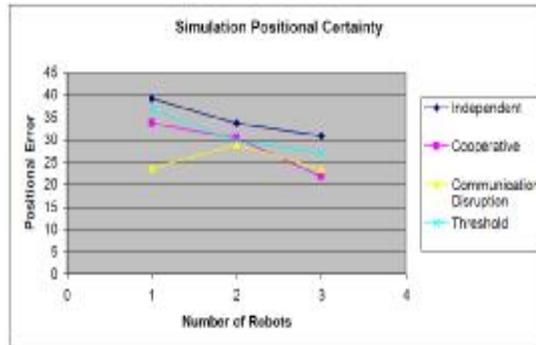


Figure (3.3.3) Physical imitation results of positional certainty [8]

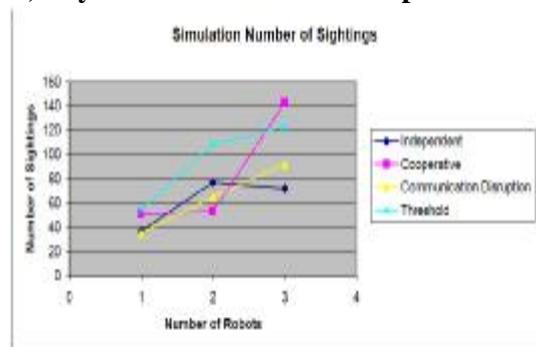


Figure (3.3.4) Physical imitation results of simulation number of sightings [8]

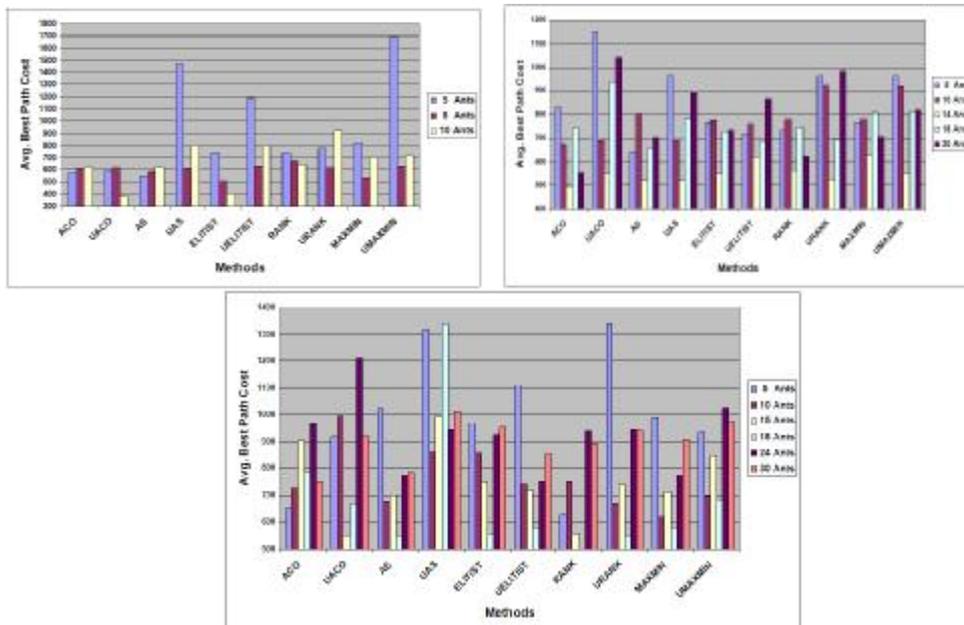


Figure (3.4.1) Avg. Best Path Cost for AntNet (Nodes =10, 20, 30 Consequently) Using Different Ant Searcher Methods [9]

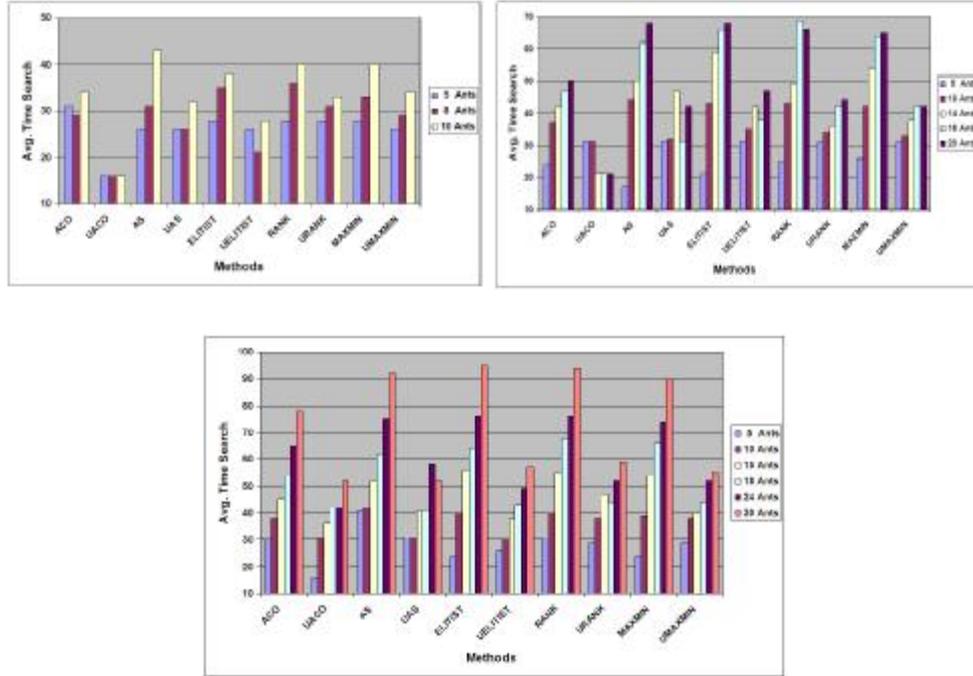


Figure (3.4.2) Avg. Time Search for AntNet (Nodes =10, 20, 30 Consequently) Using Different Ant Searcher Methods [9]