

Fuzzy Particle Swarm Optimization with Simulated Annealing and Neighborhood Information Communication for Solving TSP

Rehab F. Abdel-Kader

Electrical Engineering Department
Faculty of Engineering, Port-Said University
Port Fouad 42523, Port-Said, Egypt

Abstract— In this paper, an effective hybrid algorithm based on Particle Swarm Optimization (PSO) is proposed for solving the Traveling Salesman Problem (TSP), which is a well-known NP-complete problem. The hybrid algorithm combines the high global search efficiency of fuzzy PSO with the powerful ability to avoid being trapped in local minimum. In the fuzzy PSO system, fuzzy matrices were used to represent the position and velocity of the particles in PSO and the operators in the original PSO position and velocity formulas were redefined. Two strategies were employed in the hybrid algorithm to strengthen the diversity of the particles and to speed up the convergence process. The first strategy is based on Neighborhood Information Communication (NIC) among the particles where a particle absorbs better historical experience of the neighboring particles. This strategy does not depend on the individual experience of the particles only, but also the neighbor sharing information of the current state. The second strategy is the use of Simulated Annealing (SA) which randomizes the search algorithm in a way that allows occasional alterations that worsen the solution in an attempt to increase the probability of escaping local optima. SA is used to slow down the degeneration of the PSO swarm and increase the swarm's diversity. In SA, a new solution in the neighborhood of the original one is generated by using a designed λ search method. A new solution with fitness worse than the original solution is accepted with a probability that gradually decreases at the late stages of the search process. The hybrid algorithm is examined using a set of benchmark problems from the TSPLIB with various sizes and levels of hardness. Comparative experiments were made between the proposed algorithm and regular fuzzy PSO, SA, and basic ACO. The computational results demonstrate the effectiveness of the proposed algorithm for TSP in terms of the obtained solution quality and convergence speed.

Keywords- Information Communication; Particle Swarm Optimization; Simulated Annealing; TSP.

I. INTRODUCTION

Traveling Salesman Problem (TSP) is a well-known NP-complete problem that has important practical applications as many complicated problems in various fields can be abstracted and changed to TSP [1-3]. The problem can be described as a single salesman who wants to visit a number of cities. The main objective of TSP is to find a minimal length closed tour that visits each city exactly once. TSP has been studied

extensively over the past several decades. Although it is simple to model the TSP mathematically, there is no definite algorithm that can be used to solve TSP in polynomial time. It is evident that the computational cost of TSP by exhaustive permutations is $O(n!)$. Today many bio-inspired intelligence techniques are rapidly developing such as Genetic Algorithms (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are used to solve combinatorial optimization problems such as TSP.

PSO is a computation paradigm based on group intelligent global optimization methods initially introduced by Kennedy and Eberhart in 1995 [4 - 6] to simulate the process of birds forage for food. PSO is different from other evolutionary algorithms, as it convergences quickly, has less parameters, encodes with real numbers and can directly deal with the problem domain, without conversion. PSO is simple, easy-to implement, and is widely used, in continuous function optimization problems where it yields excellent optimization performance. Recently, some researches apply this algorithm to problems of discrete quantities. However, the basic PSO algorithm suffers a serious problem that all particles are prone to be trapped into the local minimum in the later phase of convergence. The optimal value found is often a local minimum instead of a global minimum.

Aiming at solving the shortcoming of the basic PSO algorithm, many variations, such as Fuzzy PSO [3, 6], Hybrid PSO [7], Intelligent PSO [8], Niching PSO [9] and Guarantee Locally Convergent PSO [10] have been proposed to increase the diversity of particles and improve the convergence performance. In this paper, a new algorithm that combines the fuzzy PSO algorithm with Neighborhood Information Communication (NIC) strategy and Simulated Annealing (SA) was proposed and applied to solve the TSP. The NIS strategy incorporates the shared information provided by the individual's neighborhood into the PSO update equations. SA is a kind of stochastic method that is well known for its effective capability of escaping local optima. By integrating NIC and SA to the fuzzy PSO, the new algorithm, which we call it PSO-NIC-SA can not only escape from local minimum trap in the later phases of convergence, but also simplify the implementation of the algorithm. In the experiments, three additional algorithms: fuzzy PSO, SA, and ACO have been

implemented and the results have been compared to the results of the proposed algorithm. Test results demonstrate that the PSO-NIC-SA algorithm outperforms the other algorithms in solving TSP. The rest of this paper is organized as follows: The basic particle swarm optimization algorithm is presented in Section II. The mathematical model for TSP and the Fuzzy PSO algorithm for solving the TSP are described in Section III. In Section IV, the proposed PSO-NIC-SA algorithm for TSP is explained. The experimental results are reported in Section V. Finally, we summarize the paper with some concluding remarks in Section VI.

