Solve Traveling Salesman Problem Using Particle Swarm Optimization Algorithm

Xuesong Yan[1], Can Zhang[1] , Wenjing Luo[1], Wei Li[1], Wei Chen[1] and Hanmin Liu[2]

**[1] School of Computer Science, China University of Geosciences
Wuhan, Hubei 430074, China**

**[2] Wuhan Institute of Ship Building Technology
Wuhan, Hubei 430050, China**

Abstract

The traveling salesman problem (TSP) is one of the most widely studied NP-hard combinatorial optimization problems and traditional genetic algorithm trapped into the local minimum easily for solving this problem. Particle Swarm Optimization (PSO) algorithm was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities. Compared with the genetic algorithm, PSO algorithm has high convergence speed. In this paper, aim at the disadvantages of genetic algorithm like being trapped easily into a local optimum, we use the PSO algorithm to solve the TSP and the experiment results show the new algorithm is effective for the this problem.

Keywords: *Traveling Salesman Problem, Particle Swarm Optimization, Population, Global Optimal.*

1. Introduction

The traveling salesman problem (TSP) [1] is one of the most widely studied NP-hard combinatorial optimization problems. Its statement is deceptively simple, and yet it remains one of the most challenging problems in Operational Research. The simple description of TSP is: Give a shortest path that covers all cities along. Let $G=(V;E)$ be a graph where V is a set of vertices and E is a set of edges. Let $C=(c_{ij})$ be a distance (or cost) matrix associated with E. The TSP requires determination of a minimum distance circuit (Hamiltonian circuit or cycle) passing through each vertex once and only once. C is said to satisfy the triangle inequality if and only if $c_{ij}+c_{jk}\ge c_{ik}$ for $i,j,k\in V$.

Due to its simple description and wide application in real practice such as Path Problem, Routing Problem and Distribution Problem, it has attracted researchers of various domains to work for its better solutions. Those traditional algorithms such as Cupidity Algorithm, Dynamic Programming Algorithm, are all facing the same obstacle, which is when the problem scale N reaches to a certain degree, the so-called “Combination Explosion” will occur. For example, if $N=50$, then it will take 5×10^{48} years under a super mainframe executing 100 million instructions per second to reach its approximate best solution.

A lot of algorithms have been proposed to solve TSP [2-7]. Some of them (based on dynamic programming or branch and bound methods) provide the global optimum solution. Other algorithms are heuristic ones, which are much faster, but they do not guarantee the optimal solutions. There are well known algorithms based on 2-opt or 3-opt change operators, Lin-Kerninghan algorithm (variable change) as well algorithms based on greedy principles (nearest neighbor, spanning tree, etc). The TSP was also approached by various modern heuristic methods, like simulated annealing, evolutionary algorithms and tabu search, even neural networks.

Particle Swarm Optimization (PSO) algorithm was an intelligent technology first presented in 1995 by Eberhart and Kennedy, and it was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities [8]. If we compare PSO with Genetic Algorithms (GAs), we may find that they are all maneuvered on the basis of population operated. But PSO doesn't rely on genetic operators like selection operators, crossover operators and mutation operators to operate individual, it optimizes the population through information exchange among individuals. PSO achieves its optimum solution by starting from a group of random solution and then searching repeatedly. Once PSO was presented, it invited widespread concerns among scholars in the optimization fields and shortly afterwards it had become a studying focus within only several years. A number of scientific achievements had emerged in these fields [9-11]. PSO was proved to be a sort of high efficient optimization algorithm by numerous research and experiments [12]. PSO is a meta-heuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-Newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc. This paper improves the disadvantages of standard PSO being easily trapped into a local optimum and proposed a new algorithm which proves to be more simply conducted and with more efficient global searching capability, then use the new algorithm for engineering optimization field.

2. Related Work

In recent years, since the TSP is a good ground for testing optimization techniques, many researchers in various fields such as artificial intelligence, biology, mathematics, physics, and operations research devote themselves to trying to find the efficient methods for solving the TSP, such as genetic algorithms (GAs) [13], ant colony optimization (ACO) [14], simulated annealing (SA) [15], neural networks (NN) [16], particle swarm optimization (PSO) [17], evolutionary algorithms (EA) [18], memetic computing [19], etc. Besides, there are many practical applications of the TSP in the real world [20, 21], such as data association, vehicle routing (with the additional constraints of vehicle’s route, such as capacity’s vehicles), data transmission in computer networks, job scheduling, DNA sequencing, drilling of printed circuits boards, clustering of data arrays, image processing and pattern recognition, analysis of the structure of crystals, transportation and logistics.

