Turk J Elec Eng & Comp Sci, Vol.18, No.6, 2010, © TÜBİTAK
doi:10.3906/elk-0901-6

Performance analysis of swarm optimization approaches for the generalized assignment problem in multi-target tracking applications

Ali Önder BOZDOĞAN, Asım Egemen YILMAZ, Murat EFE
Department of Electronics Engineering, Faculty of Engineering, Ankara University, 06100, Tandogan, Ankara-TURKEY
e-mail: {bozdogan, aeyilmaz, efe}@eng.ankara.edu.tr

Abstract

The aim of this study is to investigate the suitability of selected swarm optimization algorithms to the generalized assignment problem as encountered in multi-target tracking applications. For this purpose, we have tested variants of particle swarm optimization and ant colony optimization algorithms to solve the 2D generalized assignment problem with simulated dense and sparse measurement/track matrices and compared their performance to that of the auction algorithm. We observed that, although with some modification swarm optimization algorithms provide improvement in terms of speed, they still fall behind the auction algorithm in finding the optimum solution to the problem. Among the investigated colony optimization approaches, the particle swarm optimization algorithm using the proposed 1-opt local search was found to perform better than other modifications. On the other hand, it is assessed that swarm optimization algorithms might be powerful tools for multiple hypothesis target tracking applications at noisy environments, since within single execution they provide a set of numerous good solutions to the assignment problem.

Key Words: *Generalized assignment problem, ant colony optimization, particle swarm optimization, data association, target tracking*

1. Introduction

The data association problem, in which the measurements are assigned to the established tracks, is a crucial step in multi-target tracking applications. From simple nearest neighbor method to complex multiple hypothesis tracking, the tracking literature is filled with a variety of solutions proposed for associating measurements to the established targets in a complex multi-target environment. These methods show progressive advancement in performance through taking advantage of the increasing computational resources. Lately, research on assignment methods has shown great success for solving the data association problem [1- 4]. In the assignment method, the data association problem is converted to a 0-1 optimization problem where the total distance/benefit of assigning targets to measurements is minimized/maximized [1]. The early assignment algorithms use only a list of measurements from a single time scan, which will be correlated with the targets being tracked. This way, the resulting data association problem can be formulated as a 2D asymmetric assignment problem, which can be solved efficiently by polynomial time algorithms such as Munkres [5], auction [6] and JVC [7]. Availability of cheaper computational power has stimulated a desire for exploiting further lists of measurements in making data association decisions among the researchers. Unfortunately, additional lists for the assignment yield a multidimensional problem, which is well known to be NP hard [2]. Thus, a variety of approximations were proposed to address the multidimensional assignment problem. Lagrangian relaxation is used to find a solution in polynomial time [8] as the state of the art approach. This method also provides a measure of accuracy for the solution found [2, 8]. However, in [8] it is stated that with this method, a complete assignment hypothesis tree is needed and over 90 percent of the computing power is spent on the creation of this assignment hypothesis tree rather than solving the assignment problem. To reduce the computational effort, a randomized method was proposed in [8] to build the assignment hypothesis tree randomly then solve the assignment problem on this reduced tree. This method demonstrated superior performance both in computational time and accuracy. Furthermore, the randomized method was able to create multiple assignment hypotheses without any additional computation; i.e. it can produce "m" *good* solutions, which can also be exploited by multiple-target tracking algorithms. Inspired from this success, we turn to investigate different randomized methods for solving the assignment problem encountered in multi-target tracking applications.

Our interest lies in the nature-inspired and colony-based stochastic optimization algorithms, namely, the particle swarm optimization (PSO) [9] and ant colony optimization (ACO) [10]. The ACO is an algorithm designed for solving discrete combinatorial problems. Therefore, it is directly applicable to solving the assignment problem for target tracking. In this paper, we will be using the MAX-MIN ant system (MMAS) variant of the ACO.

The particle swarm optimization method on the other hand, is originally designed for the optimization of continuous functions. However, due to its increasing popularity (caused by its simplicity and power), several researchers have applied it to combinatorial optimization problems. Of these, the most important contributions have been made by applying PSO to the task assignment (symmetric assignment) problem [11, 12], traveling salesman problem [13, 14], sequencing and scheduling problems [15-17], the quadratic assignment problem [18, 19], the permutation problem [20], and more recently to the shortest path problem [21]. In this study, among the works listed above, we investigate and apply two discrete modifications of PSO (the "Clerc" approach [13] and the "Hu-Eberhart-Shi" approach [20]) that are applicable to the generalized assignment problem. Moreover, we later propose a novel PSO type method called *1-opt local search* in order to address the shortcomings of the mentioned PSO variants.

In this work, we test these nature inspired algorithms on the 2D assignment problem and compare their performances with the performance of the auction algorithm. The reason for selecting the 2D assignment problem as our test bed follows from the fact that 2D assignment problem can be solved in polynomial time. The auction algorithm provides the optimum solution, and is a well-known algorithm in the tracking community. Therefore it is used as a reference for the performance of the nature inspired heuristic algorithms.

In Section II we present mathematical definition of the asymmetric assignment problem in target tracking. Section III describes the biologically inspired optimization methods we have studied. In Section IV, we describe our numerical tests and state our results. Finally, in Section V we raise our conclusions and suggest directions for future work.

2. 2D asymmetric assignment problem in target tracking

In multi-target tracking, at each scan time a set of observed measurements is populated in order to be assigned to the existing tracks, so that each track can be updated. Thus, the non-trivial problem of finding which observed measurement is originated from which target, has to be solved. Since only the current list of measurements is being assigned to the tracks, the problem is referred to as the "*2D assignment problem.*" In what follows, the definition of the 2D assignment problem in multi-target tracking applications is given.

Given a finite set of tracks $T = \{0,\ 1,\ldots,\ n\}$, a finite set of measurements $M = \{0,\ 1,\ldots,\ m\}$, and a matrix of track scores where each element $\mathrm{B}(i,\ j)$ represents the benefit of associating track i to the measurement j. The objective of the 2D asymmetric[1] assignment problem is to find the track-measurement association with the global maximum benefit satisfying the following constraints [4]:

1. Each track (except for track zero) is to be associated with at most one measurement.
2. Each measurement (except for measurement zero) is to be associated with at most one track.

