Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003

PARTICLE SWARM OPTIMIZATION FOR TRAVELING SALESMAN PROBLEM

KANG-PING WANG, LAN HUANG, CHUN-GUANG ZHOU, WEI PANG

College of Computer Science and Technology, Jilin University, Changchun 130012, China
E-MAIL: cgzhou@mail.jlu.edu.cn

Abstract:

This paper proposes a new application of Particle Swarm Optimization for Traveling Salesman Problem. We have developed some special methods for solving TSP using PSO. We have also proposed the concept of Swap Operator and Swap Sequence, and redefined some operators on the basis of them, in this way the paper has designed a special PSO. The experiments show that it can achieve good results.

Keywords:

Particle Swarm Optimization; Traveling Salesman Problem; Combinatorial Optimization

1 Introduction

The particle swarm optimization (PSO) was originally presented by Kennedy and Eberhart[1], it has been compared with Genetic Algorithm[4,5,6], it is inspired by social behavior among individuals, these individuals (we call them particles) are moving through an n-dimensional search space, each particle represents a potential solution of the problem, and can remember the best position(solution) it has reached. All the particles can share their information about the search space, so there is a global best solution. In each of iteration, every particle calculates its velocity according to the following formula,

$$V_{id}=\omega * V_{id}+ \eta_1 * rand() * (P_{id}-X_{id})+ \eta_2 * rand() * (P_{gd}-X_{id}) \quad (1)$$

Where ω is the inertia factor, X_{id} is current position of the particle, P_{id} is the best solution this particle has reached, P_{gd} is the global best solution of all the particles. η_1 and η_2 is the weight determining the influence of P_{id} and P_{gd}.

After calculating the V_{id}, we can get the new position in next iteration:

$$X_{id} = X_{id} + V_{id} \quad (2)$$

Particle Swarm Optimization is an evolutionary computation technique because it has common evolutionary attributes:

1. It has an initializing process, in the process, there is a population that is made up of certain number of individuals, and each individual in the population will have a random solution first.

2. It searches better solution in the search space by producing new better generation.

3. The production of new generation is based on the previous generation.

Traveling Salesman Problem (TSP) is a well-known NP-hard combinatorial optimization problem. By now, TSP has been well studied by many metaheuristic approaches, such as nearest neighborhood search (NNS), simulated annealing (SA), tabu search (TS), neural networks (NN), Ant Colony System (ACS)[7], and genetic algorithm (GA)[8]. Since 1995,Particle Swarm Optimization has been proven to succeed in continuous problems, much work has been done effectively in this area, but for discrete problems, it is still a new field, especially, applying PSO algorithm to TSP is a new research direction.

2 Basic Concept

2.1 Swap Operator

Consider a normal solution sequence of TSP with n nodes,

$S = (a_i)$, i=1…n.

Here we define Swap Operator $SO(i_1,i_2)$ as exchanging node a_{i1} and node a_{i2} in solution S. Then we define $S'=S+SO(i_1,i_2)$ as a new solution on which operator $SO(i_1,i_2)$ acts. So the plus sign " + " above has its new meaning.

It can be given a concrete example: Suppose there is a TSP problem with five nodes, here is a solution: S=(1, 3, 5, 2, 4). The Swap Operator is SO(1, 2), then

$S' = S + SO(1,2) = (1,3,5,2,4) + (1,2) = (3,1,5,2,4)$

2.2 Swap Sequence

A Swap Sequence SS is made up of one or more Swap Operators.

$$SS=(SO_1, SO_2, SO_3 \ldots, SO_n) \quad (3)$$

$SO_1, SO_2, \ldots, SO_n$ are Swap Operators, here the order of the Swap Operator in SS is important.

2.3 The feature of Swap Sequence

Swap Sequence acting on a solution means all the Swap Operators of the Swap Sequence act on the solution in order. This can be described by the following formula:

$$S' = S + SS = S + (SO_1,\ SO_2,\ SO_3,\ldots,\ SO_n) = ((S+SO_1)+ SO_2)+\ldots + SO_n \quad (4)$$

Different Swap Sequences acting on the same solution may produce the same new solution. All these Swap Sequences are named the equivalent set of Swap Sequences. In the equivalent set, the sequence which has the least Swap Operator is called Basic Swap Sequence of the set or Basic Swap Sequence (BSS) in short.

Several Swap Sequences can be merged into a new Swap Sequence, we define the operator $\oplus$ as merging two Swap Sequences into a new Swap Sequence. Suppose there is two Swap Sequences, SS1 and SS2, SS1 and SS2 act on one solution S in order, namely SS1 first, SS2 second, we get a new solution S', and there is another Swap Sequence SS' acting on the same solution S, then get the same solution S', described as follows:

$$SS' = SS1 \oplus SS2 \quad (5)$$

SS' and SS1 $\oplus$ SS2 are in the same equivalent set.

2.4 The construction of Basic Swap Sequence

Suppose there is two solutions, A and B, and our task is to construct a Basic Swap Sequence SS which can act on B to get solution A, we define SS=A−B(Here the sign − also has its new meaning).We can swap the nodes in B according to A from left to right to get SS. So there must be an equation A=B+SS.

For example, two solutions are:

A:(1 2 3 4 5)

B:(2 3 1 5 4)

A(1)=B(3)=1,so the first Swap Operator is SO(1 ,3), B_1=B+SO(1,3) then we get the following result:

B_1:(1 3 2 5 4)

A(2)=B_1(3)=2 ,so the second operator is SO(2,3),and B_2=B_1+SO(2,3),then

B_2=(1 2 3 5 4)

The third operator is SO(4,5),then B_3=A. Finally we get the Basic Swap Sequence SS=A−B=(SO(1 3),SO(2 3), SO(4 5)).

