

Solving City Routing Issue with Particle Swarm Optimization

Sarman K. Hadia

Associate Professor,
Charotar University Of Science
& Technology, Changa,
Gujarat, India

Arjun H. Joshi

Chaitalee K. Patel
PG Student,
Charotar University Of Science
& Technology, Changa,
Gujarat, India

. Yogesh P Kosta

Director, Marwadi Group of
Institutions,
Rajkot, Gujarat, India

ABSTRACT

The city routing issue is the problem to find a shortest tour of minimum length on a fully connected graph. Various Nature-inspired algorithms have been proposed towards this problem.

This paper proposes an application of Particle Swarm Optimization for this Issue. Results are achieved with the concept of Swap Operator and Sequence of Swap.

Keywords

Particle Swarm Optimization (PSO), City Routing, Bird flocking, Basic Swap Sequence (BSS), Vehicular Ad-Hoc Network (VANET)

1. INTRODUCTION

City Routing Issue can be described as follows: Given a graph $G = (L, E)$, where $L = \{1, \dots, n\}$ and $E = \{1, \dots, e\}$. Here Cost K is associated with each edge linking vertices i and j and the problem consists in finding the minimal total length cycle of G . The length can be found out by the summation of the costs of the edges in the considered cycle. This problem is having different applicability for different areas such as genome map, VLSI chip fabrication, Vehicular routing and broadcast schedule, among others.

Social psychologist James Kennedy and Professor Russell C. Eberhart had simulated the natural swarming behavior of birds as they search for food to introduce particle swarm optimization [5]. The concept of swap is applied here in the PSO algorithm to solve the City Routing Issue [3]. The Shortest tour cycle is formed here with this Approach.

In this paper Section 2 describes the basics of optimization, PSO as a Member of Swarm intelligence and PSO Algorithm. City Routing Problem is discussed in section 3. Section 4 describes concept of swap and Swap Sequence. Simulation results and related discussion for 5 and 10 nodes scenario is described in section 5. Finally section 6 concludes the paper.

2. BACKGROUND

Optimization can be defined as the search for a set of variables that either minimize or maximize a cost function, $f(x)$. There are various algorithms for the optimization. Particle Swarm optimization is most popular algorithm that can be used to optimize functions.

2.1 PSO As A Member Of Swarm Intelligence

PSO converges quickly relative to other population-based optimization algorithms such as GA and also offering good solution quality [4]. Each particle in PSO flies through the search space by updating its individual velocity at regular intervals toward the best position or location it personally has found (i.e. the personal best), and toward the globally best position found by the entire swarm (i.e. the global best).

2.2 Basic PSO Algorithm

The concept of PSO roots from the social behaviour of organisms such as bird flocking and fish schooling. Through cooperation between individuals, the group often can achieve their goal more efficiently in a effective manner [5]. PSO simulates this social behaviour as an optimization tool to solve some optimization Issues. In a PSO system, each particle having two properties of position and velocity represents a candidate solution which is expressed by the objective function. In the iteration, the objective function is finding out to establish the fitness value of each particle using position as input. Fitness value indicates which position is better. Each particle flies in the search space with a velocity which 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 modify its velocity and thus arrives in the new position.

PSO can be described in mathematical terms 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 $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$; $pbest_i = (P_1, P_2, \dots, P_d)$ denotes the best position searched by the i^{th} particle and the $gbest = (g_1, g_2, \dots, g_n)$ is the best position searched by the whole swarm up-to-now.

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

$$Vid = w \times Vid + c1 \times rand() (pbestid - Xid) + c2 \times rand() (gbestid - Xid)$$

$$Xid = Xid + Vid \quad (1)$$

Where w is the inertia coefficient which is a chosen constant in interval $[0, 1]$; $c1, c2$ 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. has defined the "swapping operator" for solving this kind of problem based on PSO [10].

3. CITY ROUTING PROBLEM

City routing issue can be described as follows: a Person has to visit other Persons located in various cities in his region, and he need to find cheapest tour that ensures that all cities are visited.

