International Conference on Computer and Communication Engineering (ICCCE 2012), 3-5 July 2012, Kuala Lumpur, Malaysia

Particle Swarm Optimization with Partial Search to Solve Traveling Salesman Problem

M. A. H. Akhand
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
akhand@cse.kuet.ac.bd

Shahina Akter
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
shahinak23@gmail.com

S. Sazzadur Rahman
Dept. of Computer Science and Engineering
Khulna University of Engineering & Technology
Khulna, Bangladesh
onlysumon@yahoo.com

M. M. Hafizur Rahman
Dept. of Computer Science
KICT, International Islamic University Malaysia
Jalan Gombak, Malaysia
hafizur@iium.edu.my

Abstract— **Particle Swarm Optimization (PSO) is population based optimization technique on metaphor of social behavior of flocks of birds and/or schools of fishes. For better solution, at every step each particle changes its velocity based on its current velocity with respect to its previous best position and position of the current best particle in the population. PSO has found as an efficient method for solving function optimization problems, and recently it also studied to solve combinatorial problems such as Traveling Salesman Problem (TSP). Existing method introduced the idea of Swap Operator (SO) and Swap Sequence (SS) in PSO to handle TSP. For TSP, each particle represents a complete tour and velocity is measured as a SS consisting with several SOs. A SO indicates two positions in the tour that might be swap. In the existing method, a new tour is considered after applying a complete SS with all its SOs. Whereas, every SO implantation on a particle (i.e., a solution or a tour) gives a new solution and there might be a chance to get a better tour with some of SOs instead of all the SOs. The objective of the study is to achieve better result introducing using such partial search option for solving TSP. The proposed PSO with Partial Search (PSOPS) algorithm is shown to produce optimal solution within a less number of generation than standard PSO, Genetic Algorithm in solving benchmark TSP.**

Keywords- Particle Swarm Optimization (PSO), Traveling Salesman Problem (TSP) and Swap Sequence.

I. INTRODUCTION

Particle Swarm Optimization (PSO) is a population based optimization technique on metaphor of social behavior of flocks of birds and/or schools of fishes [1]. Particles or members of the swarm fly through a multidimensional search space looking for a potential solution. Each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors. PSO has found as an efficient method for solving function optimization problems, and recently it also studied to solve combinatorial problems such as Traveling Salesman Problem (TSP). Swap Operator based PSO [2], Enhanced Self-Tentative PSO [4], Discrete PSO [5] are the different PSO based methods for solving TSP problem. Among the methods, Swap Operator based PSO [2] is the pioneer and studied for other methods [3-7].

In the basic PSO model, the position of the i-th particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ in the D-dimensional search space. For TSP, each particle's position represents a complete tour. At every step each particle changes position based on its velocity represented as $V_i = (v_{i1}, v_{i2}, \ldots v_{iD})$. The velocity of a particle depends on its previous best position, $P_{ibest} = (p_{i1}, p_{i2} \ldots p_{iD})$ and the best one among all the particles in the population $G_{best} = (g_1, g_2 \ldots g_D)$.

The velocity for solving TSP in the Swap Operator based PSO is a Swap Sequence (SS) [2]. A SS is consists with several Swap Operators and may define as SS= (SO_1, SO_2, SO_3... SO_n), where SO denotes Swap Operator. A SO indicates two positions in the tour that might be swap. All swap operators of a SS are applied maintaining order on a particle and gives a new tour i.e., a particle having new solution in the PSO.

In the existing method, the new tour is considered after applying all the SOs of a SS and no intermediate measure is considered. It is notable that every SO implantation gives a new tour; and therefore, there might be a chance to get a better tour with some of SOs instead of all the SOs. The objective of the study is to achieve better result introducing using such partial search in the SS based PSO for solving TSP.

In the proposed PSO with Partial Search (PSOPS) method, a number of particles are initialized and at every step new position of a particle is measured with the velocity applying partial SS with some of SOs instead of all SOs. The proposed PSOPS algorithm is shown to produce optimal solution within a less number of iteration than standard PSO in solving benchmark TSP.

The outline of the paper is as follows. Section II explains our proposed PSO with Partial Search (PSOPS) for solving TSP. To make the paper independently readable this section also gives a brief description existing Swap Sequence (SS) based PSO. Section III is for experimental studies of the proposed method comparing performance with other methods in solving benchmark TSP. At last, we came up with the conclusions with discussions of our proposed methodology in Section IV.

II. PSO WITH PARTIAL SEARCH (PSOPS) FOR TSP

The proposed PSO with Partial Search (PSOPS) method is based on the PSO based on Swap Sequence (SS) [2]; we first give a short description of the method with SS and Swap Operator (SO). Finally we present our proposed PSOPS.

