Asian Journal of Information Technology 5 (11): 1191-1196, 2006

Improved Particle Swarm Algorithm for TSP Based on the Information Communication and Dynamic Work Allocation

Qiang Wang, Lei Xiong, Haiyan Liu and Jun Liang
Faculty of Computer Science and Information Engineering, Guangxi Normal University, Guilin, Guangxi, China, 541004

Abstract: An improved Particle Swarm Optimization (PSO) algorithm is designed for TSP solving based on the information communication and dynamic work allocation. A strategy of information communication among particles with greedy idea is proposed so as to make a particle gain more useful information from other particles. The general algorithm frame of PSO is also enhanced, combined with dynamic work allocation strategy to keep the balance between the searching efficiency and solution quality. Some experiments are conducted and demonstrate that our improved PSO algorithm for TSP problem is efficient and promising.

Key words: Particle Swarm Optimization (PSO), TSP, information communication, dynamic work allocation

INTRODUCTION

Traveling Salesman Problem (TSP) is a typical NP complete problem[1]. Though it is simple in describing TSP mathematically, there is no any definite algorithm can be used to solve TSP in polynomial time. Many complicated problems in different areas can be abstracted and changed to TSP. So it is very important, both in theory and in application, to find a method to solve TSP in polynomial time. Nowadays Swarm Intelligence (SI) techniques are developing quickly, many evolutionary computational algorithms, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO) are used to solve the combinatorial optimization problems such as TSP[2,3].

Particle Swarm Optimization technique[4], which is different from GA, was first proposed by Kennedy and Eberhart in 1995. The particle is defined as an individual without volume and mass and a unified simple behavior regulation is followed by each particle. Then the very complicated characteristics will be appeared for the whole particle swarm. And this phenomenon can be used to solve complicated optimization problem. It has been successfully used in many research areas such as function optimization, fussy system control, ANN training etc and has become a new and hot spot of research in the world.

TSP and PSO algorithm

The mathematical modal of TSP: TSP can be described as: A salesman is going to travel a number of cities for selling. Starting from city #1 he must passes and stops at each of all the rest cities for only once and then returns to city #1, the most cost-efficient path is to be found. If the distance is considered only as the cost, to find the best path T = (t1,t2,…,tn) is the same as to make following target function minimum

$$f(T)=\sum_{i=1}^{n-1} d(t_i,t_{i+1})+d(t_n,t_1)$$

t_i is the numbering of the city which is a natural number between 1 and N, $d(t_i,t_j)$ denote the distance from city i to city j and for symmetrical TSP $d(t_i,t_j) = d(t_j,t_i)$.

For TSP, the solution searching space increases rapidly as the total number of city increases and this leads to the “combination explosion” phenomenon. For instance, the total number of city N = 100, the total number of solution paths possible will be $4.67*10^{155}$. It is very difficult to search the best solution in such a huge space. Finding out an efficient algorithm to solve TSP has remarkable significance both in theory and in application, so, many investigators are attracted in this research area.

Basic PSO algorithm: As an evolutionary algorithm, PSO has characteristics of both the evolutionary computation and swarm intelligence. To realize the best solution searching in complicated space, the fitness of the individual particle is valued in the process of cooperation and competition among individuals. In PSO, every individual is regarded as a particle without volume and

Corresponding Author: Qiang Wang, Faculty of Computer Science and Information Engineering, Guangxi Normal University, Guilin, Guangxi, China, 541004

mass in an n-dimensional space. Each state of a particle can represent a reasonable solution for the optimization problem and a value of fitness will be calculated for the particle according to target function. The particle's moving direction and distance in solution space are decided by its velocity of flying. The particles move around the particle which has the best fitness. The optima will be found after several generations of evolution. There are two extrema in each generation, the best solution for a particle in its history *pbest* and the best solution for the whole swarm. *gbest*. A particle renews its velocity and position according to the following equation:

$$V_{ij}(t+1) = wV_{ij}(t) + c_1r_1(t)(p_{ij}(t) - x_{ij}(t)) + c_2r_2(t)(p_{gj}(t) - x_{ij}(t))$$
$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

Here, $x_i(y)$ represents the position of the current particle at time t. $V_i(t)$ denotes the velocity of the current particle. The subscript j represents the jth dimension of searching space and i represents the particle i and t represents the tth generation, c1,c2 denote the constant of acceleration, r1,r2 denote independent probability function. Please refer to referents[5] for The detailed description about PSO algorithm.