Swarm intelligence is an important research topic based on the collective behavior of decentralized and self-organized systems in computational intelligence. It consists of a population which simulates the animals’ behavior in the real world. Now there are many swarm intelligence optimization algorithms, such as genetic algorithms, particle swarm optimization, ant colony optimization, bee colony algorithm, differential evolution, fish-warm algorithm, etc. Due to the simple concept, easy implementation and quick convergence, PSO has gained much attention and been successfully applied in a variety of fields mainly for optimization problems.

So far, as for the constrained optimization problems, relatively less work based on PSO can be found than those based on other EAs. Parsopoulos and Vrahatis [22] proposed a non-stationary multi-stage assignment penalty function method to transform the constrained problem to the unconstrained problem. Simulation results showed that PSO outperformed other EAs, but the design of the multi-stage assignment penalty function is too complex. In the work of Hu and Eberhart [23], the initial swarm contains only feasible solutions and a strategy to preserve feasibility is employed. Motivated by multi-objective optimization techniques, Ray and Liew [24] proposed a swarm algorithm with a multilevel information sharing strategy to deal with constraints. In their work, a better performer list (BPL) is generated by a multilevel Pareto ranking scheme treating every constraint as an objective, while the particle which is not in the BPL gradually congregates round its closest neighbor in the BPL.

In these swarm intelligence algorithms, the most popular and widely used algorithms for solving the TSP are GAs, ACO and PSO. Clerc [25] develops several algorithm variants with those operations and methods and applies them to the asymmetric TSP instance br17.atsp. In his algorithm the positions are defined as TSP tours represented in vectors of permutations of the |N| vertices of the graph correspondent to the considered instance. This approach was applied to tackle the real problem of finding out the best path for drilling operations [27].

Particle swarm optimization is presented to solve traveling salesman problem [26], where the authors have proposed the concept of swap operator and swap sequence, and redefined some operators on the basis of them. This paper has designed a special PSO, but the special PSO does not improve the updating formula, and the experiment results are worse than ours.

Hendtlass [28] proposes the inclusion of a memory for the particles in order to improve diversity. The memory of each particle is a list of solutions (target points) that can be used as an alternative for the current local optimal point. There is a probability of choosing one of the points of the particle’s memory instead of the current best point of the whole swarm P_{gdb}. The size of the memory list and the probability are new parameters added to the standard PSO algorithm. The algorithm is applied to the benchmark TSP instance burma14. The results obtained with algorithmic versions with several parameter settings are compared with the results of an Ant Colony Optimization algorithm.

Pang et al. [29] extends the work of Wang et al. [26]. Their algorithm alternates among the continuous and the discrete (permutation) space. In order to avoid premature convergence, Pang et al. [29] use a chaotic operator. This operator changes randomly the position and velocity in the continuous space, multiplying these vectors by a random number. Four versions of their algorithm are applied to four benchmark instances with 14 to 51 cities: burma14, eil51, eil76 and berlin52. The algorithm variations comprise the presence or not of chaotic variables and the two local search procedures. In the set of instances tested, the results showed that the version that includes chaotic variables and the 2-opt local search presented the best results. Pang et al. [30] present a fuzzy-based PSO algorithm for the TSP. They apply their algorithm to instances burma14 and berlin52. No average results or comparisons with other algorithms are reported.

A hybrid approach that joins PSO, Genetic Algorithms and Fast Local Search is presented by Machado & Lopes [31] for the TSP. The positions of the particles represent TSP tours as permutations of |N| cities. The value assigned to each particle (fitness) is the rate between a constant Dmin and the cost of the tour represented in the particle's position. The hybrid PSO is applied to the following symmetric TSP benchmark instances: pr76, rat195, pr299, pr439, d657, pr1002, d1291, rl1304, d2103.

Goldbarg et al. [32] present a PSO algorithm for the TSP where the idea of distinct velocity operators is introduced. The velocity operators are defined according to the possible movements a particle is allowed to do. This algorithmic proposal obtained very promising results. It was applied to 35 benchmark TSP instances with 51 to 7397 cities. The results were comparable to the results of state-of-the-art algorithms for the TSP.