II. PARTICLE SWARM OPTIMIZATION

PSO proposed by Dr. Eberhart and Dr. Kennedy in 1995 is a computational technique based on the idea of collaborative behavior and swarming in biological populations inspired by the social behavior of bird flocking or fish schooling [4 - 6].

The algorithm, which is based on a metaphor of social interaction, searches a space by adjusting the trajectories of individual vectors, called "particles" as they are conceptualized as moving points in multidimensional space. The individual particles are drawn stochastically toward the position of their own previous best performance and the best global performance among its neighbors.

The PSO algorithm is simple, easy to implement, robust to control parameters, and computationally efficient compared to other heuristic optimization techniques. The original PSO has been applied to a learning problem of neural networks and function optimization problems, and the efficiency of the method has been confirmed.

When PSO is used to solve an optimization problem, a swarm of particles, is used to explore the solution space for an optimum solution. Each particle represents a candidate solution and is identified with specific coordinates in the D -dimensional search space. The position of the i 'th particle is represented as $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. The velocity of a particle is denoted as $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$. The fitness function is evaluated for each particle in the swarm and is compared to the fitness of the best previous result for that particle and to the fitness of the best particle among all particles in the swarm. After finding the two best values, the particles evolve by updating their velocities and positions according to the following equations:

$$V_i^{t+1} = \omega * V_i^t + c_1 * rand_1 * (p_{i_best} - X_i^t) + c_2 * rand_2 * (g_{best} - X_i^t) \quad (1)$$

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

Where $i = (1, 2, \dots, Pop_Size)$ and Pop_Size is the size of the swarm; p_{i_best} is the particle best reached solution and g_{best} is the global best solution in the swarm. c_1 and c_2 are cognitive and social parameters that are bounded between 0 and 2. $rand_1$ and $rand_2$ are two random numbers, with uniform distribution $U[0,1]$. $-V_{max} \leq V_i^{t+1} \leq V_{max}$ (V_{max} is the maximum velocity).

The inertia weight ω is a factor used to control the balance of the search algorithm between exploration and exploitation. The recursive steps will go on until we reach the termination condition.

III. TRAVELING SALESMAN PROBLEM

A. Mathematical Model of TSP

The TSP can be described as follows: In the graph $G = (X, E)$, X is the set of nodes, or cities to be visited, E is the set of edges, $E = \{(x_i, x_j): x_i, x_j \in X\}$. The objective of TSP is to find a minimal length closed tour that visits a number of cities ' n ' such that each city is visited exactly once. This closed tour is called the *Hamiltonian cycle*. When the total distance traveled is the main metric for evaluating the cost then the problem of finding the best path S is the same as the minimization of the target fitness function $F(S)$ defined as follows:

$$F(S) = \sum_{i=1}^{n-1} d(x_i, x_{i+1}) + d(x_n, x_1) \quad (3)$$

Where $d(x_i, x_{i+1})$, $0 \leq i \leq n-1$ is the *Euclidean* distance from city x_i to city x_{i+1} . If for all pairs of nodes x_i, x_j , the distances $d(x_i, x_j)$ and $d(x_j, x_i)$ are equal then the problem is said to be *symmetric*, otherwise it is said to be *asymmetric*. In the TSP the solution space increases rapidly as the total number of cities increases. For example, with number of cities $n=100$, the total number of possible solution paths will be $4.67 * 10^{155}$. Tremendous research has focused on this research problem due to its significance both in theory and applications.

B. Fuzzy Matrix to Represent TSP Solution

The Fuzzy Discrete PSO (FD-PSO) was first introduced by Wei Pang et al. [1, 2] to solve the TSP, and subsequently used and improved in [11] and was found to achieve satisfactory results.