Track zero and measurement zero are dummy variables. Track zero represents the case where no track can be found to be associated with a measurement. This may possibly be caused by a spurious measurement (false alarm) or a valid measurement from a new track initiator, i.e., a new target. Likewise, measurement zero represents the case where no measurement can be found to be associated with a given track, therefore it is a misdetection. Mathematically these criteria can be formulated as [1]

$$\max_{\rho} \sum_{i=0}^{n} \sum_{j=0}^{m} B(i,j)\rho(i,j) \tag{1}$$

where ρ is the binary assignment variable such that

$$\rho(i,j) = \begin{cases} 1 & \text{measurement j is assigned to track i} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

subject to the conditions

$$\begin{aligned} \sum_{i=0}^{m} \rho(i,j) &= 1 \text{ for } j = 1, \ldots, n \\ \sum_{j=0}^{n} \rho(i,j) &= 1 \text{ for } i = 1, \ldots, m \end{aligned} \tag{3}$$

2.1. Nature inspired heuristic algorithms

In this paper we are interested in colony based, nature inspired algorithms to find a group of solutions to the assignment problem. The algorithms studied in this work are described below.

[1]The word asymmetric indicates the fact that the number of measurements and tracks are not the same.

2.2. Particle swarm optimization

Particle swarm optimization (PSO) is a population based, biologically inspired stochastic optimization method. It was discovered by Kennedy and Eberhart while they were trying to simulate a simplified social model [9]. This simulation model was later found out to be propitious in optimization of continuous functions.

PSO, similar to other population-based algorithms, is initialized with a random set of solutions (in fact solution candidates). These solutions are referred to as *particles.* The set of particles is called a *swarm.* In the generic PSO, each particle is flown randomly in the problem space under the influence of two attractors, particle's own best position and the best position found by any member of the whole swarm. Through this random motion that is biased by the local best and the global best attractors, a search is carried out around desirable regions of the problem space.

Assuming a swarm of n particles is being used, each individual particle i, $(1 \leq i \leq n)$ at PSO iteration number t has the following attributes [22]: A position vector, $x_i(t)$, which codes parameters that need to be optimized; a velocity vector, $v_i(t)$, which operates on the position vector in order to determine the position where the particle will be flown next, and finally a personal best position vector, $y_i(t)$, which stores individual's most fit position found from the beginning of the algorithm up to iteration t. Additionally, each individual particle shares its personal best position with its peers in order to find the global best position, y_i^*, the most fit position by any particle within the swarm, therefore ensuring a swarm-wide information sharing.

At each iteration, the particle position is updated through the equations

$$\begin{aligned} v_i(t+1) = wv_i(t) + c_1u_1(t).*[y_i(t) - x_i(t)] \\ +c_2u_2(t).*[y_i^*(t) - x_i(t)] \end{aligned} \tag{4}$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \tag{5}$$

where, u_1 and u_2 are d dimensional independent identically distributed vectors whose elements are sampled independently from the uniform distribution, U(0,1). The variable w in equation (4) is called the inertia weight, which is typically linearly reduced (from 1.0 to 0.0 according to [22], from 0.95 to 0.4 according to [23]) as the algorithm progresses. The variables c_1 and c_2 denote the acceleration coefficients. They determine how far an individual particle can fly at each iteration, and typically they are set to 2.0 [24]. The operation ".*" here is defined to be an element by element product.

The standard PSO algorithm described above has been proposed for, and commonly used in, the optimization of continuous functions. For the problems of discrete nature, however, application of the PSO is not as popular. The problem of assignment, unfortunately, is a discrete combinatorial problem. It can easily be seen that random update procedure described in the formulation above will lead to invalid assignments as a result of which multiple tracks get assigned to the same measurement or multiple measurements get assigned to a unique track. Therefore, the standard PSO procedure needs to be modified for the solution of the assignment problem. The modifications implemented in this work are described below.

2.2.1. PSO Variant-1 Based on Clerc's Approach

Originally proposed for the traveling salesman problem (TSP), Clerc has redefined the terms in equations (4) and (5) in order to apply the PSO to the TSP [13]. Since 2D assignment can be thought of as a relaxation to the TSP, a similar approach is applicable to our case.

Definitions of terms and operations within the Clerc framework for PSO used in the generalized assignment problem are described below [13].

Clerc's Definition 1: Position

Each particle is encoded as a permutation vector representing the order in which tracks get assigned to the measurements. For example, the particle $x_i(t) = [1\ 2\ 3\ 4\ 5]$ will code the assignment where the first track gets assigned to the first measurement, the second track gets assigned to the third measurement and so on for all the tracks in the tracks list.

Clerc's Definition 2: Velocity

In order to apply the PSO, the velocity term has to be redefined, such that when applied to a position term, it has to change this term to another position in the permutation space. In Clerc's formulation, the velocity is defined as a set of sequential swap operations. For example, the velocity term $v_i(t) = \{(1,2), (2,3)\}$, when applied to a position vector, will dictate that two consecutive swaps in the measurement list order will be made (i.e., first, the position of the first measurement will be swapped with that of the second measurement; later this new position of the second measurement will be swapped with the position of the third measurement).

In Clerc's approach, two types of addition operations are defined.

Clerc's Definition 3: Addition of Position with Velocity

Addition of a position term with a velocity set will result in a new position by implementing swaps given in the velocity set as described above.

Clerc's Definition 4: Addition of Velocity with Velocity

Addition of a velocity with another velocity will append the new velocity set to the end of the first velocity set.

Clerc's Definition 5: Subtraction

Subtraction operation is defined exclusively in order to be used with position vectors. Subtraction of two position vectors results in a velocity set which, when applied to the subtrahend, will give the minuend.

Clerc's Definition 5: Multiplication

Multiplication operation is defined to be used exclusively with the velocity term. Let c be a real coefficient, v be a velocity and $||v||$ be the cardinality of v. According to Clerc [13], the result of the multiplication of c with v depends on the value of c:

1. Case $c = 0$

   $cv = \emptyset$

2. Case $0 < c \leq 1$

   In this case, v will be truncated in order to keep its first $c||v||$ elements. If $c||v||$ is not an integer, it will be rounded down.

3. Case $c > 1$

   In this case, c will be decomposed as:

   $c = k + c'$, $k \in \mathrm{N}^+$, $c \in [0,1[$

and cv will be defined to be:

$$cv = \underbrace{v \oplus v \oplus \ldots \oplus v}_{k \text{ times}} \oplus c'v$$

4. Case $c < 0$

   In this case, cv will be rewritten as $cv = (\text{-}c)(\text{-}v)$; where negation of velocity is defined to be the reversal of the order of the swap operations kept in the velocity set.