Yuan et al. [33] propose new concepts for “chaos variables” and memory for particles. The memory of each particle is a |N|-dimensional vector of chaos variables. The chaos variables are numbers in the interval (0, 1) and are generated with a method proposed by the authors. They apply their algorithm to four benchmark instances with 14 to 51 cities: burma14, oliver30, att48, eil51. The results obtained for instances oliver30 and att48 are compared with the results obtained by algorithms based on: Simulated Annealing, Genetic Algorithm and Ant Colony Systems. Their algorithm outperforms the others regarding quality of solution of these two instances.

Fang et al. [34] present a PSO algorithm for the TSP where an annealing scheme is used to accept the movement of a particle. They apply their algorithm to instances oliver30 and att48. The results are compared with the results of algorithms based on: Simulated Annealing, Genetic Algorithms and Ant Colony. In the two instances tested, their algorithm presents the best average results.

Preserving diversity in particle swarm optimization is used to solve traveling salesman problem [35]. This paper showed that by adding a memory capacity to each particle in a PSO algorithm performance can be significantly improved to a competitive level to ACO only on the smaller TSP problems.

Particle swarm optimization-based algorithms are presented for TSP and generalized TSP [36], where an uncertain searching strategy and a crossover eliminated technique are used to accelerate the convergence speed. Compared with the existing algorithms for solving TSP using swarm intelligence, it has been shown that the size of the solved problems could be increased by using the proposed algorithm.

A hybrid multi-swarm particle swarm optimization algorithm is presented to solve the probabilistic traveling salesman problem [37] . The Probabilistic Traveling Salesman Problem (PTSP) is a variation of the classic Traveling Salesman Problem (TSP) and one of the most significant stochastic routing problems. In the PTSP, only a subset of potential customers needs to be visited on any given instance of the problem. The number of customers to be visited each time is a random variable. In the paper, a new hybrid algorithmic nature inspired approach based on Particle Swarm Optimization (PSO), Greedy Randomized Adaptive Search Procedure (GRASP) and Expanding Neighborhood Search (ENS) Strategy is proposed for the solution of the PTSP. The proposed algorithm is tested on numerous benchmark problems from TSPLIB with very satisfactory results. Comparisons with the classic GRASP algorithm, with the classic PSO, and with a Tabu Search algorithm are also presented. Also, a comparison is performed with the results of a number of implementations of the Ant Colony Optimization algorithm from the literature. Our proposed algorithm provides a new best solution.

An efficient method based on a hybrid genetic algorithm–particle swarm optimization (GA–PSO) is presented for various types of economic dispatch (ED) problem [38]. The arithmetic crossover operator is used as crossover operator in the genetic algorithm. It can be defined to produce a new child as linear combination of two chromosomes. As randomly generated a coefficient is used to produce new child. In this method, a new approach for obtaining a coefficient is proposed benefit from the similarity of parent chromosomes. The obtained results showed that the proposed method has been applied successfully in various ED problems.

A novel two-stage hybrid swarm intelligence optimization algorithm is presented to solve traveling salesman problem

[39]. This paper presents a novel two-stage hybrid swarm intelligence optimization algorithm called GA–PSO–ACO algorithm that combines the evolution ideas of the genetic algorithms, particle swarm optimization and ant colony optimization based on the compensation for solving the traveling salesman problem.

3. Particle Swarm Optimization Algorithm

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Formally, let $f : R^n \to R$ be the cost function which must be minimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution. The gradient of f is not known. The goal is to find a solution a for which $f(a) \le f(b)$ for all b in the search-space, which would mean a is the global minimum. Maximization can be performed by considering the function $h = -f$ instead.

PSO was presented under the inspiration of bird flock immigration during the course of finding food and then be used in the optimization problems. In PSO, each optimization problem solution is taken as a bird in the searching space and it is called "particle". Every particle has a fitness value which is determined by target functions and it has also a velocity which determines its destination and distance. All particles search in the solution space for their best positions and the positions of the best particles in the swarm. PSO is initially a group of random particles (random solutions), and then the optimum solutions are found by repeated searching. In every iteration, a particle will follow two bests to renew itself: the best position found for a particle called p_{best} ; the best position found for the whole swarm called g_{best} . All particles will determine following steps through the best experiences of individuals themselves and their companions.