1) Construction of Fuzzy Matrix

Assume the solution of TSP is $T = \{(x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n), (x_n, x_1)\}$, where n is the number of cities, x_i ($i = 1, \dots, n$) is the i -th visited city (node) in this solution, and $(x_1, x_2), (x_2, x_3), \dots, (x_{n-1}, x_n), (x_n, x_1)$ are the visited directed edges in turn. Then fuzzy position matrix can be represented as an $n \times n$ matrix P as follows:

$$P = \begin{pmatrix} p_{11} & \dots & p_{1n} \\ \cdot & & \cdot \\ p_{n1} & \dots & p_{nn} \end{pmatrix} \quad (4)$$

Where $p_{ij} \in [0, 1]$ means the possibility of choosing directed edge (x_i, x_j) , after city x_i has been chosen in the TSP solution. In order to avoid directed edge (x_i, x_i) for $(i = 1, \dots, n)$ appearing in TSP solution, we need to set the elements in diagonal of fuzzy matrix to very small values, thus let p_{ii} ($i = 1, \dots, n$) = $-Max$.

The velocity of the particle is defined as:

$$V = \begin{pmatrix} v_{11} & \dots & v_{1n} \\ \cdot & & \cdot \\ v_{n1} & \dots & v_{nn} \end{pmatrix} \quad (5)$$

2) Fuzzy PSO operators

The original PSO position and velocity equations (1) and (2) should be redefined to the form of matrices. The symbol " \otimes " was used to denote the modified multiplication operator which represents a scalar matrix multiplication where all

elements in the matrix are multiplied by the scalar. The symbols " \oplus " and " \ominus " denote the addition and subtraction between matrices respectively. The modified position and velocity vectors can be rewritten as follows:
 $V_i^{t+1} = \omega * V_i^t \oplus (c_1 * rand_1) \otimes (p_{i_best} \ominus X_i^t) \oplus (c_2 * rand_2) \otimes (g_{best} \ominus X_i^t)$ (6)

$$X_i^{t+1} = X_i^t \oplus V_i^{t+1} \quad (7)$$

3) Initialization

The elements of the position matrix P given in (4) are randomly generated subject to the following constraints:

$$\sum_{j=1}^n p_{ij} = 1, \text{ for } i = \{1, 2, \dots, n\} \quad (8)$$

$$P_{ij} \in [0, 1] \quad (9)$$

Similarly, the elements of the velocity matrix V given in equation (5) are randomly generated subject to the following constraint:

$$\sum_{j=1}^n v_{ij} = 0 \text{ for } i = \{1, 2, \dots, n\} \quad (10)$$

4) Normalization of the Position Matrix

The position matrix P may violate the constraint given in (9) after the update of the position matrix in the forthcoming generations. Thus, it is necessary to normalize the position matrix. First, all negative elements in P are converted to 0. The other elements in P are recalculated according to the following transformation:

$$P = \begin{pmatrix} p_{11} / \sum_{i=1}^n p_{1i} & \dots & p_{1n} / \sum_{i=1}^n p_{1i} \\ \vdots & & \vdots \\ p_{n1} / \sum_{i=1}^n p_{ni} & \dots & p_{nn} / \sum_{i=1}^n p_{ni} \end{pmatrix} \quad (11)$$

5) Defuzzification

The fuzzy position matrix P represents the potential solution for the TSP. However, matrix P has to be decoded in order to obtain the feasible solution representing the best route (the ordered sequence of nodes or cities visited). This procedure is called "Defuzzification". In this paper the global maximum method is used for the Defuzzification of the position matrix. In this method we have a flag array to record whether we have selected the columns of the matrix and a route array to record the route solution, first all the columns are not selected, then for each row of the matrix, we choose the element which is not selected by the previous rows and has the max value, then we mark the column of the max element "selected", and the column number are recorded to the route array. After all the rows have been processed, we get the route solution from the route array and the cost of the TSP route is calculated according to (3).

6) Fuzzy PSO Algorithm for TSP

The pseudo code of the fuzzy PSO algorithm for solving the TSP is presented in Fig. 1.

IV. HYBRID ALGORITHM FOR TSP

PSO algorithm is problem-independent, which means little specific knowledge relevant to the given problem is required. This advantage makes PSO more robust than many other search algorithms. However, as a stochastic search algorithm, PSO is prone to lack global search ability at the end of its run. PSO may fail to find the required optima as it can easily get trapped into local optima in complex optimization problems. Two strategies were proposed in the hybrid algorithm to strengthen the diversity of the particles and to speed up the convergence process. The first strategy is based on NIC among the particles where a particle absorbs better historical experience of the neighboring particles. The second strategy is the use of SA which randomizes the search algorithm in a way that allows occasional alterations that worsen the solution in an attempt to increase the probability of escaping local optima. The NIC and the SA strategies are explained in more detail in the following Sections.