The cities are called nodes and roads connecting the cities are called arcs in network terminology [9]. The set of nodes to be visited are defined as $L = \{1, 2, \dots, n\}$ where n is the total number of nodes, and the set of arcs connecting the nodes is defined as $E = \{(i,j): i, j \in L, i \neq j\}$, where the pair (i,j) indicates the arc between node i and j . There are direct links between pair of nodes, which can be referred as a complete graph. The graph consisting of the nodes L and arcs E is then connected. Numbers of visits are exactly one for each node. If a person travels directly from node i to node j then a common definition of the set of decision variables is $X = \{X_{ij}: i, j \in L, i \neq j\}$ where $X_{ij}=1$, otherwise it is 0. The cost of traversing from node i to node j is defined as $K = \{k_{ij}: i, j \in L, i \neq j\}$.

4. CONCEPT OF SWAP

Take a solution sequence of this problem with n nodes, $M = (L_i), i=1 \dots n$. $MO(i_1, i_2)$ can be define as exchanging node L_{i_1} and node L_{i_2} in solution M . Then $M' = M + MO(i_1, i_2)$ can be defined as a new solution on which operator $MO(i_1, i_2)$ acts. Suppose there is a city routing problem with Five nodes, solution: $M = (6, 7, 8, 9, 10)$. The Swap Operator is $MO(3, 4)$, then

$$M' = M + MO(3,4) = (6, 7, 8, 9, 10) + (3,4) = (6, 7, 9, 8, 10) \quad (2)$$

4.1 Sequence of Swap

A Sequence of swap MM is having one or more Swap Operators. $MM = (MO_1, MO_2, MO_3, \dots, MO_n)$, where $MO_1, MO_2, MO_3, \dots, MO_n$ are Swap Operators.

4.2 Sequence of swap feature

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

$$M' = M + MM = M + (MO_1, MO_2, MO_3, \dots, MO_n) = (M + MO_1) + MO_2 + \dots + MO_n$$

Various Swap Sequences acting on the same solution may create the same new solution. All these Swap Sequences are called the equivalent set of Swap Sequences. In the equivalent set, the sequence which is having the least Swap Operator is called Basic Swap Sequence of the set. Several Swap Sequences can be merged into a new Swap Sequence. Denoting the operator \oplus as merging two Swap Sequences into a new swap Sequence. Suppose there is two Swap Sequences, $MM1$ and $MM2$, $MM1$ and $MM2$ act on one

solution M in order, namely $MM1$ first, $MM2$ second, A new solution M' can be get, and there is another Swap Sequence MM' acting on the same solution M , then the Same solution M' can be get, described as $MM' = MM1 \oplus MM2$, where MM' and $MM1 \oplus MM2$ are in the same equivalent set.

4.3 Basic Sequence of Swap Construction

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

For example, two solutions are:

$C: (1\ 2\ 3\ 4\ 5)$ and $D: (2\ 3\ 1\ 5\ 4)$

Since $C(1) = D(3) = 1$, so the first Swap Operator is $MO(1, 3)$, $D1 = D + MO(1, 3)$ then the following result can be get:

$D1: (1\ 3\ 2\ 5\ 4)$

Since $C(2) = D1(3) = 2$, So the second operator is $MO(2,3)$, and $D2 = D1 + MO(2,3)$, then

$D2 = (1\ 2\ 3\ 5\ 4)$

The third operator is $MO(4, 5)$, then $D3 = C$. Finally we get the Basic Swap Sequence $MM = C - D = (MO(1\ 3), MO(2\ 3), MO(4\ 5))$.

4.4 Modified Velocity Update Equation

Formula (1) has already no longer been suitable for the TSP problem. It can be update as follows:

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

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 . $\beta * (P_{gd} - X_{id})$ means all Swap Operators in Basic Swap Sequence $(P_{gd} - X_{id})$ should be maintained with the probability of β . From here it can be seen that the bigger the value of α the greater the influence of P_{id} is, for more Swap Operators in $(P_{id} - X_{id})$ will be maintained, it is also the same as $\beta * (P_{gd} - X_{id})$.