For particle id, its velocity and its position renewal formula are as follows:

$$V'_{id} = \omega V_{id} + \eta_1 rand()(P_{idb} - X_{id}) + \eta_2 rand()(P_{gdb} - X_{id}) \quad (1)$$

$$X'_{id} = X_{id} + V'_{id} \quad (2)$$

In here: ω is called inertia weight, it is a proportion factor that is concerned with former velocity, $0 < \omega < 1$, η_1 and η_2 are constants and are called accelerating factors, normally $\eta_1 = \eta_2 = 2$, $rand()$ are random numbers, X_{id} represents the position of particle id ; V_{id} represents the velocity of particle id ; P_{id} , P_{gd} represent separately the best position particle id has found and the position of the best particles in the whole swarm.

In formula(1), the first part represents the former velocity of the particle, it enables the particle to possess expanding tendency in the searching space and thus makes the algorithm be more capable in global searching; the second part is called cognition part, it represents the process of absorbing individual experience knowledge on the part of the particle; the third part is called social part, it represents the process of learning from the experiences of other particles on the part of certain particle, and it also shows the information sharing and social cooperation among particles.

The flow of PSO can briefly describe as following: First, to initialize a group of particles, e.g. to give randomly each particle an initial position X_i and an initial velocity V_i , and then to calculate its fitness value f. In every iteration, evaluated a particle's fitness value by analyzing the velocity and positions of renewed particles in formula (1) and (2). When a particle finds a better position than previously, it will mark this coordinate into vector P1, the vector difference between P1 and the present position of the particle will randomly be added to next velocity vector, so that the following renewed particles will search around this point, it's also called in formula (1) cognition component. The weight difference of the present position of the particle swarm and the best position of the swarm P_{gd} will also be added to velocity vector for adjusting the next population velocity. This is also called in formula (1) social component. These two adjustments will enable particles to search around two bests.

The most obvious advantage of PSO is that the convergence speed of the swarm is very high, scholars like Clerc [40] has presented proof on its convergence. In order to verify the convergence speed of the PSO algorithm, we selected four benchmarks function and compared the results with traditional genetic algorithm (GA).

F1: Schaffer function

$$\min f(x_i) = 0.5 - \frac{(\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5)}{[1 + 0.001(x_1^2 + x_2^2)]^2},$$

$$-10 \le x_i \le 10$$

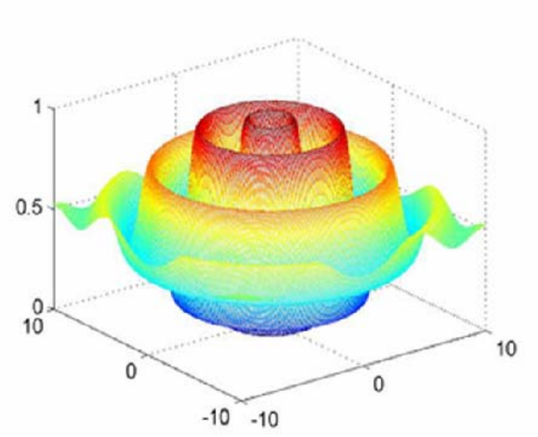

Fig. 1 Schaffer function

In this function the biggest point is in the situation where xi= (0, 0) and the global optimal value is 1.0, the largest in the overall points for the center, and 3.14 for the radius of a circle on the overall situation from numerous major points of the uplift. This function has a strong shock; therefore, it is difficult to find a general method of its global optimal solution.

F2: Shubert function

$$\min f(x,y) = \left\{ \sum_{i=1}^{5} i \cos\left[(i+1)x + i\right] \right\} \times$$
$$\left\{ \sum_{i=1}^{5} i \cos\left[(i+1)y + i\right] \right\}, x, y \in [-10,10]$$

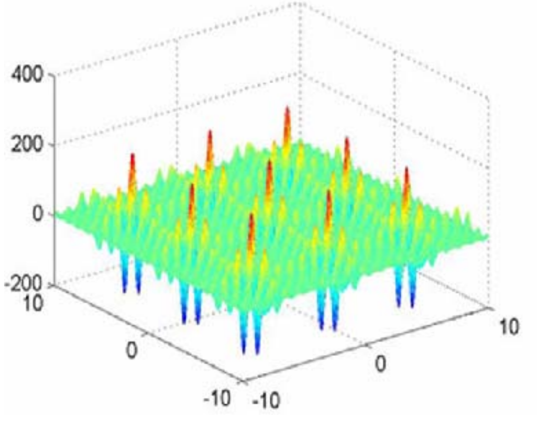

Fig. 2 Shubert function

This function has 760 local minimum and 18 global minimum, the global minimum value is -186.7309.