A. Neighborhood Information Communication

Biological results suggest that information sharing among neighboring individuals contributes to evolution as the current state of neighbors significantly impact on the decision process of group members. However, PSO, as a simulation of group foraging behavior, does not include any neighborhood information sharing into its evolutionary equations. In traditional PSO, the global best solution g_{best} is the only information shared among the particles of the swarm. In this paper, an information sharing strategy among the particles is utilized in the proposed hybrid algorithm. In the renewing process of the position and velocity matrices, a particle absorbs better historical experience of the neighboring particles with better fitness values than its own. Better particles will guide the other particles to improve their fitness. This results in a very small probability to be trapped by local optima. All particles in the swarm will be ranked according to their fitness values. In minimization problems, the particle with the smallest fitness value will be ranked 1 and similarly for all other particles. Each particle shows interest in other particles according to their rank. The modified velocity matrix can be rewritten as follows:

$$V_i^{t+1} = \omega * V_i^t \oplus \left(\frac{1}{rank(i)} * rand_1 \right) \otimes (p_{i_best} \ominus X_i^t) + rand_2 \otimes (g_{best} \ominus X_i^t) + social_info(i) \quad (12)$$

Where $social_info(i)$ is calculated as shown in Fig. 2. The $social_info()$ module gives the direction of the swarm by sharing information with all other individuals that have better fitness values. V_{max} has been set to small values to prevent abrupt variations in the solution domain.

```
Step1: initialization
1.1. Initialize the swarm size to Pop Size and the maximum number of
iterations Max iterations
1.2. Initialize the position and velocity matrices for all particles in the
swarm.
1.3. Evaluate the fitness of each particle
1.4. Initialize the local best for every particle  $P_i = X_i^0$  and the global best
 $g_{best}$  is the best among all  $P_i$ 
Step 2: if Current iteration equals Max iterations goto step 5.
Step 3: for i:=0 to Pop Size -1
3.1. Current iteration= Current iteration +1
3.2. Calculate the new velocity matrices according to equation (6)
3.3. Calculate the new position matrices according to equation (7)
3.4. Normalize the position matrices according to equation (11)
3.5. Defuzzy the new position matrices and calculate the cost for each
position matrix using(3).
3.6. If the cost of the new position of a particle is less than that of the
local best of the particle then update the local best position with
new position
Step 4: If the cost of the local best of some particles is less than the  $g_{best}$ , then
update the  $g_{best}$  with the local best particle. Goto step 2
Step 5: output  $g_{best}$  route and its corresponding cost
```

Figure 1. Fuzzy PSO Algorithm

```
social info(i)
{
    social effect =0;
    for k=1: Pop Size
        if F(k)<F(i)
            social_effect= social_effect + (1/rank(k)*rand) * (X_k' * X_i')
        if (social effect > V_max)
            return V_max
        else
            return social effect }
}
```

Figure 2. Neighborhood Information Communication Algorithm.

B. SA-based Local Search for TSP

SA is an intelligent stochastic strategy used in solving optimization problems. It was successfully applied to the optimization problems by Kirkpatrick [3, 12]. SA employs certain probability to avoid becoming trapped in a local optimum by allowing occasional alterations that increases the diversity of the particles in the swarm. In the search process, the SA accepts not only better but also worse neighboring solutions with a certain probability. Such mechanism can be regarded as a trial to explore new space for new solutions, either better or worse. The probability of accepting a worse solution is larger at higher initial temperature. As the temperature decreases, the probability of accepting worse solutions gradually approaches zero. More specifically, starting from an initial state, the system is perturbed at random to a new state in the neighborhood of the original one. Then the change ΔE of the fitness function value is calculated. For minimization problems, the new state is accepted with probability $\min\{1, \exp(-\Delta E / T)\}$, where T is a control parameter corresponding to the temperature in the analogy. The SA algorithm generally starts from a high temperature, and then the temperature is gradually lowered. At each temperature, a search is carried out for a certain number of

iterations. The above technique can increase the diversity in the particles and enable PSO to accept a bad solution with a probability that will gradually decrease to zero as the temperature decreases. In this paper a simple λ search method is designed for generating the SA neighborhood solutions, where λ is a parameter representing the depth of the local search. The λ search method includes two steps:

1. Swap the order of a pair of randomly selected rows in the position matrix
2. Perform the matrix normalization transformation according to (11)

This process is repeated for λ randomly selected pairs of rows and λ new solutions are produced. The best solution among the λ generated solutions is selected as the newly produced solution.