With the operations and terms defined above, PSO algorithm described with Equations (4) and (5) becomes applicable to both TSP and generalized assignment problems.

2.2.2. PSO variant-2 based on Hu-Eberhart-Shi approach

Originally designed to solve the n-queens problem, Eberhart et al. in [20] generalize the PSO algorithm to the permutation type problems. In their approach, a particle's position is encoded as a permutation vector similar to that of the Clerc's method. Velocity term also has a usage of swap operation acting on the position vector. However, rather than dictating sequential swap operations immediately like the Clerc velocity, velocity is treated as a vector encoding the probability of a swap of a certain assignment in the Hu-Eberhart-Shi approach. If the distance between particle and better solutions known by the swarm is large, the particle velocity becomes large and the particle gets more inclined to changing into a new permutation sequence. A detailed description of the definitions of terms and operations within the Hu-Eberhart-Shi framework for PSO used in this work is given below [20].

Eberhart et al.'s Definition 1: Position
Each particle is encoded as a permutation vector, which encodes the track to measurement associations in the same way as in the Clerc algorithm.

Eberhart et al.'s Definition 2: Velocity

Velocity in the Hu-Eberhart-Shi approach, which is calculated through equation (5), represents the probability of change in particle's position. It is encoded as a vector of probabilities and has the dimension of the position vector. However, since probabilities are defined to lie in the range [0, 1], the calculated velocity is further processed before acting on the position. First, the absolute value of the velocity is taken; then, the positive velocity vector is normalized to its maximum range so that velocity probabilities are achieved. A running example of the procedure taken from [20] is shown in Figure 1.

In Figure 1, assuming that the maximum velocity is 50, first the velocity vector is normalized and $|V|$ is found. Then for each position element, a swap between the element of the global best position and the particle position is made according to the probability calculated in $|V|$. As shown in Figure 1, the marked track of the tested particle is assigned to the seventh measurement, whereas the same track of the global best particle is assigned to the seventeenth measurement. The velocity vector shows that probability of swap for that track is 0.8. Therefore a random sampling from the uniform distribution is made and assuming that a swap is found as the outcome, position of measurement seven and seventeen in the tested particle is swapped.

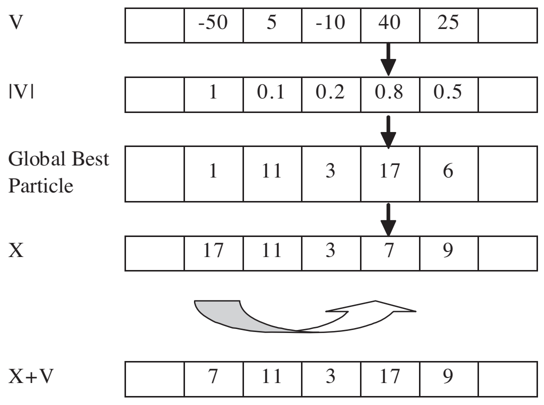


Figure 1. Description of Hu-Eberhart -Shi approach for the position element marked with arrows [20].

The drawback of this approach is that, once a particle and the global best particle becomes the same, no further change can be achieved. Eberhart et al. recommend a random swap between arbitrary two measurements if such a case is observed.

2.3. PSO variant-3 called 1-opt local search approach

Our experiments revealed that both Clerc and Hu-Eberhart-Shi approaches displayed very slow convergence, in fact, too slow to render them feasible for multi-target tracking applications. Thus, we propose a faster PSO type algorithm based on 1-opt local search method. The proposed 1-opt local search method merges PSO type group/colony search methodology with the well-known 1-opt local neighborhood search. In this framework, a group of 1-opt local search agents perform parallel random local search, and share information through the global best agent found by the whole *swarm.* It is a greedy type method; therefore, unlike the standard PSO agents do not need to store their local best positions.

The 1-opt local search algorithm starts with n random particles sampled uniformly form the permutation space described above. The particle position definition is the same as the definition used by both Hu-Eberhart-Shi and Clerc approaches. Later, a main loop consisting of local and global search steps is executed until the termination conditions are met. 1-opt local search can be thought of a PSO variant, which gets rid of velocity definition unsuitable to the assignment problem; and divides particle position update into two steps, the local search update and the global search update. The local as well as global search updates are described in detail below.

2.3.1.1 Local search update

The local search update for each particle consists of a loop of size k, which is configurable by the user. Each loop iteration starts with a random track selection. For each track that comes after the selected track in the position vector, a test is undertaken to check whether swapping measurements between two tracks increases the fitness of the particle. If an increase is observed, the swap is made and the loop is reinitialized with a new track selection. The pseudo-code of the local search update is given in Figure 2.

```
for (i = 1 : Search Iterations)
  Select a random track
  Put selected track to the taboo list
  for ( Second Track = ((Selected Track + 1) : length(Tracks)) )
      if swapping measurement of selected track and second track
      decrease cost
        do the swap
        break the second loop
     end
  end
end
```

Figure 2. Description of 1-opt local search – Local search step.

2.3.1.2. Global search update

In the global search update, particles updated by the local search procedure are compared to the global best particle of the swarm. For every track coded in the position vector of the particle, if the measurement taken by the particle and the measurement taken by the global best particle is different, it is checked whether a swap in the particle between local measurement and global measurement is leading to an increase in the fitness of the particle. If fitness is increased, the swap is performed; otherwise, the swap is still performed, but this time with a probability of p, which is another user configurable parameter. The pseudo-code for the global update procedure is given in Figure 3.

This simple algorithm, with its greedy approach, has shown superior performance compared to both Hu-Eberhart-Shi and Clerc approaches in execution time as well as in the quality of solutions found. The complexity of the algorithm is estimated as follows: Let M describe the length of the measurement vector. In each local search iteration step, particles will on the average select the track in the middle of the track list. Therefore, for the worst case assuming the track in the middle is selected, $M/2$ swap checks are performed for each k local search iteration. Adding to this, the M global search swap checks and assuming P particles are executed for I iterations. Thus, the average complexity of the algorithm is found to be $O(IP((kM/2)+2))$.