F3: Hansen function

$$\min f(x,y) = \sum_{i=1}^{5} i\cos((i-1)x + i) \sum_{j=1}^{5} j\cos((j+1)y + j),$$
$$x, y \in [-10,10]$$

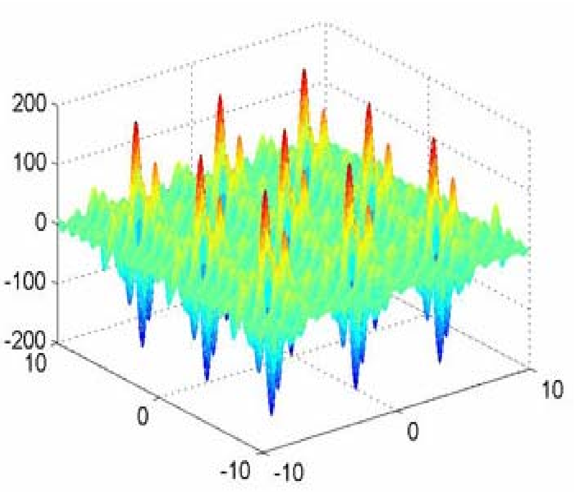

Fig. 3 Hansen function

This function has a global minimum value -176.541793, in the following nine point (-7.589893，-7.708314)、(-7.589893， -1.425128)、(-7.589893， 4.858057)、(-1.306708， -7.708314)、(-1.306708， -1.425128)、(-1.306708， 4.858057)、(4.976478， -7.708314)、(4.976478，-7.708314)、(4.976478，4.858057) can get this global minimum value, the function has 760 local minimum.

F4: Camel function

$$\min f(x,y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + \left(-4 + 4y^2\right)y^2,$$
$$x, y \in \left[-100,100\right]$$

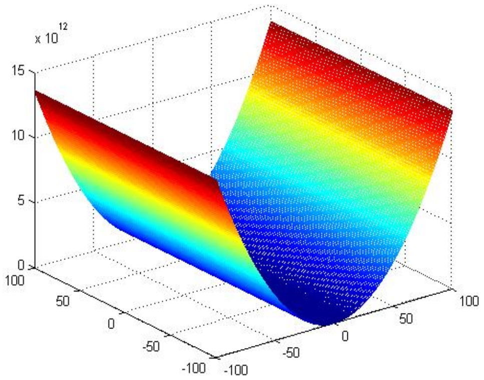

Fig. 4 Camel function

Camel function has 6 local minimum (1.607105, 0.568651)、(-1.607105, -0.568651)、(1.703607, -0.796084)、(-1.703607, 0.796084)、(-0.0898,0.7126)和(0.0898,-0.7126)， the (-0.0898,0.7126) and (0.0898,-0.7126) are the two global minimums, the value is -1.031628.

Table 1: Experiment results comparison (100 runs for each case)

Function	***Algorithm***	***Convergence Times***	***Optimal Solution***
F1	GA	72	1.0000000
	PSO	75	1.0000000
F2	GA	75	-186.730909
	PSO	80	-186.730909
F3	GA	85	-176.541793
	PSO	90	-176.541793
F4	GA	23	-1.031628
	PSO	56	-1.031628

In the experiment, each case is repeated for 100 times. Table 1 shows the statistics of our experimental results in terms of accuracy of the best solutions. GA found the known optimal solution to F1 72 times out of 100 runs, found the known optimal solution to F2 75 times out of 100 runs, found the known optimal solution to F3 85 times out of 100 runs, found the known optimal solution to F4 23 times out of 100 runs; PSO algorithm is efficiency for the four cases: found the known optimal solution to F1 75 times out of 100 runs, found the known optimal solution to F2 80 times out of 100 runs, found the known optimal solution to F3 90 times out of 100 runs and found the known optimal solution to F4 56 times out of 100 runs.