3 The transformation of the formula and the disposal of the factors

Formula (1) has already no longer been suitable for the TSP problem. We update it as follows:

$$V_{id}=V_{id} \oplus \alpha *(P_{id}-X_{id}) \oplus \beta *(P_{gd}-X_{id}) \quad \alpha, \beta \in [0,1] \quad (6)$$

where α, β are random number between 0 and 1. $\alpha *(P_{id}-X_{id})$ means all Swap Operators in Basic Swap Sequence $(P_{id}-X_{id})$ should be maintained with the probability of α, it is the same as $\beta *(P_{gd}-X_{id})$.

From here we can see that the bigger the value of α the greater the influence of P_{id} is, for more Swap Operators in (Pid-Xid) will be maintained, it is also the same as $\beta *(P_{gd}-X_{id})$.

4 TSP_PSO Algorithm Description

TSP_PSO_1

Initialization: each of the particles gets a random solution and a random Swap Sequence, namely velocity.

TSP_PSO_2

If the algorithms are ended, go to TSP_PSO_5.

TSP_PSO_3

For all the particles in position X_{id}, calculating the next position X_{id}'

3-1 Calculating difference between P_{id} and X_{id}, according to the method we have proposed above, $A=P_{id}-X_{id}$, where A is a basic sequence.

3-2 Calculating $B=P_{gd}-X_{id}$,B is also a basic sequence

3-3 Calculating velocity V_{id} according to formula (6), and then we transform Swap Sequence V_{id} to a Basic Swap Sequence.

3-4 Calculating new solution

$$X_{id}=X_{id}+V_{id} \quad (7)$$

Formula (7) means that Swap Sequence V_{id} acts on solution X_{id} to get a new solution.

3-5 Updating P_{id} if the new solution is superior to Pid

TSP_PSO_4

Updating P_{gd} if there is new best solution, which is superior to P_{gd}. Goto TSP_PSO_2.

TSP_PSO_5

Sketch the global best solution.

5 Experiment Results

We use a TSP benchmark problem with 14 nodes to test the validity of our approach. The experiment has been made on a PC (PentiumIV-2GHz CPU, 256M RAM, Win2000 OS, VC++6.0).

Table 1: Description of TSP(14nodes)

Node	1	2	3	4	5	6	7
Coord X	16.47	16.47	20.09	22.39	25.23	22.00	20.47
Coord Y	96.10	94.44	92.54	93.37	97.24	96.05	97.02
Node	8	9	10	11	12	13	14
Coord X	17.20	16.30	14.05	16.53	21.52	19.41	20.09
Coord Y	96.29	97.38	98.12	97.38	95.59	97.13	94.55

The TSP problem is described as Tab.1. The initial random solution is shown by Fig.1, and the best solution we have got is shown by Fig.2

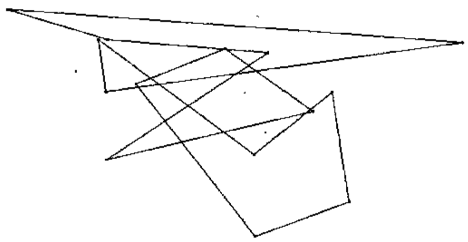

Figure 1 The initial random solution. Cost: 59.8462

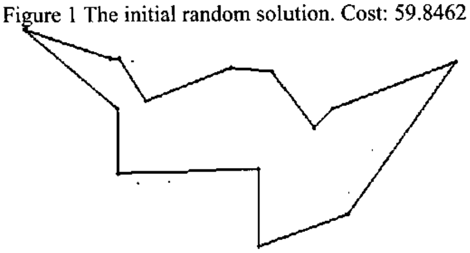

Figure 2 The best solution we have got. Cost: 30.8785

Performance of the experiment:

The size of the solution space:14! /(14*2) = 3113510400

The number of the particles: 100

The number of the iterations of the algorithm: 20000.

The size of search space: 2000000

Search space/solution space: 0.064%

The best solution we have got:

1->10->9->11->8->13->7->12->6->5->4->3->14->2

Cost: 30.8785 (Equal to the optimal solution)

6 Conclusion

The experimental results above show that our algorithm has only searched for very small proportions of the solution space, that is, the speed of convergence is fast, and the space expense of our algorithm is very small. The encouraging result we got is the same as the optimal solution. So it has given us the confidence of further study.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (Grant No.60175024), and by Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education

References

[1] Eberhart R ,Kennedy J. A New Optimizer Using Particles Swarm Theory, Proc. Sixth International Symposium on Micro Machine and Human Science (Nagoya, Japan) IEEE Service Center, Piscataway, NJ:39-43, 1995

[2] Kennedy J, Eberhart R. Particle Swarm Optimization, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948, 1995

[3] Shi Y H, Eberhart R C. Parameter Selection in Particle Swarm Optimization. The 7th Annual Conference on Evolutionary Programming, San Diego, USA,1998

[4] Eberhart R, and Shi Y. Comparison between Genetic Algorithms and Particle Swarm Optimization.The 7thAnnual Conference on Evolutionary Programming, San Diego, USA,1998

[5] Angeline P. Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Difference.The 7th Annual Conference on Evolutionary Programming, San Diego,USA,1998

[6] Kennedy J, and Spears W. Matching Algorithms to Problems: An Experimental Test of the Particle Swarm and some Genetic Algorithms on the Multimodal Problem Generator. IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, USA,1998

[7] Huang Lan,Zhou Chunguang,Wang Kangping.Hybrid Ant Colony algorithm for Traveling Salesman Problem.Progress In Natural Science.Vol 13 No.4(China),April 2003

[8] Zhou C G et al. Computing Intelligence (in Chinese) Changchun: Publishing House of Jilin University, 2001