2.4. Ant colony optimization

Ant colony optimization (ACO) is a biologically inspired, group based stochastic optimization algorithm using artificial particles called ants, which iteratively construct random candidate solutions to combinatorial optimization problems [10]. The solutions constructed by the ants are biased to be in the good regions of the problem space under the influence of two forces, namely problem dependent heuristic information and pheromone trails. The problem dependent heuristic information is gathered from the fitness function to be optimized whereas the pheromone trail is a specialty of the ACO achieved by positive feedback from ant paths constructed throughout the algorithm.

```
global_search(particles,Cost Matrix)
for (i = 1 : length(particles))
   for (j = 1 : length(particles(i))
      Compare the measurement associated to particle(i) s  Track j and
      global best
      particle s Track j
        if different
           Find the swap required to make particle(i) s  Track j equal
           to the global best  Track j
               If swap increase benefit/decrease cost
                  Do the swap
               else
                  Do the swap with a probability p
               end
         end
    end
end
```

Figure 3. Description of 1-opt local search – Global search step.

In the ACO, multiple ants construct candidate solutions by starting with an empty solution [25]. Ants then iteratively add new solution components probabilistically until a complete solution is achieved. After solution construction is finished, ants give feedback on the solutions they have constructed by depositing pheromone on the solution components they visited. In general, solution components which are part of better solutions are used by many ants. This is because of the fact that, in ACO, good paths will receive a higher amount of pheromone; thus in future iterations ants become biased to follow such paths. In the classical ACO, all ants are allowed to deposit pheromones, but more recent ACO variants prefer a more elitist strategy where only a single or a small group of ants are allowed to leave pheromone trails [26]. To avoid stagnation in the search process, all pheromone trails are faded (evaporated) by a factor before ants lay new pheromones. Evaporation also helps to avoid unlimited accumulation of pheromones.

In the standard ACO algorithm, a main loop is repeated until a termination condition, generally a maximum amount of iterations, is met. In the main loop, first, the ants construct feasible solutions, then the pheromone trails get updated. Optionally, a local search may be applied to improve the quality of solutions.

In this work, we use MAX-MIN ant system (MMAS) [25] in solving the asymmetric assignment problem. The MMAS is a successful variant of the ACO algorithm and is distinguished from the other ACO variants in setting up maximum and minimum limits for pheromone levels and assuring that such levels are not violated by successive deposition and fading of pheromones. Also, like other successful variants the MMAS uses an elitist strategy in pheromone deposition. The description of MMAS algorithm [25] fused in this work is described below.

2.4.1. Tour construction

Initially, ants are placed on randomly chosen tracks. Then, at each construction step, ants select a measurement for this track probabilistically. The selected track (as well as the selected measurement), if different from the dummy measurement, is put into taboo lists for that particular ant; then the ant moves to another randomly selected track not in the taboo list. The probabilistic choice measurement is biased by both the pheromone trail and locally available heuristic information. The pheromone trail information is described below, and the latter is defined as the reciprocal of $c_i(t)$, the cost of assigning measurement i to track t [10, 25–26]:

$$\eta_i(t) = \frac{1}{c_i(t)}. \tag{6}$$

Given that pheromone level and the heuristic information for measurement i assigned to track t to be $\tau_i(t)$ and $\eta_i(t)$, assuming that Ω represents the set of valid measurements for track t, then the probability of assigning measurement i to track t is found as follows [25]:

$$p(M_i|\, Track(t)) = \frac{\tau_i(t)^\alpha \eta_i(t)^\beta}{\sum\limits_{i\in\Omega} \tau_i(t)^\alpha \eta_i(t)^\beta}. \tag{7}$$

In Equation (7), the terms α and β determine the relative importance of heuristic and pheromone trail information.

2.4.2. Pheromone update

After all ants finish associating a measurement to each track, the pheromone trails get updated. First, all pheromone trails are lowered the by a constant factor (evaporation), then the best performing ant deposits pheromone on the track-measurement assignments it has created [25]. Pheromone update is performed through the following equaiton [25]:

$$\tau_i(t) = \rho\tau_i(t) \ + \sum\nolimits_{j=1}^{m} \Delta\tau_i(t)^{\,j}. \tag{8}$$

Here, ρ is the trail persistence (thus, 1 - ρ models the evaporation) [25], $\Delta\tau_i(t)^j$ is the amount of pheromone the best ant puts on the assignment of measurement i to track j if they are assigned and m is the number of measurements. $\Delta\tau_i(t)^j$ is defined as

$$\Delta\tau_i(t)^{\,j} = \begin{cases} \frac{1}{c_i(t)}, & \text{if measurement } j \text{ is assigned to track } t \\ 0\ , & \text{otherwise} \end{cases} \quad . \tag{9}$$

In general, good assignments will receive more pheromone, and therefore will be more likely to be chosen in future iterations of the algorithm. In the MMAS, the maximum and minimum levels for pheromone are limited. The maximum limit on the pheromones is shown in [25] to be

$$\text{pheromone}_{\max} = ((1-\rho)\text{OptimalTotalCost})^{-1}. \tag{10}$$

Since optimal total cost is unknown, the optimal cost found by the algorithm is used to estimate the maximum pheromone limit. The minimum pheromone limit is calculated as

$$\text{pheromone}_{min} = \text{pheromone}_{\max}(m)^{-1}, \tag{11}$$

where m describes the length of the measurement matrix.

3. Numerical tests and results

Although our interest lies in multi-target tracking applications, we find it unnecessary to utilize a full tracking scenario to test the assignment algorithm performances as we are only interested in the optimum measurement to track assignments. Thus, in order to focus only on the performance comparison of the 2D assignment approaches, we have created random dense and sparse measurement/track matrices in order to test the algorithms. The elements of test matrices represent the benefit resulting from associating the relevant track with the measurement. The benefit values were sampled uniformly from the integers within the interval [1,100]. The optimal assignment configuration is found by utilizing both auction and Munkres algorithms. The timing performance of the auction algorithm was found to be superior to that of Munkres algorithm, thus the auction algorithm was taken to be the benchmark for the heuristic algorithms. The parameters utilized by heuristic algorithms are summarized in Table 1.

Table 1. Parameters used in heuristic algorithms.

PSO	ACO	1-Opt PSO
w= 0.5+rand(0,2)	$\alpha = 1$	$k = 50$
c_1= 2	$\beta = 15$	$p = 10^{-6}$
c_2= 2	$\rho = 0.8$	

An Intel Pentium 4 PC with a 2.8GHz CPU and 1 GB memory was used as a test bed. All algorithms were coded on a MATLAB R2006a platform. In the simulations, 50 random benefit matrices were created, and the algorithms were tested for 300 iterations. For the results presented, "problem size" indicates total number of elements in the test matrices. Two major simulation setups were constructed in order to observe the effect of the change in some parameters. These setups will be described in the following subsections.