4. PSO Algorithm for TSP

In this paper, we will use the most direct way to denote TSP-path presentation. For example, path 4-2-1-3-4 can be denoted as (4, 2, 1, 3) or (2, 1, 3, 4) and it is referred as a chromosome. Every chromosome is regarded as a validate path. (In this paper, all paths should be considered as a ring, or closed path).

Fitness function is the only standard of judging whether an individual is “good” or not. We take the reciprocal of the length of each path as the fitness function. Length the shorter, fitness values the better. The fitness function is defined as following formula:

$$f(Si) = 1 \Big/ \sum_{i=1}^{N} d[C_{n(i)}, C_{n(i+1) \bmod N}] \qquad (3)$$

In order to verify the proposed algorithm is useful for the TSP, the experiment test we select 10 TSP test cases: berlin52, kroA100, kroA200, pr299, rd400, ali535, d657, rat783, u1060 and u1432. All experiments are performed on Intel Core(TM)2 Duo CPU 2.26GHz/4G RAM Laptop. In the experiments all test cases were chosen from TSPLIB (http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95), and the optimal solution of each test case is known in the website.

We list the test cases’ optimal solutions and compared with traditional genetic algorithm and PSO algorithm, the comparison results shown in Table 2. The comparison results demonstrate clearly the efficiency of our algorithm. Note that for the 10 test cases the optimum was found in all ten runs. The number of cities in these test cases varies from 52 to 1432.

Table 2: Optimal results comparison

Test Cases	***Optimal in TSPLIB***	***GA***	***PSO***
Berlin52	7542	7542	7542
kroA100	21282	21315	21310
kroA200	29368	30168	29968
Pr299	48191	48568	48540
Rd400	15281	15135	15135
Ali535	202310	242310	231120
D657	48912	50912	50612
Rat783	8806	8965	8905
U1060	224094	279094	269908
U1432	152970	182780	177890

5. Conclusions

This paper introduce the PSO algorithm for TSP, we use the proposed algorithm for solving the combinatorial problem: TSP, the new algorithm shows great efficiency in solving TSP with the problem scale from 52 to 1432. By analyzing the testing results, we reach the conclusion: in the optimization precision and the optimization speed, the new algorithm is efficiency than the genetic algorithm in coping with the TSP.

Acknowledgments

This paper is supported by the Provincial Natural Science Foundation of Hubei, China (No. 2011CDB334 and No.2011CDB346), National Civil Aerospace Pre-research Project of China and the Fundamental Research Founds for National University, China University of Geosciences (Wuhan).

References

[1] Durbin R, Willshaw D, "An Anlaogue Approach to the Traveling Salesman Problem Using an Elastic Net Approach", Nature, 326, 6114, 1987, pp.689-691.

[2] Tao Guo and Zbigniew Michalewize, "Inver-Over operator for the TSP", In Parallel Problem Sovling from Nature(PPSN V), Springer-Verlag press, 1998, pp.803-812.

[3] Zhangcan Huang, Xiaolin Hu and Siduo Chen, "Dynamic Traveling Salesman Problem based on Evolutionary Computation", In Congress on Evolutionary Computation(CEC'01), IEEE Press, 2001, pp.1283-1288.

[4] Aimin Zhou, Lishan Kang and Zhenyu Yan, "Solving Dynamic TSP with Evolutionary Approach in Real Time", In Congress on Evolutionary Computation(CEC'03), 2003, pp.951-957.

[5] Hui.Yang, Lishan.Kang and Yuping.Chen, "A Gene-pool Based Genetic Algorithm for TSP", Wuhan University Journal of Natural Sciences, 8(1B), 2003, pp.217-223.

[6] Xuesong Yan, Lishan.Kang, "An Approach to Dynamic Traveling Salesman Problem", Proceedings of the Third International Conference on Machine Learning and Cybernetics,2004, pp. 2418-2420.

[7] Xuesong Yan, Aimin Zhou, Lishan Kang, "TSP Problem Based on Dynamic Environment", Proceedings of the 5th World Congress on Intelligent Control and Automation, 2004, pp.2271-2274.