The modified PSO algorithm: The above equation for basic PSO has been applied to continue optimization problems successfully. But for discrete optimization problem, such as TSP, the velocity of particle is hard to be expressed because the position of a particle is coded according to visiting sequence of cities. For TSP the Eq. 2 and 3 must be modified. An attempt has been done in referent[6]. An equation for discrete optimization was constituted to renew the particle velocity and position but the testing result of the algorithm proposed in referent[6] is not efficient enough because standard PSO is adopted in that algorithm. We improve the effect by introducing the strategy of information communication among particles and dynamic work allocation among classes of particle swarm in this study.

Strategy of information communication among particles with greedy idea: To overcome the weakness that it is easy to drop into local minimum in standard PSO algorithm, a strategy of information communication among particles with greedy idea is designed to strengthen the diversity of the particles and to speed the convergence process.

In the renewing process of position and velocity, a particle absorbs the better historical experience of the neighboring particles, remains better sub-path of its own and constitutes a new particle. The new particle continues the operation of renewing position and velocity. To carry such a strategy the Eq. 2 and 3 are modified as following:

$$V_{ij}' = wV_{ij} \otimes \gamma(p_{mj} - x_{ij}) \oplus \alpha(p_{ij} - x_{ij}) \oplus \beta(p_{gj} - x_{ij}) \quad (4)$$

$$X_{ij}' = X_{ij} + V_{ij}' \quad (5)$$

Here, P_{mj}, P_{ij}, P_{gj} denote the neighboring particle positions, the best position of the particle to be renewed, and the best position of whole swarm particles respectively, "$\oplus$" represents the combination operator of city order exchanging in velocities addition[6], "-" represents subtraction operation of the particle positions[6], "+" represents addition operation of the particle position and velocity[6], "$\otimes$" represents information communication operation of the particle position and velocity and $\gamma(p_{mj}-x_{ij})$ represents information communication operation between particle i and its neighbor particle m with the probability, γ respectively. The larger the parameter is the more frequent communication takes place between particles. With this information communication strategy it is easy for particles to jump out from local optima as shown in Fig. 1.

In the basic PSO, only particle g_{best} passes information to other particles, the information transfer in single direction and the whole iteration process follow the current best solution. If the current best solution can not be renewed in a long period of time, the particles get together and their velocity will decrease to near zero. This phenomenon limits the search range of the particles, which makes algorithm fall into the local optima prematurely. With the strategy of information communication among particles proposed in this study, the historical experience of all particles is shared and the particles not only follow *gbest* simply but get beneficial information from neighbor particle as well. The information sharing and communicating are very profitable for enhancing search ability of particles and it is helpful for global optima finding.

Figure 2, shows v1, v2, v4 represent the partial velocity vectors produced under the influence of the *gbest* particle, *pbest* particle and the neighboring particle S2, respectively. V3 is the velocity of S1 itself. Under the effect of v1, v2, v3, v4, particle S1 moves to S3 with the velocity of V5. Then the particle S1 will start to move form S3 next time.

