Particle Swarm Optimization with Simulated Annealing for TSP

LUPING FANG, PAN CHEN, SHIHUA LIU
Software Institute of Zhejiang University of Technology
Zhejiang University of Technology, Zhaohui No.6 Area, Hangzhou, 310014
CHINA

Abstract: -Aiming at the shortcoming of basic PSO algorithm, that is, easily trapping into local minimum, we propose an advanced PSO algorithm with SA and apply this new algorithm for solving TSP problem. The core of algorithm is based on the PSO algorithm. SA method is used to slow down the degeneration of the PSO swarm and increase the swarm's diversity. The comparative experiments were made between PSO-SA, basic GA, basic SA and basic ACA on solving TSP problem. Results show PSO-SA is more superior to other methods.

Key-Words: - particle swarm optimization; simulating annealing algorithm; TSP; GA

1 Introduction

The travel salesman problem (TSP) has been studied extensively over the past several decades. In this paper，this problem is solved by a new method called PSO with SA.

The Particle Swarm Optimization(PSO) algorithm was first introduced by Kennedy and Eberhart in 1995[1-2] and is widely used to solve continuous quantities problems. Recently, some researches apply this algorithm to problems of discrete quantities. It has been reported that PSO has better performance in solving some optimization problems. However, 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. 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], Hybrid PSO [4], Intelligent PSO [5], Niching PSO [6] and Guarantee Locally Convergent PSO[7]. have been proposed to increase the diversity of particles and improve the convergence performance

In the paper, we proposed a new solution which combines PSO algorithm with the simulated annealing algorithm (SA) and apply it to solve TSP problem. SA is a kind of stochastic method and is well known for its feature of effective escaping from local minimum trap. By integrating SA to the PSO, the new algorithm, which we call it PSO-SA can not only escape from local minimum trap in the later phase of convergence, but also simplify the implementation of the algorithm. In the experiments, four algorithms have been used and the results have been compared, among which PSO-SA algorithm has the best performance in solving TSP problem.

2 Basic PSO Algorithm

The concept of PSO roots from the social behaviour of organisms such as bird flocking and fishing schooling. Through cooperation between individuals, the group often can achieve their goal efficiently and effectively. PSO simulates this social behaviour as an optimization tool to solve some optimization problems. In a PSO system, each particle having two properties of position and velocity represents a candidate solution to the expressed by the objective function. In the iteration, the objective function is calculated to establish the fitness value of each particle using position as input. Fitness value determines which position is better. Each particle flies in the search space with a velocity that is dynamically adjusted based on its own flying experience and its companions' flying experience. In other word, every particle will utilize both the present best position information of its own (*pbest*) and the global best position information (*gbest*) that the swarm has searched up-to-now to change its velocity and thus arrives in the new position.

PSO can be described mathematically as follows.

Suppose that the search space is of d-dimension and the number of particles is n. The i th particle is represented by a d-dimension vector $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$; $pbest_i = (p_1, p_2, ...p_D)$ denotes the best position searched by the i th particle and the $gbest = (g_1, g_2, ..., g_n)$ is the best position searched by the whole swarm up-to-now.

Each particle updates its velocity and position according the following equations.

$$V_{id} = w \times V_{id} + c_1 \times rand()(pbest_{id} - X_{id}) + c_2 \times rand()(gbest_d - X_{id}) \quad (1)$$

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

where w is the inertia coefficient which is a chosen constant in interval [0,1]; c_1, c_2 are two acceleration constants; $rand()$ is random value in interval [0,1]. The velocities of particles are restricted in interval $[V_{\min}, V_{\max}]$. If the resulting value is smaller than $V_{\min}$, one element of the velocity vector is set to $V_{\min}$; if the resulting value is greater than $V_{\max}$, one element of velocity vector is set to $V_{\max}$.

The method described above is suitable for problems of continuous quantities. However, it can't be applied directly to problems of discrete quantities. Wang et al. defined the "swapping operator" for solving TSP problem based on PSO [8]. However, the drawback of being easily trapped into local minima still exists. The following new algorithm was proposed to overcome this shortcoming.

3 Applying the Simulated Annealing to Particle Swarm Optimization in Solving TSP Problem

To avoid trapping into local minimum and increase the diversity of particle swarm, the simulated annealing algorithm is introduced into PSO to solve TSP problem

The simulated annealing (SA) is a stochastic strategy for selecting the minimal configuration of states of a system. It was first put forward by Metropolis and successfully applied to the optimization problems by Kirkpatrick [9]. The solution space S of n-city TSP problem can be described as a collection of arrangements of {1,2,…,n}. Every arrangement C_i represents a route that the salesman follows to visit every city once time. So S can be expressed that $S = \{C_i \mid C_i \text{ is a cycle array of } \{1,2,...,n\}\}$. Energy function E in SA algorithm becomes $f(C_i)$, which represents the length of route C_i . $\Delta f = f(C_{new}) - f(C_{old})$ is used as ΔE , representing the energy difference of two energy states. Starting from an initial solution, the algorithm unconditionally accepts the solution which results in smaller energy value than the last solution. The decision to accept or reject a new solution with larger energy value is made according to a probability function. In this paper, a new acceptable decision rule for TSP is designed as follows.