3.1. Simulation Setup 1

In the first simulation setup, in order to investigate the effect of particle/ant size, performance analysis was carried out with colonies consisting of 5, 10 and 15 particles/ants. For this setup, the test matrix size is fixed at 100×100.

By taking the global benefit returned from the auction algorithm to be 100 percent, the percentage average benefits returned by the heuristic methods are computed as seen in Table 2. It is observed that both the 1-opt and ant colony optimization methods show robust behavior in providing solutions of good quality, even with increasing problem dimension. In contrast, both Hu-Eberhart-Shi and Clerc algorithms do not provide good quality solutions within the fixed iterations given. Furthermore, their performance deteriorates when increasing matrix dimension.

If we compare the ant colony optimization with 1-opt local search in terms of accuracy, the ant colony algorithm seems to be more capable in providing solutions of better quality.

Table 2. Percent average benefit (computed over 50 independent executions) returned by the heuristic methods.

	Problem Size	50		100		200		400	
Particles		Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
5	1-opt Local Search Search	98.09	0.15	97.68	0.20	97.32	0.16	97.48	0.22
10		98.14	0.07	97.99	0.11	97.28	0.21	97.51	0.18
15		98.19	0.18	98.15	0.18	97.36	0.17	97.64	0.19
5	Ant Colony Optimization	99.50	0.03	98.50	0.09	97.92	0.12	97.51	0.15
10		99.69	0.08	98.98	0.16	98.22	0.21	97.60	0.18
15		99.74	0.04	99.12	0.13	98.21	0.11	97.88	0.13
5	Clerc Algorithm	74.03	0.45	66.30	0.16	61.88	0.42	58.21	0.94
10		75.30	1.11	68.45	0.39	64.61	0.30	59.53	0.89
15		77.47	0.65	71.35	0.19	67.22	0.63	61.04	0.93
5	Hu-Eberhart-Shi Algorithm	91.93	0.57	86.11	0.57	76.10	0.34	67.66	0.55
10		93.49	0.55	88.78	0.68	81.24	0.31	70.97	0.67
15		95.50	0.29	90.21	0.54	83.44	0.53	74.22	0.62

Tables 3 to 5 summarize the timing results (i.e. average timing behavior of the algorithms) obtained for this experimental setup. In terms of timing, it can be observed that both 1-opt local search and Hu-Eberhart-Shi algorithms perform better than the ant colony optimization methods and Clerc algorithms.

Even though it was previously observed that ant colony optimization provides more accurate solutions than 1-opt local search, it seems to be inappropriate for real-time applications due to low speed, and nonlinear timing behavior (i.e. abnormal increases in the execution time with increasing problem size). On the other hand, the other three other algorithms demonstrate almost-linear timing behavior. Among these, Clerc method seems to be much too slow. Hu-Eberhart-Shi and 1-opt local search methods, which are competing with each other in terms of speed and timing behavior, outperform ant colony optimization and Clerc.

In Tables 3 to 5 is shown a comparison of 1-opt local search and ant colony optimization (in terms of 95% maximum benefit achievement). It is observed that 1-opt local search still outperforms ant colony optimization in speed and linear timing behavior for convergence to 95% maximum benefit. In this comparison, the results of Clerc and Hu-Eberhart-Shi methods are not considered and included; since they have already proven to be failing in terms of solution accuracy as presented in Table 2 previously.

Table 3. Timing results (in seconds) for 5 particles and 300 iterations.

	Total Time (Completion of All Iterations)			
Problem Size	50	100	200	400
1-opt Local Search	0.94	1.8	3.7	8.73
Ant Colony Optimization	18.66	84.0	494.7	3438.2
Clerc Algorithm	14.37	30.95	65.3	135.7
Hu - Eberhart - Shi Algorithm	0.8	1.11	1.77	3.13
Auction Algorithm	0.03	0.17	1.41	1.92
	95% Max Benefit Achieving Time			
Problem Size	50	100	200	400
1-opt Local Search	0.01	0.04	0.11	0.29
Ant Colony Optimization	0.19	1.68	6.60	57.3

Table 4. Timing results (in seconds) for 10 particles and 300 iterations.

	Total Time (Completion of All Iterations)			
Problem Size	50	100	200	400
1-opt Local Search	1.99	3.91	8.23	18.42
Ant Colony Optimization	36.08	161.48	982.2	6837.9
Clerc Algorithm	30.18	65.65	136.9	280.83
Hu - Eberhart - Shi Algorithm	1.66	2.23	3.65	6.51
Auction Algorithm	0.03	0.17	1.41	1.90
	95% Max Benefit Achieving Time			
Problem Size	50	100	200	400
1-opt Local Search	0.03	0.07	0.19	0.55
Ant Colony Optimization	0.48	1.61	13.10	113.9

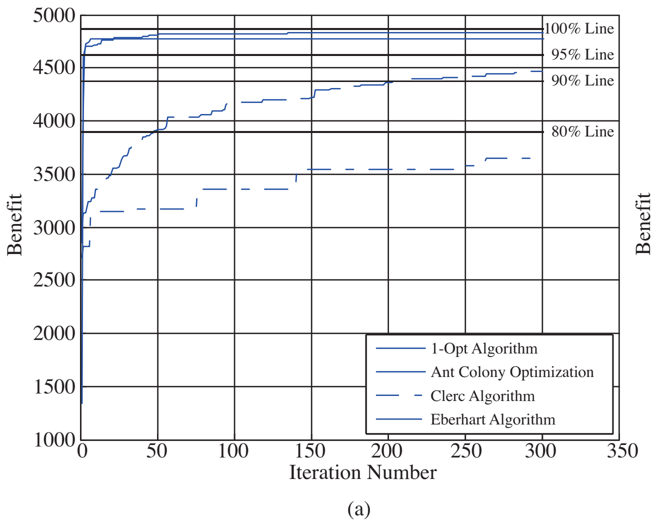
Table 5. Timing results (in seconds) for 15 particles and 300 iterations.