A. Swap Operator(SO)

A SO indicates two positions in the tour that might be swap for a new tour from a tour. Suppose there is a TSP problem with five nodes, and a solution (i.e., particle or tour): S= (l, 2, 3, 5, 4). The Swap Operator is SO (1, 2), then new solution S' is:

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

Here meaning of '+' is to conceive swap operator and change the position of cities in the solution indicated in the operator.

B. Swap Sequence(SS)

A Swap Sequence (SS) is made up of one or more SOs.

$$SS = (SO_1, SO_2, SO_3, \ldots SO_n) \quad (2)$$

SO_1, SO_2, ... SO_n are Swap Operators, here the order of the Swap Operator in SS is important [2]. Apply a SS on a solution we may get a new solution. If we get S2 apply as SS on S1 than it becomes:

$$S_2 = S_1 + SS_{12} = S_1 + (SO_1, SO_2, SO_3, \ldots SO_n) \quad (3)$$

$$SS_{12} = S_2 - S_1 = (SO_1, SO_2, SO_3, \ldots SO_n) \quad (4)$$

It is notable that different SSs acting on the same solution may produce the same new solution. All these SSs are named the equivalent set of SSs. In the equivalent set, the sequence which has the least SOs is called Basic Swap Sequence of the set or Basic Swap Sequence (BSS) in short. As an example, if S_1: (1 2 3 4 5) and S_2: (2 3 1 5 4) than SS_{12}= SO(1,3); SO(2,3); SO(4,5).

C. PSO to Solve TSP

In PSO, at every step each particle changes position based on its velocity that depends on its previous best position and the best one among all the particles in the population. For TSP it may represented as follows:

$$X_i^{(t)} = X_i^{(t-1)} + V_i^{(t)} \quad (5)$$

$$V_i^{(t)} = V_i^{(t-1)} \otimes \alpha(P_{ibest}^{(t-1)} - X_i^{(t-1)}) \otimes \beta(G_{best}^{(t-1)} - X_i^{(t-1)}) \;\; \alpha,\beta \in [1,0] \quad (6)$$

In the above equations X_i is the position of i-th particle that instance to a solution or a complete tour for TSP and P_{ibest} is its previous best position; and G_{best} is the position of the best particle in the population at that time.

In Eq (4) $P_{ibest} - X_i$ and $G_{best} - X_i$ are SSs and the operator $\otimes$ means to merge two SSs into a new SS. Therefore, V_i is a SS with one or more SOs merging three SSs. In the equation α and β are random number between 0 and 1 and larger value of any of them influence to consider more SOs in SS in V_i from its portion. Therefore, the bigger the value of α, the greater the influence of P_{ibest} is and it is also the same for β on G_{best}.

D. PSO with Partial Search (PSOPS) to Solve TSP

We introduced a partial search option in PSO of previous section to solve TSP modifying Eq. 5 and 6. The problem we identified in Eqs. 5 and 6 that $V_i^{(t)}$ does not coincide with straightforward meaning of the velocity of original PSO for function optimization where full value of $V_i^{(t)}$ usually gives better result with respect to its portion. In case of TSP, implementation every SO from SS of $V_i^{(t)}$ on $X_i^{(t-1)}$ gives a new solution and solution with some of the SOs may give better than with all the SOs. Considering the above two points we worked with partial search where velocity is calculated as like PSO of Eq. 6 but this velocity is considered as tentative velocity($V'^{(t)}_i$). SOs from this tentative velocity is then apply on $X_i^{(t-1)}$ one after another; and tour cost is calculated after applying each and every SO since each SO give new tour from previous. In the proposed method next solution is the best one among all the solutions and final velocity that require for next generation is calculated according to Eq. 9, the SS between $X_i^{(t)}$ and $X_i^{(t-1)}$. Although Eq. 8 seems to increase computational cost due intermediate fitness calculation but overall might decrease finding better result with minimal SOs as well as time.

$$V'^{(t)}_i = V_i^{(t-1)} \otimes \alpha(P_{ibest}^{(t-1)} - X_i^{(t-1)}) \otimes \beta(G_{best}^{(t-1)} - X_i^{(t-1)}) \;\; \alpha,\beta \in [1,0] \quad (7)$$

Suppose $V_i^{(t)} = SO_1, SO_2, \ldots\ldots SO_n$ then

$X^{1}{}_i^{(t)} = X_i^{(t-1)} + SO_1$; $X^{2}{}_i^{(t)} = X^{1}{}_i^{(t)} + SO_2 = X_i^{(t-1)} + SO_1 + SO_2$;

$X^{n}{}_i^{(t)} = X^{n-1}{}_i^{(t)} + SO_n$

$X_i^{(t)} = X^{j}{}_i^{(t)}$ where $X^{j}{}_i^{(t)}$ belongs minimum Tour Cost among

$$X^{1}{}_i^{(t)}, X^{2}{}_i^{(t)} \ldots X^{j}{}_i^{(t)} \ldots X^{n}{}_i^{(t)}. \quad (8)$$

$$V_i^{(t)} = X_i^{(t)} - X_i^{(t-1)} \quad (9)$$

Figure 1 shows the pseudo-code of the proposed PSO with Partial Search (PSOPS) using Eq. 7 and 8 to solve TSP. The stopping criteria might be fixed number generation or a define gradient of G_{best}.