$$P = \begin{cases} 1 & \Delta f \le 0 \\ \exp(-\frac{\Delta f}{t_i}) & \Delta f > 0 \end{cases} \quad (3)$$

Equation (3) selects solutions according the value Δf . When Δf is non-positive, which means that the new route is shorter than the old one, the new route is always accepted. In the case of Δf >0, firstly the acceptable probability p should be calculated according to equation (3). If $P > rand(0,1)$ (rand(0,1) is a number generated randomly between 0 and 1), the new route is also accepted, otherwise the new route will be rejected. In equation (3), t is an important control parameter which is referred to as temperature. The temperature decreases during each iteration which in turns affects the acceptance of new route. As the temperature decreases, the acceptance possibility of a degraded route decreases. The possible cooling schedule in this paper is

$$T = (\alpha)^i T_0 + T_\theta \quad (4)$$

where cooling coefficient α is a random constant between 0 and 1, i is the number of iterations operated so far, T_0 is the initial temperature and T_θ is the lowest temperature value.

Combining the fast optimal search ability of PSO with the probability jump property of SA, we design a new algorithm frame to solve TSP problem. The main idea is that at first every particle searches its local best route C_{ilbest} using SA algorithm to update individual personal best route C_{ipbest} and the global best route C_{gbest}. $c_1(pbest_k - x_k) + c_2(gbest_k - x_k)$ in equation (1) can be viewed as the crossover operation of genetic algorithm (GA) to generate new route.

Step1: Initialization.

Configure swarm size m and initialize all particles $C_1, C_2, ..., C_m$. C_i can be expressed as:

$C_i = \{s[1], s[2], ...s[n]\}$, where $s[i] = j$ means that city j is visited in order i and $s[n+1] = s[1]$.

Configure T_0, T_θ, α and L. T_0 is the initial temperature . The higher the initial temperature, the better the result is. T_θ is the lowest temperature value. At low temperature, every particle finds its local best route C_{ilbest} in its local area. α is cooling coefficient which is a random constant between 0 and 1. L is maximum number of iterations in a certain temperature.

Step2: Every particle C_i searches for its local best route C_{ilbest}

In one iteration, every particle C_i generates a new route C_i' in its local area and then according to the accepting rule of SA decides whether to accept the new route or not. After L iterations, every particle finds its local best route C_{ilbest}. Here, exchange method is used to generate a new route. We will describe it in the following.

N cities are numbered 1,2,3,… and n respectively. Assume s[u] is a city number corresponding to the city which the salesman visits the *u*th order in the visiting order number sequence. Two cities s[u],s[v] are selected randomly in route C_i and their visiting order u and v are exchanged, while the visiting order of the other cities remains the same. After exchanging, a new route C_i' is generated.

Assume the city visiting order sequence of route C_i is:

$\ldots s[u-1]s[u]s[u+1]\ldots s[v-1]s[v]s[v+1]\ldots$

After exchanging the visiting order of city u and city v, the new route C_i' is:

$\ldots s[u-1]s[v]s[u+1]\ldots s[v-1]s[u]s[v+1]\ldots$

Step3: Update personal best route C_{ipbest} and the global best route C_{gbest}

Compare the fitness value of C_{ipbest} and C_{ilbest}. The route with smaller fitness value will replace C_{ipbest}.

$$C_{ipbest} = \begin{cases} C_{ilbest} & f(C_{ilbest}) < f(C_{ipbest}) \\ C_{ipbest} & f(C_{ilbest}) \geq f(C_{ipbest}) \end{cases} \quad (5)$$

After updating every particle's personal best value, we can get the new global best value C_{gbest}.

Step4: Obtain the new route C_{new} by the crossing operation.

Particle C_i crosses with the new C_{ipbest} and C_{gbest} separately to update itself. There are four crossing strategies reviewed in literature [10] and one of them is adopted in our work. Suppose two routes old1=1 2 3 4 5 6 7 8 9 and old2=9 8 7 6 5 4 3 2 1. First we randomly select a crossing area in old2, then insert the city numbers in the crossing area to old1 at the same position as in old2, finally delete the duplicate cities in old1 that appear outside the crossing area. For example, the crossing area in old2 we choose is 7 6 5 4, after the crossing operation the new route will be 1 2 7 6 5 4 3 8 9.

Step5: Calculate new temperature T specified in equation (4). If $T \leq T_\theta$, the algorithm finds the best route C_{gbest} and comes to the end. Otherwise go to Step2.

4 Experimental Results

For comparison, three other artificial algorithms including basic SA, basic GA and Basic Ant Colony Algorithm are used to solve the same TSP problem (Oliver30 and Att48). We used the following configuration for our new algorithm: the swarm size m = 30, initial temperature $T_0 = 10000$, the lowest temperature $T_\theta = 1$, cooling coefficient $\alpha = 0.99$. All four algorithms were run 20 times for Oliver30 and Att48 TSP problems, the results are shown in Table 1 and Table 2.