From experiment results we found that more beneficial experience can be shared among particles if high frequency of the information communication is adopted at the beginning phase of the algorithm

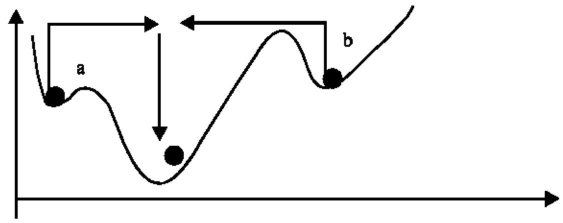

Fig. 1: Jump out from local optima in the action between particles

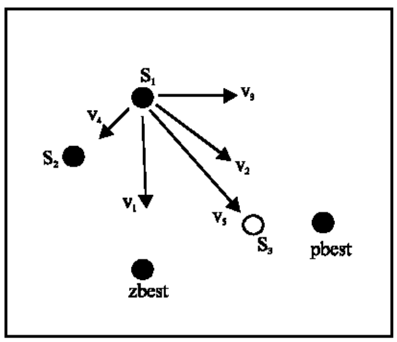

Fig. 2: Position and velocity renew in the action between particles

execution. But at the phase near to the end, the fitness of particles is similar from each other and the differences among particles get smaller and smaller. Keeping high frequency for the information communication is not helpful for space searching. So the probability p_m will be taken according to the following Eq.

$$P_m = 0.6 - \frac{t}{T_{max}}(0.6 - 0.2)$$

Here, t represents the iteration number of current generation of particle swarm and T_{max} represents the allowed maximum iteration generation number of particle swarm.

Suppose that S1 is the particle to be going to communicate with neighboring particle, S2 is the historical optima among the neighboring particles of S1, S1' is new particle constituted by the combination of S1 and S2 in the mode of roulette. An example is shown as following:

Suppose total number of city is 10, S1 and S2 are:

S_1	0	3	5	9	6	4	1	2	8	7
S_2	7	9	6	4	8	2	1	3	5	0

First manipulation: Select a city randomly (suppose city 4 is selected) and add it to S_1, the neighbors of city 4 are cities 6, 1 and 8 in S1 and S2 and also suppose city 6 and 8 are chosen to be neighbors of city 4 on the left and the right in the new particle S1'

S_1			6	4	8					

Second manipulation: Consider the left neighbor of the city 6, the neighbors of city 6 are cities 9 and 4 in S1 and S2, the city 4 has been in S_1, only city 9 can be the left neighbor of city 6○

S_1		9	6	4	8					

Third manipulation: Consider the right neighbor of the city 8, the neighbors of city 8 are cities 2, 4 and 7 in S1 and S2, the city 4 has been in S1' already and suppose to chose city 2 as the right neighbor of city 8, if $D_{4,2}$, + $D_{2,8}$, <$S_{4,8}$, + $D_{2,8}$ the city 2 is inserted between cities 4 and 8

S_1		9	6	4	2	8				

Fourth manipulation: Consider the left neighbor of city 9, insert city 5 between citied 6 and 9 due to the same reason above

S_1	9	5	6	4	2	8				

Fifth manipulation: Consider the right neighbor of city 8, insert city 7 between citied 2 and 8

S_1	9	5	6	4	2	7	8			

Sixth manipulation: Consider the left neighbor of city 9 and right neighbor of city 8, All of neighbor cities of city 8 and 9 have been in s1'already, the rest cities are 0, 1, 3 and suppose to chose city 0 as the left neighbor of the city 9

S_1	9	5	6	4	2	7	8			0

Seventh manipulation: Chose city 3 as the left neighbor of city 0

S_1	9	5	6	4	2	7	8		3	0

Eighth manipulation: Insert city 1 between city 0 and 3 due to the same reason above

S_1	9	5	6	4	2	7	8	3	1	0

Suppose 8-3 and 5-6 are two city pairs with the longest distance, disconnect the pairs and reconnect them in changed order, then we get the final result

S_1	9	5	8	7	2	4	6	3	1	0

Modified PSO with strategy of the dynamic work dividing: A strategy of the dynamic work allocation is proposed in addition to the strategy of information communication between particles in this study. The developing ability and testing ability are balanced in the modified algorithm so as to promote the searching efficiency and solution quality.