	Total Time (Completion of All Iterations)			
Problem Size	50	100	200	400
1-opt Local Search	3.1	6.04	12.79	29.24
Ant Colony Optimization	53.6	245.59	1462.8	10210.5
Clerc Algorithm	46.4	99.78	210.22	460.79
Hu - Eberhart - Shi Algorithm	2.53	3.47	5.54	10.41
Auction Algorithm	0.03	0.17	1.44	1.93
	95% Max Benefit Achieving Time			
Problem Size	50	100	200	400
n1-opt Local Search	0.03	0.08	0.30	0.90
Ant Colony Optimization	0.54	2.46	19.50	136.1
Hu - Eberhart - Shi Algorithm	1.95	-	-	-

3.2. Simulation Setup 2

In the second simulation setup, the effect of denseness/sparseness of the measurement/track matrix is investigated by varying the problem size. Convergence results taken from a random run representing the general trend observed in the experiments is shown in Figures 4 through 7. In the figures, the solid horizontal lines indicate the 100%, 95%, 90% and 80% benefit lines of the optimum global benefit value found by the auction algorithm. Speedy convergence to the top line by an optimization algorithm indicates that the algorithm produces the best solution in a fast manner. As it can be deduced from the figures, convergence speed of both 1-opt local search algorithm and the ant colony optimization is superior to both Hu-Eberhart-Shi and Clerc approaches for all problem sizes and regardless of the number of particles. Furthermore, it can be observed that performance of algorithms enhance with increasing the size of the swarm.

On the other hand, there are cases where all algorithms suffer from premature convergence and stagnation. Nevertheless, we are interested in creating a swarm of good solutions quickly. The parameter β in the ant colony optimization method was reduced in order to address the stagnation issue, likewise a probabilistic decision term allowing benefit decreasing swaps in the local search part of the 1-opt local search method may be introduced as a solution for the same problem. Unfortunately, it was observed that although such modifications helped converging to the optimum value in the long run, they showed slower converging speed and required significantly increased number of iterations, which might render them infeasible for tracking applications. Therefore, a greedier approach was undertaken for both algorithms.



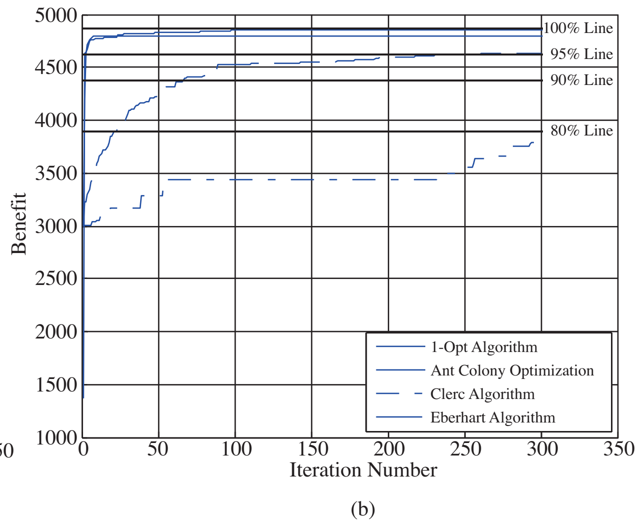
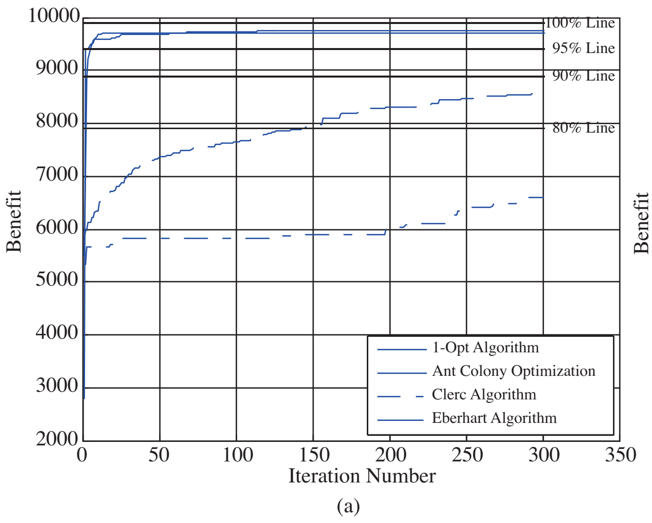


Figure 4. Algorithm performances for problem size 50. Results obtained for (a) 5 particles; (b) 15 particles.



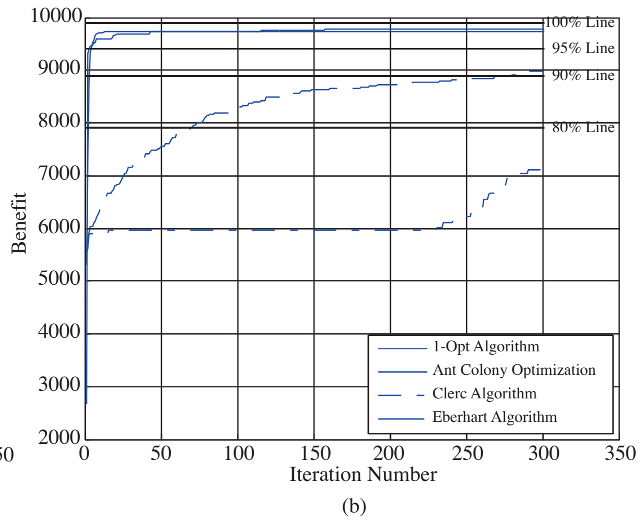
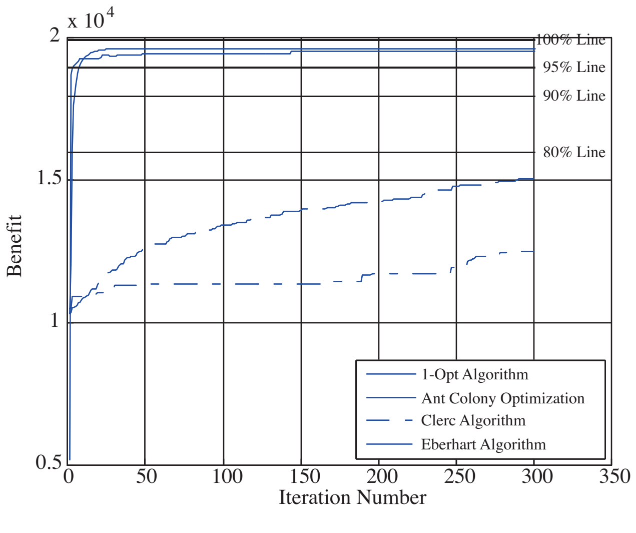