1. *Initialize particles with random tours. Calculate fitness of each particle and copy as P_{best}; identify G_{best} in the population.*
2. *For each particle*
   a. *Calculate tentative velocity $V'^{(t)}_i$ using Eq. 7.*
   b. *Calculate new solution (i.e., tour) $X_i^{(t)}$ using Eq. 8.*
   c. *Calculate final velocity $V_i^{(t)}$ using Eq. 9.*
   d. *Update P_{best} with current new solution (i.e., tour) if new one is better than exiting P_{best}.*
3. *Update G_{best} if there is a new best solution, which is superior to G_{best}.*
4. *If stopping criteria reach then take G_{best} as a solution; otherwise go to Step 2.*

Fig. 1: PSO with Partial Search (PSOPS) algorithm for TSP.

III. Experiment Studies

This section describes the benchmark problems and experimental settings and then presents experimental results. The selected problems for this study are burma14, eil51, berlin52, eil76, kroA100 and kroA200. A numeric value in the problem name represents the number of cities in that tour. The number of cities varied from 14 to 200 in the selected problems and give diverse test bed. The selected problems are also used in several previous studies. A city is represented as a coordinate in a problem. Therefore, tour cost is found after calculating distance using the coordinates. Table I describes the coordinates of 14 cities of burma14 problem.

For proper understanding, we also solved the problems with Genetic Algorithm (GA) [8-9] and PSO [2]. The experiments have been made on a PC (Intel Core 2 Duo E7500 @ 2.93 GHz CPU, 2GB RAM, Win2007 OS, C# 2010). The population size and number of generation defined as 100 and 500, respectively. For GA, we used Roulettes Wheel selection, and crossover and mutation rates were 100% and 10%, respectively. The selected parameter values are not optimal values, but were chosen for simplicity as well as for fairness in observation.

Table II compares the tour costs of our proposed PSOPS with GA and PSO. The presented results are the average of five independent runs and the best result (i.e., minimum tour cost) for a particular problem is marked in bold face. For burma14 problem tour costs for GA and PSO are 32.488 and 33.016, respectively. For the same problem, our proposed PSOPS tour cost is 30.945 that is the best among the three methods. It is also notable from the table that in some cases PSO is found worse than GA, the pioneer population based method; whereas our proposed PSOPS modifying PSO is found better than GA for any problem. Moreover, PSOPS is shown the best results for all problems.

TABLE I : DESCRIPTION OF BURMA14 TSP.

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

TABLE II : COMPARISON OF TOUR COST AMONG GA, PSO AND PSOPS FOR BENCHMARK TSP PROBLEMS. THE RESULTS ARE THE AVERAGE OF FIVE INDEPENDENT RUNS OF EACH METHOD WITH POPULATION SIZE 100 AND NUMBER OF GENERATION 500.

Problem	**GA**	**PSO**	**PSOPS**
burma14	32.488	33.016	**30.945**
eil51	860.706	870.231	**768.181**
berlin52	17547.222	15786.424	**14099.361**
eil76	1624.296	1591.035	**1263.521**
kroA100	116722.262	118089.893	**84224.503**
kroA200	211219.596	274825.008	**206281.560**

For better understanding an experimental analysis is done for burma14 problem. Several exiting studies are also considered the problem for their analysis. Fig. 2 shows tour cost minimization over generation for 1000 generation. From the figure it is found that the convergence rate of the proposed PSOPS is higher than other two methods. Although PSO converges faster than GA initially, GA is shown better than PSO after 400 generations. The faster convergence of PSOPS is achieved due to consideration of best next tour considering partial SS from velocity as described in Eq. 8. The slow convergence of PSO is also reported in the previous study that it required around 20000 generations for minimal results [2]. Fig. 3 shows the required time verses generation for the methods to solve the same burma14 problem. It is seen from the figure that GA is slower than PSO or PSOPS, and time requirement for both PSO and PSOPS is almost same. Finally the proposed PSOPS is found an efficient method for solving TSP achieving minimal tour cost within minimal time.