C. Hybrid PSO-NIC-SA Algorithm

The objective of TSP is to minimize the fitness function given in (3) that represents the cost of a particular route. Combining the fast optimal search ability of fuzzy PSO with the Neighboring Information Communication model and the probability jump property of SA, we design a new algorithm

```
Step1: initialization
  1.1. Initialize the swarm size to Pop Size and the maximum number of
        iterations Max iterations
  1.2. Initialize the position and velocity matrices for all particles in the swarm.
  1.3. Evaluate the fitness of each particle
  1.4. Initialize the local best for every particle  $P_i = X_i^0$  and the global best  $g_{best}$ 
        is the best among all particles in the swarm.
  1.5 initialize  $c_1, c_2$ , Current iteration=0
  1.6 set Simulating annealing parameters: Initial temperature  $T_n$ , Final
        temperature  $T_{end}$  and rate of cooling  $\beta$ , indicator  $m=0$ ; current
        temperature  $T = T_n$ 
Step 2: if Current iteration equals Max iterations goto step 5.
Step 3: for i:=0 to Pop Size -1
  3.1. Current iteration= Current iteration +1;
  3.2. Calculate the new velocity matrices according to equation (6)
  3.3. Calculate the new position matrices according to equation (7)
  3.4. Normalize the position matrices according to equation (11)
  3.5. Defuzzy the new position matrices and calculate the cost for each
        position matrix using(3).
  3.6. if the cost of the new position of a particle is less than that of the
        local best of the particle then update the local best position with
        new position
Step 4: if the cost of the local best of some particles is less than the  $g_{best}$ 
        position then update the  $g_{best}$  with the local best particle.
        if ( $g_{best}$  does not improve)
            m ++;
            if (m= gen){
                while ( T > Tend ){
                    Generate a neighbor solution  $g'$ best from  $g_{best}$ ;
                    Calculate fitness of  $g'$ best;
                    Calculate  $\Delta E = F(g'best) - F(g_{best})$ ;
                    if ( $\min[1, \exp(-\Delta E / T)] > \text{random}[0,1]$ )
                         $g_{best} = g'best$ ;
                         $T = \beta T$ ;}
                m=0;}
            Goto step 2
Step 5: output global best route and its corresponding cost
```

Figure 3. Hybrid PSO-NIC-SA Algorithm

PSO-NIC-SA to solve TSP problem. The pseudo code of the PSO-NIC-SA algorithm for solving the TSP is shown in Fig. 3.

In the proposed algorithm, fuzzy PSO is performed first using the position and velocity update equations that incorporates the neighborhood information communication strategy. If the best particle g_{best} does not improve for a specified number of generations gen , then SA is used. The best individual from the fuzzy PSO algorithm provides the initial solution for SA during the hybrid search process. The hybrid algorithm implements easily and reserves the generality of PSO and SA. Moreover, it can be applied to many combinatorial optimization problems by simple modification.

V. EXPERIMENTAL RESULTS

The proposed algorithm has been implemented using MATLAB 7.1 and executed on a Pentium IV, 2.8 GHz computer. To evaluate the efficiency of the proposed algorithm and for comparison purpose, three other artificial algorithms including fuzzy PSO, basic SA, and basic ACO have been implemented, and applied to the same TSP problems. Three TSP benchmark problems (Burma14, Berlin52, Eil75) selected from the TSPLIB [13] were tested. The parameters used in PSO, NIC and SA were determined

through the preliminary experiments. The following parameter setting was used in the proposed algorithm: the swarm size $Pop_Size = 50$, $Max_iterations = 1000$, initial temperature $T_0 = 1000$, final temperature $T_{end} = 1$, rate of cooling $\beta = 0.99$ and the SA search parameter $\lambda = 5$. All algorithms were run 20 times for all the TSP problems. The results shown in Table I represent the mean and best solutions found by running the various trails of each method.