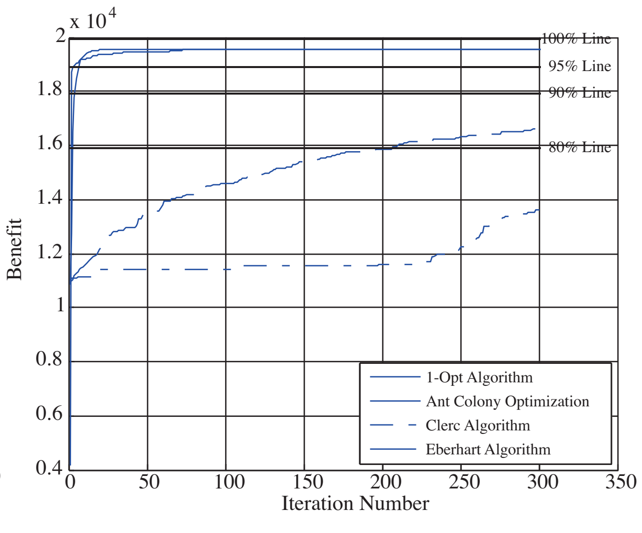
Figure 5. Algorithm performances for problem size 100. Results obtained for (a) 5 particles; (b) 15 particles.

4. Conclusions

In this paper, we tested several heuristic algorithms on an important target-tracking problem, the asymmetric assignment problem. Our experimental results indicate that although with some modification, nature inspired optimization algorithms provide improvement in terms of speed, they still fall behind the auction algorithm in

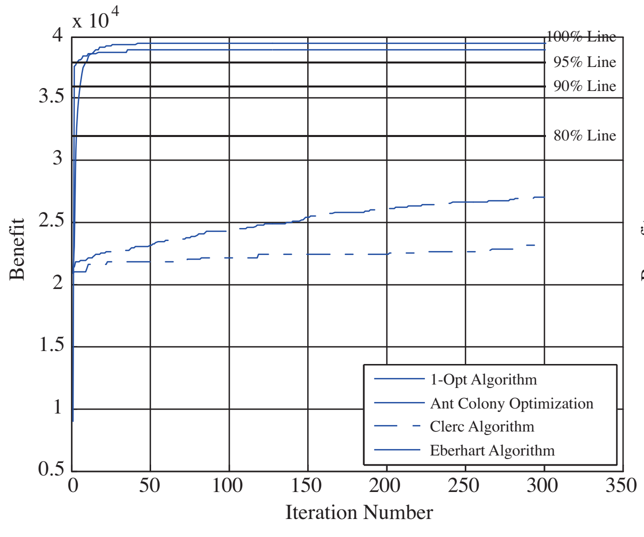


(a)

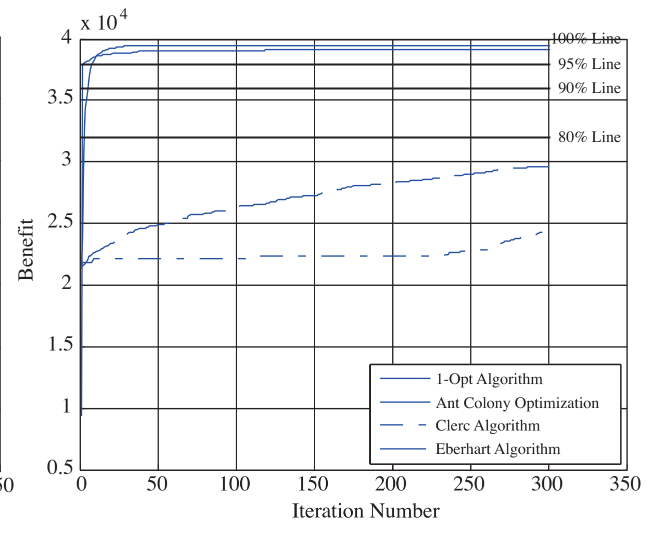


(b)

Figure 6. Algorithm performances for problem size 200. Results obtained for (a) 5 particles; (b) 15 particles.



(a)



(b)

Figure 7. Algorithm performances for problem size 400. Results obtained for (a) 5 particles; (b) 15 particles.

finding the optimum solution. An overall qualitative comparison and a summary of the results are given in Table 6. Among the heuristic approaches, the PSO algorithm using the proposed 1-opt local search has been found to perform better than other modifications in terms of convergence speed and quality of solutions found. Both Clerc and Hu-Eberhart-Shi approaches were found to be inappropriate for target tracking applications. These algorithms may not be able to provide adequate convergence response for the stringent timing requirements of real time tracking applications except only for the problems of very small dimensions. The reason of this poor performance is considered to be inefficient handling of information by both algorithms. In its each iteration, the auction algorithm uses only the best choice and the second best choice for each track. Information regarding competition for measurements between tracks is stored efficiently in a price matrix. Furthermore, tracks that were assigned to a measurement in the previous iterations do not enter the bidding phase unless challenged by other tracks. Therefore, as iterations proceeded, the dimension of the problem to be solved by the auction algorithm decreases. In contrast, the random search approach utilized by heuristic methods suffers from the increased dimension. Additionally, by their definition, no dimension reduction like the auction algorithm enjoyed, was applicable to the heuristics. Furthermore, for the Hu-Eberhart-Shi approach differentiating between two near track-measurement configurations may be very hard as the differentiating probability found by the velocity vector may be insignificant.

Table 6. Overall qualitative comparison of performances of the heuristic methods for the generalized assignment problem.

	Solution Accuracy	Speed	Timing Behavior	Overall Assessment
1-opt Local Search	Good	Fast	Linear	Suitable for target tracking applications
Ant Colony Optimization	Best	Slowest	Non Linear	Suitable for target tracking applications (but limited to non-real time applications only; due to low speed and non linear timing behavior)
Clerc Algorithm	Poor	Slow	Linear	Not suitable for target tracking applications due to poor solution accuracy
Hu - Eberhart - Shi Algorithm	Insufficient	Fastest	Linear	Not suitable for target tracking applications due to insufficient solution accuracy

In contrast, both 1-opt local search and ant colony optimization methods produced very promising outcomes. Both methods converged to over 95% of the optimal solution in just a few iterations; on the other hand, it should be noted that the execution time of the ant colony optimization is very long, and it demonstrates nonlinear timing behavior, making it inappropriate for real-time tracking applications.