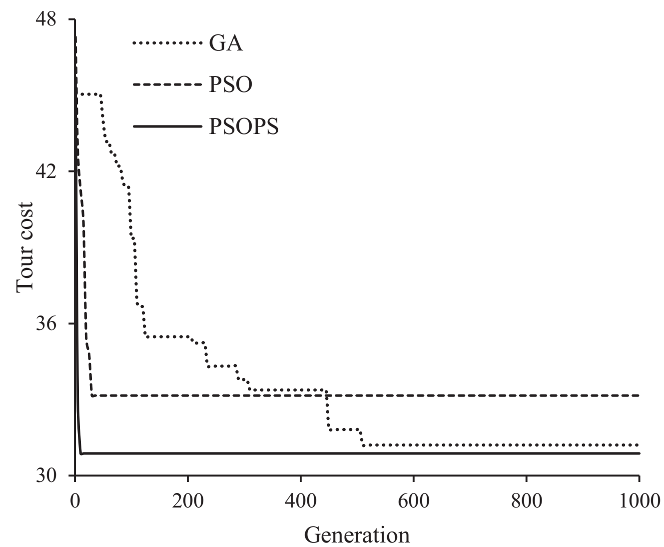

Figure 2. The tour cost vs generation for burma14 problem.

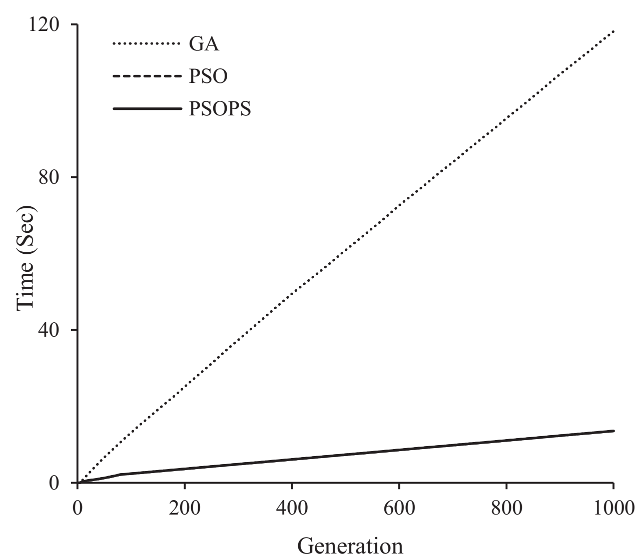

Figure 3. Required time vs generation for burma14 problem.

IV. CONCLUSIONS

This paper investigates a variant of Swap Operator (SO) based Particle Swarm Optimization (PSO) method for solving Traveling Salesman Problem (TSP). For TSP, each particle represents a complete tour and velocity is measured as a Swap Sequence (SS) consisting with several SOs. In the existing method, a new tour is considered after applying a complete SS with all its SOs of the velocity calculated. Since every SO implantation on a solution gives a new solution, we checked all the solutions that found for the velocity and the best one is considered for the update of solution. The proposed PSO with Partial Search (PSOPS) algorithm is shown to produce optimal solution within a minimal time than standard PSO, Genetic Algorithm in solving benchmark TSPs.

REFERENCES

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

[2] K. P. Wang, L. Huang, C. G. Zhou, W. Pang. “Particle swarm optimization for traveling salesman problem”. International Conference on Machine Learning and Cybernetics, Xi’an, 2-5 November (2003), pp. 1583–1585.

[3] Wen-Liang Zhong , Jun Zhang And Wei-Neng Chen. “A Novel Discrete Particle Swarm Optimization to solve Traveling Salesman problem”. IEEE Congress on Evolutionary Computation (2007), pp. 3286-3287.

[4] Jiang-Wei Zhang, Wen-Jian Si. “Improved Enhanced Self-Tentative PSO Algorithm for TSP”. Sixth IEEE International Conference on Natural Computation (2010), pp. 2638-2641.

[5] M. Fatih Tasgetiren, P. N. Suganthan, Quan-Ke Pan. “A Discrete Particle Swarm Optimization Algorithm for the Generalized Traveling Salesman Problem”. GECCO ’07 Proceedings of the 9th annual conference on Genetic And Evolutionary Computation (2007).

[6] Elizabeth F., G. Goldbarg, Marco C. Goldbarg and Givanaldo R.de Souza. “Particle Swarm Optimization Algorithm for Traveling Salesman Problem”. Traveling Salesman Problem, Book edited by: Federico Greco, ISBN (978-958-7619-10-7) pp. 202.

[7] X. H. Shi, Y. C. Liang, H. P. Lee, C. Lu, Q. X. Wang. “Particle swarm optimization-based algorithms for TSP and generalized TSP”. Information Processing Letters 103 (2007), pp. 169–176.

[8] D. E. Goldberg, Genetic Algorithms, Addison-wesley, 1998.

[9] Darrell Whitley. “A genetic algorithm tutorial”. Statistics and Computing (1994) 4, pp. 65-85.