In standard PSO there is only one swarm P composed of N particles, but in our modified PSO the swarm P is divided into two classes of sub swarms, named P1 and P2 respectively, which have different searching strategies and tasks. In this paper we suppose that there is only one group in P1 and there are two groups in P2. The P1 is in charge of searching in the local area surrounding the current best solution while P2 is divided into two sub swarms in charge of the global testing in the solution space. P1 and P2 are divided according to the average fitness.

For sub swarm P1

Step 1: current particle is reviewed to decide whether it needs to communicate with neighboring particles according to probability p_m, if need, the current particle (S1) communicates with neighboring particle which has the best solution in the neighboring area in the way described in 3.1 and constitutes a new particle S1', if need not, S1 will be S1'.

Step 2: S1' is taken as current particle and its velocity and position are renewed. The renewed particle is denoted as S1''.

Step 3: S1'' is receipted as a new particle if its fitness is better than S1, otherwise, S1'' will be processed with stimulating annulling algorithm and receipted according to probability $p_m = 1 - p_m$.

For sub swarm P2

Step 1: current particle is reviewed according to standard PSO algorithm.

In our modified PSO two classes of sub swarms, P1 and P2, cooperate to do the searching work. The different independent operations are performed inner the sub swarm. The cooperation between P1 and P2 is realized through the strategy of dynamic work allocation and the diversity is kept by the information communication between particles. The modified PSO overcomes the weakness that all of the particles are belonged to the same swarm and are easily gathered at local optima, which ensure the algorithm converges to global optima of the searching space.

In the process of iteration, the roles of P1 and P2 are exchanged from time to time. When the best solution searched by P2 is better than that by P1, the roles of P2 and P1 are exchanged to keep more efficient local search always surrounding current optima solution. The particles in P1 are not always in P1 and so are in P2. The particles will be re-divided once several generations.

A disturb mechanism is introduced in our searching process. When particles fall to local optima, they will be forced to jump out by reset states of some particles. Introducing disturb mechanism can enhance the global searching ability. In our algorithm the fitness of particles is tested if the current optima do not change for several iterations. Then the particles with similar fitness and particles with poor fitness are put into a buffer pools and disturbed in probability, the better particles will be returned to swarm again and divided to P1 or P2 according to fitness.

RESULTS AND DISCUSSION

To validate the performance of our algorithm, a set of testing data which is in the web sit (www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/) for TSP is chosen. For our experiments, the running environment is Celeorn 1.7G CPU, 256M RAM, Win2000 OS, VC6.0. Two kinds of comparison experiments are performed. One kind is to compare the other PSO algorithms, data come from referents[6-8], with ours. The other is to compare the other evolution algorithms, data come from referents[9-11], with ours.

Table 1 shows the results of the comparison, the burma14 problem provided by TSPLIB is chosen for the experiment in which the PSO algorithms proposed in referent[6,7] are compared with our algorithm. Every data in Table 1 is the average from 30 times running. The actual effect of our algorithm can be seen in Table 1 which shows that the total iteration generations for our algorithm for finding the global optima is only 0.1% to that for the algorithm proposed in referent[6] and 2% to that for the algorithm in referent[7].

Table 2 shows the results of comparison in which algorithms proposed in referent[8] are compared with our algorithm. The compared algorithms include SPSO

Table 1: Experiment results for the algorithm proposed in this paper comparing with algorithm in referents[6,7]

Scale of solution space	(14-1)!/2=3,113,510,400		
Algorithm comparison	In referent[6]	In referent[7]	Proposed in this paper
Average iteration generations	20,000	1000	20
Average searching space	20,000*100 =2,000,000	1,000*100=100,000	20*100=2,000
Proportion from searching space to solution space	2,000,000/3,113,510,400=0.064%	100,000/3,113,510,400=3.211E-5	2,000/3,113,510,400=6.42E-7
The best result	30.8785	(Equal to the best known result in the world)	

Table 2: Experiment results for the algorithm proposed in this study comparing with algorithm in referents[8]