Start Here Next Another point that should be emphasized is that the heuristic methods provide not only the best solution but also a set of good solutions. In terms of speed, considering that a group of good solutions being returned, 1-opt local search method consistently achieved results faster than the auction algorithm for each individual solution. Moreover, in terms of association accuracy, in an environment with high probability of false alarms and missed detections; for a given set of measurements at a fixed scan, the best solution to the assignment problem might be different than the exact association. In such a case, having the other good quality solutions will be useful for constructing new (and/or maintaining existing) association hypotheses for a multiple hypothesis target tracking application. Hence for hard assignment problems where multiple targets share a number of measurements that might be correlated, such as arising in tracking air targets flying in formation or ground targets moving in close proximity by a sensor with limited angle resolution, both approaches could be fruitful in providing a good set of candidate association hypotheses quickly. These hypotheses as shown in [4] might then be used in a multidimensional assignment algorithm which bases its assignment decisions on a multiple lists of measurements received in a sliding time window.

As a future work, we will test both approaches on a real target tracking simulation in a multiple dimensional assignment framework where auction algorithm's optimal 2D assignment may not be adequate in tracking targets accurately and resulting in track loss. Other probable future work areas regarding the performance issues might be as follows:

- the investigation of the effects of the parameter k in the 1-opt local search method;

- improvement of the performance of PSO for the generalized assignment problem—either by application of other discrete PSO techniques given in [11-21], or by tuning the PSO parameters in order to decrease instabilities and possible premature convergences.

References

[1] H. Wang, T. Kirubarajan, Y. Bar-Shalom, "Precision large scale air traffic surveillance using IMM/assignment estimators", IEEE Trans. Aerospace and Electronic Systems, Vol. 35, pp. 255 – 266, 1999.

[2] S. Deb, M. Yeddanapudi, K. Pattipati, Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation", IEEE Trans. Aerospace and Electronic Systems, vol. 33, pp. 523 – 538, 1999.

[3] R.L Popp, K.R. Pattipati, Y. Bar-Shalom, "m-best S-D assignment algorithm with application to multitarget tracking", IEEE Trans. Aerospace and Electronic Systems, Vol. 37, pp. 22–39, 2001.

[4] A. Sinha, T. Kirubarajan, "A randomized heuristic approach for multidimensional association in target tracking", in Proc. SPIE, Vol. 5428, pp. 202 – 210, 2004.

[5] J. Munkres, "Algorithms for the Assignment and Transportation Problems", Journal of the Society of Industrial and Applied Mathematics, Vol. 5, No. 1, pp. 32–38, 1957.

[6] D. P. Bertsekas, "An auction algorithm for shortest paths", SIAM Journal on Optimization, Vol. 1, pp. 425-447, 1991.

[7] R. Jonker and A. Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems", J. Computing, Vol. 38, pp. 325-340, 1987.

[8] Y. Bar-Shalom, Multitarget/Multisensor Tracking: Applications and Advances – Volume III, Artech House Publishers, Boston, 2000.

[9] J. Kennedy, R. Eberhart, "Particle swarm optimization", in Proc. IEEE Int. Conf. Neural Networks, Vol. 4, pp. 1942–1948, 1995.

[10] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Cambridge, MA, 2004.

[11] S. Salman, A. Imtiaz, S. Al-Madani, "Discrete particle swarm optimization for heterogeneous task assignment problem", in CD ROM Proc. World Multiconf. Syst. Cyb. & Inf. (SCI 2001), 2001.

[12] S. Salman, A. Imtiaz, S. Al-Madani, "Particle swarm optimization for task assignment problem", Microprocess. Microsyst., Vol. 26, No. 8, pp. 363–371, 2002.

[13] M. Clerc, "Particle Swarm Optimization, illustrated by the Traveling Salesman Problem", New Optimization Techniques in Engineering, Springer Verlag, pp. 219-239, 2004.

[14] K.P. Wang, L. Huang, C.G. Zhou, W. Pang, "Particle swarm optimization for traveling salesman problem", in: Proc. Int. Conf. Mach. Learning and Cybernetics, pp. 1583–1585, 2003.

[15] L. Cagnina, S. Esquivel, R. Gallard, "Particle swarm optimization for sequencing problem: a case study", in Proc. IEEE Conf. Evol. Comp., pp. 536–541, 2004.

[16] T.O. Ting, M.V.C. Rao, C.K. Loo, S.S. Ngu, "Solving unit commitment problem using hybrid particle swarm optimization", J. Heuristics, Vol. 9, No. 6, pp. 507–520, 2003.

[17] H.H. Balcı, J.F. Valenzuela, "Scheduling Electric Power Generators Using Particle Swarm Optimization Combined With The Lagrangian Relaxation Method", Int. J. Appl. Math. Comput. Sci., Vol. 14, No. 3, pp. 411–421, 2004.

[18] H. Liu, A. Abraham, J. Zhang, "A Particle Swarm Approach to Quadratic Assignment Problems", A. Saad et al. (Eds.): Soft Computing in Industrial Applications, ASC 39, pp. 213–222, 2007.

[19] T. Gong, A. L. Tuson, "Particle Swarm Optimization for Quadratic Assignment Problems – A Forma Analysis Approach", Int. J. Comp. Int. Rsrch (IJCIR), Vol. 4, No. 8, pp. 177-185, 2008.

[20] X. Hu, R.C. Eberhart, Y. Shi, "Swarm intelligence for permutation optimization: a case study of n-queens problem", in Proc. IEEE Swarm Int. Symp., pp. 243-246, 2003.

[21] A.W. Mohemmed, N.C. Sahoo, T.K. Geok, "Solving shortest path problem using particle swarm optimization", Appl. Soft Comput. J., Vol. 8, No. 4, pp. 1643-1653, 2008.

[22] F. van den Bergh, A.P. Engelbrecht, "A Cooperative approach to particle swarm optimization", IEEE Trans. Evolutionary Computation, Vol. 8, pp. 225 – 239, 2004.

[23] Y. Shi, R.C. Eberhart, "Parameter selection in particle swarm optimization", Evolutionary Programming VII, Lecture Notes in Computer Science, Vol. 1447, Springer, New York, pp. 591–600, 1998.

[24] R. C. Eberhart, P. Simpson, R. Dobbins, "Computational Intelligence, PC Tools", Academic Press International, San Diego, CA, 1996.

[25] T. Stützle, H.H. Hoos, "MAX-MIN Ant system", Future Generat. Comput. Syst., Vol. 16, pp. 889–914, 2000.

[26] C. Blum, "Ant colony optimization: Introduction and recent trends", Physics of Life Reviews, Vol. 2, No. 4, pp. 353-373, 2005.