Table 1 Results for Oliver30 problem

Algorithm	Oliver30		
	Average Value	Best Solution	Worst Solution
Basic SA	437.6632	424.9918	480.1452
Basic GA	482.4671	468.1532	504.5256
Basic ACA	447.3256	440.8645	502.3694
PSO-SA	426.6856	423.7405	430.7122

Table 2 Results for Att48 problem

Algorithm	Att48		
	Average Value	Best Solution	Worst Solution
Basic SA	34980	35745	41864
Basic GA	37548	36759	43561
Basic ACA	35684	34559	42256
PSO -SA	34512	33966	35101

From the results of the four algorithms, it is clear that our algorithm is significantly better than the other algorithms. For Oliver30 problem, the best fitness value 423.7405 achieved by PSO with SA algorithm is not only the smallest value in these four algorithms but also the best value that has been obtained so far we know. Meanwhile in the worst value case and average 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. Figure 4.1 illustrates the best solution of Oliver30 by applying PSO with SA algorithm. Figure 4.2 shows the best solution of att48 by applying PSO with SA algorithm.

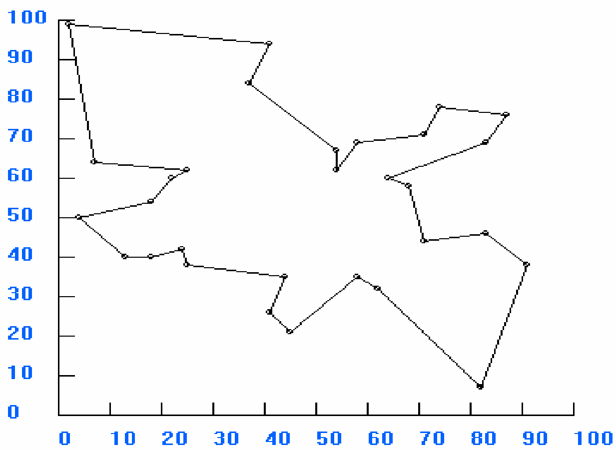

Fig.1 The best solution of Oliver30 by applying PSO with SA algorithm.

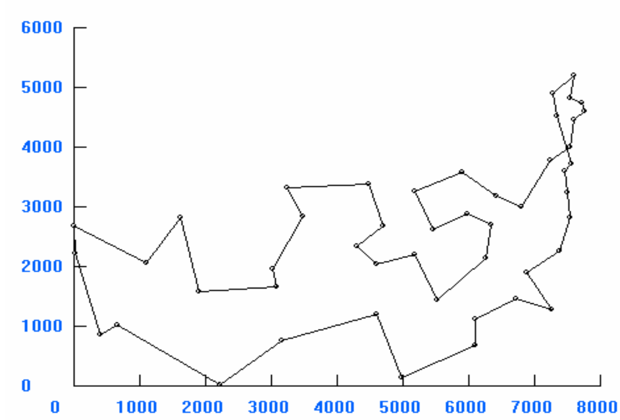

Fig.2 The best solution of att48 by applying PSO with SA algorithm

4 Conclusions

In this paper, we proposed a hybrid algorithm which integrates particle swarm optimization with simulating annealing algorithm to solve the well-known TSP problem. This new algorithm is simple and easily to implement. The experimental results show that the new algorithm has competitive potential for solving discrete optimization problems.

References:

[1] Eberhart RC, Kennedy J, A new optimizer using particles swarm theory[A], *Proc Sixth Int Symposium on Micro Machine and Human Science[C]*, 1995, pp. 39-43.

[2] Shi YH, Eberhart RC, A modified particle swarm optimizer[A],*IEEE Int Conf on Evalutionary Computation [C]*, 1998, pp. 63-73

[3] Shi YH, Eberhart RC, Fuzzy Adaptive Particle SwarmOptimization, *Congress on Evolutionary Computation*, 2001, pp.101-106

[4] Lovbjerg M, Rasmussen TK, Krink T, Hybrid Particle Swarm Optimizer with Breeding and Subpopulation, *Proceedings of Evolutionary Computation Conference*, 2001

[5] Ciuprina G, Ioan D, Munteanu I, Use of Intelligent-Particle Swarm Optimization in Electmmagnetics, *IEEE Transactions on Magnetics*, Vol.38, No.2, 2002, pp. 1037-1040

[6] Brits R, Engelbrecht AP, A NichingParticle Swarm Optimizer, *The 4th Asia-Pacific Conference on Simulated Evolution and Learning*, 2002

[7] Bergh ED, Engelbrecht AP, A New Locally Convergent Particle Swarm Optimizer, *IEEE Conference on Systems, Man, and Cybernetics*,2002

[8] Wang KP et al, Particle Swarm Optimization for Traveling Salesman Problem, *International Conference on Machine Learning and Cybernetics*, Vol.3, 2003, pp. 1583-1585

[9] Kirkpatrick S ,Simulated Annealing, *Sci*,Vol.220, 1983,pp.671-680.

[10] Gao S, Han B, Wu XJ, Solving travelling salesman problem by hybrid particle swarm optimization algorithm, *Control and Decision*,Vol.19, No.11, 2004, pp. 1286-1289