[8] J. Kennedy and R. C.Eberhart, "Particle Swarm Optimization", IEEE International Conference on Neural Networks, 1995, pp.1942-1948.

[9] Clare M, Kennedy J, "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space", IEEE Trans. on Evolution2ary Computation, vol.6(1), 2002, pp.58-73.

[10] C.A.Coello and M.S.Lechuga, Mopso, "A proposal for multiple objective particle swarm optimization", In IEEE Proceedings World Congress on Computational Intelligence, 2002, pp.1051-1056.

[11] J.Kennedy, "The particle swarm: social adaptation of knowledge", In Proc. IEEE Conf. on evolutionary computation, 1997, pp.3003-3008.

[12] E. Oscan and C. K.Mohan, "Analysis of A Simple Particle Swarm Optimization System", Intelligence Engineering Systems Through Artificial Neural Networks, 1998, pp.253-258.

[13] Hua Z, Huang F, "A variable-grouping based genetic algorithm for large-scale integer programming", Inf Sci 176(19), 2006, pp.2869–2885.

[14] Ellabib I, Calamai P, Basir O, "Exchange strategies for multiple ant colony system", Inf Sci 177(5), 2007, pp.1248–1264.

[15] Lo CC, Hus CC, "Annealing framework with learning memory", IEEE Trans Syst Man Cybern Part A 28(5), 1998, pp.1–13.

[16] Masutti TA, de Castro LN, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", Inf Sci 179(10), 2009, pp.1454–1468.

[17] Onwubolu GC, Clerc M, "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization", Int J Prod Res 42(3), 2004, pp.473–491.

[18] Shen G, Zhang YQ, "A new evolutionary algorithm using shadow price guided operators", Appl Soft Comput 11(2), 2011, pp.1983–1992.

[19] Acampora G, Gaeta M, Loia V, "Combining multi agent paradigm and memetic computing for personalized and adaptive learning experiences", Comput Intell J 27(2), 2011, pp.141–165.

[20] Marinakis Y, Marinaki M, Dounias G, "A hybrid particle swarm optimization algorithm for the vehicle routing problem", Eng Appl Artif Intell 23(4), 2010, pp.463–472

[21] Banaszak D, Dale GA, Watkins AN, Jordan JD, "An optical technique for detecting fatigue cracks in aerospace structures", In: Proc 18th ICIASF, 2009, pp.1–7.

[22] Parsopoulos, K.E., Vrahatis, M.N., "Particle swarm optimization method for constrained optimization problems", In: Kvasnic ˇ ka, V. et al. (Eds), Proceedings of the second Euro-International Symposium on Computational Intelligence, Kos ˇ ice, Slovakia, 2002, pp.214 – 220.

[23] Hu, X., Eberhart, R.C., "Solving constrained nonlinear optimization. problems with particle swarm optimization", In: Callaos, N. (Ed.), Proceedings of the Sixth World Multiconference on Systemics. Cybernetics and Informatics, Orlando, Fl, 2002, pp.203 – 206.

[24] Ray, T., Liew, K.M., "A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimization problems", In: Kim, J.H. et al. (Eds.), Proceedings of the 2001 Congress on Evolutionary Computation, IEEE Service Center, Piscataway, NJ, 2001, pp.75 – 80.

[25] Clerc, M, "Discrete particle swarm optimization, illustrated by the traveling salesman problem", In: Studies in Fuzziness and Soft Computing New optimization techniques in engineering, Babu, B.V. & Onwubolu, G.C. (Eds.), Vol. 141, 2004, pp.219-239.

[26] Wang KP, Huang L, Zhou CG, Pang W, "Particle swarm optimization for traveling salesman problem", In:

International conference on machine learning and cybernetics, 2003, pp.1583–1585.

[27] Onwubolu, G.C. & Clerc, M, "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization", International Journal of Production Research, Vol. 42, No. 3, 2004, pp.473－491.

[28] Hendtlass, T, "Preserving diversity in particle swarm optimization, Developments in Applied Artificial Intelligence", Proceedings of the 16th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems IEA/AIE 2003, Laughborough, UK, June 2003, In: Lecture Notes in Computer Science, Vol. 2718, 2003, pp.4104-4108.