From the results of the four algorithms, it is clear that our algorithm is significantly better than the other algorithms. For example, in Burma14 problem, the best fitness value 30.87 achieved by the hybrid PSO-NIC-SA algorithm is not only the smallest value among the four algorithms but also the optimal solution for the problem. Meanwhile the mean value case, performance is also greatly improved by the new algorithm. It is shown that the new algorithm is a better and more effective means to solve TSP problem. Fig. 4 presents the convergence speed for the various algorithms in the Burma14 benchmark. The Figure shows the mean fitness found versus the number of iterations in the four implemented algorithms. As can be seen from Fig. 4, the proposed PSO-NIC-SA algorithm was able to reach good solutions faster than other methods in the early stages of the search process, and reach better solutions than others at the end of the search process.

TABLE I. COMPARISONS OF FUZZY PSO, SA, ACO AND PSO-NIC-SA FOR BURMA14, BERLIN52, EIL75 BENCHMARKS

TSP Problem	Optimal Solution	Fuzzy PSO			SA			ACO			Hybrid PSO-NIC-SA		
		Mean	Best	Time (ms)	Mean	Best	Time (ms)	Mean	Best	Time (ms)	Mean	Best	Time (ms)
Burma14	30.87	34.6	30.87	444	34.98	33.56	199	30.95	30.87	107	30.87	30.87	354
Berlin52	7542	7645	7543	8976	7331	7389	5890	7601.5	7542	5432	7593.8	7542	6766
Eil75	542.37	593.83	587.65	13543	598.76	580	10250	544.7	542.37	10876	543.9	542.37	11240

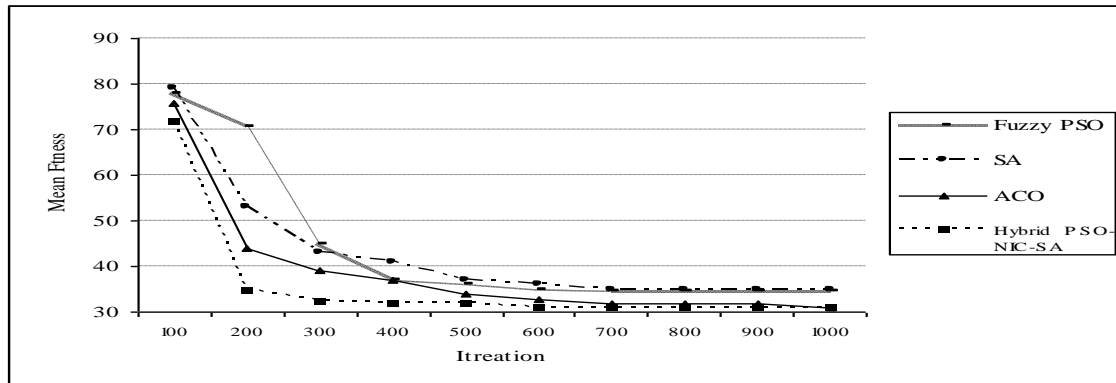


Figure 4. Convergence Curves for the Different Algorithms in Burma14.

VI. CONCLUSIONS

In this paper, an effective hybrid algorithm based on fuzzy PSO is proposed for solving the TSP, which is a well-known NP-complete problem. The proposed algorithm combines the high global search efficiency of fuzzy PSO with the powerful ability to avoid being trapped in local minimum. In the fuzzy PSO system, fuzzy matrices were used to represent the position and velocity of the particles in PSO and the operators in the original PSO position and velocity formulas were redefined.

Two strategies were employed in the hybrid algorithm to strengthen the diversity of the particles and to speed up the convergence process. The first strategy is based on neighborhood information communication among the particles where a particle absorbs better historical experience of the neighboring particles. This strategy integrates the individual experience of the particles with the neighbor sharing information of the current state.

The second strategy is the use of simulated annealing which randomizes the search algorithm in a way that allows occasional alterations that worsen the solution in an attempt to increase the probability of escaping local optima. SA is used to slow down the degeneration of the PSO swarm and increase the swarm's diversity. In SA, a new solution in the neighborhood of the original one is generated by using a designed λ search method. A new solution with fitness worse than the original solution is accepted with a probability that gradually decreases at the late stages of the search process. The hybrid algorithm is examined using a set of benchmark problems from TSPLIB with various sizes and levels of hardness.