4.5 Steps For Solution Algorithm

Step 1:

Each of the particles gets a random solution and a random Swap Sequence, namely velocity.

Step 2:

If the algorithms are ended, go to step 5.

Step 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 that has been Shown 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 (3), and then transform swap Sequence V_{id} to a basic Swap Sequence,
- 3.4 Calculating new solution

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

Equation (4) means that Swap Sequence V_{id} acts on solution X_{id} to get a new solution.

- 3.5 If the new solution is superior to P_{id} then update P_{id} .

Step 4:

If there is new best solution; which is Superior to P_{gd} then update P_{gd} . Go to step 2.

Step 5:

Draw the global best solution.

5. SIMULATION RESULTS

Here Simulation results are divided in two cases for 5 nodes and 10 nodes.

Case 1:

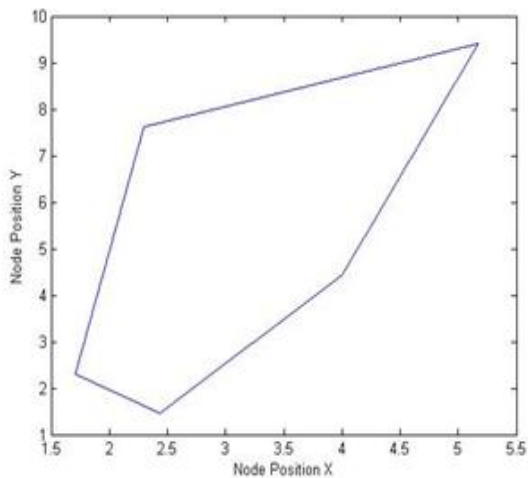


Figure 1: 5 Nodes Complete cycle

Here the Number of Nodes Used is 5. Maximum no. Of iterations used are 200. Value of α and β is 0.85. Figure 1 shows the 5 Nodes Complete Cycle for the given Node positions. Figure 2 shows the plot between Iteration and Tour Distance and it also shows plot between Iteration and Best Tour length. Here the best shortest length for the tour found is 18.3240 which is average of all 30 simulation Runs. Tour Distance is going to be decreased with Iteration.

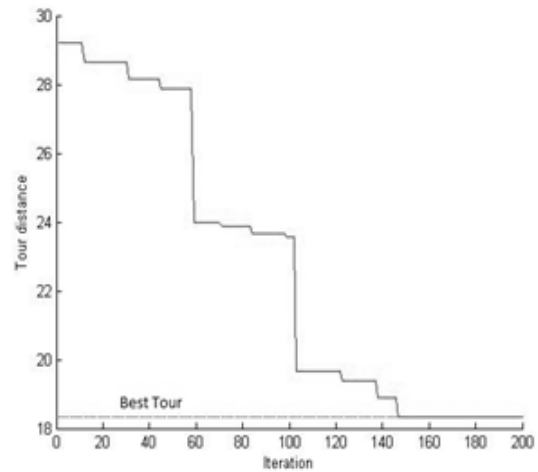


Figure 2: Iteration vs. Tour Distance

Case 2:

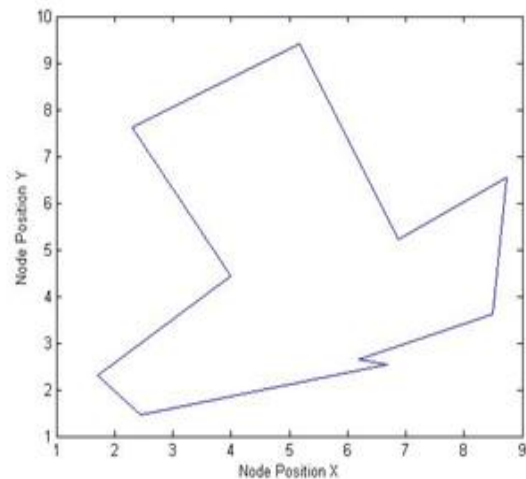


Figure 3: 10 Nodes Complete cycle

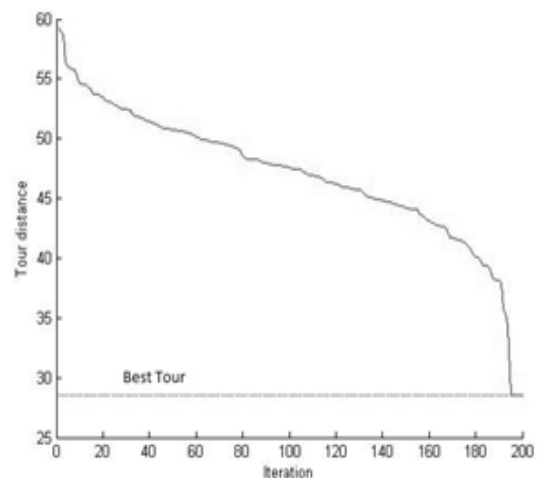


Figure 4: Iteration vs. Tour Distance

Table 1: Tour best Distance value for particular Simulation Run

Run No	Tour Best value	Run No	Tour Best value	Run No	Tour Best value
1	28.1599	11	27.7822	21	28.304
2	29.6691	12	26.9067	22	28.304
3	28.304	13	27.7418	23	28.304
4	28.3617	14	27.6933	24	28.304
5	27.7418	15	26.9067	25	28.304
6	27.7822	16	26.9067	26	28.3617
7	28.1599	17	28.3524	27	28.1599
8	26.9067	18	27.5174	28	28.304
9	28.9465	19	28.3928	29	27.7822
10	28.3928	20	28.3617	30	28.304

Here the Number of Nodes Used is 10. Maximum no. Of iterations used are 200. Value of α and β is 0.85. Figure 3 shows the 10 Nodes Complete Cycle for the given Node positions. Figure 4 shows the plot between Iteration and Tour Distance and it also shows plot between Iteration and Best Tour length. Table 1 shows Tour best Distance value for particular Simulation Run. Here the best shortest length for the tour found out is 28.01809 which is average of all 30 simulation runs. Tour Distance is going to be decreased with Iteration.

6. CONCLUSION

Particle swarm optimization algorithm is hereby used to solve the city routing issue. Here the algorithm performs well for small no. of Cities. This can be improved to work for more no. of cities. This algorithm can also be applied for Vehicular Ad-Hoc Network (VANET) routing issue as a further work.

7. REFERENCES

- [1] Angeline P. Evolutionary Optimization versus Particle Swarm Optimization: Philosophy and Performance Difference. The 7th Annual Conference. On Evolutionary Programming, San Diego, USA, 1998.
- [2] Huang Lan Zhou Chunguang, Wang Kangping. Hybrid Ant Colony algorithm for Traveling Salesman Problem. Progress In Natural Science. Vol 13 No.4(Cbina), April 2003.
- [3] Kang-Ping Wang, lan Huang, Chun-Guang Zhou, Wei Pang, “ Particle Swarm Optimization For Travelling Salesman Problem”, IEEE 2003.
- [4] 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.
- [5] Kennedy J, Eberhart R. Particle Swarm Optimization, IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, IV: 1942-1948, 1995.
- [6] KP. Wang, L. Huang, C.G. Zhou, W. Pang, Particle swarm optimization for traveling salesman problem, International conference on Machine Learning and Cybernetics 3 (2003) 1583-1585
- [7] M. Clerc, in: Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem New Optimization Techniques in Engineering, Springer, 2004, pp. 219–239.
- [8] M. Clerc, Discrete particle swarm optimization illustrated by the traveling salesman problem, <http://www.mauriceclerc.net>, 2000.
- [9] Shuang Cong, Yajun Jia and Ke Deng, “ Particle Swarm And Ant Colony Algorithms and their Applications in Chinese Traveling Salesman Problem”
- [10] Zhou C G et al. Computing Intelligence (in Chinese) Changchun: Publishing House of Jilin University, 1585.