TSP Problem	The best known solution for TSP	The average solution of classic PSO	The best solution of RPAC	The average solution of SCPSO	The average solution by algorithm of this paper	The best solution by algorithm of this paper
gr24	1272	1305	1278	1274	1272	1272
Bayes29	2020	2102	2042	2028	2020	2020
gr48	5046	5528	5265	5199	5048	5046

(standard PSO), RPAC (random probability ants group) and SCPSO (sub swarm cross based PSO). The input data come from three classic TSP, gr24, bayes 29 and gr 48 and all of them are provided in TSPLIB. The swarm scale for all algorithms in[8] is 500 but 100 for our algorithm. The results in the table are the average of 30 times running in which each is limited to 200 iterating generations. The algorithm proposed in this study converges to the best known solution for problems gr24 and bayes29 in the 30 times running and the percentage for problem gr48 obtaining the best known solution reaches 93%. In 30 times running, all of the results are much better than the algorithms in referent[8].

Figure 3 shows the converging curves for different algorithms including the classic PSO, the PSO with genetic cross, the PSO with multi swarms cross and the algorithm proposed in this study in solving the gr24 problem. Comparing to the classic PSO and the PSO with genetic cross, the PSO with multi swarms cross is better in solution quality and converging speed. But it is very hard for the PSO with multi swarms cross to improve farther because, at the beginning, all of the sub swarms evolutes independently, particles exchanging between sub swarms is taken place only at the time when the fitness of particle stops changing. Our algorithm is much better than the three algorithms mentioned above both in solution quality and converging speed, due to the adopting of the information communication and dynamic work allocation strategy.

In following tests the algorithm in this paper is compared with the other evolution optimization algorithms and the tested problems include Eil50, Eil70, KroA100 provided in TSPLIB also. Table 3 shows the results of comparing our algorithm with ACS (ants colony), GA (genetic algorithm), EP (evolution planting) and SA (simulating annulling). Table 4 shows the results of comparing our algorithm with GASA (genetic algorithm with simulate annulling), SA +2-opt, PSA+2-opt and GA+2-opt.

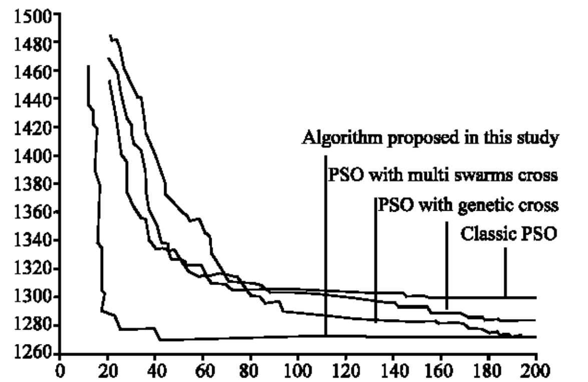

Fig. 3: The converging curves for different algorithms

The problems provided in TSPLIB for algorithm testing are in two expression forms. The distance between two cities is expressed in integer if the computing result is given in the parenthesis and the distance between two cities is expressed in real number if the computing result is given in the square brackets. We use the data form in real number in the experiments.

For the tests in Table 3, the swarm scale is 100. The results in the square brackets refer to average generations of 20 times running. The results in parenthesis are the solution in integer form and N/A refers to that the result can not be gained. From the results shown in table 3 we can see that our algorithm is much better than GA, EP and SA both in solution quality and converging speed and it is better than ACS also in the converging speed.

From the results shown in Table 4 we can see, for the TSP problem of 75 cities, GA+2-opt converges to local optima after 100 generations prematurely, SA+2-opt and PSA+2-opt are going to converge to

Table 3: Our algorithm comparing with other evolution algorithms

TSP problem	The best known solution	ACS	GA	EP	SA	Algorithm proposed in this paper
Eil50	427.86	427.96	N/A	427.86	N/A	427.86
	(425)	(N/A)	(428)	(426)	(443)	(425)
		[1830]	[25,000]	[100,000]	[68512]	[332]
Eil75	542.37	542.37	N/A	549.18	N/A	542.37
	(535)	(535)	(545)	(542)	(580)	(535)
		[3480]	[80,000]	[325,000]	[173,250]	[624]
KroA 100	21285.44	21285.44	N/A	N/A	N/A	21285.44
	(21282)	(21282)	(21761)	(N/A)	(N/A)	(21282)
		[4820]	[103,000]	N/A	N/A	[996]