Comparative experiments were made between the proposed algorithm and PSO, SA, and basic ACO. The computational results validate the effectiveness of the proposed approach.

REFERENCES

- [1] W. Pang, K.-P. Wang, C.-G. Zhou, and L.-J. Dong, "Fuzzy Discrete Particle Swarm Optimization for Solving Traveling Salesman Problem," In Proceedings of the Fourth International Conference on Computer and Information Technology, pp. 796-800, 2004.
- [2] W. Pang, K.-P. Wang, C.-G. Zhou, and L.-J. Dong "Modified Particle Swarm Optimization Based on Space Transformation for Solving Traveling Salesman Problem," In Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, pp. 2342-2346, 2004.
- [3] L. Fang, P. Chen, and S. Liu, "Particle Swarm Optimization with Simulated Annealing for TSP," In Proceedings of 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Databases, pp. 206-209, 2007.
- [4] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," In Proceedings of IEEE International Conf. on Neural Networks, Piscataway, NJ, Vol. 4: pp. 1942-1948, 1995.
- [5] R. Eberhart and Y. Shi, "Comparison Between Genetic Algorithms and Particle Swarm Optimization," In e. a. V. William Porto, editor, Evolutionary Programming, Lecture Notes in Computer Science, Vol. 1447, pp. 611-616. Springer, 1998.
- [6] Y. Shi and R. Eberhart, "Fuzzy Adaptive Particle Swarm Optimization," In Proceedings of Congress on Evolutionary Computation, pp.101-106, 2001.
- [7] M. Lovbjerg, T. Rasmussen, and T. Krink, "Hybrid Particle Swarm Optimizer with Breeding and Subpopulation," In Proceedings of Evolutionary Computation Conference, 2001.
- [8] G. Ciuprina, D. Ioan, and I. Munteanu, "Use of Intelligent-Particle Swarm Optimization in Electromagnetic," IEEE Transactions on Magnetics, Vol. 38, No. 2, pp. 1037-1040, 2002.
- [9] R. Brits, A. Engelbrecht, and A. Niching, "Particle Swarm Optimizer," In Proceedings of 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002.
- [10] E. Bergh and A. Engelbrecht, "A New Locally Convergent Particle Swarm Optimizer," In Proceedings of IEEE Conference on Systems, Man, and Cybernetics, 2002.
- [11] B. Shen, M. Yao, and W. Yi, "Heuristic Information Based Improved Fuzzy Discrete PSO Method for Solving TSP," Lecture Notes in Artificial Intelligence, Vol. 4099, Springer, Berlin, pp. 859-863, 2006.

- [12] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science Journal*, Vol. 220, No. 4598, pp. 671-680, 1983.
- [13] TSPLIB: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsplib.html>
- [14] Mohsen, F. M. A. (2011). A new Optimization-Based Image Segmentation method By Particle Swarm Optimization. *International Journal of Advanced Computer Science and Applications - IJACSA*, (Special Issue), 10-18.
- [15] Trivedi, J. A. (2011). Framework for Automatic Development of Type 2 Fuzzy , Neuro and Neuro-Fuzzy Systems. *International Journal of Advanced Computer Science and Applications - IJACSA*, 2(1), 131-137.
- [16] Khan, A. R., Rehman, Z. U., & Amin, H. U. (2011). Application of Expert System with Fuzzy Logic in Teachers ' Performance Evaluation. *International Journal of Advanced Computer Science and Applications -*

IJACSA, 2(2), 51-57.

AUTHOR PROFILE

Rehab Farouk Abdel-Kader, attended Suez Canal University, Port-Said, Egypt majoring in Computer Engineering, earning the BS degree in 1996. She graduated from Tuskegee University, Tuskegee, Alabama with a MS degree in Electrical Engineering in 1999. She joined the Ph.D. program at Auburn University and earned her Doctor of Philosophy degree in 2003. She worked as an assistant Professor in the Engineering Studies Program in Georgia Southern University, Statesboro, Georgia from 2003 to 2005. She is currently an assistant professor in the Electrical Engineering department, Faculty of Engineering at Port-Said, Port-Said University, Port-Said, Egypt. Her current research interests include Signal Processing, Artificial Intelligence, Pattern Recognition, and Computer Vision