[29] Pang, W.; Wang, K.; Zhou, C.; Dong, L.; Liu, M.; Zhang, H. & Wang, J, "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, Shangai, China, August 2004, pp.2342-2346.

[30] Pang, W.; Wang, K.; Zhou, C. & Dong, L, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem", In: Proceedings of the Fourth International Conference on Computer and Information Technology, Wuhan, China, September 2004, pp.796-800.

[31] Machado, T.R. & Lopes, H.S, "A hybrid particle swarm optimization model for the traveling salesman problem", In: Natural Computing Algorithms, Ribeiro, H.; Albrecht, R.F. & Dobnikar, A. (Eds.), 2005, pp.255-258.

[32] Goldbarg, E.F.G; Souza, G.R. & Goldbarg, M.C, "Particle swarm for the traveling salesman problem", Proceedings of the EvoCOP 2006, Gottlieb, J. & Raidl, G.R. (Ed.), Lecture Notes in Computer Science, Vol. 3906, 2006, pp.99-110.

[33] Yuan, Z.; Yang, L.; Wu, Y.; Liao, L. & Li, G, "Chaotic particle swarm optimization algorithm for Traveling Salesman Problem", In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, 2007, pp.1121-1124.

[34] Fang, L.; Chen, P. & Liu, S, "Particle swarm optimization with simulated annealing for TSP", Proceedings of the 6th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Long, C. A.; Mladenov, V. M. & Bojkovic Z. (Eds.), 2007, pp.206-210.

[35] Hendtlass T, "Preserving diversity in particle swarm optimization", Lect Notes Comput Sci 2718, 2003, pp.4104–4108.

[36] Shi XH, Liang YC, Lee HP, Lu C, Wang QX, "Particle swarm optimization-based algorithms for TSP and generalized TSP", Inf Process Lett 103(5), 2007, pp.169–176.

[37] Yannis M, Magdalene M, "A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem", Comput Oper Res 37(3), 2010, pp.432–442.

[38] Guvenc U, Duman S, Saracoglu B, Ozturk A, "A hybrid GA–PSO approach based on similarity for various types of economic dispatch problems", Electron Electr Eng Kaunas: Technologija 2(108), 2010, pp.109–114

[39] Wu Deng, Rong Chen, Bing He, Yaqing Liu, Lifeng Yin, Jinghuan Guo, "A novel two-stage hybrid swarm intelligence optimization algorithm and application", Soft Computing 16, 2012, pp.1707-1722.

[40] M.Clerc and J.Kennedy, "The Particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space", IEEE Trans. on Evolutionary Computation, Vol.6, 2002, pp.58-73.

[41] Xuesong Yan, Hui Li, "A Fast Evolutionary Algorithm for Combinatorial Optimization Problems", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics,2005, pp.3288-3292.

[42] Xuesong Yan, Qing Hua Wu, "A New Optimizaiton Algorithm for Function Optimization", Proceedings of the 3rd International Symposium on Intelligence Computation & Applications, 2009, pp. 144-150.

[43] Xue Song Yan, Qing Hua Wu, Cheng Yu Hu, Qing Zhong Liang, "Circuit Design Based on Particle Swarm Optimization Algorithms", Key Engineering Materials, Vols. 474-476, 2011, pp.1093-1098.

Xuesong Yan associate professor received him B.E. degree in Computer Science and Technology in 2000 and M.E. degree in Computer Application from China University of Geosciences in 2003, received he Ph.D. degree in Computer Software and Theory from Wuhan University in 2006. He is currently with School of Computer Science, China University of Geosciences, Wuhan, China and now as a visiting scholar with Department of Computer Science, University of Central Arkansas, Conway, USA. He research interests include evolutionary computation, data mining and computer application.

Can Zhang received him B.E. degree in Computer Science and Technology in 2011. He is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Wenjing Luo received her B.E. degree in Computer Science and Technology in 2012. She is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Wei Li received her B.E. degree in Computer Science and Technology in 2012. She is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Wei Chen received him B.E. degree in Computer Science and Technology in 2012. He is currently is the M.E. degree candidate with School of Computer Science, China University of Geosciences, Wuhan, China. Her research interests include evolutionary computation.

Hanmin Liu associate professor. He is currently as a Ph.D candidate of School of Computer Science, China University of Geosciences, Wuhan, China. He research interests include evolutionary computation and applications.