Table 4: Our algorithm comparing with other evolution algorithms (continued)

TSP problem	The best known solution	GASA	SA+2-opt	PSA+2-opt	GA+2-opt	Algorithm proposed in this paper
Eil50	427.86	427.855	431.556	431.89	443.155	427.86
	(425)	(N/A)	(N/A)	(N/A)	(N/A)	(425)
		[738]	[1345]	[1343]	[409]	[332]
Eil75	542.37	542.309	560.339	560.85	566.27	542.37
	(535)	(N/A)	(N/A)	(N/A)	(N/A)	(535)
		[870]	[1693]	[1718]	[428]	[624]

better solution from 1000 generations. Though GASA is better than GA and SA, it can be not compared to our algorithm.

CONCLUSION

The PSO algorithms have been using in the problem of continuing optimization successfully in recent years. But it is still a challenge in using PSO to solve discrete optimization problems. In this paper we propose a modified PSO based on the information communication between particles with the greedy idea and dynamic work allocation between the sub swarms. The modified algorithm proposed in this paper can keep the balance between the searching efficiency and solution quality. Experiments prove that our modified PSO algorithm is better than several evolution optimization algorithms in solving TSP problems and its results are satisfactory.

REFERENCES

1. Johnson, D.S. and L.A. Mc Geoch, 1997. The traveling salesman problem: A case study in local optimization. Aarts E H L, Lenstra J K. Local search in combinatorial optimization. John Wiley and Sons, pp: 215-310.
2. Freisleben, B. and P. Merz, 1996. New Genetic Local Search Operators for the Traveling Salesman Problem. Proc. 4th Conf. Parallel Problem Solving from Nature. pp: 616-621.
3. Gambardella, L.M. and M. Dorigo, 1995. Ant-Q: A Reinforcement Learning Approach to the Traveling Salesman Problem. Proc. 12th Intl. Conf. on Machine Learning. Morgan Kaufmann, pp: 252-260.
4. Eberhart, R.C. and J. Kennedy, 1995. A New Optimizer Using Particles Swarm Theory. Proc. Sixth Int Symposium on Micro Machine and Human Science. Nagoya, pp: 39-43.
5. Shi, Y. and R. Eberhart, 1998. A Modified Particle Swarm Optimizer. Anchorage, Alaska: IEEE Intl. Conference on Evolutionary Computation, 05: 69-73.
6. Wang Kang-ping, Huang Lan and Zhou Chun-guang *et al.*, 2003. Particle Swarm Optimization for Traveling Salesman Problem. IEEE Service Center, Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an: IEEE Press, 5: 1583-1585.
7. Pang Wei, Wang Kang-ping and Zhou Chun-guang *et al.*, 2005. Fuzzy Discrete Particle Swarm Optimization for Solving Travel Salesman Problem, Mini-micro Systems, 8: 1331-1334
8. Tan Hao, Wang Jin-yan and He Yi-zhen *et al.*, 2005. Sub swarm and crossbreed strategy PSO for TSP, Systems Engineering, 4: 83-87.
9. Dorigo, M. and L.M. Gambardella, 1997. Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Service Center, IEEE Transactions on Evolutionary Computation, Piscataway, NJ: IEEE Press, 1: 53-66.
10. Okada, M., K. Taji and Fu M. kushima, 1998. Probabilistic analysis of 2-op for traveling salesman problems. Intl. J. Systems Sci., 29: 297-310.
11. Wang Ling and Zheng Dazhong, 2001. A Kind of GASA Hybrid Optimization Strategy, Control Theory and Applications, 8: 552-554.